(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-529410
(P2004-529410A)
(43)公表日　平成16年9月24日(2004.9.24)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 13/00	G06F 13/00	520B	5C064
G09C 1/00	G06F 13/00	547T	5J104
H04N 7/173	G09C 1/00	660D	
	H04N 7/173	610Z	

審査請求 未請求　予備審査請求 有　（全 188 頁）

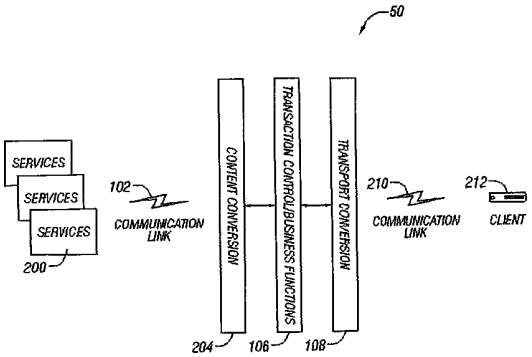
(21)出願番号　特願2002-563701 (P2002-563701)
(86)(22)出願日　平成14年2月1日 (2002.2.1)
(85)翻訳文提出日　平成15年8月4日 (2003.8.4)
(86)国際出願番号　PCT/US2002/002725
(87)国際公開番号　WO2002/063879
(87)国際公開日　平成14年8月15日 (2002.8.15)
(31)優先権主張番号　60/265,986
(32)優先日　平成13年2月2日 (2001.2.2)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　60/266,210
(32)優先日　平成13年2月2日 (2001.2.2)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　60/267,876
(32)優先日　平成13年2月9日 (2001.2.9)
(33)優先権主張国　米国 (US)

(71)出願人　500200845
オープンティブイ・インコーポレーテッド
アメリカ合衆国・９４１１１・カリフォルニア州・サンフランシスコ・サクラメント　ストリート・２７５
(74)代理人　100064621
弁理士　山川　政樹
(72)発明者　アラオ，ラチャッド
アメリカ合衆国・９４０８６・カリフォルニア州・サニイベイル・エンジェル　アベニュ・３３０
(72)発明者　デルプチ，アラン
フランス国・エフ－９２９２７　パリ　ラ　デファンス・アベニュ　アンドレ　プロシン・２０

最終頁に続く

(54)【発明の名称】インタラクティブ・テレビ用のサービス・ゲートウェイ

(57)【要約】
サービス・ゲートウェイは、クライアント・プロトコルと複数の標準通信プロトコルとの間のプロキシの役割を果たす。このサービス・ゲートウェイは、非対称ルーティング、データ圧縮、および暗号化の機能を備え、これにより、最適なクライアント処理能力および通信リンク帯域幅を実現する。サービス・ゲートウェイでは、クライアントとサービス・プロバイダとの間コンテンツ変換を実行することができる。サービス・ゲートウェイは、クライアントで利用可能なメモリとメッセージ内のシーケンス番号を追跡記録して、必要な場合にエラー・コードを生成する。ストア・アンド・フォワードメッセージ機能を抽象セッション識別子とともに用意する。サービス・ゲートウェイは、ユーザ・データグラム・プロトコルをサポートする。



【特許請求の範囲】
【請求項１】
コンピュータ読取り可能媒体であって、命令が実行されたときに、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバ側で受信し、
サービス・プロバイダ互換プロトコルと異なるクライアント・デバイス互換プロトコルに第１のメッセージを変換し、
第１のメッセージをサーバ側で圧縮し、
圧縮された第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つでクライアント・デバイスに送信する動作をコンピュータに実行させる命令が格納されるコンピュータ読取り可能媒体。
【請求項２】
さらに、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信し、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、
変換された未圧縮の第２のメッセージをサーバからサービス・プロバイダに送信する動作をコンピュータに実行させる命令を格納する請求項１に記載の媒体。
【請求項３】
さらに、
クライアント・デバイスに第１のメッセージを送信する前に、
サーバ側で第１のメッセージを暗号化し、暗号化されたデータをクライアント・デバイス互換プロトコル・メッセージにカプセル化し、
暗号化された第１のメッセージ内にフラグを立てて、暗号化されている第１のメッセージが暗号化されていることを示し、
暗号化された第１のメッセージをクライアント・デバイスに送信し、
暗号化された第１のメッセージをクライアント・デバイス側で暗号解読する動作をコンピュータに実行させる命令を格納する請求項２に記載の媒体。
【請求項４】
さらに、
サーバ側で第１のメッセージの断片をクライアント・デバイス互換プロトコルに個別に暗号化する動作をコンピュータに実行させる命令を格納する請求項３に記載の媒体。
【請求項５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項１に記載の媒体。
【請求項６】
さらに、
伝送ビット・レートを制御することによりメッセージ流量を制御する動作をコンピュータに実行させる命令を格納する請求項１に記載の媒体。
【請求項７】
コンピュータ読取り可能媒体であって、
クライアント・デバイス依存トランスポート層からクライアント・デバイスのクライアント・デバイス・ハードウェア識別子を取り出し、
クライアント・デバイス・ハードウェア識別子をネットワーク・オペレータのハードウェア識別子リストに格納し、
サーバとクライアント・デバイスとの間の通信セッションを確立する前にクライアント・デバイスのハードウェア識別子を認証し、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およ

びオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバ側で受信し、
第１のメッセージをクライアント・デバイス互換プロトコルに変換し、
第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つでクライアント・デバイスに送信し、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信し、
第２のメッセージをサーバ側で受信し、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成し、
第２のメッセージ内のクライアント・デバイスのハードウェア識別子のある場所にセッション識別子を挿入し、
第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、
変換されたメッセージをサーバからサービス・プロバイダに送信する動作をコンピュータに実行させる命令を格納するコンピュータ読取り可能媒体。
【請求項８】
さらに、
第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項９】
さらに、
できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１０】
第１のメッセージと第２のメッセージが電子商取引を含む請求項９に記載の媒体。
【請求項１１】
さらに、
第２のメッセージをサーバに送信する前にクライアント・デバイス側で第２のメッセージ内にデバイス互換プロトコルによるＨＴＴＰをカプセル化し、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前にサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換し、
変換されたＨＴＴＰメッセージへの応答としてサーバ側でＨＴＴＰサーバを介してサービス・プロバイダからｃｏｏｋｉｅを受信し、
ｃｏｏｋｉｅをサーバ側にキャッシュし、
セッション識別子変換テーブルへのｃｏｏｋｉｅを生成し、
セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、
クライアント・デバイスのハードウェア識別子を使用して、中央レジストリからユーザ情報を抽出する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１２】
第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項１１に記載の媒体。
【請求項１３】
サーバがビジネス・フィルタをクライアント・デバイスに送信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項７に記載の媒体。

【請求項１４】
さらに、
クライアント・デバイス側のユーザとクライアント・デバイスがオフラインになったときのサービス・プロバイダとの間の電子商取引を遂行する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１５】
さらに、
クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスに送信し、
クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成し、
クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納し、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するためにデータ名サービスへの要求を解決する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項１８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）データに対応するソケット型抽象層を実装する動作をコンピュータに実行させる命令を格納し、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項７に記載の媒体。
【請求項１９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイスで複数のユーザを認証する動作をコンピュータに実行させる命令を格納する請求項７に記載の媒体。
【請求項２０】
さらに、
ビジネス・エージェントにおいて、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引のクライアント・デバイス・ユーザ情報へのアクセスを制御し、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項７に記載の媒体。
【請求項２１】
コンピュータ読取り可能媒体であって、命令が実行されたときに、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをクライア

ント・デバイス側でサーバから受信する動作をコンピュータに実行させる命令を格納し、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なるコンピュータ読取り可能媒体。

【請求項２２】

さらに、

未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信する動作をコンピュータに実行させる命令を格納する請求項２１に記載の媒体。

【請求項２３】

第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスで第１のメッセージを受信する前にサーバ側でメッセージが暗号化されていることを示し、さらに、

暗号化された第１のメッセージをクライアント・デバイス側で暗号解読する動作をコンピュータに実行させる命令を格納する請求項２２に記載の媒体。

【請求項２４】

第１のメッセージが、クライアント・デバイス側で第１のメッセージを受信する前にサーバ側で個別に暗号化されている断片に分割されている請求項２３に記載の媒体。

【請求項２５】

クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項２１に記載の媒体。

【請求項２６】

さらに、

伝送ビット・レートを制御することによりメッセージ流量を制御する動作をコンピュータに実行させる命令を格納する請求項２１に記載の媒体。

【請求項２７】

さらに、

クライアント・デバイス依存トランスポート層からクライアント・デバイスのクライアント・デバイス・ハードウェア識別子を取り出し、

クライアント・デバイスとサーバとの間で通信セッションを確立する前にクライアント・デバイス・ハードウェア識別子の認証用に格納するため識別子リストにクライアント・デバイスのハードウェア識別子を送り、

クライアント・デバイスでサーバから、アプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージを受信し、

サーバがサービス・プロバイダ互換プロトコルからのメッセージをクライアント・デバイス互換プロトコルに変換し、

第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを使用してクライアント・デバイスで受信し、

クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信する動作をコンピュータに実行させる命令を格納し、サーバが第２のメッセージ内のクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換されたメッセージをサービス・プロバイダに送信する請求項２１に記載の媒体。

【請求項２８】

さらに、

第１のメッセージの受信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。

【請求項２９】

さらに、

できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。

【請求項３０】

第１のメッセージと第２のメッセージが電子商取引を含む請求項２９に記載の媒体。

【請求項３１】

さらに、

デバイス互換プロトコルのＨＴＴＰをクライアント・デバイス側で第２のメッセージにカプセル化し、第２のメッセージをサーバに送信してサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換してから、変換されたＨＴＴＰメッセージをサービス・プロバイダに送信し、

クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答としてＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバにセッション識別子を送信する動作をコンピュータに実行させる命令を格納し、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項２７に記載の媒体。

【請求項３２】

第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項３１に記載の媒体。

【請求項３３】

クライアント・デバイスがビジネス・フィルタをサーバから受信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項２７に記載の媒体。

【請求項３４】

さらに、

クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。

【請求項３５】

さらに、

クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスで受信し、

クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。

【請求項３６】

さらに、

第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成し、

クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納し、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。
【請求項３７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するためにデータ名サービスへの要求を解決する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。
【請求項３８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）データに対応するソケット型抽象層を実装する動作をコンピュータに実行させる命令を格納し、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項２７に記載の媒体。
【請求項３９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイスで複数のユーザを認証する動作をコンピュータに実行させる命令を格納する請求項２７に記載の媒体。
【請求項４０】
さらに、
ビジネス・エージェントにおいて、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御する動作をコンピュータに実行させる命令を格納し、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項２７に記載の媒体。
【請求項４１】
命令が実行されたときに、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバからクライアント・デバイスに送信する動作をコンピュータに実行させる命令を格納し、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なるコンピュータ読取り可能媒体。
【請求項４２】
さらに、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをクライアント・デバイスから受信する動作をコンピュータに実行させる命令を格納する請求項４１に記載の媒体。
【請求項４３】
第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスに第１のメッセージを送信し、クライアント・デバイス側で暗号化された第１のメッセージを暗号解読する前にサーバ側でメッセージが暗号化されていることを示す請求項４２に記載の媒体。
【請求項４４】

第１のメッセージを断片に分割し、各断片をサーバ側で個別に暗号化してから、第１のメッセージをクライアント・デバイスに送信する請求項４３に記載の媒体。
【請求項４５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項４１に記載の媒体。
【請求項４６】
さらに、
サーバ側で伝送ビット・レートを制御することによりメッセージ流量を制御する動作をコンピュータに実行させる命令を格納する請求項４１に記載の媒体。
【請求項４７】
命令が実行されたときに、
クライアント・デバイスとサーバとの間で通信セッションを確立する前に、クライアント・デバイスのハードウェア識別子の格納および認証用に、クライアント・デバイス依存トランスポート層から取り出したクライアント・デバイスのハードウェア識別子をサーバ識別子リストに送り、
放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを介してクライアント・デバイスにサーバから、アプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージを送信し、サーバがサービス・プロバイダ互換プロトコルからメッセージをクライアント・デバイス互換プロトコルに変換し、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに受信する動作をコンピュータに実行させる命令を格納し、サーバが第２のメッセージ内のクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換されたメッセージをサービス・プロバイダに送信するコンピュータ読取り可能媒体。
【請求項４８】
さらに、
サーバからクライアント・デバイスへの第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。
【請求項４９】
さらに、
できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに、送信用の第２のメッセージをサーバ側で格納する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。
【請求項５０】
第１のメッセージと第２のメッセージが電子商取引を含む請求項４９に記載の媒体。
【請求項５１】
さらに、
サーバ側で、デバイス互換プロトコルで第２のメッセージにカプセル化されているＨＴＴＰを受信し、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前にサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換し、
クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答として

ＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバ側でセッション識別子を受信する動作をコンピュータに実行させる命令を格納し、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項４７に記載の媒体。

【請求項５２】

第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項５１に記載の媒体。

【請求項５３】

さらに、

サーバからビジネス・フィルタをクライアント・デバイスに送信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。

【請求項５４】

さらに、

クライアント・デバイスがオフラインのときにクライアント・デバイス側のユーザとサービス・プロバイダとの間で電子商取引を遂行する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。

【請求項５５】

さらに、

クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスに送信し、

クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。

【請求項５６】

さらに、

第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成し、クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにそのシーケンス番号を格納し、そのシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイス側で第１のメッセージを拒絶し、第１のメッセージの重複受信を回避する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。

【請求項５７】

さらに、

トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するためにデータ名サービスへの要求を解決する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。

【請求項５８】

さらに、

ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するサーバ内のソケット型抽象層を実装する動作をコンピュータに実行させる命令を格納し、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項４７に記載の媒体。

【請求項５９】

さらに、

サーバ側で、ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア

識別子を使用してクライアント・デバイス側の複数のユーザを認証する動作をコンピュータに実行させる命令を格納する請求項４７に記載の媒体。
【請求項６０】
さらに、
サーバ内のビジネス・エージェントで、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引に含まれるクライアント・デバイス・ユーザ情報へのアクセスを制御する動作をコンピュータに実行させる命令を格納し、
サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類が、サービス・プロバイダとネットワーク・オペレータとの契約に応じてビジネス規則により定められる請求項４７に記載の媒体。
【請求項６１】
インタラクティブ・テレビ・システムにおける非対称通信の方法であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバ側で受信することと、
サービス・プロバイダ互換プロトコルと異なるクライアント・デバイス互換プロトコルに第１のメッセージを変換することと、
第１のメッセージをサーバ側で圧縮することと、
圧縮された第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つでクライアント・デバイスに送信することを含む方法。
【請求項６２】
さらに、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信することと、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換することと、
変換された未圧縮の第２のメッセージをサーバからサービス・プロバイダに送信することを含む請求項６１に記載の方法。
【請求項６３】
さらに、
クライアント・デバイスに第１のメッセージを送信する前に、
サーバ側で第１のメッセージを暗号化し、暗号化されたデータをクライアント・デバイス互換プロトコル・メッセージにカプセル化することと、
暗号化された第１のメッセージ内にフラグを立てて、暗号化されている第１のメッセージが暗号化されていることを示すことと、
暗号化された第１のメッセージをクライアント・デバイスに送信することと、
暗号化された第１のメッセージをクライアント・デバイス側で暗号解読することを含む請求項６２に記載の方法。
【請求項６４】
さらに、
サーバ側で第１のメッセージの断片をクライアント・デバイス互換プロトコルに個別に暗号化することを含む請求項６３に記載の方法。
【請求項６５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項６１に記載の方法。
【請求項６６】
さらに、
伝送ビット・レートを制御することによりメッセージ流量を制御することを含む請求項６１に記載の方法。

【請求項６７】
分散コンピューティング・システムにおける通信の方法であって、
クライアント・デバイス依存トランスポート層からクライアント・デバイスのクライアント・デバイス・ハードウェア識別子を取り出すことと、
クライアント・デバイス・ハードウェア識別子をネットワーク・オペレータのハードウェア識別子リストに格納することと、
サーバとクライアント・デバイスとの間の通信セッションを確立する前にクライアント・デバイスのハードウェア識別子を認証することと、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバ側で受信することと、
第１のメッセージをクライアント・デバイス互換プロトコルに変換することと、
第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つでクライアント・デバイスに送信することと、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信することと、
第２のメッセージをサーバ側で受信することと、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成することと、
第２のメッセージ内のクライアント・デバイスのハードウェア識別子のある場所にセッション識別子を挿入することと、
第２のメッセージをサービス・プロバイダ互換プロトコルに変換することと、
変換されたメッセージをサーバからサービス・プロバイダに送信することを含む方法。
【請求項６８】
さらに、
第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定することを含む請求項６７に記載の方法。
【請求項６９】
さらに、
できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納することを含む請求項６７に記載の方法。
【請求項７０】
第１のメッセージと第２のメッセージが電子商取引を含む請求項６９に記載の方法。
【請求項７１】
さらに、
第２のメッセージをサーバに送信する前にクライアント・デバイス側で第２のメッセージ内にデバイス互換プロトコルによるＨＴＴＰをカプセル化することと、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前にサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換することと、
変換されたＨＴＴＰメッセージへの応答としてサーバ側でＨＴＴＰサーバを介してサービス・プロバイダからｃｏｏｋｉｅを受信することと、
ｃｏｏｋｉｅをサーバ側にキャッシュすることと、
セッション識別子変換テーブルへのｃｏｏｋｉｅを生成することと、
セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答することと、
クライアント・デバイスのハードウェア識別子を使用して、中央レジストリからユーザ情

報を抽出することを含む請求項６７に記載の方法。
【請求項７２】
第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項７１に記載の方法。
【請求項７３】
サーバがビジネス・フィルタをクライアント・デバイスに送信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項６７に記載の方法。
【請求項７４】
さらに、
クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行することを含む請求項６７に記載の方法。
【請求項７５】
さらに、
クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスに送信することと、
クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認することを含む請求項６７に記載の方法。
【請求項７６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成することと、
クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納することと、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避することを含む請求項６７に記載の方法。
【請求項７７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービスを含む請求項６７に記載の方法。
【請求項７８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）データに対応するソケット型抽象層を実装することを含み、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化することを含む請求項６７に記載の方法。
【請求項７９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイスで複数のユーザを認証することを含む請求項６７に記載の方法。
【請求項８０】
さらに、
ビジネス・エージェントにおいて、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御することを含み、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規

則により定められる請求項６７に記載の方法。
【請求項８１】
非対称通信の方法であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをクライアント・デバイス側でサーバから受信することを含み、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なる方法。
【請求項８２】
さらに、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信することを含む請求項８１に記載の方法。
【請求項８３】
さらに、
第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスで第１のメッセージを受信する前にサーバ側でメッセージが暗号化されていることを示し、
暗号化された第１のメッセージをクライアント・デバイス側で暗号解読することを含む請求項８２に記載の方法。
【請求項８４】
さらに、
第１のメッセージが、クライアント・デバイス側で第１のメッセージを受信する前にサーバ側で個別に暗号化されている断片にすでに分割されている請求項８３に記載の方法。
【請求項８５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項８１に記載の方法。
【請求項８６】
さらに、
伝送ビット・レートを制御することによりメッセージ流量を制御することを含む請求項８１に記載の方法。
【請求項８７】
分散コンピューティング・システムにおける通信の方法であって、
クライアント・デバイス依存トランスポート層からクライアント・デバイスのクライアント・デバイス・ハードウェア識別子を取り出すことと、
クライアント・デバイスとサーバとの間で通信セッションを確立する前にクライアント・デバイス・ハードウェア識別子の認証用に格納するため識別子リストにクライアント・デバイスのハードウェア識別子を送ることと、
クライアント・デバイスでサーバから、アプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージを受信し、サーバがサービス・プロバイダ互換プロトコルからメッセージをクライアント・デバイス互換プロトコルに変換することと、
第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを使用してクライアント・デバイスで受信することと、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・

コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信することを含み、サーバが第２のメッセージ内のクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換されたメッセージをサービス・プロバイダに送信する方法。

【請求項８８】

さらに、

第１のメッセージの受信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定することを含む請求項８７に記載の方法。

【請求項８９】

さらに、

できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納することを含む請求項８７に記載の方法。

【請求項９０】

第１のメッセージと第２のメッセージが電子商取引を含む請求項８９に記載の方法。

【請求項９１】

さらに、

デバイス互換プロトコルのＨＴＴＰをクライアント・デバイス側で第２のメッセージにカプセル化し、第２のメッセージをサーバに送信してサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージ変換してから、変換されたＨＴＴＰメッセージをサービス・プロバイダに送信することと、

クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答としてＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバにセッション識別子を送信することを含み、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項８７に記載の方法。

【請求項９２】

第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項９１に記載の方法。

【請求項９３】

さらに、

クライアント・デバイスがサーバからビジネス・フィルタを受信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項８７に記載の方法。

【請求項９４】

さらに、

クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行することを含む請求項８７に記載の方法。

【請求項９５】

さらに、

クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスで受信することと、

クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、

クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認することを含む請求項８７に記載の方法。
【請求項９６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成することと、
クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納することと、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避することを含む請求項８７に記載の方法。
【請求項９７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービスを含む請求項８７に記載の方法。
【請求項９８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するソケット型抽象層を実装することを含み、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化することを含む請求項８７に記載の方法。
【請求項９９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイスで複数のユーザを認証することを含む請求項８７に記載の方法。
【請求項１００】
さらに、
ビジネス・エージェントにおいて、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御することを含み、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項８７に記載の方法。
【請求項１０１】
非対称通信の方法であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサーバからクライアント・デバイスに送信することを含み、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なる方法。
【請求項１０２】
さらに、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをクライアント・デバイスから受信することを含む請求項１０１に記載の方法。
【請求項１０３】
さらに、

第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスに第１のメッセージを送信し、クライアント・デバイス側で暗号化された第１のメッセージを暗号解読する前にサーバ側でメッセージが暗号化されていることを示す請求項１０２に記載の方法。

【請求項１０４】

さらに、

第１のメッセージを断片に分割し、各断片をサーバ側で個別に暗号化してから、第１のメッセージをクライアント・デバイスに送信する請求項１０３に記載の方法。

【請求項１０５】

クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項１０１に記載の方法。

【請求項１０６】

さらに、

サーバ側で伝送ビット・レートを制御することによりメッセージ流量を制御することを含む請求項１０１に記載の方法。

【請求項１０７】

分散コンピューティング・システムにおける通信の方法であって、

クライアント・デバイスとサーバとの間で通信セッションを確立する前に、クライアント・デバイスのハードウェア識別子の格納および認証用に、クライアント・デバイス依存トランスポート層から取り出したクライアント・デバイスのハードウェア識別子をサーバ識別子リストに送ることと、

放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを介してクライアント・デバイスにサーバから、アプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージを送信し、サーバがサービス・プロバイダ互換プロトコルからメッセージをクライアント・デバイス互換プロトコルに変換することと、

クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信することを含み、サーバが第２のメッセージ内のクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換されたメッセージをサービス・プロバイダに送信する方法。

【請求項１０８】

さらに、

サーバからクライアント・デバイスへの第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定することを含む請求項１０７に記載の方法。

【請求項１０９】

さらに、

できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに、送信用の第２のメッセージをサーバ側で格納することを含む請求項１０７に記載の方法。

【請求項１１０】

第１のメッセージと第２のメッセージが電子商取引を含む請求項１０９に記載の方法。

【請求項１１１】

さらに、

サーバ側で、デバイス互換プロトコルで第２のメッセージにカプセル化されているＨＴＴ

Ｐを受信することと、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前にサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換することと、
クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答としてＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバ側でセッション識別子を受信することを含み、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項１０７に記載の方法。

【請求項１１２】
第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項１１１に記載の方法。

【請求項１１３】
さらに、
サーバからビジネス・フィルタをクライアント・デバイスに送信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択することを含む請求項１０７に記載の方法。

【請求項１１４】
さらに、
クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行することを含む請求項１０７に記載の方法。

【請求項１１５】
さらに、
クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージをクライアント・デバイスに送信することと、
クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認することを含む請求項１０７に記載の方法。

【請求項１１６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内にシーケンス番号を生成し、クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにそのシーケンス番号を格納し、そのシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイス側で第１のメッセージを拒絶し、第１のメッセージの重複受信を回避することを含む請求項１０７に記載の方法。

【請求項１１７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービスを含む請求項１０７に記載の方法。

【請求項１１８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するサーバ内のソケット型抽象層を実装することを含み、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化することを含む請求項１０７に記載の方法。

【請求項１１９】

さらに、
サーバ側で、ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイス側の複数のユーザを認証することを含む請求項１０７に記載の方法。
【請求項１２０】
さらに、
サーバ内のビジネス・エージェントで、サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御することを含み、
サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類が、サービス・プロバイダとネットワーク・オペレータとの契約に応じてビジネス規則により定められる請求項１０７に記載の方法。
【請求項１２１】
インタラクティブ・テレビ・システムにおける非対称通信のための装置であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージを受信し、サービス・プロバイダ互換プロトコルと異なるクライアント・デバイス互換プロトコルに第１のメッセージを変換するサーバと、
第１のメッセージをサーバ側で圧縮する圧縮機構と、
圧縮された第１のメッセージを放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つでクライアント・デバイスに送信するメッセージ伝送コンポーネントと
を備える装置。
【請求項１２２】
さらに、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信する通信リンクと、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換する変換機能コンポーネントと、
変換された未圧縮の第２のメッセージをサーバからサービス・プロバイダに送信するメッセージ伝送コンポーネントを備える請求項１２１に記載の装置。
【請求項１２３】
さらに、
第１のメッセージをクライアント・デバイスに送信する前に、第１のメッセージをサーバ側で暗号化し、暗号化されたデータをクライアント・デバイス互換プロトコル・メッセージにカプセル化し、暗号化された第１のメッセージ内にフラグを立てて、暗号化された第１のメッセージが暗号化されていることを示し、暗号化された第１のメッセージをクライアント・デバイスに送信する暗号化コンポーネントと、
暗号化された第１のメッセージをクライアント・デバイス側で暗号解読する暗号解読コンポーネントを備える請求項１２２に記載の装置。
【請求項１２４】
さらに、
サーバ側で第１のメッセージの断片をクライアント・デバイス互換プロトコルに個別に暗号化する断片暗号化コンポーネントを備える請求項１２３に記載の装置。
【請求項１２５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項１２１に記載の装置。
【請求項１２６】
さらに、

送信のビット・レートを制御することによりメッセージ・フロー・レートを制御するメッセージ・フロー・レート・コントローラを備える請求項１２１に記載の装置。
【請求項１２７】
分散コンピューティング・システムにおける通信のための装置であって、
クライアント・デバイス依存トランスポート層から取り出されるクライアント・デバイスのクライアント・デバイス・ハードウェア識別子と、
クライアント・デバイス・ハードウェア識別子をネットワーク・オペレータのハードウェア識別子リストに格納するコンピュータ・メモリと、
サーバとクライアント・デバイスとの間の通信セッションを確立する前にクライアント・デバイスのハードウェア識別子を認証する認証コンポーネントと、
サーバ側でアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをサービス・プロバイダ互換プロトコルで受信し、放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを介して第１のメッセージをクライアント・デバイスに送信し、さらにクライアント・デバイスから受信した第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換された第２のメッセージをサーバからサービス・プロバイダに送信するサーバ・メッセージ処理コンポーネントと、
第１のメッセージをクライアント・デバイス互換プロトコルに変換するサーバ変換コンポーネントと、
クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信するクライアント・デバイス・メッセージ処理コンポーネントと、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成し、クライアント・デバイスのハードウェア識別子の代わりにそのセッション識別子を第２のメッセージに挿入してから、その第２のメッセージをサービス・プロバイダに送信するサーバ・コンポーネントを備える装置。
【請求項１２８】
さらに、
第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定するサーバ・コンポーネントを備える請求項１２７に記載の装置。
【請求項１２９】
さらに、
できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納するコンピュータ・メモリを備える請求項１２７に記載の装置。
【請求項１３０】
第１のメッセージと第２のメッセージが電子商取引を含む請求項１２９に記載の装置。
【請求項１３１】
さらに、
第２のメッセージをサーバに送信する前にクライアント・デバイス側で第２のメッセージ内にデバイス互換プロトコルによるＨＴＴＰをカプセル化するクライアント・デバイス・コンポーネントと、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前にサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換するＨＴＴＰ変換コンポーネントと、
変換されたＨＴＴＰメッセージへの応答としてサーバ側でＨＴＴＰサーバを介してサービ

ス・プロバイダから受け取ったｃｏｏｋｉｅをキャッシュするためのコンピュータ・メモリと、
セッション識別子変換テーブルへのｃｏｏｋｉｅを生成し、セッション識別子テーブルへのｃｏｏｋｉｅを使用してＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、クライアント・デバイスのハードウェア識別子を使用して中央レジストリからユーザ情報を抽出するサーバ・コンポーネントを備える請求項１２７に記載の装置。

【請求項１３２】
第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項１３１に記載の装置。

【請求項１３３】
サーバがビジネス・フィルタをクライアント・デバイスに送信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項１２７に記載の装置。

【請求項１３４】
さらに、
クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行するサーバ・コンポーネントを備える請求項１２７に記載の装置。

【請求項１３５】
さらに、
クライアント・デバイスに送信された、クライアント・デバイス側で利用できるメモリの量を要求する第３のメッセージと、
クライアント・デバイスに向かう第４のメッセージのメッセージ・サイズをチェックし、クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認するコンポーネントを備える請求項１２７に記載の装置。

【請求項１３６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側の第１のメッセージ内に生成されるシーケンス番号と、
クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納するコンピュータ・メモリと、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避する拒絶コンポーネントを備える請求項１２７に記載の装置。

【請求項１３７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービス・コンポーネントを含む請求項１２７に記載の装置。

【請求項１３８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するソケット型抽象層コンポーネントを備え、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項１２７に記載の装置。

【請求項１３９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用

してクライアント・デバイスで複数のユーザを認証する認証コンポーネントを備える請求項１２７に記載の装置。
【請求項１４０】
さらに、
サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御するビジネス・エージェントを備え、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項１２７に記載の装置。
【請求項１４１】
インタラクティブ・テレビ・システムにおける非対称通信のための装置であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージをクライアント・デバイス側でサーバから受信する通信リンクを備え、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なる装置。
【請求項１４２】
さらに、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージをサーバに送信するクライアント・メッセージ・ハンドラ・コンポーネントを備える請求項１４１に記載の装置。
【請求項１４３】
第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスで第１のメッセージを受信する前にサーバ側でメッセージが暗号化されていることを示し、
暗号化された第１のメッセージをクライアント・デバイス側で暗号解読する暗号解読コンポーネントを備える請求項１４２に記載の装置。
【請求項１４４】
第１のメッセージが、クライアント・デバイス側で第１のメッセージを受信する前にサーバ側で個別に暗号化されている断片に分割されている請求項１４３に記載の装置。
【請求項１４５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項１４１に記載の装置。
【請求項１４６】
さらに、
送信ビット・レートを制御することによりメッセージ・フロー・レートを制御するメッセージ・フロー・レート・コントローラを備える請求項１４１に記載の装置。
【請求項１４７】
分散コンピューティング・システムにおける通信のための装置であって、
サーバがサービス・プロバイダ互換プロトコルからのメッセージをクライアント・デバイス互換プロトコルに変換し、放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを介して第１のメッセージをクライアント・デバイスに送信するようになっており、そのサーバからクライアント・デバイスで受信する、アプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージと、
クライアント・デバイス依存トランスポート層から取り出されるクライアント・デバイスのクライアント・デバイス・ハードウェア識別子と、

クライアント・デバイスとサーバとの間で通信セッションを確立する前にクライアント・デバイス・ハードウェア識別子の認証用に格納するため識別子リストにクライアント・デバイスのハードウェア識別子を送信し、クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスからアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージをサーバに送信するクライアント・デバイス・メッセージ・ハンドラとを備え、サーバが第２のメッセージにクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、第２のメッセージをサービス・プロバイダ互換プロトコルに変換し、変換されたメッセージをサービス・プロバイダに送信する装置。

【請求項１４８】

さらに、

第１のメッセージの受信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定するクライアント・コンポーネントを備える請求項１４７に記載の装置。

【請求項１４９】

さらに、

できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージを格納するコンピュータ・メモリを備える請求項１４７に記載の装置。

【請求項１５０】

第１のメッセージと第２のメッセージが電子商取引を含む請求項１４９に記載の装置。

【請求項１５１】

クライアント・デバイス側で第２のメッセージがデバイス互換プロトコルでＨＴＴＰをカプセル化してから、第２のメッセージをサーバに送信し、サーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージ変換してから、変換されたＨＴＴＰメッセージをサービス・プロバイダに送信し、

クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答としてＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバにセッション識別子を送信するセッション識別子コンポーネントを備え、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項１４７に記載の装置。

【請求項１５２】

第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項１５１に記載の装置。

【請求項１５３】

クライアント・デバイスがビジネス・フィルタをサーバから受信して、クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択する請求項１４７に記載の装置。

【請求項１５４】

さらに、

クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行するクライアント・コンポーネントを備える請求項１４７に記載の装置。

【請求項１５５】

さらに、
クライアント・デバイス側で利用できるメモリの量を要求するクライアント・デバイス側の第３のメッセージと、
クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認するためのクライアント・デバイスに向かう第４のメッセージのメッセージ・サイズを格納する請求項１４７に記載の装置。
【請求項１５６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前のサーバ側の第１のメッセージ内のシーケンス番号と、
クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納するコンピュータ・メモリと、
このシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイスで第１のメッセージを拒絶し、第１のメッセージを重複受信することを回避する拒絶コンポーネントを備える請求項１４７に記載の装置。
【請求項１５７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービスを備える請求項１４７に記載の装置。
【請求項１５８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するソケット型抽象層を備え、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項１４７に記載の装置。
【請求項１５９】
さらに、
ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイスで認証される複数のユーザ識別子を備える請求項１４７に記載の装置。
【請求項１６０】
さらに、
サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御するビジネス・エージェントを備え、
サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項１４７に記載の装置。
【請求項１６１】
インタラクティブ・テレビ・システムにおける非対称通信のための装置であって、
サービス・プロバイダ互換プロトコルでアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含むサーバからクライアント・デバイスへの第１のメッセージを格納し、第１のメッセージが圧縮され、サービス・プロバイダ互換プロトコルからサーバ側のクライアント・デバイス互換プロトコルに変換され、クライアント・デバイス互換プロトコルはサービス・プロバイダ互換プロトコルと異なる装置。
【請求項１６２】
さらに、
未圧縮の第２のメッセージをサービス・プロバイダ互換プロトコルに変換してサービス・プロバイダに送信するために、クライアント・デバイス互換プロトコルでクライアント・デバイスからサーバに送信されるアプリケーション・コード、制御、データ、およびオー

ディオ／視覚的データのうちの少なくとも１つを含む未圧縮の第２のメッセージを格納する請求項１６１に記載の装置。
【請求項１６３】
第１のメッセージを暗号化し、クライアント・デバイス互換プロトコルにカプセル化し、第１のメッセージ内にフラグを立てて、クライアント・デバイスに第１のメッセージを送信し、クライアント・デバイス側で暗号化された第１のメッセージを暗号解読する前にサーバ側でメッセージが暗号化されていることを示す請求項１６２に記載の装置。
【請求項１６４】
第１のメッセージを断片に分割し、各断片をサーバ側で個別に暗号化してから、第１のメッセージをクライアント・デバイスに送信する請求項１６３に記載の装置。
【請求項１６５】
クライアント・デバイス互換プロトコルがクライアント・デバイス内のネイティブ・トランスポート層と互換性がある請求項１６１に記載の装置。
【請求項１６６】
さらに、
サーバ側の送信ビット・レートを制御することによりメッセージ・フロー・レートを制御するメッセージ・フロー・レート・コントローラを備える請求項１６１に記載の装置。
【請求項１６７】
分散コンピューティング・システムにおける通信のための装置であって、
クライアント・デバイスとサーバとの間で通信セッションを確立する前に、クライアント・デバイスのハードウェア識別子の格納および認証用に、クライアント・デバイス依存トランスポート層からサーバ識別子リストへ取り出したクライアント・デバイスのハードウェア識別子を備え、
サービス・プロバイダ互換プロトコルからメッセージをクライアント・デバイス互換プロトコルに変換するサーバからクライアント・デバイスに送信される、放送用搬送波、ローカル・エリア・ネットワーク、およびポイントツーポイント接続のうちの少なくとも１つを介して送信されたアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第１のメッセージと、
クライアント・デバイスのハードウェア識別子からセッション識別子を生成するためにクライアント・デバイス互換プロトコルでクライアント・デバイスから受信したサーバへのアプリケーション・コード、制御、データ、およびオーディオ／視覚的データのうちの少なくとも１つを含む第２のメッセージとを備え、サーバが第２のメッセージ内のクライアント・デバイスのハードウェア識別子の代わりにセッション識別子を挿入し、かつ変換されたメッセージをサービス・プロバイダに送信するために第２のメッセージをサービス・プロバイダ互換プロトコルに変換する装置。
【請求項１６８】
さらに、
サーバからクライアント・デバイスへの第１のメッセージの送信に、放送用搬送波とポイントツーポイント接続待ち時間、および放送ストリームとポイントツーポイント接続負荷条件、およびメッセージ・サイズに基づいて放送用搬送波を使用するか、ポイントツーポイント接続を使用するかを決定するサーバ・ルーティング・コンポーネントを備える請求項１６７に記載の装置。
【請求項１６９】
さらに、
できる限り速やかに、接続時、ランダムな期間経過後、設定された期間後、イベントの発生後、メッセージの発生後、および拡散された利用可能な帯域幅のうちの少なくとも１つを含むタイミング制約条件を満たしたときに送信用の第２のメッセージをサーバ側で格納するコンピュータ・メモリを備える請求項１６７に記載の装置。
【請求項１７０】
第１のメッセージと第２のメッセージが電子商取引を含む請求項１６９に記載の装置。

【請求項１７１】
さらに、
変換されたＨＴＴＰメッセージをサービス・プロバイダに送信する前に、デバイス互換プロトコルで第２のメッセージ内にカプセル化されているＨＴＴＰを含む第２のメッセージを受信してサーバ側で第２のメッセージを標準ＨＴＴＰ通信プロトコル・メッセージに変換するコンバータと、
クライアント・デバイスによって送信され変換されたＨＴＴＰメッセージへの応答としてＨＴＴＰサーバを介してサービス・プロバイダから受信したキャッシュされているｃｏｏｋｉｅと関連してサーバ側でセッション識別子を受信するセッション識別子コンポーネントを備え、セッション識別子変換テーブルへのｃｏｏｋｉｅでｃｏｏｋｉｅが識別され、セッション識別子テーブルへのｃｏｏｋｉｅを使用して、ＨＴＴＰサーバからのクライアント・デバイスのハードウェア識別子名要求に応答し、中央レジストリからユーザ情報を抽出する請求項１６７に記載の装置。
【請求項１７２】
第１のメッセージがＴＣＰ／ＩＰ上のＨＴＴＰを使用し、第２のメッセージがＤＡＴＰ上のＬＨＴＴＰを使用する請求項１７１に記載の装置。
【請求項１７３】
さらに、
クライアントの好み、視聴者プロファイル、およびトランザクション履歴のうち少なくとも１つに基づき入力からクライアント・デバイスにキャプチャする情報を選択するためのサーバからクライアント・デバイスへのビジネス・フィルタを備える請求項１６７に記載の装置。
【請求項１７４】
さらに、
クライアント・デバイスがオフラインのときにクライアント・デバイスのユーザとサービス・プロバイダとの間の電子商取引を遂行するサーバ・コンポーネントを備える請求項１６７に記載の装置。
【請求項１７５】
さらに、
クライアント・デバイス側で利用できるメモリの量を要求する、クライアント・デバイスへの第３のメッセージと、
クライアント・デバイス側の利用可能メモリの量がクライアント・デバイスに第４のメッセージを転送する前に第４のメッセージを受け取れるだけ十分あることを確認するためのクライアント・デバイスに向かう第４のメッセージのメッセージ・サイズを備える請求項１６７に記載の装置。
【請求項１７６】
さらに、
第１のメッセージをクライアント・デバイスに送信する前にサーバ側で第１のメッセージ内に生成されたシーケンス番号を備え、クライアント・デバイスで第１のメッセージを受信した後クライアント・デバイス内にタイムスタンプとともにシーケンス番号を格納し、そのシーケンス番号がスライディング・タイム・リジェクション・ウィンドウ内に出現した場合にクライアント・デバイス側で第１のメッセージを拒絶し、第１のメッセージの重複受信を回避する請求項１６７に記載の装置。
【請求項１７７】
さらに、
トランスポート通信プロトコル・メッセージ内のサービス・プロバイダを識別するサービス識別子を解決するデータ名サービスを備える請求項１６７に記載の装置。
【請求項１７８】
さらに、
ユーザ・データグラム・プロトコル（ＵＤＰ）に対応するソケット型抽象層をサーバ内に

備え、ソケット型抽象層をＵＤＰの上で実行して、ＵＤＰをトランスポート・レベル・プロトコル・メッセージ内にカプセル化する請求項１６７に記載の装置。
【請求項１７９】
さらに、
サーバ側で、ニックネーム、個人識別子、およびクライアント・デバイスのハードウェア識別子を使用してクライアント・デバイス側の複数のユーザを認証する認証コンポーネントを備える請求項１６７に記載の装置。
【請求項１８０】
さらに、
サービス・プロバイダとクライアント・デバイス・ユーザとの間の電子商取引内のクライアント・デバイス・ユーザ情報へのアクセスを制御するビジネス・エージェントをサーバ内に備え、サービス・プロバイダに提供されるクライアント・ビジネス情報の量と種類がサービス・プロバイダとネットワーク・オペレータとの間の契約に応じてビジネス規則により定められる請求項１６７に記載の装置。
【発明の詳細な説明】
【技術分野】
【０００１】
（著作権表示）
本出願文書の開示の一部に、著作権保護の主張の対象となる資料（コード・リスティングおよびメッセージ・リスティング）が含まれている。著作権所有者は、米国特許商標庁のファイルまたは記録の記載に従い誰が特許文書または特許開示のそのままの複製を行おうと異存はないが、いかなる形であれ他のすべての著作権を留保する。Ｃｏｐｙｒｉｇｈｔ　２００１　ＯｐｅｎＴＶ，Ｉｎｃ．。
【０００２】
本発明は、インタラクティブ・テレビ環境における通信の分野に関するものであり、特に、インタラクティブ・テレビ用の汎用メタ言語およびデジタル・テレビ・アプリケーション・プロトコルを実装する方法と装置に関するものである。
【背景技術】
【０００３】
インタラクティブ・テレビ・システムを利用して、さまざまなサービスを視聴者に提供することができる。インタラクティブ・テレビ・システムは、標準的なテレビ番組ストリーム、インタラクティブ・テレビ・アプリケーション、テキストおよびグラフィック画像、Ｗｅｂページおよびその他の種類の情報を配信することができる。インタラクティブ・テレビ・システムはさらに、視聴者のアクションまたは応答を登録することもでき、マーケティング、娯楽、および教育などの目的に使用できる。ユーザまたは視聴者によるシステムのインタラクティブな操作では、広告の製品またはサービスの注文、ゲーム番組出場者との競争、特定の番組に関する専門情報の要求、または数ページ分の情報の検索などを行うことができる。
【０００４】
通常、放送サービス・プロバイダまたはネットワーク・オペレータが、視聴者のテレビに送信するためのインタラクティブ・テレビ信号を発信する。インタラクティブ・テレビ信号には、アプリケーション・コードまたは制御情報を含むインタラクティブ部分とテレビ番組またはその他の情報表示を含むオーディオ／ビデオ部分が含まれる。放送サービス・プロバイダは、オーディオ／ビデオ（Ａ／Ｖ）およびインタラクティブ部分を単一の信号にまとめて、ユーザのテレビに接続されている受信機に送信する。この信号は、一般に、送信前に圧縮され、ケーブル・テレビ（ＣＡＴＶ）回線や直接衛星送信システムなどの通常の放送チャンネルを通じて送信される。
【０００５】
通常、テレビのインタラクティブ機能の制御には、テレビに接続されているセットトップ・ボックス（ＳＴＢ）を使用する。ＳＴＢは、放送サービス・プロバイダによって送信さ

れた放送信号を受信し、信号のＡ／Ｖ部分から信号のインタラクティブ部分を分離し、信号のそれぞれの部分を圧縮解除する。ＳＴＢでは、インタラクティブ情報を使用して、Ａ／Ｖ情報がテレビに送信されている間に、例えばアプリケーションを実行する。ＳＴＢは、情報をテレビに送信する前に、Ａ／Ｖ情報をインタラクティブ・アプリケーションによって生成されたインタラクティブ・グラフィックスまたはオーディオと組み合わせることができる。インタラクティブ・グラフィックスおよびオーディオでは、視聴者に追加情報を提示したり、視聴者に入力を求めたりすることができる。ＳＴＢは、モデム接続またはケーブルを介して視聴者の入力またはその他の情報を放送サービス・プロバイダに送ることができる。

【０００６】

寄せ集め的な性質に従って、インタラクティブ・テレビジョン・システムは、放送サービス・プロバイダ／ネットワーク・オペレータから情報を受け取るクライアントまたは視聴者が理解できるさまざまな異なる通信プロトコルでコンテンツを提供する。典型的には、クライアントは、限られた処理能力と通信帯域幅しか持たないプロセッサを備えたＳＴＢである。各種コンテンツおよびプロトコルの変換は、通常のＳＴＢプロセッサに備わっている限られた処理能力では対応できない。そこで、クライアント／ＳＴＢプロセッサが容易に認識でき、サービス・プロバイダが使用するさまざまなプロトコルで通信できる単純な通信プロトコルが必要である。また、インタラクティブ・テレビ環境においてアクセス、コンテンツ、およびスケジューリングの適応制御を行うソフトウェアおよびハードウェアのアーキテクチャも必要である。

【発明の開示】

【発明が解決しようとする課題】

【０００７】

本発明は、上述のインタラクティブ・テレビ環境に必要な機能の問題を解決するものである。本発明は、長い間切実に求められていたＳＴＢプロセッサにより容易に取り扱える単純なコンテンツおよび通信プロトコルの提供に応え、ヘッドエンド・オペレータのサービス・プラットフォーム（ＳＰ）またはサーバ、その受信契約者クライアントおよび複数のサービス・プロバイダとの複雑な通信を可能にする。以下の説明ではクライアント／ＳＴＢの例を使用するが、本発明は、デジタル・アシスタント、携帯電話、ポケット・パーソナル・コンピュータ、または電子信号を受信することができる他の種類の電子デバイスをはじめとするすべてのクライアント・デバイスに適用される。本発明は、サービス・プラットフォーム（ＳＰ）内に配置される。テレビ信号を受信契約者に送るネットワーク・オペレータは、ＳＰまたはサーバを使用し、ビジネス、トランスポート、および通信機能を作成、提供して、サービス・ゲートウェイを介してサービス・プロバイダとクライアントまたはＳＴＢ視聴者との間で通信を行うことができる。

【課題を解決するための手段】

【０００８】

インタラクティブ・テレビ環境では、クライアントからＳＰへの間欠的リターン・パスなど、インタラクティブ・テレビに固有の問題に対応し、解決する必要がある。つまり、クライアント・デバイスは、ＳＴＢをオフにした場合のように、通信リンクに常に接続されているわけではないということである。そのため、クライアントからのアクティブなリターン・パスが常にあるわけではない。本発明は、このような間欠的リターン・パス問題を緩和するストア・アンド・フォワード機能を実現する。

【０００９】

帯域幅および処理の制限および通信の複雑さもまた、インタラクティブ・テレビ環境では問題になる。一方、ネットワーク・オペレータは、通常、データおよび番組をクライアントに送信するのに比較的大きな伝送容量（通常、衛星放送受信アンテナ）を放送チャネルに持たせている。他方、クライアントのリターン・パスは、通常ＳＴＢシナリオではデータ伝送容量が比較的低く、電話回線はリターン・パスである。リターン・パスがたまたま大きな帯域幅を持つとしても、ＳＴＢ／クライアントは通常、リターン・パスでデータを

送信するのに低速なモデムを用いている。処理限界があるため、ＳＴＢすなわちクライアントが、ＳＴＢと通信しているサービス・プロバイダによって利用される多数の通信プロトコルを処理する能力が限られる。

【００１０】

本発明はさらに、帯域幅の割り当てが最適になるように、ヘッドエンド・サーバからセットトップ・ボックスに最適な形でコンテンツを転送する方法を実現する。本発明のこの態様は、セットトップ・ボックス内に全体が配置され、インタラクティブ・テレビ環境で異なるチャネルの待ち時間を計算する統計的計算を実行するアプリケーションを含む。本発明のこの態様はさらに、これらの計算の結果を使用してヘッドエンド・サーバからコンテンツを抜き出してセットトップ・ボックスに送り、コンテンツを転送するのに要する時間を短縮することも含む。チャネルの待ち時間の好ましい定義は、要求が発行されてからその要求が遂行されるまでの時間の長さである。利用可能なすべてのチャネルにわたるファイル要求の小さな部分、好ましくは５％未満をランダムに発行することでレポートを生成し、待ち時間のサンプリングを行う。チャネルを使用する際に周波数を加減することができ、使用する周波数は統計の結果に基づく。より詳細な結果は、統計に対する一組の優先度をさらに細かく分けることにより得られる。例えば、本発明では、ファイルのサイズと所定のチャネルでの待ち時間との関係を記録する。ヘッドエンド・サーバは、ネットワークの要求に応じて放送カルーセル内のリソースの出現を加減する。本発明はさらに、送信のビット・レートを制御することによりメッセージ・フロー・レートを制御するメッセージ・フロー・レート・コントローラを提示する。これらの課題および他の課題を、本発明で取りあげる。

【発明を実施するための最良の形態】

【００１１】

本発明の他の目的および利点は、以下の詳細な説明を読み、添付図面を参照することで明白になるであろう。

【００１２】

本発明はさまざまな修正を加えることができ、また他の形態も可能であるが、特定の実施形態を図面の例を用いて示しており、これらについて以下で詳述する。ただし、図面および詳細な説明は本発明を開示されている特定の形態に制限する意図はなく、むしろその反対に、付属の請求項によって定義されているように、発明は本発明の精神と範囲にあるすべての修正、等価物、および代替え物を対象とすることは理解されるであろう。

【００１３】

本発明は、サービス・ゲートウェイがヘッドエンド・オペレータのサービス・プラットフォーム（ＳＰ）内に配置され、コンテンツ・トランスコーダ、Ｈ２Ｏおよびデジタル・テレビ・アプリケーション・トランスポート・プロトコルと相互作用する。代表的なインタラクティブ・テレビ環境には、複数のクライアント／受信契約者、典型的には、さまざまな通信プロトコルを使用して複数のネットワーク上でコンテンツを提供する複数のアプリケーション・サーバと通信しなければならないＳＴＢがある。通常、ＳＴＢが備える処理能力は限られており、ＳＴＢプロセッサあるいはＳＴＢスタック内に多数の通信プロトコル・ハンドラを入れるのは望ましくない。したがって、すべてのＳＴＢおよびアプリケーション・サーバに対応できる共通通信インターフェイスが必要である。本発明は、サービス・ゲートウェイが、典型的な処理能力の限られているＳＴＢおよびＳＰに適し、プロセッサ稼働率が低くてもよい通信プロトコル・プロキシを備える。サービス・ゲートウェイは、代表的なインターネット通信プロトコルと比べて処理サイクル数が比較的少ないＤＡＴＰを使用できる。ＤＡＴＰを使用すると、ＳＴＢでの通信プロトコル・ハンドラのオーバーヘッドが低減され、すべてのＳＴＢに対して通信プロトコル・ハンドラを共通にすることができる。ＤＡＴＰプロトコルは、ＳＴＢのオペレーティング・システムとインターフェイスするＳＴＢから独立しているバイト・コードであるＯコードで書かれているため、すべてのＳＴＢに移植可能である。

【００１４】

本発明では、ＳＧＷは、ＤＡＴＰサーバ通信プロキシおよび非同期ルーターとして稼働する。ＳＧＷにより、ＳＴＢにあるＳＰクライアントはＤＡＴＰプロトコルを使用してアプリケーション・サーバに接続することができる。ＨＴＭＬとネイティブ・コードとの間のプロキシ、すなわちＨ２Ｏが用意されているが、これは、この文脈では、ＳＰアプリケーション・サーバとみなすことができる。Ｈ２Ｏは、ＨＴＭＬからＳＰ　Ｏコードへなど、特定のコンテンツ変換を実行する。Ｏコードは、ＳＰ上で実行されている仮想マシンのＳＴＢ独立のバイト・コードである。好ましい実施形態では、ＤＡＴＰプロトコル・スタックのＯコード実装がクライアント、通常はＳＴＢ内に存在する。クライアントは、ＤＡＴＰプロトコルを使用してＤＡＴＰサーバ、ＳＧＷと通信する。Ｈ２Ｏプロキシは、ＨＴＭＬからＯコードへなどのコンテンツ変換を実行するＳＧＷの片方の側に存在する。クライアント／ＳＴＢ内のＤＡＴＰスタックのＯコード実装は、通信要求を発行し、ＤＡＴＰプロトコルを使用してＳＧＷと通信する。Ｈ２Ｏによって変換されたコンテンツは、ＳＧＷを通して、コンテンツが表示されるクライアントに渡される。

【００１５】

ＳＧＷは、各個別ＳＴＢを扱い、各関連コンテンツを処理する実行スレッドを作成するＤＡＴＰサーバ機能を備える。ＳＧＷサーバ・スタックは、ＤＡＴＰプロトコルを使用してクライアント／ＳＴＢと通信する。ＳＧＷはさらに、ＳＧＷを介してＳＴＢと異なるアプリケーション・サーバとの間の通信をＳＴＢが行えるようにするのに必要な適切なプロトコルを用いる。インタラクティブ・テレビ・アプリケーションは通常、よく知られているインターネット・ベースのプロトコル（ＨＴＭＬなど）を使用して、クライアント／ＳＴＢとアプリケーション・サーバとの間の通信を行う。本発明では、ＳＧＷが、ＳＧＷを介したクライアント／ＳＴＢとアプリケーション・サーバとの間の汎用の適切な非対称通信プロトコルを備える。本発明は、クライアント／ＳＴＢで利用可能な最小限の処理およびメモリに適応する。

【００１６】

ＳＧＷはデータ圧縮に対して非対称ソリューションを採用している。クライアント／ＳＴＢからネットワーク・オペレータへの双方向経路の帯域幅は比較的小さく、通常の電話回線またはケーブルのリターン・チャネルが低速のモデムに接続されているのがふつうである。したがって、低速モデムで利用可能な帯域幅を上げるには、サーバからクライアント／ＳＴＢにダウンロードするコンテンツを圧縮する。しかし、クライアント／ＳＴＢでは、データ圧縮を実行しない方が好ましい。返されるクライアント／ＳＴＢデータは比較的小さく、データ圧縮を実行しようにも処理能力がないのがふつうであるＳＴＢプロセッサでデータ圧縮をする必要がない。しかし、他の実施形態では、クライアント／ＳＴＢからのデータ圧縮が望ましい場合があり、このような場合には、ＳＧＷ側でデータ圧縮を実行する。データ圧縮は、クライアント／ＳＴＢに関して、データが下流のクライアント／ＳＴＢへ流れるときに圧縮され、ＳＴＢから上流に流れるときには圧縮されないという点で、非対称である。そのため、本発明のアーキテクチャは、通信している両方のエンティティが対称的な能力を持つと仮定している通常のインターネット・ベースのプロトコルと異なり非対称である。

【００１７】

ＳＧＷとクライアント／ＳＴＢは、ユーザ識別子ではなくクライアントのセッション識別子を使用してアプリケーション・サーバと通信するので、クライアント・ユーザは匿名ユーザのままでよい。本発明はさらに、クライアントへのマルチキャスト機能も備える。放送帯域幅であり、チューナーがＳＴＢ内にあり、放送メッセージが利用でき、ＳＴＢ内の特定のフィルタ・セットアップにより感知されるときに、放送リンクを介してマルチキャスト・メッセージを複数のクライアントに送信することができる。ＤＡＴＰを介するＳＧＷは、放送時に特定のエントリからＳＴＢがメッセージを受信するよう要求する。ＳＴＢで放送を受信するのにチューナーを利用できない場合、チューナーなしでメッセージ断片も各ポイントツーポイントの個別リンクでＳＴＢに送信される。ＳＴＢがＬＡＮ上にある場合、メッセージがＳＴＢのＬＡＮ上のよく知られているアドレスに送信される。

【００１８】
本発明はさらに、インターネット・アプリケーションからのｃｏｏｋｉｅを処理するための新規性のある構造と方法も提示し、またＨＴＴＰ要求をＤＡＴＰメッセージ内にカプセル化する「軽量」ＨＴＴＰプロトコルであるＬＨＴＴＰも実装する。ＬＨＴＴＰは、ＤＡＴＰの上で実行されるＨＴＴＰの簡易バージョンであり、ＳＧＷがサービス・プロバイダとの通信のためＨＴＴＰに変換する。この新規性のあるＬＨＴＴＰは、ＤＡＴＰの上で実行され、ＴＣＰ／ＩＰ規格の一部を実装したものではない。
【００１９】
ＳＧＷは、ＳＴＢとのリンクつまりソケット接続を確立する。しかし、ユーザ・データグラム・プロトコル（ＵＤＰ）を実装する場合も、ＵＤＰは直接実行されない。ＵＤＰを出力できるＳＴＢでは、ＵＤＰにＤＡＴＰをカプセル化する。ＤＡＴＰカプセル化ＵＤＰはＳＧＷに送信される。ＵＤＰの場合、ＳＧＷのソケットとＳＴＢのソケットは、実際には、ＵＤＰの上のシミュレートされた接続に結合されまとめられる。このＳＧＷのシミュレートされた接続を通じて、ＤＡＴＰパケットがＳＴＢからＳＧＷサーバへ、またＳＧＷサーバからＳＴＢへと送信される。
【００２０】
多くのＳＴＢモデムは、データ圧縮機能を持たず、備える処理能力も最低限度であるため、ＳＴＢのデータ圧縮を実行する処理コストに応じる余裕がない。したがって、好ましい実施形態では、ＳＧＷは非対称データ圧縮機能を備え、この機能はＳＧＷ側で実行される。ＳＴＢではデータを圧縮しない。ＳＴＢは、圧縮データを受け取って圧縮解除するが、ＳＴＢはＳＧＷによって実行されるデータ圧縮を実行しない。ただし、データの圧縮解除は、データの圧縮に比べて計算能力をあまり必要としないため、ＳＴＢで圧縮解除を実行するのが好ましい。圧縮されたデータはＳＴＢのＤＡＴＰスタックに送られるが、未圧縮データはＳＴＢからＳＧＷに送られる。ＳＧＷは、ＳＴＢから送られた未圧縮データにデータ圧縮を実行し、ＳＧＷはその圧縮されたデータをアプリケーション・サーバに返す。したがって、好ましいＤＡＴＰ／ＳＧＷ非対称圧縮では、ＳＴＢからＳＧＷを通りアプリケーション・サーバへ向かうリターン・パスの帯域幅が増える。
【００２１】
本発明のＳＧＷは、非対称ルーティングを行う。非対称ルーティングでは、データが放送のため放送ストリームに送られるように帯域幅の一部がＳＧＷに割り当てられる。ＳＧＷは、放送ストリームで、またはＳＧＷとＳＴＢとの間のポイントツーポイント（ＰＴＰ）接続で１つまたは複数のＳＴＢにデータを送信するかどうかを決定することができる。ＳＧＷは、データの量、ＳＴＢへのポイントツーポイント・リンクの速度、および現在の通信リンクの負荷状態に基づいて、放送ストリームかまたはＰＴＰ接続を介するかデータの送信経路を決定する。そこで、データ・セットが大きすぎるので、ポイントツーポイント・リンクでデータ・セットを送信せず、その代わりに放送ストリームを介して送信することをＳＧＷが決めることもできる。受信ストリームまたはポイントツーポイント・リンクに送信する前に、データをＳＧＷで圧縮して、ＳＧＷとリンクまたはストリームとの間のリンクの帯域幅を高くし、ＳＴＢ内のメモリ限界に適応するようにできる。
【００２２】
ＳＧＷでは、すべてのＳＴＢスタックのオペレーションに最低限の処理能力が要求されるように設計されているためＤＡＴＰを計算量の少ないものにできる。たとえば、ＤＡＴＰ暗号化方式では、Ｒｉｖｅｓｔ、Ｓｈａｍｉｒ、およびＡｌｄｅｒｍａｎ（ＲＳＡ）公開鍵暗号化を使用したときに、サーバから送られてくる鍵に対し、指数演算段階の時間と処理能力が最低限で済むように小さい指数値（３またはそれ以上）を選択する。そのため、大きな計算はＳＧＷに割り当てられ、ＳＴＢすなわちクライアントのプロセッサは最低限の処理能力で済む。同様に、ＳＴＢ内のＤＡＴＰの上にあるＬＨＴＴＰ層では、重い解析や他の大きな処理能力を必要とするオペレーションを実行する必要がない。その代わりに、ＬＨＴＴＰによりＨＴＴＰデータをＤＡＴＰメッセージにカプセル化し、ＨＴＴＰプロトコルへの変換などの大きな計算能力を必要とするＨＴＴＰ機能についてはＳＧＷ側で処

理する。
【００２３】
ＤＡＴＰが実行するトランザクションはさらに多い。というよりは、ＤＡＴＰはトランザクション指向のプロトコルではなくメッセージ・ベースのプロトコルだということであり、したがって、ユーザがメッセージをＳＴＢからアプリケーション・サーバに送信しても、アプリケーション・サーバは応答しなくてもよい。つまり、ＳＴＢとサービス・プロバイダのメッセージとの間に１対１の対応関係がない。信頼性のないＤＡＴＰメッセージのクラスを除くすべてのＤＡＴＰが信頼性のあるＤＡＴＰ層を通じて処理される。すべてのＤＡＴＰメッセージには、トランザクションの基盤として使用できる一意の識別子が設定される。
【００２４】
ＤＡＴＰを使用したトランザクション、たとえばＨＴＴＰ要求では、ＳＴＢはＷｅｂページを要求するＤＡＴＰメッセージをＳＧＷに送信する。ＳＧＷは、ＬＨＴＴＰをＨＴＴＰに変換し、Ｈ２Ｏを介してインターネットに送信する。Ｗｅｂページを含む応答がコンテンツを変換するＨ２Ｏを介してインターネットからＳＧＷに返されると、ＳＧＷはＬＨＴＴＰ　ＤＡＴＰメッセージをＳＴＢに送信し、要求されたＷｅｂページのコンテンツをＳＴＢに返す。トランザクションの別の実施例として、ＳＴＢから送信されるＦｅｔｃｈｍａｉｌがある。Ｆｅｔｃｈｍａｉｌ要求は、ＤＡＴＰメッセージ内にカプセル化される。ＤＡＴＰメッセージの上でＤＡＭＬが使用される。ＤＡＭＬは、ＸＭＬのドメイン固有のインスタンスである。
【００２５】
そこで、ＳＴＢはＤＡＴＰメッセージをＤＡＭＬ（ＸＭＬ）要求を含むＦｅｔｃｈｍａｉｌに送信する。Ｆｅｔｃｈｍａｉｌは、ＤＡＴＰメッセージを読み込み、メッセージからコンテンツを抽出し、そのコンテンツをアプリケーション・サーバに渡し、そのサーバでトランザクションを処理して、メッセージをＦｅｔｃｈｍａｉｌに返す。次に、Ｆｅｔｃｈｍａｉｌは、要求された内容を含むＤＡＴＰメッセージをＳＴＢに送信する。
【００２６】
ＳＧＷはさらに、ユーザ注文要求にすばやく反応しながら、複数のユーザから送られた多数の注文のピークに応じられるストア・アンド・フォワード機能も備える。ＳＧＷは、ユーザの注文への応答としてユーザに「注文アクノリッジ」を送信し、注文トランザクションを実際に処理するアプリケーション・サーバに後で送信するためその注文を格納する。後で注文送信することにより、多数の注文を時間をかけて分散させるため、アプリケーション・サーバに一度に送信しなくてよくなる。したがって、帯域幅を効率よく使用できる。ＤＡＴＰ／ＳＧＷはさらに、時間に対するメッセージ断片のシーケンス番号に基づくスライディング・リジェクション・ウィンドウを備える。ＤＡＴＰ／ＳＧＷについて、以下で詳述する。
【００２７】
サービス・プラットフォーム
そこで図１を参照すると、本発明のＳＧＷが配置されるＳＰがこの図に示されている。ＳＰ　５０は、コンテンツ変換２０４、トランザクション制御／ビジネス機能１０６、およびトランスポート変換１０８という３つのカテゴリに大まかに分けられるアプリケーションのグループからなる。ＳＰにより、サービス２００はクライアント２１２とやり取り（対話）することができる。サービス２００は、通信リンク１０２を通じてＳＰ　５０と通信する。次に、ＳＰ　５０がクライアント２１２と交信する。クライアント２１２はＳＴＢ、デジタル・アシスタント、携帯電話、または通信リンク２１０を通じてＳＰと通信できるその他の通信デバイスとすることができる。コンテンツ変換２０４およびトランスポート変換１０８サービスは、トランスポートおよび通信機能を備え、ビジネス機能サービスはビジネス制御機能を備える。
【００２８】
図２は、サービス・プラットフォーム５０の好ましい実装の一実施例を示している。サー

ビス２００は、インターネットまたは他のネットワークまたはネットワーク・オペレータからアクセス可能な通信チャネルを介してショッピング、チャット、およびその他のサービスを提供する。ネットワーク・オペレータは、ＳＰを使用して、これらのサービスにアクセスする。サービス・マネージャ２３８を含むビジネス機能２０６は、カルーセル・マネージャ２５４とやり取りして、サービス２００からコンテンツを取り出す。カルーセルは、ＳＰ　５０からクライアントへのオーディオ／ビデオ／インタラクティブ・データ放送の反復ストリームを含む。カルーセル・マネージャ２５４、トランザクション・マネージャ２４２、およびサービス・マネージャ２３８は、放送カルーセルにコンテンツを挿入したり、削除したりする操作を制御する。サービス・コンテンツが取り出され、Ｈ２Ｏ　２４８によりＳＰに適した形式に変換される。Ｈ２Ｏ　２４８は、コンテンツ変換２０４の可能な一例である。Ｈ２Ｏは、ＨＴＭＬコンテンツをＳＰ／クライアント読取り可能コンテンツに変換する。変換されたコンテンツは、データ・カルーセルにフォーマットされ、オープン・ストリーマ２５６により多重化され、クライアント２１２に放送される。クライアント２１２は、サービスとやり取りし、必要ならば、ＳＰおよびサービス２００と通信する。ＰＴＰの通信は、ＳＧＷ　２４６を通る。ＳＧＷ　２４６は、ＳＴＢ　ＤＡＴＰプロトコルをプラットフォーム・ビジネス・エージェント２２６とＨ２Ｏ　２４８が認識する標準通信プロトコルに変換するトランスポート変換を実行する。負荷分散機能２３６は、ビジネス機能２０６、カルーセル・マネージャ２５４、およびＳＧＷ２４６とやり取りし、放送リンク２４１とＰＴＰ通信リンク２１０との間の最適な負荷分散を決定する。ビジネス機能２０６は、プラットフォーム・ビジネス・エージェント２２６とやり取りして、サービス２００とクライアント２１２の間のアクセスおよび情報交換を制御する。

【００２９】

サービス２００はネットワーク・オペレータとネゴシエーションを行って、オペレータのサービス・プラットフォームを介して受信契約者にサービスを提供する。図３に示されているように、ネットワーク・オペレータは、サービス・マネージャ２３８を使用してサービスおよびそのサービスと関連するネゴシエーション済みビジネス規則２２２（例えば、スケジュール、帯域幅要件、視聴者情報へのサービス・アクセス）を登録する。サービス・マネージャ２３８は、サービス・データ２１６（例えば、ＵＲＬアドレス、コンテンツ）を格納する。ビジネス規則２２２およびサービス・データ２１６に基づいて、サービス・マネージャ２３８は放送通信２３４機能と通信して、コンテンツ・プロバイダからコンテンツを取り出す。

【００３０】

コンテンツは、サービス２００から取り出されると、コンテンツ変換２０４とコンテンツ・フィルタ２２４により処理され、クライアント・デバイス２１２に適した形式に変換される。放送２３４機能により、コンテンツが放送２３４ネットワークに適した形式に変換される。変換されたコンテンツは、放送リンク２４１を介してクライアント２１２に届く。クライアント２１２およびサービス２００は、トランスポート変換２０７の一部であるポイントツーポイント・リンク２１０およびポイントツーポイント機能２３２を介してやり取りする。サービス２００は、ショッピング、オーディオ／ビデオ、ゲーム、投票、広告、メッセージ送受信、またはその他のサービスを含む。

【００３１】

クライアント２１２は、ポイントツーポイント２３２通信リンクを通して、サービス・プラットフォーム５０およびサービス２００と通信する。負荷分散機能２３６は、ビジネス機能２０６とやり取りし、放送２３４通信リンク２４１とポイントツーポイント２３２通信リンク２１０の間の最適な負荷分散を決定する。プラットフォーム・ビジネス・エージェント２２６は、ビジネス規則２２２を使用して、サービス２００とクライアント２１２の間の情報のやり取りおよび交換を制御する。例えば、ネットワーク・オペレータは選択により、サービス２００がユーザ情報にアクセスするのを禁止することができる。サービス２００は、ユーザ情報にアクセスするにあたってビジネス規則２２２およびサービス・データ２１６に基づく料金を支払わなければならない。

【００３２】
視聴者マネージャ２４０は、クライアント／ユーザ情報をユーザ・データ２２０に格納する。ビジネス・エージェント２２６は、サービス２００への視聴者情報の流れを制御する。トランザクション・マネージャ２４２は、サービス２００とクライアント２１２との間で交換されるトランザクション情報を記録する。広告マネージャ２４４は、ビジネス規則２２２およびユーザ・データ２２０に基づいて、放送２３４リンク２４１およびポイントツーポイント２３２リンク２１０を介してクライアントに提示される広告と広告の種類を決定する。サービス・プラットフォーム・トランザクション・マネージャは、トランザクション・データベースにすべてのトランザクションを記録し、正確なオペレータ収益回収（ＳＴＢがオフになっている場合でも）および視聴者プロファイル１６２および視聴者カテゴリ１６０内の受信契約者プロファイル（視聴者購入および視聴習慣）を確実なものとするが、これにより、ネットワーク・オペレータは付加価値データを得られる。

【００３３】
トランザクション・ログも、累積的ユーザ・プロファイルの生成のためユーザの視聴およびトランザクション・データをマイニングするのに役立ち、また協調フィルタリングなどのより高度なプロファイリング手法にも使用される。視聴者またはクライアントは、視聴者ユーザ・プロファイルに基づき１つまたは複数のカテゴリ（例えば、「スポーツ・ファン」、「フランス人シェフ」）に分けることができる。カテゴリを使用することで、ネットワーク・オペレータが視聴者／クライアントの長期および短期視聴および購入動向に基づき適応型ターゲット広告作成および放送を行うことができる。

【００３４】
サービス・プラットフォームは、財布機能を備えるが、これはチェックアウト／購入機能を持つ。この財布機能は、サービス・プラットフォームによってサポートされているが、選択されたショップまたはサービスは自分のチェックアウト・プロシージャに有利なようにこの財布機能をバイパスすることもできる。財布機能は、視聴者プロファイル、視聴者カテゴリ、およびトランザクション・ログに関するデータを記録し、そのようなデータにアクセスする。そこで、財布機能により、ユーザによる入力データのタイピングが最小限に抑えられる。この機能は、特に、ユーザがデータ入力容量に制限のあるＴＶリモート・コントロールで注文するときに役立つ。通常、財布機能向けのコンテンツは下４桁を取り除いたクレジット・カード番号であり、ユーザは確認のためその下４桁を補うか、または出荷先住所を入力するだけでよい。

【００３５】
オプションにより、すべてのユーザ情報が財布機能に入れられ、サービスから隠される。財布情報は、サービス・プラットフォームとクライアントの両方に存在する。サービス・プラットフォーム上の財布情報に、クライアント出荷先住所、クライアント完全クレジット・カード番号など、できる限りの情報を収めるのが好ましい。ＳＴＢ内の財布情報には、どのクレジット・カードで購入を行ったかを視聴者に覚えておいてもらうためのクレジット・カードの下４桁などの部分情報のみを含めることができる。部分情報は、ＳＴＢデータにアクセス可能なＳＴＢを使用する信頼できない人および信頼できないアプリケーションから保護するためにＳＴＢ内に格納される。

【００３６】
視聴者がクリックして電子商取引アプリケーションまたはサービスの呼び出しがトリガされたときにリンクが発生すると、サービス・プラットフォームは、受信契約者のナビゲーション位置を決定して、それをトランザクション・データベース１５８に記録する。サービス・プラットフォームはさらに、視聴者がストアへのリンクを辿ったとき、または視聴者が購入（「衝動買い」と呼ぶ）を決定したときに見ていた番組を決定し、記録する。トランザクション・データベース２５８を使用すると、オペレータは詳細なコンテキストおよび購入履歴を格納し受信契約者に提供することができる。さらに、コンテキストおよび購入履歴のこのような格納は、受信契約者プロファイルおよびカテゴリ情報を改善するのに役立ち、追加収益を発生しかつ／または電子商取引プロバイダとのチャンネル取引の一

部となることができる。

【００３７】

サービス・プラットフォームを使用することで、ネットワーク・オペレータは自ネットワークへの電子商取引の配備を円滑化し、電子商取引収益の分け前を捕捉できる。サービス・プラットフォーム・パーソナル財布機能を使用すると、ネットワーク・オペレータはクレジットを管理し、支出限度を課し、少額決済を有効にすることができる。サービスはサービス・マネージャおよび視聴者マネージャとやり取りして、視聴者プロファイルおよびカテゴリに基づいてクライアントに提示するために、サービス・プラットフォームが選択するサービスのグループを提供することができる。オペレータは、逆に、サービスに対してその受信契約者プロファイルを知らせて、受信契約者への専用視聴者ターゲット提供物および広告を要求することができる。

【００３８】

視聴者マネージャ２５２は受信契約者／ユーザ登録２６４、好み、およびプロファイル情報２６２を管理する。視聴者マネージャ２５２により、ユーザはデータベース内の個人情報を登録し記録することができる。個人情報には、視聴パターン、販促に関する好み、個人、財布、および人口統計情報などが含まれる。この記録された情報およびユーザの活動に基づいて、視聴者マネージャ２５２はプロファイル情報を生成し、ユーザ・プロファイル２６２および予想される好みおよびニーズに合わせてユーザをカテゴリに分類し、ターゲット・サービス、コンテンツおよび広告を製作する。さらに視聴者マネージャ２５２は、サービスおよび視聴者パラメータの一括更新も実行する。

【００３９】

ＳＧＷおよび支援機能を介して、サービス・プラットフォームを使用することで、ネットワーク・オペレータが視聴者データベースへのアクセスを制御し、ネットワーク・オペレータと特権情報（例えば、クレジット・カード番号、視聴者の実際の氏名、家の住所、電話番号、社会保険番号など）にアクセスできる契約を事前に交わしているサービス・プロバイダのみを許可するようにできる。分散機能については、つまり、クライアントが十分な処理能力と記憶容量を備えている場合、視聴者マネージャ２５２により、クライアント・デバイスに格納されている個人およびプロファイル情報にアクセスすることができ、またクライアント・デバイスがユーザの好みのコンテンツを選択するようにできる。クライアントは、クライアント・デバイス（例えば、ＳＴＢ）内のビジネス・フィルタを介してユーザの好みのコンテンツを選択する。

【００４０】

視聴者マネージャ２５２は、家庭／受信契約者／ＳＴＢ（またはその他のクライアント・デバイス）識別および認証を行うことができ、サービス・ゲートウェイおよび親制御機能をサポートする。視聴者マネージャ２５２は、ニックネームおよび個人識別番号を介した単一ＳＴＢでの複数視聴者識別および登録認証をサポートする。視聴者識別子は、クライアント・デバイス識別番号から導くのが好ましい。視聴者マネージャ２５２は、ＳＧＷをサポートする観察されている累積ＴＶ視聴および購入習慣にリンクされているロギング、生成、および組み合わせ決定を通じて家庭および個人の視聴者プロファイリングを提供する。視聴者マネージャは、サービス・プラットフォームとＳＴＢとの間の分散データ捕捉および格納をサポートしており、また双方向同期もサポートする。視聴者マネージャ２５２は、すべてのサービス・プラットフォーム・アプリケーションの間で分散プロファイルを使用することができ、外部ＳＭＳ／ＣＲＭとの同期処理を行う。

【００４１】

視聴者マネージャ２５２は、ＳＴＢまたはその他のクライアント・デバイスへのニックネーム、フルネーム、およびＰＩＮ格納を含む抽象視聴者識別子を使用した単一のＳＴＢすなわちクライアント・デバイスに対する複数の視聴者登録が可能である。ビジネス・エージェント２２６は、サービス・プロバイダと視聴者との間のやり取りを行うためのトランザクション・ビジネス規則を強制する。ネットワーク・オペレータによって定められているビジネス規則に基づき、またサービス・プロバイダとの契約に基づき、ビジネス・エー

ジェント２２６はユーザ情報へのトランザクションおよびサービス・プロバイダによるアクセスを制御する。ビジネス・エージェント２２６は、トランザクションの実行時に視聴者情報の挿入、置き換え、および削除を行う。

【００４２】

ビジネス・エージェント２２６はＳＧＷ２４６とともには、受信契約者とサービス・プロバイダとの間のセッションを確立する。ＳＧＷ／ビジネス・エージェント２２６は、視聴者情報詳細へのアクセスを制御し、サービス・プロバイダに提示される視聴者情報の取り込み、置き換え、および削除により視聴者情報を操作することができる。ＳＧＷ／ビジネス・エージェントは、デフォルト値を定め、ユーザ情報へのアクセスを制御する。ＳＧＷ／ビジネス・エージェントはさらに、トランザクション・ロギング、メッセージ・ロギング、負荷／トランザクション監視を実行する。

【００４３】

広告キャンペーン管理では、視聴者データ・マイニングおよび分析システムを使用して、製品、広告、および放送のタイミングの最良の選択を提案することができる。サービス・プラットフォームは、規則ベースのシステムを備え、「スマート」広告キャンペーンを作成する。これらのキャンペーンは、ユーザの好み、プロファイル、購入および表示習慣、および人口統計調査に基づく適応型である。広告マネージャは、広告コンテンツ・データベース、キャンペーン規則データベース、サービス・マネージャ、およびカルーセル・マネージャから得られる情報に基づき、視聴者に掲示する最良の製品を決定する。これは、カルーセル・マネージャが放送カタログを再構築するのをトリガする。広告マネージャはさらに、ビジネス・エージェントとインターフェイスし、視聴者がオンラインになっている間、視聴者に提示する広告コンテンツを提案する。

【００４４】

オープン・ストリーマは、広告をＮ個のサービス・プラットフォーム・カルーセルとして、トランスポート・ストリーム毎に１つずつパッケージし、帯域幅の使用を最適化する。ＳＴＢクライアント・アプリケーションは広告ライブラリとともに放送される。このライブラリは、キャンペーン獲得、組み合わせ決定、追跡、レポート作成の機能を実行する。キャンペーン獲得クライアント・コンポーネントは、クライアント・アプリケーションと並列に実行され、キャンペーン・カルーセルを監視し、情報をキャッシュし、資産を事前フェッチする。組み合わせ決定クライアント・コンポーネントは、ローカル・パラメータ（表示されるページの種類、ユーザ情報、キャンペーンを打った回数など）でそれぞれの広告キャンペーンを評価し、最良の広告にアクセスして表示する。

【００４５】

本発明が利用されるサービス・プラットフォームは、インタラクティブ・テレビ環境におけるコンテンツ、広告、メッセージング・サービス、電子商取引、およびテレビ電子商取引（Ｔ－Ｃｏｍｍｅｒｃｅ）の規制のための包括的収益ソリューションを規定するシステム・アーキテクチャを備える。本発明が属するサービス・プラットフォーム収益ソリューションは、ネットワーク・オペレータが制御し、業者、サービス・プロバイダ、ネットワーク・オペレータ、およびサービス・プラットフォーム・プロバイダが最適な形で収益業務に加わることができる。サービス・プラットフォームは、ネットワーク・オペレータ、ソリューション・プロバイダ、およびサービス・プロバイダ向けの新しい収益ストリームを生み出すことができる中央集中構造を持つ。

【００４６】

ＳＧＷを使用することで、ネットワーク・オペレータによって、かつネットワーク・オペレータの制御に従ってのみ視聴者情報をサービスに送らなければならないようにすることで、ＳＰがヘッドエンド・オペレータの貴重な受信契約者プロファイル・データベースを隠すことができる。受信契約者の識別情報を保護するために、サービスがトランザクション詳細をＳＰに送信するセッションで、抽象化されたユーザ識別子（つまり、セッション識別子）をサービスに送信する。ユーザ識別子はセッション固有のものである。家族が同じＳＴＢを使用する場合のように、クライアントと関連する複数のユーザ識別子がありえ

る。家族の各メンバおよび家庭ＳＴＢは、ＳＰ視聴者マネージャにより個別に視聴者識別子、カテゴリが割り当てられ、購入／映画要求／視聴習慣／などに関するトランザクションに関して追跡され、プロファイルが作成される。ＳＧＷで視聴識別子を利用できるようになる。サービス・プロバイダは、セッション識別子を通じてのみクライアントまたはＳＴＢ識別子を知る。ネットワーク・オペレータまたはヘッドエンド・オペレータのみが、ＳＧＷを使って、セッション識別子を注文の履行に必要な視聴者情報詳細（氏名、住所、出荷情報など）とすることができる。オペレータ側でクレジット・カード集金またはその他のトランザクションの実行を望んでいない場合には、クレジット・カード番号またはその他の情報については例外とすることができる。

【００４７】

本発明は、ネットワーク・オペレータが視聴者情報データベースへのアクセスを制御し、ネットワーク・オペレータと特権情報（例えば、クレジット・カード番号、視聴者の実際の氏名、家の住所、電話番号、社会保険番号など）にアクセスできる契約を交わしているサービス・プロバイダのみを許可するようにできる。視聴者マネージャ２５２は、クライアント・デバイスに格納されている個人情報およびプロファイル情報にアクセスすることができ、またクライアント・デバイスまたはＳＰは視聴者プロファイルに格納されている視聴結果に基づいてユーザのお好みのコンテンツおよび購入習慣を選択することができる。クライアント、ＳＧＷ、またはＳＰは、クライアント、ＳＧＷ、または他のＳＰコンポーネントによりクライアント・デバイス内でアクティブにされているビジネス・フィルタを介して視聴者プロファイリングに基づきユーザのお好みのコンテンツを選択する。

【００４８】

視聴者マネージャ２５２は、家庭／受信契約者／ＳＴＢ（またはその他のクライアント・デバイス）識別および認証を行うことができ、ＳＧＷおよび親制御機能をサポートする。視聴者マネージャ２５２は、ニックネームおよび／または個人識別番号（ＰＩＮ）、さらにクライアント・デバイス識別番号、トランザクション履歴、視聴者プロファイル、ニックネーム、および個人識別番号から得られる視聴者識別子を使用して単一ＳＴＢでの複数視聴者識別および登録認証をサポートする。視聴者マネージャ２５２は、観察されている累積ＴＶ視聴および購入習慣にリンクされているロギング、生成、および組み合わせ決定を通じて家庭および個人の視聴者プロファイリングを実行する。視聴者マネージャは、ＳＰとＳＴＢとの間の分散データ捕捉および格納をサポートしており、また双方向同期もサポートする。

【００４９】

視聴者マネージャ２５２は、すべてのＳＰアプリケーションの間で分散プロファイルを使用することができ、外部ＳＭＳ／ＣＲＭとの同期処理を行う。視聴者マネージャ２５２は、ＳＴＢまたはその他のクライアント・デバイスへの仮名またはニックネーム、フルネームおよびＰＩＮ格納を含む抽象視聴者識別子を使用した単一のＳＴＢすなわちクライアント・デバイスに対する複数の視聴者登録が可能である。ビジネス・エージェント２２６は、サービス・プロバイダと視聴者との間のやり取りを行うためのトランザクション・ビジネス規則を強制する。ネットワーク・オペレータによって定められているビジネス規則に基づき、またサービス・プロバイダとの契約に基づき、ビジネス・エージェント２２６はユーザ情報へのトランザクションおよびサービス・プロバイダによるアクセスを制御する。ＳＧＷをサポートするビジネス・エージェント２２６は、サービス・プロバイダの契約と抽象セッション識別子に基づいてトランザクション中に聴取情報の補足、追加、置き換え、および削除を行う。

【００５０】

ビジネス・エージェント２２６は、受信契約者クライアントとサービス・プロバイダとの間のセッションを確立する。ビジネス・エージェント２２６は、視聴者情報詳細へのアクセスを制御し、サービス・プロバイダに提示される視聴者情報の取り込み、置き換え、および削除により視聴者情報を操作する。ビジネス・エージェント２２６は、デフォルト値を定め、ユーザ情報へのアクセスを制御する。ビジネス・エージェント２２６はさらに、

トランザクション・ロギング、メッセージ・ロギング、負荷／トランザクション監視を実行する。

【００５１】

広告マネージャ２４４は、放送とＰＴＰリンクの両方とのインターフェイスを備え、２つの配信チャネル間の相補的な広告のやり取りを行えるようにする。例えば、放送（プッシュ）広告では、ＳＰＳを介して広告サービスへのＰＴＰ接続がトリガされ、ユーザは製品を購入したり、製品に関係する詳細な情報を取得したりすることができる。また、放送広告をＰＴＰコンテンツに入れて、放送サービスを利用できることをユーザに知らせることもできる（例えば、インフォマーシャル）。

【００５２】

場合によっては、クライアントが情報を要求しなくても、複数の製品または広告セグメントがクライアントにプッシュまたは放送される。クライアントに関連するビジネス・フィルタはＳＴＢに置かれるのが好ましいが、これを使用して、ユーザ・プロファイルに基づき視聴者にぴったりの広告を選択する。例えば、ＳＰは、料理番組のときに視聴者に放送する料理広告のグループをスケジュールすることができる。この広告グループには、イタリア料理、フランス料理、インド料理、およびドイツ料理に料理広告が含まれる。ＳＧＷは、視聴者プロファイルに基づきクライアントに提示する料理広告の種類が選択されるように、ＳＴＢすなわちクライアントと関連するまたはそこに配置されているビジネス・フィルタを設定する。視聴者プロファイル、ユーザの好み、および／またはクライアント・プロファイルに基づき、ＳＧＷ、クライアント、またはＳＰによって設定されたＳＴＢフィルタに応じて、ある視聴者はフランス料理の広告を見ることができるが、他の視聴者はインド料理の広告を見ることができる。

【００５３】

ＳＰでは、Ｗｅｂ商取引インフラストラクチャを再利用することができる。ＳＰ内に置かれているＳＧＷは、「通常の」ＨＴＭＬテンプレートをＳＰ対応形式で置き換える。ビジネス・エージェントは、ＳＧＷを通じてＳＴＢすなわちクライアントから注文要求を受け取る。ＳＧＷがメッセージをキューに入れると（ピークの管理のため）、一部の注文が遅延を伴ってビジネス・エージェントに届く（どのような形式の確認も必要としない注文であればこの方式を使用するのが好ましい）。ＳＧＷビジネス・エージェントは、視聴者情報を注文に追加する。注文／メッセージに記載されている視聴者情報の量と種類は、サービス／小売り契約に応じてビジネス規則により決められる。

【００５４】

サービスと視聴者／クライアントとの間の通信のように、情報はトランスポート・ストリーム毎にカルーセルを単一として別々のカルーセルに送信されるか、または既存のアプリケーション・カルーセルにまとめられる。その後、注文処理が進行するが、必要に応じて、ＳＰに備えられている「クレジット・カード精算」機能を使用することができる。確認が小売り業者から送り返されると、注文がリアルタイムでユーザに送り返されるか、電子メールでユーザに送信されるか、またはＳＧＷを通じてオンデマンドで入手できるようにされる。

【００５５】

ＳＰはさらに、ＳＧＷを介して、オフライン視聴者識別（ＯＶＩ）を実行し、オンライン視聴者接続を確立することなく視聴者を識別または認証することができる。したがって、接続遅延（例えば、１０～４０秒）を購入プロセス内の最も適切な場所に設定することができる。さらに、ストア・アンド・フォワード機能とともに視聴者の識別も行える。ＯＶＩにより、オンとオフを間欠的に繰り返すクライアント・デバイスで注文／オペレーションの通信および完了を行える。

【００５６】

ＳＰ／ＳＧＷで提供されるテレビ商取引サービスを視聴者が利用して、オンラインでなくても、品目を注文フォーム（ショッピング・カート）に追加することができるオフライン注文フォーム機能を備える。ストア・アンド・フォワード機能により拡張性を高めること

ができる。ストア・アンド・フォワードは、ピーク外の時間に転送するか、または単純に、トランザクションが開始した後所定の期間に負荷を散らすことができる。完全ストア・アンド・フォワード・ソリューションが組み込まれており、いつでも任意のチャネルから応答を転送することができる。ストア・アンド・フォワードは、機能強化された電子商取引、テレビ商取引トランザクションに使用できる。オフライン視聴者認証により、オフライン支払い選択を実行できる。オフライン支払い選択がＳＰ／ＳＧＷに用意されており、購入プロセスを改善し、ストア・アンド・フォワード機能をテレビ商取引／電子商取引とともに使用できるようになっている。

【００５７】

ＳＰ／ＳＧＷは、状況に応じて標準のＷｅｂトランスポートを使用する、つまり、リアルタイム要求についてはＨＴＴＰを使用し、状況により非同期通信の場合にはＳＭＴＰを使用する（たとえば、業者報告、ストア・アンド・フォワード）。オンラインになったときでも、ＳＰは短時間だけ接続し、データ（例えば、電子メール）にアクセスし、それからローカルでそのデータを使用することができる。ＳＰ／ＳＧＷは、オペレータ視聴者データベースを保護するため、通常のＷｅｂ　ｃｏｏｋｉｅではなくセッション・ベースの識別子を用意している。ＳＰ／ＳＧＷは、Ｗｅｂ　ｃｏｏｋｉｅの代わりに、ユーザを識別するためにサービスで使用できない、セッションでのみ使用できるセッション・ベースの識別子を備えている。サービス側では、ＳＧＷに視聴者情報を要求しなければならない（そして、ネットワーク・オペレータによって課金されなければならない）。

【００５８】

ＳＰ／ＳＧＷは、オプションにより、視聴者に、いつ接続が行われたを通知し、またオプションにより、接続を維持する視聴者の承認を求めることもできる。ＳＰはさらに、視聴者の画面に「Ｃｏｎｎｅｃｔｉｏｎ　ＯＮ」というステータスを表示する。ＳＰは、効率が高ければＰＴＰ通信の放送用帯域幅を使用する。どの情報を放送に載せ、どの情報をＰＴＰ接続で送るかを決める負荷分散機能が用意されている。負荷分散決定は、データの緊急性、ＰＴＰ伝送リンクに対する放送の配信待ち時間、放送およびＰＴＰ経路に対する比較した負荷、およびデータを受け取る視聴者の数に基づいて行われる。一般に、多数の視聴者に送られるデータは放送で送られ、即座に送る必要のある少量のデータはＰＴＰリンクで送られる。ブロードバンド・チューナーを持たないＳＴＢは、ブロードバンドとともに送出されたＰＴＰメッセージを受信する。

【００５９】

図２によると、本発明は、コンテンツ（例えば、番組、広告）をクライアント・デバイス２１２（例えば、セットトップ・ボックス）に引き込むために使用する放送ストリームの選択を最適制御する方法である。「コンテンツ」とは、通常ではインターネット接続からは得られないテレビ用コンテンツを意味する。インタラクティブ・テレビ環境では、ヘッドエンド・サーバ５０およびセットトップ・ボックス２１２は、コンテンツを転送するのに複数の放送用チャネルを利用できる。例えば、ヘッドエンド・サーバ５０がコンテンツを放送カルーセル・ストリームに入れ、選択したチャネル２６１（つまり、地上波、ケーブル、衛星放送アンテナ）で転送し、セットトップ・ボックス２１２がこのコンテンツをこのストリームから引き出すことができる。戻りチャネル２１０（例えば、ＰＴＰ）で、セットトップ・ボックス２１２がクライアント／サーバ方式でヘッドエンド・サーバ５０と通信するが、セットトップ・ボックス２１２はヘッドエンド・サーバ５０にコンテンツを要求する。

【００６０】

本発明では、ファイルのすべてのサイズおよび要求の時刻に関する要求を遂行する際の待ち時間の記録を取る。これにより、ファイル要求のわずかな部分、好ましくは５％未満を利用可能なすべてのチャネルにランダムに分散させることができる。ファイル・サイズは待ち時間への寄与要因であるため、この方法では他のチャネルから類似のサイズのファイルを引き出せるのが好ましい。例えば、２００ｋＢのファイルは、戻り（ポイントツーポイント）チャネル２１０と放送ストリーム２６１の両方から引き出すことができる。所定

の時刻、例えば、午後５時に、ポイントツーポイント接続２１０の平均待ち時間は１．５秒であるが、放送ストリーム２６１の平均待ち時間は０．５秒である。しばらくして、例えば午後９時に、これらのチャネルで同じサイズのファイルが引き出される。次に、２００ｋＢのファイルでは、ポイントツーポイント接続２１０の平均待ち時間は１．０秒であり、放送ストリーム２６１の平均待ち時間は５秒となる。このような結果であれば、セットトップ・ボックス２１２は午後５時に放送ストリーム２６１からコンテンツのうち多くを取り出し、午後９時にポイントツーポイント接続２１０からそのコンテンツの多くを取り出すことになる。

【００６１】

生成された待ち時間データは、特定の時間の間保持され、その後、古くなったと判断されると解放される。時間枠は、許容可能なレベルの統計サンプル点が得られる十分な長さ、かつ使用しているサンプル点が最近のもので、現在時刻に関連するようにできる十分な短さとなるように選択される。古すぎるサンプル点は、次の統計計算反復の前に、サンプル・グループから取り除かれる。

【００６２】

放送ストリーム２６１の待ち時間は、主に、反復率により決定され、ネットワーク・トラフィックの影響を受ける場合がある。放送側では、ヘッドエンド・サーバ５０が、このサーバがデータ・リソースを放送するのか、それとも受信機がデータ・サーバから戻りチャネル２１０でそのデータ・リソースを取得するかを決定するフィードバック・メカニズムを実装する。各クライアント・デバイス２１２およびデータ・サーバは、使用するあるいは提供する異なるデータ・リソースに対する需要変化を追跡する。データ・リソースの需要が低下すると、ヘッドエンド・サーバ５０は自動的に、カルーセル内のリソースの出現頻度を下げる（ＩＰ戻りチャネル接続および関連するデータ・サーバにその需要の負荷を任せる）。リソースの需要が上昇すると、ヘッドエンド・サーバ５０は自動的に、カルーセル内のリソースの出現頻度を上げる（戻りチャネル２１０および関連するデータ・サーバの負荷を減らす）。本発明はさらに、送信のビット・レートを制御することによりメッセージ・フロー・レートを制御するメッセージ・フロー・レート・コントローラを提供する。

【００６３】

ＳＰは、ＳＴＢおよび／またはクライアントのために視聴者プロファイリングに基づいて放送経路内の情報を選択的に受信するフィルタを備えており、特定のフィルタがＳＴＢ内に設定されている選択された視聴者のみが放送ストリームのコンテンツ（広告、情報、またはＡ／Ｖ番組など）をキャプチャするようになっている。これらのフィルタは、ＳＰの適応および選択的配信の側面を強化するものである。カルーセル・マネージャは、オープン・ストリーマ用のデータ・カルーセルを備えている。カルーセル・マネージャは、データのカルーセルをリアルタイムで管理する。カルーセル・マネージャは、オープン・ストリーマを補完する。カルーセル・マネージャは、サーバ・コンポーネントとＳＴＢクライアントＯＣＯＤライブラリを備える。カルーセル・サーバが、カルーセル・コンテンツへの追加、削除、または他の何らかの変更を行う要求をアプリケーションから受け取る。カルーセル・マネージャは、要求を受け取ると、それを単一トランザクションとして取り扱い、必要なすべてのデータを（通常は、ＨＴＴＰを介して）取得する。カルーセル・マネージャは、必要に応じて新しいカルーセル・インデックスまたはカルーセル・ディレクトリ・ファイルを生成する。カルーセル・マネージャは、更新されたカルーセル・ディレクトリをオープン・ストリーマに公開し、それによりオープン・ストリーマの放送優先度およびトラックを制御する。

【００６４】

オープン・ストリーマは、ネットワーク・オペレータがＳＰアプリケーションおよびデータをその放送網で放送するためのソフトウェア／ハードウェア製品である。オープン・ストリーマを使用することにより、ネットワーク・オペレータＡ／Ｖ番組と同時にＳＰデータおよびアプリケーションを送信することができる。オープン・ストリーマでは、データ

・ストリームをリアルタイムで更新し、Ａ／Ｖコンテンツと一致するようにできる。例えば、ネットワーク・オペレータは、スポーツ・イベントのライブ放送とともにインタラクティブ・スポーツ・アプリケーションを放送することができる。オープン・ストリーマは、共通サーバＤＬＬおよび放送ストリーマの２つのコンポーネントを含む。アプリケーション・サーバ（例えば、天候アプリケーション・サーバ）またはＳＰ内のカルーセル・ビルダーは共通サーバＤＬＬを呼び出して、カルーセル・データを放送ストリーマに送信する。放送ストリーマは、次に、アプリケーションとＡ／Ｖデータの多重化（コード／データ比およびビット・レート要件に従って）を実行し、多重化されたそのデータを放送のため放送機器に送信する。

【００６５】

ＤＡＰ／ＤＡＴＰプロトコル方式の概要

本発明を利用すると、ＤＡＴＰを使用するＳＴＢとＳＧＷを介して標準プロトコルを使用するサービス・プロバイダとの間の通信が可能になる。ＤＡＴＰプロトコルは、エンティティがメッセージを他のエンティティに送信するとともに配信保証のあるメッセージ・ベースのプロトコルである。ＳＴＢは、メッセージをＳＧＷに送信するときにはいつでも、メッセージがその最終の宛先に到達すると受領メッセージを受け取る（ＳＧＷはアプリケーション・サーバの機能を持つ）。アプリケーション・サーバによってメッセージが処理されると、ＳＴＢとＳＧＷとのセッションがまだ開いている場合に、応答メッセージをＳＴＢに送信することができる。ＤＡＴＰメッセージ伝送段階の前にＤＡＴＰログイン段階にがあり、その後にＤＡＴＰセッションを確立するために必要なＤＡＴＰログアウト段階がある。ＤＡＴＰは、セッション指向のプロトコルである。図１０は、ＤＡＴＰセッションの単純な例を示している。

【００６６】

ＤＡＴＰは、同じＳＴＢトランスポート層接続の上の複数のセッションをサポートする。ＳＴＢクライアントは、ＳＧＷとのセッションが開いている間に、最初のセッションで使用されているのと同じＳＴＢトランスポート・リンク上で新しいセッションを開始するログイン・パケットを送信することができる。ＳＴＢクライアントとＳＧＷ内の両方のＤＡＴＰセッション管理モジュールにより、同じリンク上でさまざまなセッション・メッセージが多重化される。

【００６７】

ＤＡＴＰパケット・コンテンツの概要

ＤＡＴＰプロトコル・パケットは、固定サイズのヘッダ、可変サイズのデータ・ペイロード（ＤＡＭＬメッセージ）、およびトレーラを含む。ヘッダは、プロトコル・バージョン番号、パケット種別（ログイン／ログアウト・ハンドシェーク、Ｐｉｎｇ、データ、アクノリッジなど）、実際のパスポート情報（Ｒａｗ、ＴＣＰ／ＩＰ、ＵＤＰなど）、メッセージ・シーケンス番号（ＳＴＢまたはＳＧによって生成されるＤＡＴＰメッセージ番号）、サービス識別子（データを受信するためのサービスのＩＤ）の要素を含む。サービスＩＤは、ＤＡＴＰプロトコル内で定義されている８ビット識別子である。さらに、セッションＩＤ（セッションＩＤはハンドシェーク時にＳＧＷによって与えられる）、暗号化されたセッション用の暗号化フラグ、およびペイロード・データ・サイズもある。

【００６８】

ペイロード・データには、パケット種別によって、ハンドシェイク・パケット用のログイン／ログアウト情報、アクノリッジ・パケット用のアクノリッジ情報、データ／パケット用のデータ・ペイロードを含めることができる。トレーラには、ＤＡＴＰパケットの少なくとも３２ビットのＣＲＣチェックサムが含まれる。ＤＡＴＰプロトコルのバイト順はビッグ・エンディアンである。

【００６９】

パケット・フィールドの仕様

「プロトコル・バージョン」フィールドは、送信側エンティティによって使用されるバージョンのＤＡＴＰプロトコルである。これは、ＤＡＴＰパケットの最初のバイトである。

ＤＡＴＰパケット形式は、ＤＡＴＰプロトコル・バージョン番号によって異なることがある。新しいバージョンのＤＡＴＰプロトコルが指定されている場合、このバージョン番号を繰り上げて、変更を反映させる。２つのエンティティの間のＤＡＴＰ通信では、両方のエンティティで利用できる最高バージョンのＤＡＴＰを使用する。バージョン・ネゴシエーションは、ログイン・プロセスの一部である。

【００７０】

「パケット種別情報」フィールドは、ＤＡＴＰパケットの第２バイトである。これは、どのような種類のＤＡＴＰパケットが送信されているかを示す。「ＳＴＢトランスポート情報」フィールドは、ＤＡＴＰパケットの第３バイトである。これは、ＳＴＢ側で使用するトランスポートに関する情報を提供する。これは、ＳＴＢネイティブ・トランスポート・プロトコル種別を表すフィールドの４つのＭＳＢビットであるＳＴＢ＿ｔｒａｎｓｐｏｒｔ＿ｉｎｆｏ［７．．４］、基本のトランスポートが信頼できものでるかどうかを示すビットであるが、ただしネイティブ・トランスポート・プロトコル種別の値がプロトコルの信頼性を適切に示すことができる場合であっても正しい値に設定されるビットであるＳＴＢ＿ｔｒａｎｓｐｏｒｔ＿ｉｎｆｏ［３］、ネイティブＳＴＢトランスポートのスピード・クラスを示すビットであるＳＴＢ＿ｔｒａｎｓｐｏｒｔ＿ｉｎｆｏ［２．．１］の３つのサブフィールドに分割される。

【００７１】

サービスＩＤは、ＤＡＴＰパケットの第４バイトであり、ＤＡＴＰパケットの宛先（ＳＴＢからＳＧＷへのパケット）または送信（ＳＧＷからＳＴＢへのパケット）ホストのＩＤを示す。「セッションＩＤ」は、ＤＡＴＰパケットの第２クオドレット（ダブル・ワード）である。これは、ＤＡＴＰパケットのセッションＩＤを示す。セッションＩＤ値は、ログイン・プロセスでＳＧＷにより生成される。ログイン・パケットでは、そのセッションＩＤフィールドが０に設定されている。

【００７２】

ＤＡＴＰでは、シーケンス番号はＤＡＴＰパケットの第３クオドレットの先頭ワードである。これは、ＤＡＴＰメッセージのシーケンス番号を示している。この番号は、対応するアクノリッジに送られたパケットからのＤＡＴＰ「トランザクション」を識別する番号である。メッセージ・シーケンス番号は、送信側エンティティにより生成され、ＤＡＴＰ接続の一方の側で送信されたメッセージ間でのみ一意である。つまり、ＳＴＢクライアントからＳＧＷに送信されるＤＡＴＰメッセージとＳＧＷからＳＴＢクライアントに送信されるメッセージは、同じシーケンス番号を持つが、それでも、２つ別々の「トランザクション」に対応する。

【００７３】

ＤＡＴＰでは、データ・サイズはＤＡＴＰパケットの第３クオドレットの第２のダブル・ワードである。これは、パケットのペイロード・データのサイズをバイト単位で示す。解釈により、このサイズは６４ＫＢに制限され、低速モデム・リンク、極端に雑音の多い通信チャネル、限られているＲＡＭメモリ・リソースなどローエンドＳＴＢに対するさまざまな共通要因に対応できる。ＤＡＴＰでは、暗号化フラグはＤＡＴＰパケットの第４クオドレットの先頭バイトである。ＤＡＴＰデータ・ペイロードは、１６バイトの固定サイズ・ヘッダの後の先頭バイトから始まり、ヘッダ・データ・サイズ・フィールドで指定されているデータ・ペイロードのサイズ分まで続く。ＤＡＴＰでは、ＣＲＣはデータ・ペイロードの後の第１クオドレットである。これは、ＤＡＴＰパケット全体（ヘッダを含む）の３２　ＣＲＣチェックサムの値を含む。

【００７４】

ＳＧＷとのＤＡＴＰセッションを開始するログイン・パケットがＳＴＢクライアントによって送信される。これは、ＳＴＢがＳＧＷに自己紹介するログイン・プロセスのネゴシエーションの第１段階を表す。ＳＧＷは、成功の場合にアクノリッジ・パケットでログイン要求に応答する。これは、ＤＡＴＰ接続のネゴシエーション可能属性を決定し、セッションＩＤを新規作成セッションに割り当てる。

【００７５】
ＳＧＷは、失敗の場合にネガティブ・アクノリッジ・パケットでログイン要求に応答する。このパケットはＳＴＢによって送信され、ＳＧＷとのＤＡＴＰセッションが閉じられる。ＳＧＷは、成功の場合にログアウト・アクノリッジ・パケットでログアウト要求に応答する。
【００７６】
ＳＧＷは、失敗の場合にログアウト・ネガティブ・アクノリッジ・パケットでログアウト要求に応答する。失敗の場合として、セッションＩＤが不明であったり、ＣＲＣが正しくない場合などがある。データ・パケットは、ＤＡＴＰ接続のどのエンティティでも送信できる。ＳＴＢクライアント・アプリケーションは、ＤＡＴＰデータ・パケットをアプリケーション・サーバに送信し、アプリケーション・サーバはＳＴＢに返答し、ＳＧＷからクライアントＳＴＢにデータ・パケットを強制的に送信させることができる。データ・パケットを受信したエンティティは、受信に成功した場合、データ・アクノリッジ・パケットで応答する。データ・パケットを受信したエンティティは、受信に成功しなかった場合、データ・ネガティブ・アクノリッジ・パケットで応答する。リモートＤＡＴＰエンティティから受信していない期間が続く場合に、ただし、この期間は設定可能であるとすると、他のリモート・エンティティがＤＡＴＰ　ｐｉｎｇパケットを送信して、応答を待つことによりＤＡＴＰリンクをテストすることができる。ｐｉｎｇパケットを受信したリモート・エンティティは、ｐｉｎｇパケットが正常に受信された場合に、Ｐｉｎｇアクノリッジ・パケットをリモート・ピアに送信しなければならない。ｐｉｎｇパケットを受信したリモート・エンティティは、ｐｉｎｇパケットが正常に受信されなかった場合に、Ｐｉｎｇネガティブ・アクノリッジ・パケットをリモート・ピアに送信しなければならない。失敗の場合としては、セッションＩＤが不明であったり、ＣＲＣが正しくない場合などがある。
【００７７】
図４では、ＤＡＴＰ／ＳＧＷのアーキテクチャが示されている。多数のＳＰおよびＳＴＢクライアント・アプリケーションに対する一般的ニーズは、ＤＡＴＰ／ＳＧＷアーキクチャにアドレスされるアプリケーションに固有であるという以上にトランスポートに固有のものである。ＤＡＴＰ／ＳＧＷは、暗号化、データ圧縮、ＨＴＴＰルーティング、および後述の他の多数の機能を実行する。ＤＡＴＰ／ＳＧＷアプリケーション・バックエンド・フレームワークのアーキテクチャが図４に示されている。ＤＡＴＰ／ＳＧＷは、Ｏコード・アプリケーション・レベルのストア・アンド・フォワード機能での軽量ＨＴＴＰ（ＬＨＴＴＰ）、ＳＴＢ識別（ＯｐｅｎＴＶ中央レジストリ［ＯＣＲ］）、および他の多数の機能を備える。これらの機能は、ＤＡＴＰ／ＳＧＷプロトコルの一部としてまたはその上に実装される。
【００７８】
図４に示されているように、ＳＧＷ　１０１８はＳＴＢ　１００８とＦｅｔｃｈＭａｉｌサーバ１０２６などのさまざまなアプリケーション・サーバ１０２６、１０２８、１０３０および１０３２との間に堅牢な通信リンクを構築する。ＳＧＷ　１０１８は、ＳＴＢとアプリケーション・サービスとの間で行き来する要求のルートを決定する。ＳＧＷはクライアント／ＳＴＢ　１００８からＤＡＴＰパケットを受信し、適切なアプリケーション・サーバとコンタクトをとり、ＴＣＰ／ＩＰ接続でデータをアプリケーション・サーバに送信／受信する。ＳＧＷにより、サード・パーティ製サーバまたは、ＦｅｔｃｈＭａｉｌサーバ１０２６などのＳＰ固有のサーバでＳＴＢにメッセージを送信することができる。
【００７９】
図５に示されているように、ＳＴＢ／クライアント・スタック・アーキテクチャは、複数のモジュールだけでなく、特別な層である、アプリケーションとネイティブＳＴＢ／クライアント・トランスポートとの間のメッセージ・マネージャ１１０４も備える。ＬＨＴＴＰ　ＡＰＩ　１１０６およびストア・アンド・フォワードＡＰＩ　１１０８などのＳＴＢアプリケーションに対するＡＰＩが用意されている。ＳＧＷは、非同期バージョンのＰＡ

Ｌ層を持ち、スレッドのプールとプロセス絶縁手法を実装している。

【００８０】

好ましい実施形態では、ＤＡＴＰ／ＳＧＷは大きなメッセージ・サイズに対応する一方で、配信信頼性を保証し、ＳＴＢ内の組み込み環境の制約から生じる複雑なメモリ問題を解決する。ＤＡＴＰメッセージ・サイズを増やすために、大きなメッセージを小さなセクションに分割して、送信、順序変更を行い、再構築されたＤＡＴＰメッセージで配信する。バイナリ・エラー率（ＢＥＲ）10^{-64}の信頼性のないリンクでは、６４ＫＢのメッセージにエラーが発生する確率はおおよそ７％（メッセージ１４通あたり１通）である。６４ＫＢの転送に２４００ビット／秒のモデムで５分ちょっと要することがわかれば、ＤＡＴＰは、ビット群のうちの１つが破損しているからといってさらに５分かけて同じメッセージを再送することを回避できる。再送を回避するには、ＤＡＴＰの以下の実装ガイドラインに従うのが好ましい。

【００８１】

好ましい実施形態では、大きなメッセージ、つまり６４Ｋｂを超えるメッセージをより小さなＤＡＴＰパケットに断片化する。小さな断片のしきい値を６４Ｋｂ未満とすることができる。各ＤＡＴＰ断片は個別にアクノリッジされる。図９に示されているように、ＤＡＴＰ／ＳＧＷはメッセージ・シーケンス番号と、そのシーケンス番号を最後に使用した時刻を追跡する。「最近」使用されたシーケンス番号のＤＡＴＰメッセージは「すでに受信されている」ものとして拒絶される。このポリシーを実現するために、ＤＡＴＰ／ＳＧＷホストは、各シーケンス番号のタイムスタンプとともに最近使用したシーケンス番号のスライディング・ウィンドウを保持する。古いシーケンス番号は、（ｈｏｓｔ＿ｍａｘ＿ｒｅｔｒｙ＋１）＊ｈｏｓｔ＿ｔｉｍｅｏｕｔよりも古い場合にリモート・ホストのウィンドウから削除される。好ましい実施形態では、タイムアウト値はプログラム可能であり、必要な任意の値に設定することができる。

【００８２】

リジェクション・ウィンドウは、現在時刻から始まるある時間枠内で受信したパケットのシーケンス番号を追跡し続ける。ＤＡＴＰコア層でパケットを受信すると、そのシーケンス番号がリジェクション・ウィンドウ内で検索される。シーケンス番号がそのウィンドウ内で見つかれば、破棄される、つまりシーケンス番号と関連するパケットまたは断片は無視される。パケットのシーケンス番号がウィンドウ内に見つからない場合、新しいシーケンス番号がウィンドウに追加される。ウィンドウまたは「リジェクション・ウィンドウ」は、定期的にクリーニングされ、通信リンクで使用された時間に応じてある日付よりも古いパケット番号が除去される。パケット・リジェクション・ウィンドウ・アルゴリズムは、再送／タイムアウト・ベースの信頼できるメッセージ指向トランスポート・プロトコルで頻繁に発生する同一パケットを何回も受信するという事態を効率よく防止することができる。

【００８３】

ＤＡＴＰプロトコルは、エンティティがメッセージを配信保証とともに他のエンティティに送信するメッセージ・ベースのプロトコルである。ＳＴＢは、メッセージをサービス・ゲートウェイに送信するときにはいつでも、メッセージがその最終の宛先（サービス・ゲートウェイ自体またはアプリケーション・サーバ）に到達するとアクノリッジ・メッセージを受け取る。アプリケーション・サーバによってメッセージが処理されると、ＳＴＢとサービス・ゲートウェイとのＳＴＢセッションがまだ開いている場合に、応答メッセージをＳＴＢに送信することができる。ＤＡＴＰメッセージ伝送段階の前にＤＡＴＰログイン段階があり、その後にＤＡＴＰセッションを確立するために必要なＤＡＴＰログアウト段階がある。ＤＡＴＰを通じて送信されるメッセージは、独立に送信され、アクノリッジされる高々ＭＴＵ（Ｍｅｄｉｕｍ　Ｔｒａｎｓｍｉｓｓｉｏｎ　Ｕｎｉｔ）バイトのＤＡＴＰパケットに断片化されることに留意されたい。そこで、ＤＡＴＰメッセージは、ＤＡＴＰエンティティにより物理的に管理可能な大きさにできる。図１０は、ＤＡＴＰセッションの単純な例を示している。

【００８４】
ＤＡＴＰは、同じＳＴＢトランスポート層接続の上の複数のセッションをサポートする。ＳＴＢクライアントは、サービス・ゲートウェイとのセッションが開いている間に、最初のセッションで使用しているのとまったく同じＳＴＢトランスポート・リンク上で新しいセッションを開始するログイン・パケットを送信することができる。ＳＴＢクライアントとサービス・ゲートウェイ内の両方のＤＡＴＰセッション管理モジュールが、同じリンク上でさまざまなセッション・メッセージを多重化する役割を持つ。
【００８５】
大きなＤＡＴＰメッセージ送信をサポートするために、ＤＡＴＰはパケット断片化／再構築方式を利用する。大きなメッセージが高々ＭＴＵサイズの小さなＤＡＴＰパケットに断片化される。各ホストは、ＭＴＵサイズが設定され、各ＤＡＴＰエンティティには異なるＭＴＵサイズを設定できる。ＤＡＴＰメッセージの各断片（ＤＡＴＰパケット）は個別にアクノリッジされる。
【００８６】
「最近」使用されたシーケンス番号のあるＤＡＴＰメッセージは拒絶され、「同一断片の繰り返し受信」タイプの競合条件が回避される。このポリシーを実現するために、ＤＡＴＰホストは、ウィンドウ内の各エントリのタイムスタンプとともに最近使用した（シーケンス番号、断片ＩＤ）のスライディング・ウィンドウを保持する。古い（シーケンス番号、断片ＩＤ）エントリは、（ｈｏｓｔ＿ｍａｘ＿ｒｅｔｒｙ＋１）＊ｈｏｓｔ＿ｔｉｍｅｏｕｔよりも古い場合にＤＡＴＰホストのウィンドウから削除される。
【００８７】
デフォルトのＤＡＴＰ断片サイズ（つまり、ＭＴＵサイズ）は４ＫＢに制限されており、メモリ断片化が問題になる制約されたＳＴＢ環境に対応できる。断片サイズは、アプリケーション側で、最大６４ＫＢまで増やすことができる。
【００８８】
ＤＡＴＰは、ＤＡＴＰメッセージ１つあたり最大６５５３６個の断片をサポートする。これにより、理論上最大メッセージ・サイズを４Ｇとすることができる。ＤＡＴＰメッセージの最初の断片は、断片が新しいメッセージの最初の断片であり、その断片識別（ＩＤ）フィールドがこのＤＡＴＰメッセージを構成する断片の数に設定されていることを示すマーカーとなっている。不完全なＤＡＴＰメッセージは、（ｈｏｓｔ＿ｍａｘ＿ｒｅｔｒｙ＋１）＊ｈｏｓｔ＿ｔｉｍｅｏｕｔ経過後にリモート・エンティティにより破棄されなければならない。
【００８９】
ＤＡＴＰメッセージは、リモート・ホスト・メモリ条件に基づいて送信される。ＤＡＴＰメッセージのアクノリッジされた各パケットには、受信側エンティティの現在のメモリ状態を示すメモリ使用可能フィールドが含まれる。アクティブなエンティティは、ＳＧＷに収められている利用可能なメモリに関してメッセージをＳＧＷに送信する。ＳＧＷは、メッセージを受信側エンティティに転送する前にメモリが利用可能かどうかをチェックする。送信側エンティティは、ＳＧＷに問い合わせて、受信側エンティティで十分なメモリが利用可能かどうかを調べることができる。ＤＡＴＰを利用してメッセージをピアに送信する場合、リモート・エンティティはまず、ＳＧＷまたは受信側エンティティで、ＤＡＴＰメッセージのサイズが受信側エンティティで利用できるメモリよりも小さいかどうかを調べる。受信側エンティティにメッセージを受け取れる十分なメモリがあれば、ＤＡＴＰメッセージの断片が受信側ホストに送信される。メッセージを受信すると、受信側ホストはメッセージの受信に関するアクノリッジを送る。そうでない場合、送信側ホストは制御パケットを受信側ホストに送信し、リモートまたは受信側ホストのメモリを利用できるかどうかを問い合わせる。メッセージの一部のみを保持する利用可能なメモリに基づく部分的配信も、必要に応じて実装できる。この場合、部分的メッセージが完了までキャッシュされている。制御パケットは、リモート・エンティティ内に十分なメモリが確保されるか、メッセージ送信再試行最大回数を超えるまで送信される。最大再試行回数を超え、それで

も受信側ホストにメッセージ送信を完了できるだけの十分なメモリがない場合、メッセージ送信は失敗する（部分的メッセージ配信が許可されていない限り）。

【００９０】

ＤＡＴＰプロトコル・スタックは、与えられたメッセージの最初の断片を受信したときにＤＡＭＬメッセージのメモリを確保する。デフォルトのＤＡＴＰ断片サイズは、メモリ断片化が問題になる通常のＳＴＢすなわちクライアント環境の制約されている容量に対応できるように４ＫＢに制限するのが好ましい。断片サイズは、このメッセージを送信するアプリケーション側またはメッセージを受信するアプリケーション側で最大６４ＫＢまで増やせるのが好ましい。ＤＡＴＰは、ＤＡＴＰメッセージ１つあたり最大６５５３６個までの断片をサポートするこのが好ましい。これにより、理論上、単一のメッセージに対して最大メッセージ・サイズ４ＧＢが可能になる。必要ならば、設定メッセージ・サイズを大きくすることもできる。

【００９１】

好ましい実施形態では、ＤＡＴＰメッセージの最初の断片は、断片が新しいメッセージの最初の断片であり、断片識別フィールドがこのＤＡＴＰメッセージを構成する断片の数に設定されていることを示すマーカーを含む。不完全なＤＡＴＰメッセージは、さらにメッセージ送受信時間が要求されない限り、（ｈｏｓｔ＿ｍａｘ＿ｒｅｔｒｙ＋１）＊ｈｏｓｔ＿ｔｉｍｅｏｕｔ経過後にリモート・エンティティにより破棄されなければならない。

【００９２】

ＳＧＷ／ＤＡＴＰは、アプリケーションが機密性の高いデータをそれぞれのアプリケーション・サーバに送り返せる暗号化機能を備える。トランスポート・レベルに暗号化を実装することで、ＳＴＢすなわちクライアントの処理能力の低い環境で暗号化を実装するという問題に対処している。そのため、慎重に設計された暗号化方式と好ましいＤＡＴＰセキュアＡＰＩにより暗号化に取り組んでいる。セキュリティ／暗号化はセッション・レベルで実装される。アプリケーションでは、ＤＡＴＰセキュアＡＰＩを使用してセキュア・セッションを開く。ＤＡＴＰ暗号化パラメータについては、セッション・ログイン時にネゴシエーションが行われる。セキュア・セッション・ネゴシエーションには少なくとも、標準ＤＡＴＰログイン段階と鍵ネゴシエーション段階の２つの段階が用意される。

【００９３】

以下では鍵ネゴシエーション段階の主要ステップについて簡単に説明する。ＳＧＷは、公開鍵ｓｅｒｖｅｒ＿ｅｐｋをクライアントまたはＳＴＢに送信する。ＤＡＴＰは、Ｒｉｖｅｓｔ、Ｓｈａｍｉｒ、およびＡｄｌｅｍａｎ（公開鍵暗号化技術）ＲＳＡ（他のものも使用できる）を使用するのが好ましい。ＤＡＴＰでは、ｅ＝３となるようにＲＳＡ指数ｓｅｒｖｅｒ＿ｅｐｋ＝（ｅ，ｎ）を選択し、確実なセキュリティ・レベル（セキュリティはｎによって決まる）が維持されるようにする。ＲＳＡでメッセージを暗号化するために、ＳＴＢで（m^e）ｍｏｄ　ｎを計算する必要がある。「ｅ」が小さいと、累乗段階が小さくなり、暗号化されたメッセージを高速に実行できる。ＳＴＢすなわちクライアントは、その乱数発生器をシステム時間とＯコード層（例：現在のビデオ・フレームなど）に用意されているランダム・ソースで初期化する。ＳＴＢ／クライアントは、ＳＴＢ／クライアント秘密鍵（ｓｔｂ＿ｓｋ）を選択する。ＳＴＢは、ＲＳＡを使って、秘密鍵ｓｔｂ＿ｓｋをｓｅｒｖｅｒ＿ｅｐｋで暗号化する。ＳＴＢは、暗号化された秘密鍵ｓｔｂ＿ｓｋをＳＧＷに送信する。ＳＧＷは、暗号化されているｓｔｂ＿ｓｋをその秘密鍵ｓｅｒｖｅｒ＿ｄｐｋで暗号解読する。

【００９４】

ＳＧＷは、乱数発生器を初期化し、サーバ秘密鍵ｓｅｒｖｅｒ＿ｓｋを選択する。ＳＧＷは、秘密鍵暗号化方式を使用してｓｔｂ＿ｓｋでｓｅｒｖｅｒ＿ｓｋを暗号化する。ＳＧＷは、暗号化されたｓｅｒｖｅｒ＿ｓｋをＳＴＢに送信する。ＳＴＢは、暗号化されているｓｅｒｖｅｒ＿ｓｋをその秘密鍵ｓｔｂ＿ｓｋで暗号解読する。鍵の交換が正常に行われると、互いの秘密鍵を使用しＤＡＴＰを介して２つのエンティティの間で秘密暗号化データを交換することができる。好ましい実施形態では、ＤＡＴＰ／ＳＧＷサーバ認証ステ

ップをプロトコルに追加し、鍵交換方式を強化し、「争いの仲裁者」攻撃から防御することができる。したがって、ＤＡＴＰスタックに署名し、認証証明証を管理する機能がＤＡＴＰプロトコルに用意されている。

【００９５】

ＳＧＷとの通信時間を短縮するために、サーバの公開鍵をスタック内に埋め込んで、ＳＴＢ秘密鍵の暗号化をオフラインで実行できるようにすることが好ましい。これにより、ＳＧＷがＳＴＢすなわちクライアントにより使用されるサーバ公開鍵を知っている必要があるため、新しい鍵管理問題が生じることになる。セキュア・セッションで送信されるメッセージは、断片レベルで暗号化するのが好ましい。これは、ＤＡＴＰメッセージの個々の断片が独立に暗号化されることを意味する。

【００９６】

ＤＡＴＰセキュアＡＰＩが用意されており、これにより、セキュアＤＡＴＰセッションで暗号化されていないメッセージを送信し、ＳＰアプリケーション側では、セキュア・セッションで送信された非機密データを暗号化しないことによりＣＰＵサイクルを節約することができる。この方法は、Ｍｏｔｏｒｏｌａ　ＤＣＴ　２０００などの処理能力が限られているクライアントまたはＳＴＢの場合に役立つ。

【００９７】

セキュア・セッションがＳＧＷとＤＡＴＰクライアントまたはＳＴＢとの間で確立されると、クライアント／ＳＴＢによってアプリケーション・サーバに送信されたメッセージは、最初にＳＧＷ内に暗号解読され、その後セキュア・ソケット層（ＳＳＬ）接続を使用してアプリケーション・サーバに転送される。暗号化層は、Ｏコード開発者だけでなくアプリケーション・サーバ開発者からも利用できる暗号化ライブラリに基づいている。このライブラリはアプリケーションから使用し、アプリケーション・レベルで暗号化を管理することができる。この機能は、銀行業務などの重要アプリケーションにおけるセキュリティのため実行されるエンドツーエンドの暗号化を管理する場合に役立つ。

【００９８】

ほとんどのＳＴＢおよびクライアントで利用可能とされているような低速な（２４００～３３６００ｂｐｓ）リンク上ではデータ圧縮が用意されており、回線の全スループットを高めるために圧縮データを送信することが望ましい。場合によっては、モデム・データ圧縮がＯＳＩリンク・レベルで利用できることもある。上位プロトコルには、ペイロードを圧縮しても目立った利益があるわけではない。多数のクライアント／ＳＴＢモデムは、リンク・レベルでの圧縮機構を持たず、したがって、上位プロトコルに圧縮機能を設ける。本発明は、データ圧縮機能を提示する。

【００９９】

ＳＴＢすなわちクライアント・プロセッサがほとんどの圧縮アルゴリズムで必要としている効率のよいパターン探索（またはその他のＣＰＵを酷使するオペレーション）を実行する能力を欠いているという問題がある。しかし、圧縮解除は比較的容易なタスクであり、圧縮解除ＡＰＩがＯコード・レベルでクライアント／ＳＴＢに用意されている。これらの考慮事項に基づいて、ＤＡＴＰの圧縮サポートは非対称的である、つまり、ＳＧＷからＳＴＢすなわちクライアントへのダウンリンクのみを標準ＳＰ圧縮ツールを使用して圧縮するのが好ましい。

【０１００】

圧縮されたＤＡＴＰパケットは、ペイロード・データが圧縮されていることを示す「データ圧縮済み」フラグをパケット・ヘッダ内に立てる。パケット・ヘッダは圧縮されない。圧縮および圧縮解除では、標準装備のＳＰ圧縮および圧縮解除ツールおよびＡＰＩを使用する。ＤＡＴＰパケット・サイズは、圧縮されたペイロードのサイズを示す。ペイロードの圧縮解除されたサイズは、ペイロードの圧縮ヘッダ内に示される。ＤＡＴＰメッセージの圧縮は、断片レベルで実行される。ＤＡＴＰメッセージの各ＤＡＴＰパケットは個別に圧縮される。これは、ＤＡＴＰメッセージ断片が受信時に必ずしも連続して格納されるわけではないため好ましく、したがって、ＤＡＴＰは各断片を別々に圧縮解除することが好

ましい。各ＤＡＴＰ断片は独立に圧縮されるため、独立の圧縮解除が可能である。ＤＡＴＰ　ＳＴＢスタックとＤＡＴＰアプリケーション・サーバＡＰＩでは、ダウンリンクでのデータ圧縮を禁止または有効にできる。この機能により、アプリケーション・サーバは、高速な放送チャンネルを使用してクライアントまたはＳＴＢに大量のデータを転送することと、ＳＰ帯域幅全体を節約する放送チャンネルを通じてクライアントまたはＳＴＢの集まりにマルチキャスト・データを送信することという少なくとも２つの重要な機能を備えることができる。

【０１０１】

ＳＧＷは、設定可能な数の放送ストリームを管理するオープン・ストリーマ・アプリケーション・サーバ・モジュールを備える。これらのストリームを使用して、大量のデータ・チャンクだけでなくマルチキャスト・データをクライアントおよび／またはＳＴＢに送信する。マルチキャスト機能は、放送でルーティングを行う重要な機能として用意されており、それは、アプリケーション・サーバ側で各ＳＴＢを個別に扱わずにＳＴＢのグループにデータを送信することができるためである。ＤＡＴＰのマルチキャスト・サポートでは、信頼できないＤＡＴＰパケットが送られる。ＳＰは、セッション識別子のマルチキャスト・グループのリストを保持し、ブロードキャスト・チューナーが利用できないＳＴＢすなわちクライアントが多数のマルチキャスト・グループのメンバである場合を扱う。

【０１０２】

ＤＡＴＰネーム・サービス（ＤＮＳ）は、アプリケーション・サーバ名とサービス識別子との間のマッピングを定めるものである。よく知られているサービスではサービス識別子を予約しているが、多数のユーザ定義サービス識別子が利用でき、さまざまなアプリケーションで使用することができる。ＳＴＢまたはＯコード・アプリケーションへのサービス識別子のハードコーディングを避けるために、アプリケーション側で名前レゾリューション段階の後に名前でサービスを参照する機能を備える。この方法でアプリケーションは、ＳＧＷ設定ファイルにあまり依存しなくなる。

【０１０３】

ＤＮＳ機能をＤＡＴＰクライアントに提供する方法について以下に説明する。ＤＮＳは、ＤＡＴＰプロトコルの観点からは別のサービスとみなされる。特定のサービス識別子がＤＮＳサービス用に予約されている。ＤＮＳサービスのホストはＳＧＷ内にあるか、またはＳＰまたはＳＴＢまたは他のクライアントの別のところにあってもよい。ＤＡＴＰクライアントは、アプリケーション・サーバの名前を解決するための単純なＡＰＩを備えている。主呼び出し（ｄａｔｐ＿ｇｅｔ＿ａｓｉｄ＿ｂｙ＿ｎａｍｅ（ａｓ＿ｎａｍｅ））が要求番号を同期して返すのが好ましい。非同期通知は、成功した場合には、アプリケーション・サーバ識別子を含む名前レゾリューションのステータスを返す。パフォーマンスを著しく損なうことのない同時実行の名前レゾリューションも可能である。ユーザは、それぞれの要求にタグ付けされている要求識別子に基づいてネーム・サーバ通知をディスパッチすることができる。アプリケーション・サーバの名前パラメータを現在のＤＮＳ設定ファイルに追加する。異なるサービス識別子に対して同じ名前を使用しない。冗長性をもたらすか、または拡張性問題に対処するために、１つのサービス識別子について複数のマシンを登録することができる。

【０１０４】

好ましい実装では、ＤＮＳは定義しなければならないディレクトリ・サービスのインスタンスとみなされる。ＤＮＳ要求パケット形式は、クエリ種別（クエリの種類を示す（例えばＤＮＳクエリに対しては０））、クエリ・タグ（ディレクトリ・サービス応答と照合するユーザ提供タグ）、クエリ・データ（クエリー・オペレーションを実行するために使用するデータ（通常は、ＤＮＳのサービスの名前））の各フィールドを含む。ＤＮＳ応答パケット形式は、応答種別（応答の種類を示す（ＤＮＳ　ｒｅｓｏｌｖｅ　ＯＫには０））、応答タグ（その応答を生成したクエリ・タグと同じ）、および応答データ（クエリの応答データ（通常は、ＤＮＳのサービスのＩＤ）の各フィールドを含む。

【０１０５】

ＤＡＴＰの他の実施形態では、すべてのＤＡＴＰクライアントがモデム・ラックの背後にあると想定され、接続されているクライアント毎に、モデム・ラック・ターミナル・サーバがＳＧＷとの専用ＴＣＰ／ＩＰ接続を開き、所定のＳＴＢから受け取ったものをこのＴＣＰ接続に転送する。ＴＣＰ／ＩＰがサポートされていないが、ユーザ・データグラム・プロトコル（ＵＤＰ）がサポートされている古い世代のケーブル・ボックスで配備が可能な場合、ＤＡＴＰサーバ（例えば、ＳＧＷ）はＵＤＰポート上で着信を待つ。ＵＤＰは以下のようにサポートされている。ＳＧＷが、ＵＤＰ接続を取り扱うように新しいｄａｔｐ＿ｓｏｃｋｅｔ＿ｌｉｓｔｅｎｅｒクラスを作成する。ソケット・タイプ抽象層が作成され、ＵＤＰソケット（ＰＡＬ＿ｕｄｐ＿ｓｏｃｋｅｔ）に対応できるようになる。

【０１０６】

ＵＤＰ接続は、次のように処理される。ＵＤＰ＿ｌｉｓｔｅｎｅｒは、新しい接続要求ダイアグラムを読み込んで、新しいＡＬ＿ｕｄｐ＿ｓｏｃｋｅｔを作成する。ＵＤＰ＿ｌｉｓｔｎｅｒは接続に応答し、新規作成されたＰＡＬ＿ｕｄｐ＿ｓｏｃｋｅｔを使用してそのダイアグラムを送信する。ＵＤＰ＿ｌｉｓｔｅｎｅｒは、新しいセッション・マネージャ・スレッドを作成し、新規作成ＰＡＬ＿ｕｄｐ＿ｓｏｃｋｅｔを属性として渡す。新しいセッション・マネージャは、提供されているＰＡＬ＿ｕｄｐ＿ｓｏｃｋｅｔとともにｐａｌ＿ｕｄｐ＿ｓｏｃｋｅｔ＿ｓｅｎｄを使用してＤＡＴＰクライアントと直接交信する。データグラムのリモート・アドレスを指定する必要はないことに留意されたい。接続要求に応答している間にＵＤＰ＿ｌｉｓｔｅｎｅｒによりすでに設定されている。

【０１０７】

クライアント側では、すでに指定されているｓｔｂ＿ｔｒａｎｓｐｏｒｔ　ＡＰＩをターゲットのＳＴＢすなわちクライアントで利用できる何らかのＵＤＰ　ＡＰＩに実装するＵＤＰ　ｓｔｂ＿ｔｒａｎｓｐｏｒｔモジュールが作成される。このＵＤＰ　ｓｔｂ＿ｔｒａｎｓｐｏｒｔは、ＳＧＷ　ＵＤＰリスナー・ポートに接続要求データグラムを送信し、ＳＴＢトランスポート・リンクが稼働していることをＤＡＴＰコアに通知する前にＳＧＷから応答を受け取るまで待機する。後のデータグラムは、ＳＧＷからの接続要求応答で指定されたポートを使用して送信される。

【０１０８】

Ｗｅｂサーバをフロントエンドとして使用する標準アプリケーション・サーバとのＳＧＷ用のインターフェイスを提供するためＨＴＴＰルーティングが用意される。この場合、ＤＡＴＰは、アプリケーション・サーバ開発者に提供される標準ＤＡＴＰアプリケーション・サーバＡＰＩを使用せず、その代わりに、ＨＴＴＰ　ＰＯＳＴ（ＨＴＴＰＰ）メカニズムを使用してＤＡＴＰメッセージをＷｅｂサーバのフロントエンドに転送することによりこれらのアプリケーション・サーバと直接インターフェイスする。この方式では、クライアントおよび／またはＳＴＢアプリケーションは、ＨＴＴＰサーバと交信していることを意識しないＤＡＴＰ　ＡＰＩを使用する。

【０１０９】

ＨＴＴＰＰをサポートするために、ＤＡＴＰアプリケーション・サーバ・タイプの機能がＳＧＷによって提供される。このタイプのサーバはすべて、ポストＵＲＬを指定するためにネーム・サーバ設定ファイル内に特別なエントリを設ける。アプリケーション・サーバの通信モジュールでは、ターゲットのサーバ・タイプに応じてＨＴＴＰサーバにＤＡＴＰメッセージを掲示することができる。このモジュールは、アプリケーション・サーバ（ＡＳ）通信マネージャと２つのＡＳデータ送信側に分けるのが好ましい。一方のＡＳデータ送信側がデータをＤＡＴＰ　ＡＳ　ＡＰＩ互換アプリケーション・サーバに送信し、他方の送信側がデータをＨＴＴＰベースのアプリケーション・サーバに送信する。ＨＴＴＰサーバから受信したＨＴＴＰ　ｃｏｏｋｉｅは、ＳＧＷに格納され、必要に応じてＨＴＴＰサーバに再送される。セキュアＤＡＴＰセッションで受信したＤＡＴＰメッセージは、ＨＴＴＰＳを使用してＨＴＴＰサーバに転送される。ＳＧＷがＳＰインタラクティブ・サービスへのアクセスを制御し、接続されているクライアントのＩＤに対しアプリケーション・サーバからアクセスするための手段を提供するため、ＤＡＴＰログインおよびログアウ

トはアノニマスでないのが好ましい。
【０１１０】
ＤＡＴＰの一部としてのＳＴＢすなわちクライアント識別について以下で詳述する。ＤＡＴＰスタックには、ＳＴＢすなわちクライアント依存の固有ハードウェア識別子（ＨＩＤ）が含まれる。ＳＴＢの場合、このハードウェア識別子はＳＴＢ／ネットワーク依存のＳＴＢトランスポート層から取得する。ＨＩＤ形式は可変長文字列である。指定されたネットワークのＨＩＤは、ＨＩＤリストに格納される。ネットワーク・オペレータは、ＳＰを介し、ＡＰＩを使用して顧客データベースからＨＩＤリストを更新する。直接ネットワーク・オペレータ受信契約者データベースとインターフェイスできない場合には、ＳＰは構造のないふつうのファイルから受信契約者情報（ＨＩＤを含む）をインポートする。
【０１１１】
ＤＡＴＰセッションを確立するために、ＳＴＢすなわちクライアントＤＡＴＰスタックは、ＤＡＴＰログイン・パケット内にＨＩＤを格納する。ＳＧＷは、中央リポジトリを使ってＨＩＤが有効なものかどうかを確認する。ＨＩＤが中央リポジトリによりクリアされると、ＳＴＢスタックへのアクセスが許可される。ＨＩＤにより、ＳＧＷは接続されているＳＴＢすなわちクライアントのＩＤを判別することができる。ＨＴＴＰ　ｃｏｏｋｉｅの場合のように、ＨＩＤはリモートＳＴＢすなわちクライアントに対し「強い」認証を行わない。したがって、リモート・ユーザの正式な認証は、アプリケーション側でリモート・ピアの厳格な認証を必要とするときにＳＧＷにより実行される。
【０１１２】
ＤＡＴＰ／ＳＧＷはＨＴＴＰ機能のＬＨＴＴＰをＯコード・アプリケーション開発者へ提供し、開発者がリモートＨＴＴＰサーバとやり取りすることができるようにする。ＬＨＴＴＰは、Ｗｅｂ風のＨＴＴＰベースのアプリケーションを開発することを目的として提供されている。ＬＨＴＴＰは、Ｈ２Ｏ戦略を補完するものであり、クライアント、ネットワーク・オペレータ、およびサービスの間の裏チャネル通信に使用するＯＳ独立の簡素化されたＨＴＴＰインターフェイスを提供する。ＬＨＴＴＰインターフェイスは、ＤＡＴＰスタックに基づいており、ＨＴＴＰ要求をＤＡＴＰメッセージにカプセル化している。特別なＤＡＴＰサービス識別子がこのサービス識別子で受信されるＬＨＴＴＰ層とＤＡＴＰメッセージに割り当てられ、ＳＧＷ内の特定のＬＨＴＴＰデータ送信側を使用して宛先ＨＴＴＰサーバへのルートが設定される。
【０１１３】
ＧＥＴおよびＰＯＳＴコマンドを含む、限られた数のＨＴＴＰコマンドをサポートするのが好ましい。追加ＨＴＴＰコマンドをＬＨＴＴＰに追加することもできる。ＬＨＴＴＰ要求は、ＳＧＷで標準ＨＴＴＰ要求に変換される。ＨＴＴＰ要求は、ＳＧＷがＬＨＴＴＰクライアントの代わりに生成する。ｃｏｏｋｉｅがＬＨＴＴＰクライアントに転送される。ＳＧＷは、ｃｏｏｋｉｅをキャッシュし、セッションＩＤ変換テーブルへのｃｏｏｋｉｅを保持する。ＤＮＳ応答ＨＩＤでは、この変換テーブルを使用してＨＴＴＰサーバからの要求を解決する。ＨＴＴＰサーバは、ＨＩＤを使用して、中央レジストリ・サーバからユーザ情報を抽出するのが好ましい。ＬＨＴＴＰはさらに、セキュアＡＰＩ、ＬＨＴＴＰＳを備えている。このＡＰＩはＤＡＴＰ暗号化層に基づく。ＬＨＴＴＰＳ要求は、ＳＧＷでＨＴＴＰＳ要求に自動的に変換される。
【０１１４】
ＳＭＴＰ（Ｓｉｍｐｌｅ　Ｍａｉｌ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ）ルーティングつまり電子メールによる単純なメッセージ転送がＳＧＷとアプリケーション・サーバとの間のインターフェイスに用意されている。このインターフェイスは、アプリケーションがＤＡＴＰメッセージをＳＭＴＰベースのアプリケーション・サーバに送信するリアルタイムでないトランザクションに使用することができ、またこれらのメッセージは電子メールによりターゲットのアプリケーション・サーバに転送される。
【０１１５】
ＳＭＴＰルーティングをサポートするために、ＤＡＴＰアプリケーション・サーバ・タイ

プがＳＭＴＰアプリケーション・サーバ用に作成される。このタイプのサーバは、ネーム・サーバの設定ファイルに特別なエントリを持ち、これにより、電子メール・アドレスだけでなく転送されたメッセージの電子メール件名も指定する。アプリケーション・サーバの通信モジュールは、ターゲットのサーバ・タイプに応じてＳＭＴＰベースのアプリケーション・サーバにＤＡＴＰメッセージを公示する。ＳＭＴＰアプリケーション・サーバのデータ送信側モジュールを用意して、このタイプのトランザクションをサポートする。ＳＭＴＰアプリケーション・サーバに送信されたＤＡＴＰメッセージは、ＭＩＭＥ（多目的インターネット・メール拡張仕様）でエンコードされた電子メールに添付される。メッセージの最初の部分には、送信者のハードウェア識別子と転送されるメッセージのＤＡＴＰメッセージＩＤが含まれる。メッセージの第２の部分には、ＭＩＭＥエンコードされたＤＡＴＰメッセージが含まれる。

【０１１６】

ＳＭＴＰアプリケーション・サーバに送信されたＤＡＴＰメッセージに対して、そのメッセージがセッション・マネージャによってデコードされ、電子メールでターゲットのアプリケーション・サーバに送信できる準備が整うとアクノリッジが送られる。ＳＧＷがターゲットのアプリケーション・サーバにＤＡＴＰメッセージの電子メール配信を試みると、その後のＳＭＴＰ関連のエラーが発生することがある。ＤＡＴＰ暗号化層を使用して送信されるメッセージは、最終ホストに暗号解読された状態で転送される。ＳＭＴＰで安全にＤＡＴＰメッセージを送るためＰＧＰ暗号化もサポートされている。

【０１１７】

ＤＡＴＰ／ＳＧＷストア・アンド・フォワードサービスは、非リアルタイム・メッセージを特定のアプリケーション・サーバに送信するアプリケーション用の機能を備えている。ストア・アンド・フォワードライブラリがＤＡＴＰに実装されている。アプリケーションでは、このストア・アンド・フォワードモジュールを使用し、要求条件に応じて、異なるタイミング制約でメッセージを送信する。タイミング制約条件は、「できる限り早く」、「指定時刻」、「指定出来事、イベント、またはメッセージ」から、「ランダムな期間経過後」を含む「接続したときに必ず」までさまざまである。

【０１１８】

ストア・アンド・フォワード・モジュールは、未配信のＤＡＴＰメッセージを一部の特定の属性（タイムスタンプ、タイミング制約条件、ターゲットＡＳ識別子など）とともにファイル・システムに格納する。ファイル・システムの格納経路は、少なくともコンパイル時に設定可能であり、特定のネットワークに対応させることができる。指定されたＤＡＴＰストア・アンド・フォワード対応アプリケーションが実行されている間に転送されないメッセージは、他のストア・アンド・フォワード対応アプリケーションが実行開始しない限り転送されない。ストア・アンド・フォワード・モジュールでは、転送されたＤＡＴＰメッセージの内容は変更されない。メッセージは、ターゲットのアプリケーション・サーバに変更を加えることなく転送される。

【０１１９】

図５には、複数のモジュールを含むクライアント・スタックのＤＡＴＰアーキテクチャが示されている。線１１２１の下のモジュールは、ネイティブ・クライアント・コードで書かれているが、線１１２１の上のモジュールはＯコードで書かれている。軽量ＨＴＴＰモジュール１１０６は、軽量ＨＴＴＰ機能をＯコード・アプリケーションに提供する。これは、ＤＡＴＰ　ＡＰＩの上に設けられている。ストア・アンド・フォワードモジュール１１０８は、ストア・アンド・フォワード機能をＯコード・アプリケーションに提供する。これは、ＤＡＴＰ　ＡＰＩの上に設けられている。ＤＮＳモジュール１１１０は、ＤＡＴＰメッセージ・マネージャ・モジュール１１０４を使用して、ＤＡＴＰ名前レゾリューション・サービスを提供する。ＤＡＴＰメッセージ・マネージャ・モジュール１１０４は、ＤＡＴＰのフロントエンドを提供している。ＤＡＴＰメッセージ関連のＡＰＩ呼び出しはすべて、ＤＡＴＰメッセージ・マネージャ・モジュールを通る。このモジュールは、メッセージを複数のＤＡＴＰパケットに分けて、ＤＡＴＰパケットをメッセージに再構築する

。ＤＡＴＰトランスポート・コア・モジュール１１０２は、ＤＡＴＰセッションを管理し、ＤＡＴＰパケットの送受信、放送からのＤＡＴＰモジュール受信の管理を行う。ＤＡＴＰセキュア・トランスポート拡張モジュール１１２０は、セキュアＤＡＴＰセッションを処理する。ＤＡＴＰパケット・ライブラリ１１３４は、ＤＡＴＰパケット形式仕様に基づきＤＡＴＰ　ＳＴＢトランスポート・モジュール１１３２との間のＤＡＴＰパケットの読み込み（解析）および書き込み（作成）を行う機能を提供する。完全なＤＡＴＰパケットを読み込んだ後、このモジュールは、ＤＡＴＰトランスポート・コアに解析済みＤＡＴＰパケットを通知する。

【０１２０】

ＤＡＴＰ放送ライブラリ１１２６は、ＤＡＴＰトランスポート・コア１１０２仕様に基づいて選択されているＳＰストリームで着信を監視し、指定されたＳＴＢすなわちクライアントを対象とするモジュールを待ち、ＤＡＴＰトランスポート・コア１１０２に解析済みＤＡＴＰモジュールを通知する。ＤＡＴＰ　ＳＴＢトランスポート・モジュール１１３２は、ＤＡＴＰホスト上で利用可能なネイティブ・トランスポートまたはデータ・リンクの上にリンク・レベル・パケット・インターフェイスを備える。イベント・ループ・スタブ１１１６は、ＤＡＴＰポータビリティ層で仕様が定められているメッセージＡＰＩのスタブ・バージョンを備える。このスタブは、共通ライブラリ・イベント・ループに基づく。ポータビリティ層１１１４の役割は、メッセージ・ディスパッチ・メカニズム、暗号化ＡＰＩなどアプリケーションに左右される問題からＤＡＴＰスタックを抽出するというものである。暗号ライブラリ・スタブ１１１８は、ＤＡＴＰポータビリティ層で仕様が定められている暗号ＡＰＩのスタブ・バージョンである。このスタブは、共通ライブラリ暗号パッケージに基づく。モジュール・ライブラリ・スタブ１１２４は、ＤＡＴＰポータビリティ層で仕様が定められているマルチトラック・モジュール・ダウンロードＡＰＩのスタブ・バージョンである。このスタブは、共通ライブラリのマルチトラック・モジュール・ダウンロード・パッケージに基づく。

【０１２１】

図７に示されているＤＡＴＰはデジタルＴＶアプリケーション・プロトコル（ＤＡＰ）のサブセットである。ＤＡＰ／ＤＡＴＰが図７に示されている。ＤＡＰは、ＳＰアプリケーションとＳＧＷとの間の裏チャネル通信を標準化するのに使用される。すべてのＳＰ対応ＳＴＢがＴＣＰ／ＩＰスタック拡張を備えているわけではないので、ＤＡＴＰおよびＳＧＷがＳＰアプリケーションに汎用仮想トランスポート・メカニズムを提供する。さらに、ＳＴＢの一部はそれ専用のスタックを実行するか、または通信スタックをまったく備えていない。

【０１２２】

ＤＡＰは、単純な軽量アプリケーション・プロトコル・スイートである。ＤＡＰの主な目的は、ローエンドのＳＴＢでＰＯＰ３、ＳＭＴＰ、インターネット・メッセージ・アクセス・プロトコル（ＩＭＡＰ）などの既存のアプリケーション・プロトコルを利用する単純でかつ効果的な手段を提供することである。ＳＴＢは多くの場合、低容量処理リソースおよび／または専用通信プロトコルを持つ。ＤＡＰは、アプリケーション・プロバイダからの通信の複雑さを抽出し、そして、既存のネットワーク・インフラストラクチャを今日のアプリケーション標準に利用するように設計されている。

【０１２３】

図７に示されているように、ＤＡＰはＤＡＭＬ　１６１０－デジタルＴＶアプリケーション・メタ言語およびＤＡＴＰ　１６２０－デジタルＴＶアプリケーション・トランスポート・プロトコルの２つの部分に分けられる。ＤＡＭＬ　１６１０は、多くのＳＰアプリケーションを対象とするメタ言語である。各ＳＰアプリケーションにはＤＡＭＬのそれ専用のドメインがある。クライアント・アプリケーションは、ＤＡＭＬドメイン内にカプセル化されているメッセージへの応答および要求を行う。これらの要求メッセージは、アプリケーション・サーバにより、既存のアプリケーションにあった、ＳＭＴＰやＩＭＡＰなどの適切なプロトコルに変換される。

【０１２４】
ＤＡＴＰ　１６２０は、ＴＣＰ／ＩＰまたは他の知られているプロトコルが利用できないときに低帯域幅のアプリケーション用に設計された軽量で単純なトランスポート・プロトコルである。ＤＡＴＰは、現在のＳＴＢにおける既存の通信プロトコルとインターフェイスするように設計されている。ＤＡＰは、ＤＡＴＰ、ＤＡＭＬ－Ｍａｉｌ（メール用ＸＭＬドメイン）、ＤＡＭＬ－Ｒｅｇｉ（アカウント登録用ＸＭＬドメイン）、およびＤＡＭＬ－Ａｃｃｔ（ＳＰ　ＶＭＳ／ＡＭＳシステムへのアクセス用のＸＭＬドメイン）からなる。

【０１２５】
通常のＳＴＢは、シン(thin)・クライアント・アーキテクチャに基づいている、つまり、最小限の処理能力を持つ。今日のＳＴＢによって提供されるサービスは、多くの場合、ローエンドの「ダム(dumb)」アプリケーションである。メール、チャット、およびインターネット・ブラウザなどのリソースを大量に消費する今日のアプリケーションは、より強力な処理手段を必要としている。現在のＳＴＢでは、このタイプの処理能力を備えることはできず、したがって、ローエンドの軽量アプリケーション・プロトコルが必要である。ＤＡＰは、アプリケーション開発者にクライアント／サーバ・ネットワークの複雑さが見えないように隠す／抽象化できるくらい十分に単純である。

【０１２６】
ＤＡＰは、モジュール式で、柔軟性があり、今日新たに登場するソフトウェア・アーキテクチャに適応できる。Ｃｏｍｍｏｎ　Ｏｂｊｅｃｔ　Ｒｅｑｕｅｓｔ　Ｂｒｏｋｅｒ　Ａｒｃｈｉｔｅｃｔｕｒｅ（Ｏｂｊｅｃｔ　Ｍａｎａｇｅｍｅｎｔ　Ｇｒｏｕｐ）（ＣＯＲＢＡ）ベース・モデルまたはＣｏｍｍｏｎ　Ｏｂｊｅｃｔ　Ｍｏｄｕｌｅ（ＣＯＭ）／Ｄｉｓｔｒｉｂｕｔｅｄ　Ｃｏｍｐｏｎｅｎｔ　Ｏｂｊｅｃｔ　Ｍｏｄｕｌｅ（ＤＣＯＭ）モデルのいずれかがある。ＤＡＰは、既存のサード・パーティ製のレガシー・システムに対応し、組み込むことができる十分な柔軟性を持つ。ＤＡＰは、各種のオープンの専用プロトコルとのインターフェイスを備える。これらのプロトコルは、ＰＣが主クライアント、例えば、ＩＭＡＰまたはＰＯＰ３サービスであるサービス・システムに存在している。ＤＡＰでは、ＳＰミドル・ウェア・テクノロジを利用している。ＤＡＰサーバ・ウェアは、ＤＡＰプロトコルを既存のアプリケーション固有プロトコルに変換する。

【０１２７】
ＤＡＰおよびそのサブセットＤＡＭＬ１６１０は、軽量で、ローエンド帯域幅の機密性の高いＳＴＢをサポートできるように設計されている。ＤＡＭＬタグは、４文字以内であるのが好ましく、できれば、２または３文字に制限する。ＤＡＭＬは、バイナリＸＭＬを組み込み、ＤＡＭＬタグを使いやすくしている。ＤＡＰは、ＳＴＢとサービス・サブシステム上で実行されているアプリケーション間の通信プロトコルとして使用される。ＤＡＴＰ　１６２０は、通信ハンドシェーク、ルーティング、およびトランスポート固有認証を制御するが、ＤＡＭＬはアプリケーション固有の要求条件を管理する。ＤＡＭＬ要求と応答は、既存の通信プロトコル、例えば、ＴＣＰ、ＵＤＰ、ＤＡＴＰ、または専用通信プロトコルを使用してＳＴＢクライアントとサービス・プロバイダとの間でやり取りされる。

【０１２８】
ＤＡＰプロトコルおよびそのサブセットであるＤＡＭＬは、セッション指向または「セッションレス」プロトコル・スイートである。ＤＡＭＬドメインは、アプリケーションによって決まる。ＤＡＰプロトコルの新しいドメインは、新しいタイプのアプリケーションに使用できる。新しいＤＡＰドメインを追加しても、既存のＤＡＰドメインへの影響はほとんどない。したがって、ＤＡＰは、既存のサービスに影響を与えずにサービスを追加するネットワーク・オペレータ向けの独自のかつ簡素化されたＳＰを提供する。各ＤＡＭＬドメインは、単純な人間可読タグまたは暗号短縮タグのいずれかをベースとし、パフォーマンスが重要な要因である場合にパケット・サイズを縮小することによりプロトコルのパフォーマンスを高めることができる。

【０１２９】

ＤＡＰアーキテクチャにおけるＤＡＭＬの役割の概要を述べる。ＤＡＭＬは、インタラクティブ・テレビ・サービス用の通信動作および通信データを指定するために使用されるアプリケーション・レベルの通信プロトコルである。サービス・レベル通信プロトコルは、トランスポート・レベル・プロトコルの上位である。これは、クライアント／サーバ通信についてアプリケーション固有のコンテンツをカプセル化する方法を定義する。

【０１３０】

ＤＡＭＬは、ＳＰのモジュール式設計に対応できるドメイン固有のプロトコルの集合体である。例えば、ＤＡＭＬ－ＭａｉｌはＤＡＰのサブセットである。ＤＡＭＬ－Ｍａｉｌは、メール・ドメイン固有のプロトコルである。新しいＤＴＤを作成するだけで、新しいドメイン固有のプロトコルをＤＡＰのサブセットとして追加できる。ＤＡＰにより、ＤＡＰメッセージの送受信による通信動作を指定する。アプリケーション固有データは、ＸＭＬ形式でカプセル化される。各ＸＭＬアプリケーション・ドメインの構文で、アプリケーション・サーバが実行するアクションを指定する。これにより、ＳＭＴＰサービスおよびＩＭＡＰサービスなどの既存のインフラストラクチャとインターフェイスするために今日のＳＴＢで利用できる非常に軽量で簡素なプロトコルを設計できる。

【０１３１】

ＤＡＴＰは、ＳＧＷと複数のＳＴＢすなわちクライアントとの間の通信プラットフォームを実装するトランスポート／サービス・レベルのプロトコルである。ＤＡＭＬは、ＤＡＴＰパケット内にカプセル化される。一般に、サービス・レベル・プロトコルは、搬送プロトコルの上位であるが、ＤＡＴＰはサービス・レベル、データ・リンク・レベル、またはトランスポート・レベルのいずれかの標準的なネットワーク・モデルに入れられるという点でユニークである。このため、ＤＡＴＰは非常に柔軟性が高い。ＤＡＴＰは、ＴＣＰ、ＵＤＰ、Ｘ．２５、ロー・ソケット、またはその他のプロトコルなどの基礎となる搬送プロトコルとインターフェイスする。

【０１３２】

ＳＧＷは、ローエンドＳＴＢがネットワーク・バックエンド・インフラストラクチャに接続するためのルーティングおよびＳＧＷ技術を備える。ＳＧＷでは、ＳＴＢ／クライアントとＳＧＷとの間のトランスポート・レベルのプロトコル、例えば、ロー・ソケット上のシーケンシャル・ストリーム・プロトコルをサポートする。ＤＡＭＬではこの機能を利用する。

【０１３３】

ＤＡＭＬ－ＭａｉｌはＤＡＰプロトコルのサブセットである。ＤＡＭＬ－Ｍａｉｌは、メール・ドメイン固有のプロトコルである。このプロトコルは、ＳＴＢとＩＭＡＰ、ＰＯＰ３、およびＳＭＴＰサービスとリンクするために使用される。ＤＡＭＬ－Ｒｅｇｉは、複数のサービスについてアカウントを登録するための汎用の方法の仕様を定めるＤＡＰサービス・ドメイン・プロトコルである。ＤＡＭＬ－Ｒｅｇｉは、ＳＴＢと登録サーバとの間の単純なプロトコルである。これにより、ＳＴＢと各種のアプリケーション・システムとの間の複雑なやり取りが、単一の統合点のみ、つまり登録サーバで行うことができる。

【０１３４】

ＤＡＭＬ－Ａｃｃｔは、ＳＰ　ＶＭＳ／ＡＭＳデータベースと通信するＤＡＰサービス・ドメイン・プロトコルである。ＤＡＭＬ－Ａｃｃｔを使用することで、ＳＴＢ／クライアントはＶＭＳ／ＡＭＳシステムからのユーザ固有データについてクエリを実行し、返すことができる。すべてのＤＡＭＬドメインは、ＸＭＬ文書型定義（ＤＴＤ）構文を使用して定義される。ＤＴＤでは、メッセージ構文を記述するが、要求と応答の交換用のロジックについて記述しない。ＸＭＬは、１ブロック分のテキストのマークアップを定義する際に使用する。特定のＤＡＭＬ要求と応答は互いに関連する相互作用である。この相互作用の規則はＳＴＢとアプリケーション・サーバ・コンポーネントとにモジュール化される。

【０１３５】

メッセージング・マネージャは、ユーザ間の各種のメッセージ通信および外部者（ネットワーク・サービス受信契約者でない人たち）との各種のメッセージ通信を扱う。例えば、

これにより、ユーザは電子メールの送受信、他の非受信契約者とのチャット、および非受信契約者からのインスタント・メッセージの受信を行える。メッセージング・マネージャの電子メール部分には、ＩＭＡＰ、ＰＯＰ３、および適切なメール・ホスト・サーバ用のその他のＷｅｂメール・メッセージなどのインターネット・ベースの電子メール・サーバに接続されているＦｅｔｃｈｍａｉｌコンポーネントが含まれる。

【０１３６】

ＦｅｔｃｈｍａｉｌはすべてのＳＰサーバ・サイドのメール管理を行う。Ｆｅｔｃｈｍａｉｌは、ＤＡＰメッセージを適切なメール・ホスト・サーバ用のＩＭＡＰ、ＰＯＰ３、またはＷｅｂメール・メッセージに変換する。ＳＧＷは、ＤＡＰメール・メッセージを処理のため「Ｆｅｔｃｈｍａｉｌ」に送る。Ｆｅｔｃｈｍａｉｌは、要求に対する適切な応答で返答する。Ｆｅｔｃｈｍａｉｌは、ＩＭＡＰサーバとインターフェイスする。電子メール・アプリケーションはＳＰによって提供される。すべてのＳＰアプリケーションは、ＳＧＷによって提供される電子メール・サービスを介して電子メールを「送る」ことができる。

【０１３７】

チャット・サーバなどとのチャットＳＰサービス・インターフェイスには、チャット・サーバが含まれる。チャット・サービスにアクセスするには、専用チャット・アプリケーションを使用するが、ＳＰチャット・クライアントＤＬＬを使って任意のＳＰからもアクセスできる。チャットとプログラム・リストとのインターフェイスを実現するために、チャット・ルームを放送番組で動的に作成することができる。アプリケーションおよびその他のサービスは、ＳＰ「アラート」サービスを使用して、ＳＴＢ常駐ミニ・アプリケーションをトリガすることができる。アラートでは、オープン・ストリーマのＳＰ　ＯＭＭ拡張および機能を利用する。電子メール・サービスでは、アラート・トリガを使用して、視聴者に着信メッセージを通知する。

【０１３８】

図６では、ＳＧＷに、ＤＡＴＰ機能をサポートする複数のモジュールが組み込まれている。ＳＧＷアーキテクチャは、スレッドのプールを備えるマルチプロセス・ベースのアーキテクチャである。サーバ全体が、非同期バージョンのプラットフォーム抽象層（ＰＡＬ）で実行される。ＰＡＬでは、メッセージ・キュー・プロセスを実行する。ＰＡＬは、メッセージ受渡手法を使用して通信を行う。ＳＧＷでは、図６に示されているように、３種類のプロセッサを使用する。

【０１３９】

図６に示されているように、アプリケーション・サーバまたはサービスは、ドメイン固有のＤＡＰプロトコルを使用してＳＧＷを通じて複数のクライアント／ＳＴＢと通信する。場合によっては、クライアント／ＳＴＢはアプリケーション・サービスに直接接続できる。例えば、ＳＴＢとネットワークとの間のトランスポート・プロトコルがＴＣＰ／ＩＰの場合、ＳＴＢはＴＣＰ／ＩＰ対応であり、ＳＧＷによって提供される複雑な共通サービスを実行する必要はなく、ＴＣＰ／ＩＰを介してサービスと直接通信するクライアント／ＳＴＢを通じてネットワークの高速性能を改善できる。

【０１４０】

図８では、ＤＡＴＰサーバ、ＳＧＷの主プロセスは、上述の主ＤＡＴＰサーバ・プロセスである。ＳＧＷは複数の重要モジュールのホストである。ＴＣＰソケット・リスナー・モジュール１２０４は、ＤＡＴＰ　ＴＣＰ監視ポートで接続を待ち、その接続を受け付け、新規接続を処理する新しいセッション・マネージャの作成を要求する単純なＴＣＰソケット・リスナー・スレッドである。ＵＤＰソケット・リスナー１２０２は、ＵＤＰ接続のよく知られているポートで待機する。接続要求を受け取ると、ＵＤＰソケット・リスナー１２０２は新しいソケットを作成して、接続要求アクノリッジをリモート・ホストに送信する。その後、ＵＤＰソケット・リスナー１２０２は接続を処理する新しいセッション・マネージャの作成を要求する。

【０１４１】

セッション・マネージャ・モニタ１２０６モジュールはメイン・スレッドの一部である。このコンポーネントの主要な役割は、セッション・マネージャ（ＳＭ）プロセッサ１２１４分布（負荷に基づいてＳＭプロセッサを作成、削除する）を監視し、セッション・マネージャ作成要求を最も使われていないＳＭプロセッサ１２１５に転送する。各ＳＭプロセッサ（０～ｎ）１２１５は、ＤＡＴＰ、ＨＴＴＰＰ、ＬＨＴＴＰ、およびＳＭＴＰ用にＤＡＴＰアプリケーション・サーバ通信モジュール（ＡＳＣＭ）１２１７および独立のアプリケーション・サーバ・データ・センダー（ＡＳＤＳ）を備える。

【０１４２】

ＤＮＳネーム・サーバ１２１２スレッドは、アプリケーション・サーバ識別子とその属性（ホスト名、ポート、タイプなど）との間の照合テーブルだけでなく、セッション識別子とセッション・マネージャ・メッセージ・キュー識別子との間の照合テーブルも保持する。ネーム・サーバ・モジュールＤＮＳは、メッセージ・キューにポストされた名前レゾリューションクエリに応答する。アプリケーション・サーバのソケット・リスナー／スレッド１２０８は、メッセージ・ポスト要求がアプリケーション・サーバから入ってくるのを待つ役割を持つ。その後、ネーム・サーバ１２１２は、ポスト要求セッション識別子に基づいてポスト要求をターゲットのセッション・マネージャに転送する。

【０１４３】

セッション・マネージャのプロセッサ・プロセス１２１４、１２１６は、セッション・マネージャ・スレッド１２１５のプールのホストとなる。新しいセッション・マネージャ・スレッドが、セッション・マネージャのモニタ１２０６からセッション・マネージャのプロセッサ・スレッドへの要求に基づいて作成される。セッション・マネージャ・プロセッサ１２１４、１２１６のスレッドは、セッション・マネージャ・プロセッサ１２１４、１２１６から要求を受け取り、ＳＭモニタからの要求に基づいてセッション・マネージャを作成または削除し、要求の結果をセッション・マネージャ・プロセッサに通知する。セッション・マネージャ・スレッド１２１５は、ＤＡＴＰセッションを管理し、ＳＴＢすなわちクライアントからＤＡＴＰメッセージをアプリケーション・サーバに、アプリケーション・サーバからＳＴＢすなわちクライアントに転送する。ＳＴＢすなわちクライアント毎に１つのスレッドがある。これらのスレッドでは、複数の重要モジュールを使用してＤＡＴＰセッションを処理する（パケット・ライブラリ、アプリケーション・サーバ通信モジュール、ＤＡＴＰアプリケーション・サーバ・データ・センダー、ＨＴＴＰＰアプリケーション・サーバ・データ・センダー、ＬＨＴＴＰアプリケーション・サーバ・データ・センダー、およびＳＭＴＰアプリケーション・サーバ・データ・センダー）。

【０１４４】

放送マネージャ・プロセス１２１０は、放送によるＤＡＴＰルーティングの主要コンポーネントである。このプロセスは、ＤＡＴＰサーバ・カルーセルを管理するオープン・ストリーマ・アプリケーション・サーバである。放送マネージャ・プロセスは、ＤＡＴＰサーバの他のコンポーネントから受信した要求に応じて動的にこれらのカルーセルを更新する。

【０１４５】

ＳＰおよびＳＧＷは、当業ではよく知られており、Ｓｕｎ　Ｍｉｃｒｏｓｙｓｔｅｍｓから市販されているメモリ、モニタ、ＧＵＩ、マウス、キーボード、およびプロセッサを備えるＳｕｎ　Ｓｏｌａｒｉｓ　７データ処理システムでサポートされている。ＳＧＷは、ＵＮＩＸ（登録商標）デーモンとして実行され、設定には設定ファイルを使用し、コマンド・ラインから起動する。ネットワーク上でＳＧＷとＳＴＢ／クライアントとの間に接続が確立されると、ＴＣＰ／ＩＰにより他のサービス間のすべての通信が処理される。異なるトランスポート・プロトコルを処理するほかに、ＳＧＷではさらに、ＳＧＷの設定に応じて別のサービス・サブシステムへのメッセージのルートを設定する。

【０１４６】

ＳＧＷは、アプリケーション・サーバの入口点でその機能を実行する。このため、ネットワークおよびメッセージング機能がＳＧＷ上で分離されているので、機能の設定／追加は

簡単である。したがって、サービス・サブシステムがコア・アプリケーション機能で動作する負担から解放され、ネットワークの接続の問題はＳＧＷに任される。また、特定の機能を分離して別のホストに割り当てる、例えば、ＳＧＷを使用して電子メール・メッセージ配信と受信（ＦｅｔｃｈＭａｉｌサーバ）をネットワーク・ルーティングとセキュリティから分離することにより、拡張性を高めることができる。

【０１４７】

ＳＧＷは、単一サーバ上で同時接続を数百回実行できる規模である。ＳＧＷは、ＳＧＷのホストになるプロセッサの処理能力に応じて、処理できる設定接続数を増やすことができる。この限界は、主要ＩＳＰに対して１つのＰＯＰ（Point Of Presence）当たりのモデム数（通常、数百）に基づく。ＳＧＷが１つの中心点に配置されたＷＡＮアーキテクチャの場合、ハードウェア・ネットワーク・アドレス変換（ＮＡＴ）ベースの負荷分散デバイスが負荷分散のため複数のＳＧＷを並列に接続するように設けられている。

【０１４８】

以下に、Ｈ２Ｏアーキテクチャおよびサンプル・トランザクションの論理図を使用してＨ２Ｏプロキシ環境の概要を示す。ＵＲＩの要求は別のＨ２Ｏコンポーネント、例えば、ＳＴＢ／ＳＧＷおよびカルーセルから届く。以下の文脈では、要求を発行するＳＴＢ／ＳＧＷに関する重点事項の概要を述べるが、情報の一般的流れは同じままである。

【０１４９】

視聴者は、ＴＶ　Ｗｅｂページとやり取りすることを選択し、ＳＴＢからＨ２Ｏシステムに要求を発行し、応答を待つ。ＳＴＢ要求はＳＧＷに送信されるが、その際に、トランスポート・プロトコルとしてＤＡＴＰメッセージ内にカプセル化されている軽量ＨＴＴＰ要求（ＬＨＴＴＰ）を使用する。要求されたオブジェクトは、同じチャネルとプロトコルを使用して返される。ＳＧＷが、ＴＣＰ／ＩＰでＬＨＴＴＰプロトコルを標準ＨＴＴＰに変換し、要求をＷｅｂキャッシュに発行する。

【０１５０】

コンパイルド（コンパイルされた）・オブジェクト・キャッシュ（ＣＯＣ）は、内部ディスク領域を使用して、取り扱い対象の要求を処理する（オブジェクトの存続時間を考慮して発見的手法に従う）。その役割は、Ｈ２Ｏプロキシにクエリを実行せずに、すべての静的オブジェクト（クエリなし標準ＵＲＬ、ポストされない形式）を処理し、処理負荷を軽減することである。このアーキテクチャでは、ＣＯＣはコンパイルド・オブジェクト（Ｈ２Ｏモジュール）のみを格納する。ＣＯＣマシンはＩ／Ｏ駆動方式である。

【０１５１】

図１１を参照すると、Ｈ２Ｏプロキシ２４８は、異なるＨ２Ｏコンパイラ（またはフィルタ）を実行するための拡張可能な環境を備えている。これは、ＨＴＴＰ要求と応答を「オンザフライ」で処理する。したがってＨ２Ｏプロキシマシンはプロセス駆動方式である。Ｈ２Ｏ　ＨＴＭＬコンパイラ１４２０は、ＨＴＭＬからＳＰへのリソース・コンパイルを受け持つ。描画するＴＶレイアウト１４２２を計算するために、このコンポーネントは、組み込まれているイメージのサイズに基づき自動的にＨＴＴＰ要求を発行する。このコンパイラは、クライアント表示デバイス、たとえばテレビに合わせて、Ｗｅｂベースのイメージを再配列する。

【０１５２】

ＭＰＥＧコンパイラ１４２６は、通常のＷｅｂイメージ形式からＳＰ　Ｈ２Ｏ　ＭＰＥＧリソースに変換する操作を受け持つ。ソース形式としては、ＪＰＥＧおよびＧＩＦがあり、ＰＮＧも含むことがある。変換プロセスは、ＵＲＬを使用して引数を渡すことにより駆動できる。ＰＩＸＭＡＰコンパイラは、通常のＷｅｂイメージ形式からＳＰ　Ｈ２Ｏリソースに変換する操作を受け持つ。ソース形式としては、ＧＩＦがあり、ＰＮＧも含むことがある。

【０１５３】

リクエスト・パッチャ１４２４は、他のシステムから届いたデータ（例えば、クレジット・カード番号など）を組み込む要求または応答を完了または修正する役割を持つ。これは

、外部プロセスまたはデータベースと通信して、顧客情報をフェッチする。ＳＰコンポーネントは、ユーザ情報の中央リポジトリを備える。リクエスト・パッチャは、このコンポーネントと通信して、要求／応答をパッチするために必要なデータを取得する。

【０１５４】

ノットコンパイルド・オブジェクト・キャッシュ１４３０は、内部ディスク領域を使用して、取り扱い対象の要求を処理する（オブジェクトの存続時間を考慮して発見的手法に従う）。キャッシュされたオブジェクトは、静的ＨＴＭＬ、ＧＩＦイメージ、ＪＰＥＧイメージ、および標準のすべてのＷｅｂ形式ファイルを含む。その役割は、インターネットにクエリを実行せずに、すべての静的オブジェクト（クエリなし標準ＵＲＬ、ポストされない形式）を処理し、オブジェクトを取得する待ち時間を短縮し、ある種のフォルトトレラント能力をシステムに与えることである。顧客のＷｅｂサイトはＨ２Ｏシステムを通じてパブリッシュするＷｅｂサイトを保持する。

【０１５５】

図１２は、すでにキャッシュされている静的ページの要求を示している。ＳＴＢユーザが、ＨＴＭＬページ１５２０をロードする要求を発行する。この要求は、ＤＡＴＰ上でＬＨＴＴＰを使用してＳＧＷ　２４８に送られる。ＳＧＷは、ＴＣＰ／ＩＰでこの要求をＨＴＴＰに変換し、それをコンパイルド・オブジェクト・キャッシュ１４１０に転送する（１５２２）。コンパイルド・オブジェクト・キャッシュ１４１０は、要求された（モジュールにコンパイルされた）ＨＴＭＬページを内部ハードディスク領域に格納する。ただしオブジェクトの存続時間が期限切れになっておらず、コンパイルド・オブジェクト・キャッシュがコンパイルドＨＴＭＬページで要求を処理する場合である。ＴＣＰ／ＩＰ上でＨＴＴＰを使用してＨＴＴＰ応答１４２４をＳＧＷに送信する。ＳＧＷは、ＴＣＰ／ＩＰ上のＨＴＴＰからプロトコルをＤＡＴＰ上のＬＨＴＴＰに変換する。ＳＴＢは要求されたページ１５２６（コンパイルされた）をメモリ内にロードし、解釈できるようにＨ２Ｏブラウザ・エンジンに渡す。Ｈ２Ｏブラウザ・エンジンは、変換オプション（ｍｐｅｇまたはｐｉｘｍａｐ、幅、高さなど）で、テレビの画面に描画するために必要なイメージをＵＲＬ上で取得するようＳＧＷに要求する（１５２８）。ＳＧＷは、ＨＴＴＰ要求１５３０をコンパイルド・オブジェクト・キャッシュに送信する。コンパイルド・オブジェクト・キャッシュでは、要求された（モジュールにコンパイルされた）イメージを内部ハードディスク領域に格納する。オブジェクトの存続時間が期限切れになっておらず、コンパイルド・オブジェクト・キャッシュ１５３２および１５３４がコンパイルド・イメージとともに要求を処理する。このシナリオでは、Ｈ２Ｏプロキシは要求を免れており、他の要求を処理することができる。

【０１５６】

図１３に示されているように、ＳＴＢ　２１２ユーザは、ＨＴＭＬページ（ｈｏｍｅ．ａｓｐ）をロードする要求１６１０を発行し、要求ヘッダのホストおよびユーザ情報フィールドに、［ＳＴＢモデル＋ＳＴＢシリアル番号］およびユーザの［アクセス・カードＩＤ］が保持される。この要求１６１０は、ＤＡＴＰ上でＬＨＴＴＰを使用してＳＧＷに送られる。ＳＧＷは、ＴＣＰ／ＩＰでこの要求をＨＴＴＰに変換し、それをコンパイルド・オブジェクト・キャッシュに転送する（１６１２）。要求されたオブジェクトは、Ｗｅｂキャッシュのディスク領域では利用できない。その後、Ｗｅｂキャッシュは、要求１６１４をＨ２Ｏプロキシに転送する。Ｈ２Ｏプロキシはユーザの名前を返す（１６２０）ようＳＰに求める（１６１６）（ａｍａｚｏｎ．ｃｏｍサービスの場合）。Ｈ２Ｏプロキシは、要求にユーザの名前のパッチを当て、この要求１６２２を「ノットコンパイルド・オブジェクト・キャッシュ」に発行する。「ノットコンパイルド・オブジェクト・キャッシュ」は、ディスク領域内に要求されたＨＴＭＬページを保持せず、要求１６２４をターゲットのＷｅｂサーバ、ここではａｍａｚｏｎ．ｃｏｍに発行する。ターゲットのＷｅｂサーバは、ユーザ情報があればＨＴＭＬページを計算し、それ１６２６を「ノットコンパイルド・オブジェクト・キャッシュ」に返す。「ノットコンパイルド・オブジェクト・キャッシュ」はＨＴＭＬページ１６２８をＨ２Ｏプロキシに返す。

【０１５７】
Ｈ２Ｏプロキシは、ＨＴＴＰ要求１６３０を「ノットコンパイルド・オブジェクト・キャッシュ」に送り、レイアウト計算に必要なイメージ１６３２、１６３４、１６３６（ｇｉｆ、：ｊｐｅｇなど）を取得する。Ｈ２Ｏプロキシは、ＨＴＭＬページをコンパイルし、レイアウトを計算し、埋め込まれているイメージのＵＲＬをパッチし、その結果のＯｐｅｎＴＶリソース１６４６（ＳＰリソースＭＩＭＥタイプ付きで）を「コンパイルド・オブジェクト・キャッシュ」に送り返す。コンパイルド・オブジェクト・キャッシュでは、オブジェクトをその内部ディスク領域に格納し、コンパイルドＨＴＭＬ１６４８ページをＳＧＷに送り返す。ＳＧＷは、ＤＡＴＰでこの応答をＬＨＴＴＰに変換し、それをＳＴＢに送り返す（１６５０）。ＳＴＢは要求されたオブジェクトをメモリ内にロードし、解釈できるようにＨ２Ｏブラウザ・エンジンに渡す。

【０１５８】
Ｈ２Ｏブラウザ・エンジンは、要求１６５２をＳＧＷに発行し、（パッチが当てられたｕｒｌを通じて、ここでは、ｌｏｇｏ．ｇｉｆ　ｕｒｌはｐｉｘｍａｐリソース形式のディレクティブを含む）描画に必要なイメージｐｉｘ／ｌｏｇｏ．ｇｉｆを取得する。ＳＧＷは、ＴＣＰ／ＩＰで要求１６５２をＨＴＴＰに変換し、それをコンパイルド・オブジェクト・キャッシュに転送する。「コンパイルド・オブジェクト・キャッシュ」にはすでに、ユーザがすでにこのイメージを前回に要求していたため、正しいリソース形式で要求されたｇｉｆイメージがあり、このイメージは直接ＳＧＷに返される（１６５４）。ＳＧＷは、ＤＡＴＰで応答をＬＨＴＴＰに変換し、それをＳＴＢに送り返す（１６５６）。Ｈ２Ｏブラウザ・エンジンは、要求１６５８をＳＧＷに発行し、描画に必要なイメージｍｐｇ／ｂａｎｎｅｒ．ｊｐｇを取得する。「コンパイルド・オブジェクト・キャッシュ」は、ディスク領域内に要求されたイメージを保持せず、したがって、要求１６６０をＨ２Ｏプロキシに発行する。Ｈ２Ｏプロキシは、ＨＴＴＰ要求１６６２を「ノットコンパイルド・オブジェクト・キャッシュ」に送り、／ｂａｎｎｅｒ．ｊｐｇイメージを取得する。

【０１５９】
「ノットコンパイルド・オブジェクト・キャッシュ」はイメージをキャッシュに保持し、それ１６６４を即座にＨ２Ｏプロキシに返す。Ｈ２Ｏプロキシは、ｕｒｌで与えられたパラメータ（ｍｐｇ形式、幅、高さなど）を使用してイメージを変換し、その結果をコンパイルド・オブジェクト・キャッシュ１６６８に返す。コンパイルド・オブジェクト・キャッシュでは、オブジェクトをその内部ディスク領域に格納し、変換されたｍｐｅｇイメージをＳＧＷに送り返す（１６６８）。ＳＧＷは、ＤＡＴＰで応答をＬＨＴＴＰに変換し、それをＳＴＢに送り返す（１６７０）。ＳＴＢが、ＨＴＭＬページを画面に描画する。

【０１６０】
Ｈ２Ｏプロキシ・コンポーネントは、他のＨ２Ｏコンポーネントまたはコンパイラに、堅牢で拡張可能なアーキテクチャおよび「コンパイラ」設定用のインターフェイスを提供する。提供される他のサービスとしては、エラーのログ機能、定義済みイベントに関する管理者へのアラート機能、「コンパイラ」のデバッグ・トレース機能がある。提供されるＨ２Ｏプロキシ環境とＡＰＩから、コンパイラはＨＴＴＰ要求と応答に対しオンザフライで「パッチ」を当て、最終的に、そのために外部データベース、ファイル、またはプロセスにアクセスする。コンパイラは、特定のＨＴＴＰヘッダ（ＳＴＢ識別子、アクセス・カード識別子．．．）の削除、特定のＨＴＴＰヘッダ（ユーザ名、クレジット・カード番号．．．）の追加、ＨＴＭＬ　Ｆｏｒｍフィールドの着信ポスト要求（Ｖｉｓａカード番号．．．）への追加、文字列置換（＄ＵＩＤ＄－＞ユーザ識別子）を実行することで、ＨＴＴＰ要求にパッチを当て、コンパイラがＨＴＴＰリソース応答においてＷｅｂオブジェクト形式およびｍｉｍｅタイプを「オンザフライ」で変換し、ＨＴＴＰ要求を自動的に発行し、その引き換えに応答オブジェクトを取得する。

【０１６１】
図１４に示されているように、好ましい実施形態では、Ｈ２Ｏプロキシは内蔵しているソフトウェア（Ｗｅｂプロキシ、ファイヤウォール、Ｗｅｂサーバ、またはその他のもの．

．．）の拡張を開発することにより実装される。このホスト・ソフトウェアは、Ｈ２Ｏスレッド機能およびＨ２Ｏタスクのスケジュール機能だけでなく、Ｈ２Ｏ「コンパイラ」およびパッチング・コンポーネントの実装に必要な機能もいくつか備える。

【０１６２】

用意されているＡＰＩを使用し、プロキシ・ホスト・ソフトウェアにより一組のＡＰＩ（Ｈ２ＯプロキシＡＰＩ）を提供し、Ｈ２Ｏプロキシ・ホスト・ソフトウェア・サービスに欠けているＨ２Ｏコンパイラで必要な機能を実装し、Ｈ２Ｏプロキシ・ホスト・ソフトウェアから利用できるサービス用に上位の抽象レベルを提供する。リクエスト・パッチャ１４２４コンポーネントが、ＨＴＭＬページに対する着信ＨＴＴＰ要求を読み込み、他のプロセスまたはファイルまたはデータベースの情報で補う。ＨＴＭＬ２ＲＥＳコンパイラ１４２０は、返されたＨＴＭＬページをＳＰリソース・ファイルにコンパイルし、ＨＴＴＰ応答ヘッダのｍｉｍｅタイプを新しい形式、Ｍｉｍｅ－Ｔｙｐｅ：ｔｅｘｔ／ｏｔｖｒｅｓに合わせて変更する。ＧＩＦ２ＰＩＸコンパイラ１４２２は、返されたＧＩＦイメージをＳＰリソース・ファイルに変換し、ＨＴＴＰ応答ヘッダのｍｉｍｅタイプを新しい形式、Ｍｉｍｅ－Ｔｙｐｅ：ｉｍａｇｅ／ｏｔｖｐｉｘに合わせて変更する。２ＭＰＥＧコンパイラ１４２６は、返されたＧＩＦまたはＪＰＥＧイメージをＳＰリソース・ファイルに変換し、ＨＴＴＰ応答ヘッダのｍｉｍｅタイプを新しい形式、Ｍｉｍｅ－Ｔｙｐｅ：ｉｍａｇｅ／ｏｔｖｍｐｇに合わせて変更する。

【０１６３】

図１５には、ＨＴＭＬページ・シーケンス図の動的要求が示されている。オブジェクト・キャッシュは、シーケンス図には表示されず、このやり取りでは「受動的」コンポーネントである。ユーザＳＴＢ２１２が、ＨＴＴＰ要求を通じてページ（ｈｏｍｅ．ａｓｐ）の要求１８１０を発行する。リクエスト・パッチャ１４２４は、外部プロセス／ファイル／データベース／ｕｒｌ　１８１２、１８１４にアクセスして、ユーザ名を取得し、要求にパッチを当て、それをＨＴＭＬ２ＲＥＳコンパイラに送る（１８１６）。ＨＴＭＬ２ＲＥＳコンパイラは、要求１８１８をターゲットのＷｅｂサイト（ａｍａｚｏｎ．ｃｏｍ）に送る。Ｗｅｂサイトでは、その要求を計算し、その結果のＨＴＭＬページをＨＴＭＬ２ＲＥＳコンパイラに送り返す（１８２０）。ＨＴＭＬ２ＲＥＳコンパイラは、ファイルを解析して、イメージ・リンクＵＲＬを取得し、要求１８２２をＷｅｂサイト発行して、イメージ・ファイル（ｌｏｇｏ．ｇｉｆ、ｂａｎｎｅｒ．ｊｐｇ）を取得する（１８２４）。ＨＴＭＬ２ＲＥＳコンパイラでは、そのページのＴＶレイアウトを計算し、ＳＰリソース・ファイルにコンパイルし、ＳＴＢに送り返す（１８３０）。ＳＴＢが、ＨＴＭＬページをテレビに描画する。

【０１６４】

図１６には、イメージ・ファイルの要求、シーケンス図が示されている。ユーザＳＴＢにロードされるＨＴＭＬページには、画面に描画するイメージが必要である。これは、イメージ用のＨＴＴＰ要求１９１０（ＵＲＬ埋め込み変換オプション）を２ＭＰＧコンパイラに発行する。２ＭＰＧコンパイラは、ターゲット・サイト（ａｍａｚｏｎ．ｃｏｍ）にイメージ１９１２を要求する。ターゲット・サイトがｂａｎｎｅｒ．ｊｐｇイメージ・ファイル１９１４を２ＭＰＧコンパイラに返す。２ＭＰＧコンパイラが、ＵＲＬで指定されたオプションを使用して、ｂａｎｎｅｒ．ｊｐｇファイルを変換し、ｉｍａｇｅ／ｏｔｖｍｐｇ　ｍｉｍｅタイプ付きでＳＴＢに結果１９１６を返す。ＳＴＢが、イメージを画面に描画する。

【０１６５】

識別された異なるＨ２Ｏコンパイラが、クラスからＨ２Ｏコンパイラを継承し、そのクラスの異なる純然たる仮想エントリ・ポイントを実装する。要求／応答バッファの割り当ておよび解放を行うメモリ機能がコンパイラに与えられる。割り当てられたバッファのサイズがＦｒｅｅＢｕｆｆｅｒ機能に送られるため、異なる方式を使用してバッファを解放することができる（特定のサイズに関して、バッファをメモリ内にマッピングされたテンポラリ・ファイルとして実装することもできるが、サイズが小さい場合は、メモリ・バッフ

ァとして実装するのが好ましい）。
【０１６６】
完全ＨＴＴＰ要求／応答を含むバッファがＥｘｅｃｕｔｅ機能に渡され、コンパイラは、要求ヘッダ、ｍｉｍｅ－ｔｙｐｅを解析し、適切な処置を講じる。このバッファは読み取り専用であることが好ましい。バッファは、コンパイラまたはその他の後の機能により拡大できるように書き込み可能とすることもできる。Ｅｘｅｃｕｔｅ機能によって返されるバッファには有効なＨＴＴＰ要求／応答が格納され、メモリはＨ２Ｏプロキシにより、適切なＦｒｅｅＢｕｆｆｅｒ機能を使用して解放し、与えられたＡｌｌｏｃＢｕｆｆｅｒ機能により割り当てる必要がある。コンパイラ実装者がＨ２Ｏプロキシ環境内からデバッグ・トレースを実行できるようにデバッグ・メンバが用意されている。
【０１６７】
パラメータ機能を使用して、パラメータの名前を取得し、パラメータの現在値（文字列）を取得し、パラメータの新しい値を設定し、パラメータ・セットの確認を行う。イメージをフェッチするＨＴＭＬコンパイラ用にＵＲＬ機能が用意されている。これらの機能は、他のコンパイラでも使用でき、これにより、特別なサービスを必要に応じてコンポーネントに提供することができる。
【０１６８】
例えば、１００万個のＳＴＢがあり、平均２０，０００人のユーザが接続されているネットワークでは、ＳＧＷおよび「コンパイルド・オブジェクト・キャッシュ」に対し毎秒２，０００回のＨＴＭＬページ要求が発生する（要求されたページの一部がブロードバンドからでない限り）。これらのページが静的でなければならず、また「コンパイルド・オブジェクト・キャッシュ」から即座に提供されなければならないと仮定すると、Ｈ２Ｏプロキシでは、毎秒２００件の要求に応じなければならない。標準的なＨＴＭＬページに１０個のイメージが埋め込まれ、１０個のうち８個がＪＰＥＧであると仮定すると、Ｈ２Ｏプロキシは１０個の送出する要求を着信要求毎に発行する。「ノットコンパイルド・オブジェクト・キャッシュ」では毎秒２，０００件の要求に応じられる。
【０１６９】
可能であればＭＰＧ変換をあらかじめ実行しておくことが好ましい。Ｗｅｂクローラー(crawler)は、この問題を利用し、ＨＴＭＬページとイメージを夜間に要求して、あらかじめ変換しておくことができる。そこでコンパイラは、Ｈ２Ｏとやり取りする。Ｈ２Ｏ ２４８は、ＳＰ内に用意される、インターネット・コンテンツ開発者がＳＰ上で実行されるネットワーク・オペレータ向けのインタラクティブ・テレビ、アプリケーション、およびサービスを製作できるようにする、好ましいクライアント／サーバ・ソリューションである。そこで、Ｈ２Ｏを利用すると、インターネット上の才能とコンテンツを集めた大きなプールを拡大一途のインタラクティブ・テレビ・アプリケーションの世界市場で活用することができる。Ｈ２Ｏサーバが、インターネット・コンテンツ（ＨＴＭＬページ、ＥＣＭＡスクリプト、およびＨＴＭＬページ書式設定）をＳＰアセットに変換する。Ｈ２Ｏクライアントであるｈ２ＯＣが、それらのアセットを描画し、クライアントとやり取りする。テレビ商取引／電子商取引事例のシナリオでは、Ｈ２Ｏを使用することで、テレビ商取引／電子商取引の店舗が既存のＷｅｂツールを使用して、ショッピング・サービスを構築し、標準のＷｅｂプロトコルを使用することにより好ましいＳＰ（オペレータ）とのインターフェイスを作成することができる。そこで、本発明では、知られている方法を用いて使い勝手のよいＡＰＩを提供する。
【０１７０】
Ｈ２Ｏは、ＳＧＷと、ＷｅｂコンテンツをＳＰコンテンツに変換するための放送ツールへのプロキシとして機能する。そのため、Ｗｅｂサイト開発者は、現在のＨＴＴＰサーバおよびアプリケーション・サーバを使用して、インタラクティブ・テレビ・コンテンツを安価に製作することができる。好ましい実施形態では、Ｈ２Ｏは、ＨＴＭＬ、Ｊａｖａ（登録商標）Ｓｃｒｉｐｔ、およびインターネット・グラフィックスを変換するが、他の知られているまたは開発されているインターネットまたはその他のコンテンツまたはプロトコ

ルもＨ２Ｏのプロキシ機能に追加することができる。Ｈ２Ｏを使用すると、ＳＰはブラウザ完全対応でないＳＴＢにＷｅｂページを表示し、オリジナルのユーザ・インターフェースを作成することができる。Ｈ２Ｏでは、ＳＰはＨＴＭＬのみを使用する商取引エンジンと接続することができる。Ｈ２Ｏは、ＨＴＭＬページ、ＪＰＧ画像、ｗａｖオーディオ・ファイルなどの現在または将来のブロードバンドおよびＷｅｂコンテンツすべてをＳＰリソースに変換する役目を持つ。

【０１７１】

Ｈ２Ｏのサーバ・サイドであるＨ２ＯＳはＨＴＴＰプロキシである。他の目的については、ダイナミック・リンク・ライブラリ（ＤＬＬ）またはバッチ・ツールとしてパッケージ化することができる。Ｈ２Ｏのクライアント・サイドであるＨ２ＯＣはＳＴＢ　Ｏコード・アプリケーションである。Ｈ２ＯＣは、ＳＧＷライブラリやカルーセル・ロード・ライブラリなどの他のＳＰクライアント・コンポーネントの上に構築される。Ｈ２Ｏでは、ＵＲＬを使用して、ドキュメントおよびサービスを取り扱うことができる。Ｈ２Ｏではさらに、放送およびオンライン環境で追跡することもできる。Ｈ２ＯＳは、ＨＴＴＰプロキシ機能を備えている。ＳＰアプリケーションは、Ｈ２Ｏを通じてドキュメントを要求し、その後、Ｈ２Ｏはそのドキュメントを取り出し、解析して、コンパイルし、要求側に返す。このＨ２Ｏ機能により、オンラインおよび放送の異なる用途に同じエンジンを使用できるため、拡張性が高く、Ｈ２Ｏを自由自在に使用することができる。解析は、ドキュメントの型によって異なり、解析にはＨＴＭＬ解析、ＧＩＦ画像、またはＪＰＥＧ画像などがある。拡張可能にするために、Ｈ２Ｏでは、「プラグイン」方式をとることができ、新しいサード・パーティ製のフィルタを実行することができる。

【０１７２】

Ｈ２Ｏでは、異なる言語を使用してスクリプトを作成できる。すべてのＳＰサーバ・コンポーネントで、監視機能周辺、特に異なるプロセスをリモートで管理する機能を標準化するのが好ましい。基本機能の処理にはＳＮＭＰを使用する（「プロセスＯＫ」、および主要な問題に対するトラップ）。ステータス検査用にコマンド・ライン・インタプリタをソケットで利用できるようにする。パラメータを設定して、リモート管理を行えるようにし、ＷｅｂスクリプトでＷｅｂとインターフェイスできるようにする。好ましい実施形態では、標準化された警告およびエラー・ログを用意する。

【０１７３】

ＨＴＭＬ／ＪＳでは、サーバ側でコンテキストを処理する際に、Ｗｅｂのページ同士で情報の共有を行うことができない。放送モードでは、この方式だと不足である。本発明では、好ましくは、グローバルな永続的オブジェクトが利用できる、つまり新規ページを開始してもクリアされない放送モードを実現する。永続的オブジェクトは、ページ間のコンテキストを保持する。ＳＰによって提供される他のベース・オブジェクトも、遷移後永続的にされる（例えば、局制御、ＯＳＤ）。インターフェイス定義言語を使用してガジェットを定義することで、新規ガジェットの作成、ガジェットの修正を行うことができ、またコンパイラを修正せずにメソッドの追加も可能になる。

【０１７４】

Ｈ２Ｏカルーセル機能を使用すると、カタログのリアルタイム更新、更新時のカタログ一貫性維持、安全なトランザクション・モデルの構築を行える。Ｈ２Ｏカルーセルにより、単一ページ、または単一トランザクションでのページ・セット全体の更新を行える。カルーセル管理機能では、カルーセル・インデックスまたはディレクトリの管理を行える。インデックスには、カルーセルのデータにアクセスし、フェッチするための情報が格納される。つまり、指定されたページでは、カルーセル管理に、必要なすべてのモジュールのリストが含まれ、Ｈ２ＯＣはプロセスを円滑に進めるため必要なすべてのモジュールを一度に要求する。

【０１７５】

カルーセルは、データ圧縮、ページ上のメタ・データ（例えば、ページ相対優先度、ページ送信頻度）、ページ追跡（基本ストリーム）などの機能を備える。カルーセル・クライ

アントはＳＴＢ　ＯＣＯＤライブラリであり、リソースのロードを取り扱う。カルーセル・クライアントは、リソースの動的変化、つまり新規リソース、リソース削除、およびリソース変更を管理する。動的リソース管理により、このライブラリのクライアント（Ｈ２ＯＣ）は動的コンテンツを提示できる。カルーセル・クライアントでは、メモリ割り当て、リソースのプリフェッチおよびキャッシュ、モジュールの圧縮解除を管理する。カルーセル・クライアントは、サブインデックス／ディレクトリを管理し、リソースの「ツリー」ではなくリソースの「セット」を管理するため、アセットの共有が容易になる。リソースの単一ツリーのサブセットを別々のプロセスに割り当てて、共有リソースを有効にすることができる。

【０１７６】

Ｈ２Ｏでは、共有モジュールで、テレビのトリガ・パフォーマンスおよび帯域幅、例えば、共有リソースを監視する。Ｈ２Ｏは帯域幅利用度を最適化する。Ｈ２Ｏは、データの入札ボリュームのマルチトラック、マルチプライオリティ、および管理の機能を持つ。Ｈ２Ｏは、モジュール・レベルで、リアルタイムのプリフェッチおよびキャッシングを実行する。Ｈ２Ｏは圧縮モジュールをサポートする。Ｈ２Ｏでは、矢印キーと直接キーによるナビゲーション（例えば、数字または色）をサポートしており、国際言語（中国語）、ページ上のメタ・データ（例えば、ページ相対優先度、送信頻度）、およびページ追跡（基本ストリーム）を扱える。グローバルＧＵＩは共有される、つまり、直接キー・リンクが用意されており、任意の情報ページを他のすべてのページから共有できる。

【０１７７】

Ｈ２Ｏでは、ページと下位ページを管理し、ページが大きすぎて１画面に収まりきらずスクロールしなければならない場合を処理する。Ｈ２Ｏでは、ＨＴＭＬを使用して、オンライン、ポイントツーポイント、および放送のコンテンツを表示できる。Ｈ２Ｏにより、放送およびオンライン・コンポーネントを混ぜたページを製作できる。例えば、オンライン・サーバからのページであるが、背景が放送される場合である。Ｈ２Ｏでは、ＳＴＢのコンテンツをマージすることができる。例えば、銀行アプリケーションは、視聴者の最近２０件のクレジット・カード・トランザクションをサーバから送信しながら、ＨＴＭＬページを放送することができる。ＪａｖａＳｃｒｉｐｔ機能でサーバに何らかのＸＭＬを要求し（ＨＴＴＰのように）、結果とＤＯＭ機能を使ってテーブルにその結果をパッチとして当てるのが好ましい。

【０１７８】

視聴者の認証を安全に行えるようにセキュリティ対策を施すことが好ましく、これは、Ｈ２ＯではなくＳＧＷで実行する。しかし、Ｈ２Ｏは、それと別の認証機能を備えることもできる。Ｈ２Ｏはさらに、暗号化されたデータを（例えば、クレジット・カード番号の送信で）ＳＴＢからオンライン・サーバに送信する。一部のサービスについては、ＨＴＭＬからＳＰへの変換近くのセキュリティ・プロキシに通すのが適切である。ＳＰでは、プロキシからサービス・プロバイダまでＨＴＴＰＳを利用し、ＳＴＢからプロキシまではＯＣＯＤライブラリのようなＳＳＬを使用することができる。他の場合（例えば、銀行）では、端から端までセキュリティ対策を講じ、その場合、Ｈ２Ｏは通常、変換を実行しない。このシナリオは、Ｈ２Ｏを通じて変換を行うことなくＳＴＢが処理できるデータ用に取っておくのが好ましい。それとは別に、ＳＧＷまたはＳＴＢで暗号化を実行することもできる。

【０１７９】

本発明は、好ましい実施形態のインタラクティブ・テレビについて説明されているが、本発明は、サーバとクライアント・デバイスを含む分散コンピュータ・システムで実現することもできる。他の実施形態では、本発明は、現在知られている、あるいは知られていない、ＲＯＭ、ＲＡＭ、ＣＤ　ＲＯＭ、フラッシュ、またはその他のコンピュータ読取り可能媒体などのコンピュータ読取り可能媒体上の一組の命令として実装され、これらの命令を実行するとコンピュータが本発明の方法を実行する。

【０１８０】

本発明の好ましい実施形態は上の発明により示されているが、例を示すことのみを目的としており、請求項で定義されている本発明の範囲を制限する意図はない。

【図面の簡単な説明】

【０１８１】

【図１】本発明が配置されるサービス・プラットフォームの好ましい実施形態を示す高水準アーキテクチャ図である。

【図２】本発明が配置されるサービス・プラットフォームのより詳細なアーキテクチャを示す図である。

【図３】データベースを示している好ましい実施形態の中間水準のアーキテクチャを示す図である。

【図４】本発明の好ましいアプリケーション・バックエンド・フレームワークの一実施例を示す図である。

【図５】本発明の好ましいＤＡＴＰ　ＳＴＢスタック・アーキテクチャの一実施例を示す図である。

【図６】アプリケーション・サーバとＳＧＷとの間の裏チャネル通信を標準化するために使用されるデジタルＴＶアプリケーション・プロトコル（ＤＡＰ）のサブセットとしての本発明のサービス・ゲートウェイ（ＳＧＷ）、デジタルＴＶアプリケーション・トランスポート・プロトコル（ＤＡＴＰ）の好ましい実施形態を示す図である。

【図７】ＤＡＰのサブセットとしてのＤＡＭＬおよびＤＡＴＰを示す図である。

【図８】本発明のＳＧＷの好ましいアーキテクチャの一実施例を示す図である。

【図９】本発明のスライディング・リジェクション・ウィンドウを示す図である。

【図１０】本発明の好ましい実施形態におけるアプリケーション・サーバとしてのＳＴＢとＳＧＷとの間のＤＡＴＰセッション例を示す図である。

【図１１】コンテンツ変換のアーキテクチャＨ２Ｏを示す図である。

【図１２】クライアント／ＳＴＢ、ＳＧＷ、Ｈ２Ｏ、およびアプリケーション・サービス・プロバイダの間のメッセージ・シナリオを示す図である。

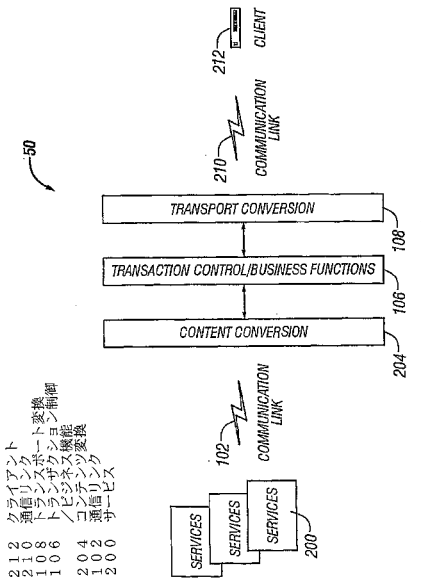
【図１３】クライアント／ＳＴＢ、ＳＧＷ、Ｈ２Ｏ、およびアプリケーション・サービス・プロバイダの間のメッセージ・シナリオを示す図である。

【図１４】クライアント／ＳＴＢ、ＳＧＷ、Ｈ２Ｏ、およびアプリケーション・サービス・プロバイダの間のメッセージ・シナリオを示す図である。

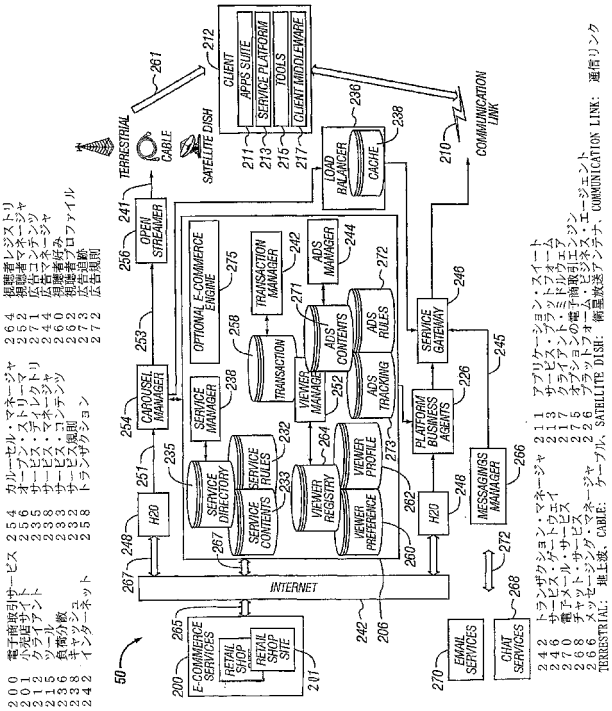
【図１５】クライアント／ＳＴＢ、ＳＧＷ、Ｈ２Ｏ、およびアプリケーション・サービス・プロバイダの間のメッセージ・シナリオを示す図である。

【図１６】クライアント／ＳＴＢ、ＳＧＷ、Ｈ２Ｏ、およびアプリケーション・サービス・プロバイダの間のメッセージ・シナリオを示す図である。

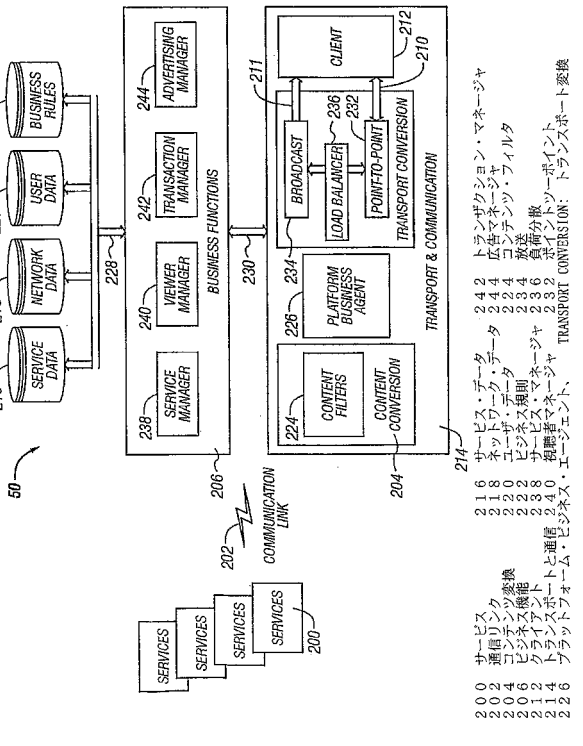
【図１】


50
TRANSPORT CONVERSION
TRANSACTION CONTROL/BUSINESS FUNCTIONS
CONTENT CONVERSION
108
106
204
210
COMMUNICATION LINK
212
CLIENT
102
COMMUNICATION LINK
SERVICES
SERVICES
SERVICES
200
212 クライアント
210 通信リンク
108 トランスポート変換
106 トランザクション制御／ビジネス機能
204 コンテンツ変換
102 通信リンク
200 サービス

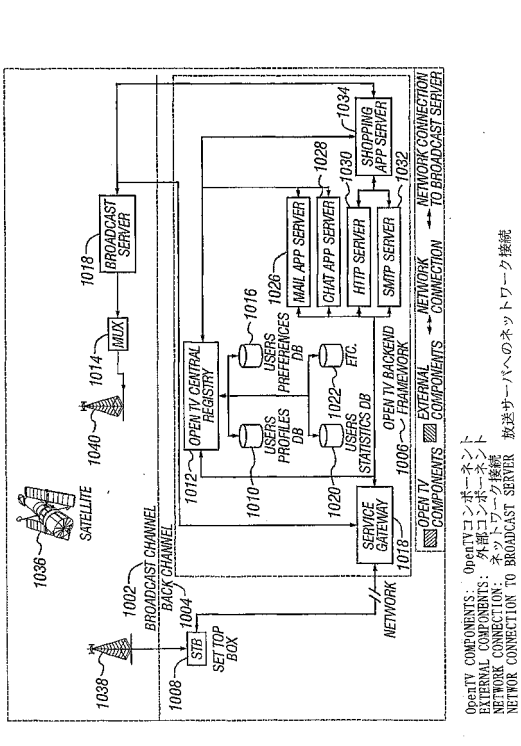
【図２】


50
INTERNET
TERRESTRIAL
SATELLITE DISH
CABLE
261
212
CLIENT
APPS SUITE
SERVICE PLATFORM
TOOLS
CLIENT MIDDLEWARE
211
213
215
217
236
238
LOAD BALANCER
CACHE
210
COMMUNICATION LINK
241
OPEN STREAMER
256
253
CAROUSEL MANAGER
254
OPTIONAL E-COMMERCE ENGINE
275
SERVICE MANAGER
238
SERVICE DIRECTORY
SERVICE CONTENTS
233
SERVICE RULES
232
TRANSACTION MANAGER
242
TRANSACTION
258
ADS MANAGER
244
ADS CONTENTS
ADS RULES
272
VIEWER MANAGER
252
VIEWER REGISTRY
VIEWER PREFERENCE
VIEWER PROFILE
273
ADS TRACKING
271
SERVICE GATEWAY
246
PLATFORM BUSINESS AGENTS
226
245
MESSAGINGS MANAGER
266
H20
248
251
235
267
262
260
264
E-COMMERCE SERVICES
265
200
RETAIL SHOP SITE
RETAIL SHOP SITE
207
242
206
EMAIL SERVICES
270
CHAT SERVICES
268
272
200 電子商取引サービス
201 小売サイト
215 ツール
236 負荷分散
238 キャッシュ
242 インターネット
254 カルーセル・マネージャ
256 オープン・ストリーマ
233 サービス・ディレクトリ
238 サービス・マネージャ
232 サービス・コンテンツ
258 トランザクション
264 視聴者レジストリ
252 視聴者マネージャ
271 広告マネージャ
244 視聴者好みプロファイル
260 視聴者ID
262 広告規則
272 広告情報
241 OPEN STREAMER
246 SERVICE GATEWAY
226 PLATFORM BUSINESS AGENTS
245 MESSAGINGS MANAGER
266 H20
248 CAROUSEL MANAGER
254 SERVICE MANAGER
238 SERVICE DIRECTORY
233 SERVICE CONTENTS
232 SERVICE RULES
TRANSACTION MANAGER
242 TRANSACTION
258 OPTIONAL E-COMMERCE ENGINE
275 ADS MANAGER
244 ADS CONTENTS
ADS RULES
272 VIEWER MANAGER
252 VIEWER REGISTRY
VIEWER PREFERENCE
VIEWER PROFILE
273 ADS TRACKING
271 LOAD BALANCER
CACHE
236
238
CLIENT
APPS SUITE
SERVICE PLATFORM
TOOLS
CLIENT MIDDLEWARE
211
213
215
217
COMMUNICATION LINK
210
TERRESTRIAL
SATELLITE DISH
CABLE
261
241 OPEN STREAMER
253 CAROUSEL MANAGER
254
255 SERVICE MANAGER
238
257 SERVICE DIRECTORY
SERVICE CONTENTS
233
SERVICE RULES
232
H20
251
235
267
248
INTERNET
265
E-COMMERCE SERVICES
200
RETAIL SHOP SITE
RETAIL SHOP SITE
207
242
206
50
EMAIL SERVICES
270
CHAT SERVICES
268
272
246 サービス・ゲートウェイ
213 サービス・プラットフォーム
215 クライアントのモバイル・ミドルウェア・エージェント
217 クライアントのモバイル・ミドルウェア・エージェント
236 負荷分散
238 キャッシュ
210 通信リンク
261 地上波、CABLE: ケーブル、SATELLITE DISH: 衛星放送アンテナ、COMMUNICATION LINK: 通信リンク

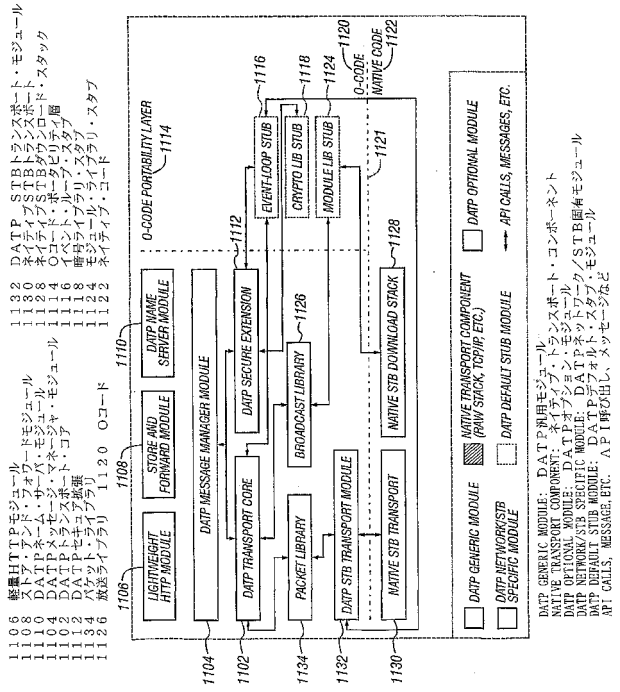
【図３】


50
SERVICE DATA
216
NETWORK DATA
218
USER DATA
220
BUSINESS RULES
222
228
BUSINESS FUNCTIONS
230
238 SERVICE MANAGER
240 VIEWER MANAGER
242 TRANSACTION MANAGER
244 ADVERTISING MANAGER
206
211
CLIENT
212
210
BROADCAST
LOAD BALANCER
236
232
POINT-TO-POINT
TRANSPORT CONVERSION
226 PLATFORM BUSINESS AGENT
224 CONTENT FILTERS
CONTENT CONVERSION
TRANSPORT & COMMUNICATION
214
204
202
COMMUNICATION LINK
SERVICES
SERVICES
SERVICES
SERVICES
200
200 サービス
202 通信リンク
204 コンテンツ変換
206 ビジネス機能
212 クライアント
214 トランスポートと通信
226 プラットフォーム・ビジネス・エージェント、
216 サービス・データ
218 ネットワーク・データ
220 ユーザ・データ
222 ビジネス規則
228 サービス・マネージャ
230 ビジネス・マネージャ
232 ポイントツーポイント
234 負荷分散
236 放送
240 視聴者マネージャ
242 トランザクション・マネージャ
244 広告マネージャ
224 コンテンツ・フィルタ
TRANSPORT CONVERSION: トランスポート変換

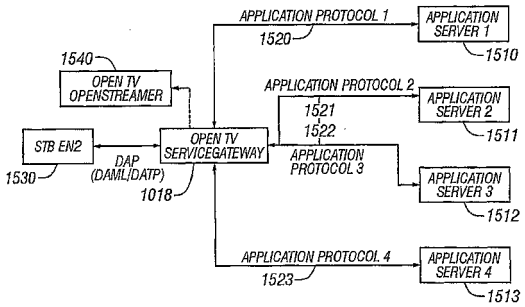
【図４】


1030 HTTPサーバ
1032 SMTPサーバ
1036 衛星
1014 マルチプレクサ
1018 放送サーバ
1002 放送チャネル
1004 裏チャネル
1008 セットトップ・ボックス
1012 OpenTV中央レジストリ
1016 ユーザ好みデータベース
1020 ユーザ統計データベース
1010 ユーザ・プロファイル・データベース
1006 OpenTVバックエンド・フレームワーク
1026 メール・アプリケーション・サーバ
1028 チャット・アプリケーション・サーバ
1034 ショッピング・アプリケーション・サーバ
1038 NETWORK: ネットワーク
SATELLITE
1036
1040
1014
MUX
1018
BROADCAST SERVER
1002
BROADCAST CHANNEL
BACK CHANNEL
1004
STB
SET TOP BOX
1038
1008
1012
OPEN TV CENTRAL REGISTRY
1016
USERS PREFERENCES DB
1010
USERS PROFILES DB
1022
USERS STATISTICS DB
ETC.
1020
SERVICE GATEWAY
1018
NETWORK
1006
OPEN TV BACKEND FRAMEWORK
1026
MAIL APP SERVER
1028
CHAT APP SERVER
1030
HTTP SERVER
1034
SHOPPING APP SERVER
SMTP SERVER
1032
OPEN TV COMPONENTS
EXTERNAL COMPONENTS
NETWORK CONNECTION
NETWORK CONNECTION TO BROADCAST SERVER
OpenTV COMPONENTS: OpenTVコンポーネント
EXTERNAL COMPONENTS: 外部コンポーネント
NETWORK CONNECTION: ネットワーク接続
NETWORK CONNECTION TO BROADCAST SERVER 放送サーバへのネットワーク接続

【図５】

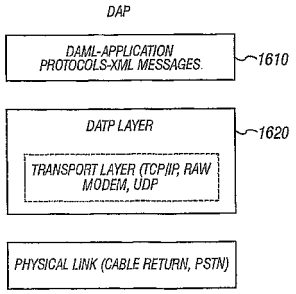


【図６】



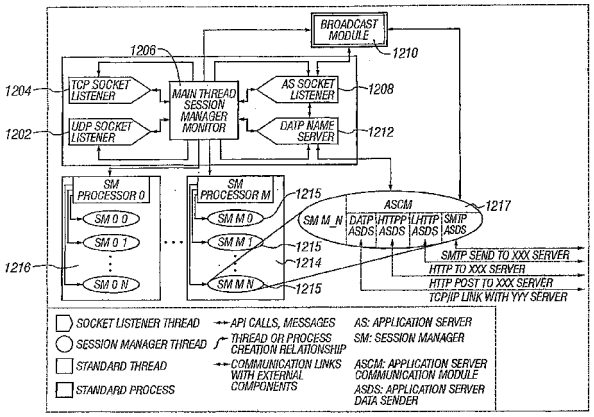
1540 OpenTVオープン・ストリーマ
1018 OpenTVサービス・ゲートウェイ
1520 アプリケーション・プロトコル1
1510 アプリケーション・サーバ1
1521 アプリケーション・プロトコル2
1511 アプリケーション・サーバ2
1522 アプリケーション・プロトコル3
1512 アプリケーション・サーバ3
1523 アプリケーション・プロトコル4
1513 アプリケーション・サーバ4

【図７】



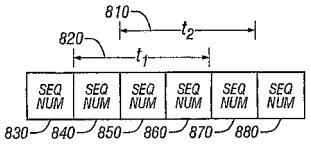
1610 DAML－アプリケーション・プロトコルーXMLメッセージ
1620 DATP層
TRANSPORT LAYER: トランスポート層
PHYSICAL LINK(CABLE RETURN, PSTN): 物理リンク（ケーブル、リターン、PSTN）

【図８】



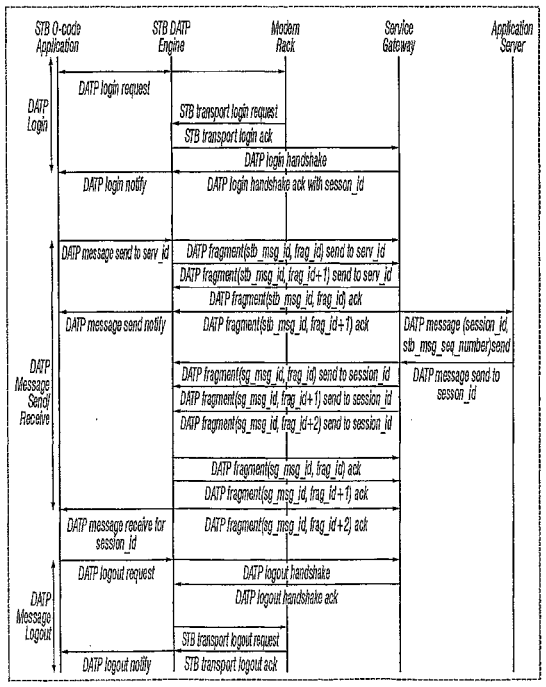
1210 放送モジュール　1206 メイン・スレッド・セッション・マネージャ・モニタ
1204 TCPソケット・リスナー　1208 ASソケット・リスナー
1202 UDPソケット・リスナー　1212 DATPネーム・サーバ
SMTP SEND TO XXX SERVER: SMTPでXXXサーバに送信
HTTP TO XXX SERVER: HTTP、XXXサーバへ
HTTP POST TO XXX SERVER: HTTP、XXXサーバにポスト
TCP/IP LINK WITH YYY SERVER: YYYサーバとTCP／IPリンク
SOCKET LISTNER THREAD: ソケット・リスナー・スレッド
SESSION MANAGER THREAD: セッション・マネージャ・スレッド
STANDARD THREAD: 標準スレッド、STANDARD PROCESS: 標準プロセス
API CALLS, MESSAGES: API呼び出し、メッセージ
THREAD OR PROCESS CREATION RELATIONSHIP: スレッドまたはプロセス作成関係
COMMUNICATION LINKS WITH EXTERNAL COMPONENTS: 外部コンポーネントとの通信リンク
AS: APPLICATION SERVER: AS：アプリケーション・サーバ
SM:SESSION MANAGER: SM：セッション・マネージャ
ASCM: APPLICATION SERVER COMMUNICATION MODULE: ASCM：アプリケーション・サーバ通信モジュール
ASDS: APPLICATION SERVER DATA SENDER: ASDS：アプリケーション・サーバのデータ・センダー

【図９】



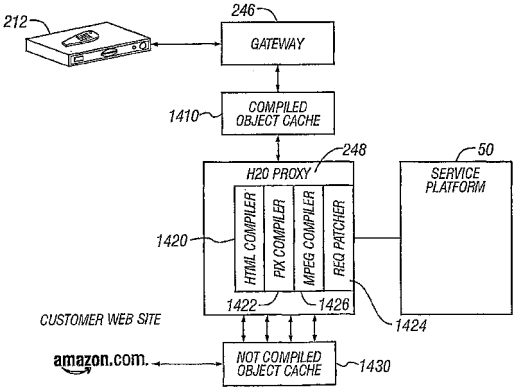
拒絶リスト

【図１０】



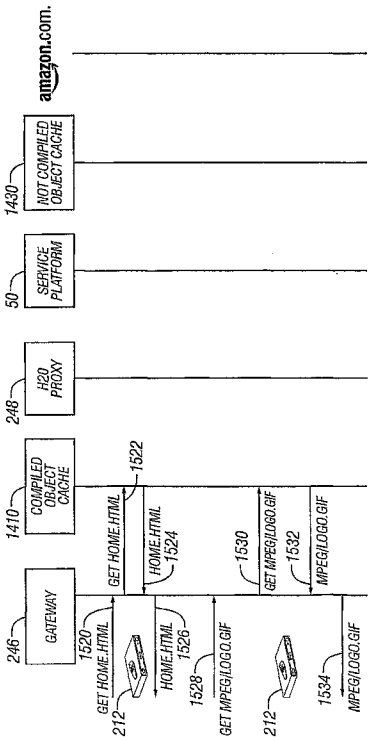
STB O-Code Application: ＳＴＢ　Ｏコード・アプリケーション
STB DATP Engine: ＳＴＢ　ＤＡＴＰエンジン
Modem Pack: モデム・ラック
Service Gateway: サービス・ゲートウェイ
Application Server: アプリケーション・サーバ

【図１１】



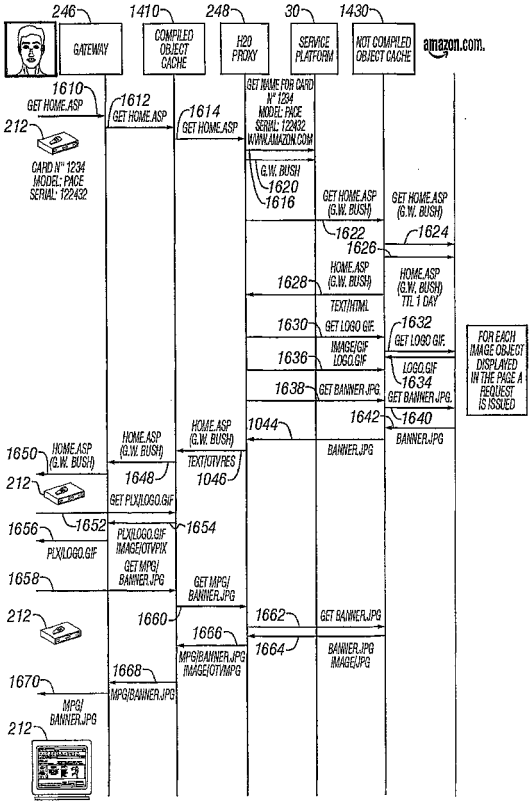
２４６　ゲートウェイ　　１４２２　ＰＩＸコンパイラ
２４８　Ｈ２Ｏプロキシ　　１４２６　ＭＰＥＧコンパイラ
１４２０　ＨＴＭＬコンパイラ　　１４２４　ＲＥＱパッチャ
５０　サービス・プラットフォーム
１４１０　コンパイルド・オブジェクト・キャッシュ
１４３０　ノットコンパイルド・オブジェクト・キャッシュ
CUSTOMER WEB SITE:　顧客Ｗｅｂサイト

【図１２】



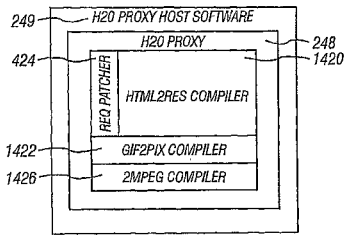
２４６　ゲートウェイ
１４１０　コンパイルド・オブジェクト・キャッシュ
２４８　Ｈ２Ｏプロキシ
５０　サービス・プラットフォーム
１４３０　ノットコンパイルド・オブジェクト・キャッシュ

【図１３】



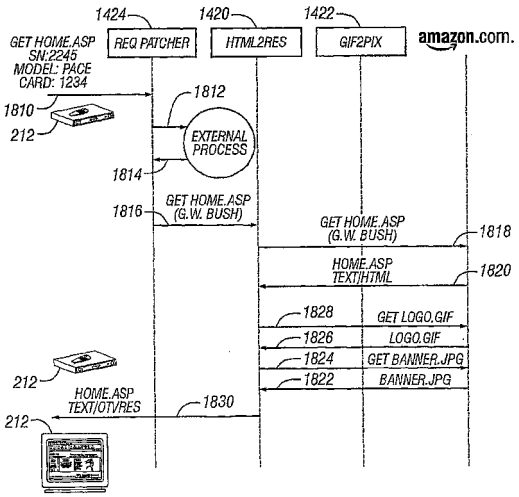
２４６　ゲートウェイ
１４１０　コンパイルド・オブジェクト・キャッシュ
２４８　Ｈ２Ｏプロキシ
３０　サービス・プラットフォーム
１４３０　ノットコンパイルド・オブジェクト・キャッシュ
FOR EACH IMAGE OBJECT DISPLAYED IN THE PAGE A REQUEST IS ISSUED:
ページ内に表示されるイメージ・オブジェクト毎に要求が発行される

【図１４】



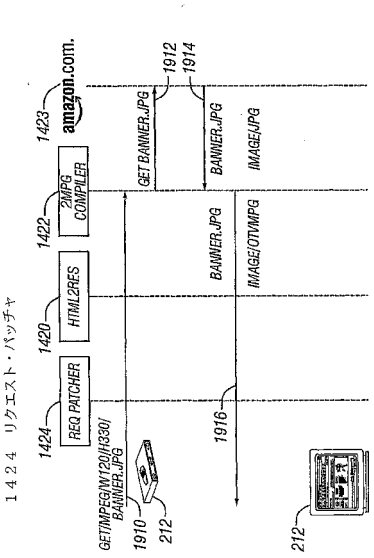
２４９　Ｈ２Ｏプロキシ・ホスト・ソフトウェア
２４８　Ｈ２Ｏプロキシ
４２４　リクエスト・パッチャ
１４２０　ＨＴＭＬ２ＲＥＳコンパイラ
１４２２　ＧＩＦ２ＰＩＸコンパイラ
１４２６　２ＭＰＥＧコンパイラ

【図１５】



１４２４　リクエスト・パッチャ
１８１２　外部プロセス

【図１６】



【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 August 2002 (15.08.2002)

PCT

(10) International Publication Number
WO 02/063879 A2

(51) International Patent Classification[7]: H04N 7/173

(21) International Application Number: PCT/US02/02725

(22) International Filing Date: 1 February 2002 (01.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/265,986	2 February 2001 (02.02.2001)	US
60/266,210	2 February 2001 (02.02.2001)	US
60/267,876	9 February 2001 (09.02.2001)	US
60/269,261	15 February 2001 (15.02.2001)	US
60/279,543	28 March 2001 (28.03.2001)	US
09/858,436	16 May 2001 (16.05.2001)	US

(71) Applicant: OPENTV, INC. [US/US]; 401 E. Middlefield Road, Mountain View, CA 94043-4005 (US).

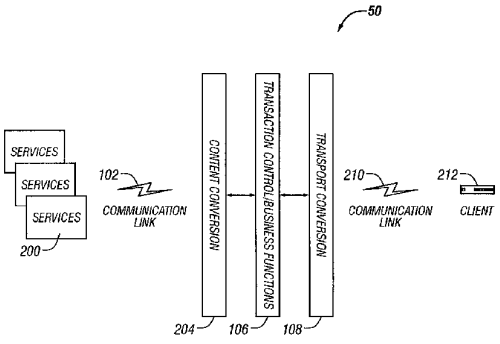
(72) Inventors: ALAO, Rachad; 330 Angel Avenue, Sunnyvale, CA 94086 (US). DELPUCH, Alain; 20, avenue André Prothin, F-92927 Paris la Défense (FR). DUREAU, Vincent; 3519 South Court, Palo Alto, CA 94306 (US). HENRARD, Jose; 14, rue de Liège, F-75005 Paris (FR). HUNTINGTON, Matthew; 23 Gordon Avenue, Twickenham TW1 1NH (GB). LAM, Waiman; 2137 Sunsprite Drive, Union City, CA 94587 (US). KIDD, Taylor; 977 Upland Road, Redwood City, CA 94062 (US).

(74) Agent: ROEBUCK, G., Michael; Madan, Mossman & Sriram, P.C., 2603 Augusta, Suite 700, Houston, TX 77057 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

[Continued on next page]

(54) Title: A SERVICE GATEWAY FOR INTERACTIVE TELEVISION



(57) Abstract: A service gateway provides a proxy between a client protocol and a plurality of standard communication protocols. The service gateway provides asymmetrical routing, data compression and encryption to optimize client processing power and communication link bandwidth. The service gateway enables content translation between clients and service providers. The service gateway keeps track of client available memory and sequence numbers in messages to generate error codes when applicable. A store and forward message capability is provided along with abstract session identifiers. The service gateway supports user datagram protocol.

WO 02/063879 A2

WO 02/063879 A2

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

TITLE: **A SERVICE GATEWAY FOR INTERACTIVE TELEVISION**

INVENTORS: **RACHAD ALAO; ALAIN DELPUCH; VINCENT DUREAU; JOSE HENRARD; MATTHEW HUNTINGTON; WAIMAN LAM; TAYLOR KIDD**

Copyright Notice

Background of the Invention

Field of the Invention

The present invention relates to the field of communications in the interactive television environment and specifically relates to a method and apparatus for providing a generic meta language and digital television application protocol for interactive television.

Summary of the Related Art

Interactive television systems can be used to provide a wide variety of services to viewers. Interactive television systems are capable of delivering typical video program streams, interactive television applications, text and graphic images, web pages and other types of information. Interactive television systems are also capable of registering viewer actions or responses and can be used for such purposes as marketing, entertainment and education. Users or viewers may interact with the systems by ordering advertised products or services, competing against contestants in a game show, requesting specialized information regarding particular programs, or navigating through pages of information.

Typically, a broadcast service provider or network operator generates an interactive television signal for transmission to a viewer's television. The interactive television signal may include an interactive portion comprising of application code or control information, as well as an audio/video portion comprising a television program or other informational displays. The broadcast service provider combines the audio/video (A/V) and interactive portions into a single signal for transmission to a receiver connected to the user's television. The signal is generally compressed prior to transmission and transmitted through typical broadcast channels, such as cable television (CATV) lines or direct satellite transmission systems.

Typically, the interactive functionality of the television is controlled by a set top box (STB) connected to the television. The STB receives a broadcast signal transmitted by the broadcast service provider, separates the interactive portion of the signal from the A/V portion of the signal and decompresses the respective portions of the signal. The STB uses the interactive information, for example, to execute an application while the A/V information is transmitted to the television. The STB may combine the A/V information with interactive graphics or audio generated by the interactive application prior to transmitting the information to the television. The interactive graphics and audio may present additional information to the viewer or may prompt the viewer for input. The STB may provide viewer input or other information to the broadcast service provider via a modem connection or cable.

In accordance with their aggregate nature, interactive television systems provide content in various content forms and communication protocols the must be understood and displayed by the STB/client that receives the information from the broadcast service provider/network operator. Typically the client is a STB having a processor possessing limited processing power. Translation of the various contents and protocols is beyond the limited processing capability available in the typical STB processor. Thus there is a need for a service gateway that receives a simple communication protocol which can be easily understood by the client/STB processor and tnanslates the simple protocol into a plurality of standard protocols used by service providers. There is also a need for a software and

hardware architecture that provides adaptive control of access, content and scheduling in an interactive television environment.

Summary of the Invention

The present invention addresses the needs of the interactive television environment discussed above. The present invention satisfies a long felt need to provide a simple content and communication protocol than can be easily handled by a STB processor and enables complex communication with the head-end operator's service platform (SP) or a server, its subscriber clients and a plurality of service providers. While the following discussion uses the example of a client/STB, the present invention applies to all client devices including digital assistants, cell phones, pocket personal computers or any other types of electronic device capable of receiving an electronic signal. The present invention resides in a service platform (SP). The SP or server enables a network operator, who provides television signals to its subscriber clients, to create and provide business, transport and communication functions that enable communication between service providers and the client or STB viewer via the service gateway.

The interactive television environment must deal with and solve problems that are unique to interactive television, such as the intermittent return path from the client to the SP. That is, the client device is not always connected to the communication link as when the STB is turned off. Thus, there is not always an active return path from the client. The present invention provides a store and forward function to alleviate this intermittent return path problem.

Bandwidth and processing limitations and communication complexities are also problematic in the interactive television environment. On one hand the network operator typically provides a broadcast channel with a relatively large data transmission capacity (typically a satellite and dish) to send data and programming to the client. On the other hand, the client return path has a relatively low data transmission capacity, usually in the STB scenario, a telephone line is the return path. Even if the return path happens to have a larger bandwidth, STBs/clients typically possess a low speed modem to send data on

the return path. Processing limitations limit the ability of a STB or client to process the spate of communication protocols utilized by the service providers communicating with the STB.

The present invention also provides a method of optimally transferring content from the Head End Server to the set top box in a manner that optimally allocates bandwidth. This aspect of the invention comprises an application for generating statistical calculations, residing entirely within the set top box, that calculates the latency of different channels in an interactive television environment. This aspect of the invention further comprises using the results of these calculations to pull content from the head end server into a set top box in a manner that minimizes the amount of time required for transferring content. A preferable definition of the latency of a channel is the length of time between the issuance of a request and the fulfillment of that request. A report is generated by randomizing the issuance of a small portion, preferably less than 5%, of file requests over all available channels, in order to sample latency times. A channel can be used with increased or decreased frequency, with the frequency of use based upon the results of the statistics. More detailed results can be obtained by further delineating a set of priorities to the statistics. As an example, the invention records the relation between the size of a file and latency on a given channel. The Head End Server increases or decreases the appearance of a resource in the broadcast carousel depending on network demand. The present invention also provides a message flow rate controller for controlling message flow rate by controlling the bit rate of transmission. These and other issues are addressed by the present invention.

Brief Description of the Drawings

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

Figure 1 illustrates a high level architecture diagram for a preferred embodiment of a service platform in which the present invention resides;

Figure 2 illustrates a more detailed architecture for a service platform in which the present invention resides;

Figure 3 illustrates a mid level architecture for a preferred embodiment showing data bases;

Figure 4 illustrates an example of a preferred application backend framework for the present invention;

Figure 5 illustrates an example of a preferred DATP STB stack architecture of the present invention;

Figure 6 illustrates a preferred embodiment of the Service Gateway (SGW), Digital TV Application Transport Protocol (DATP) of the present invention as a subset of the Digital TV Application Protocol (DAP) used to standardize back channel communications between application servers and the SGW;

Figure 7 illustrates DAML and DATP as a subset of DAP;

Figure 8 illustrates an example of a preferred architecture for the SGW of the present invention;

Figure 9 illustrates the sliding rejection window of the present invention;

Figure 10 illustrates a sample DATP session between a STB and the SGW, as an application server in a preferred embodiment of the present invention;

Figure 11 illustrates an architecture for content translation, H2O; and

Figures 12-16 - illustrate message scenarios between the client/STB, SGW, H2O and application service providers.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

Detailed Description of A Preferred Embodiment

Overview

The present invention, a service gateway resides in head-end operator's service platform (SP) and interacts with the content transcoder, H2O and a digital television application transport protocol. In a typical interactive television environment, there are a multitude of clients/subscribers, typically STBs that must communicate with a multitude of application servers providing content over a multitude of networks using various communication protocols. Typically the STB has limited processing power so that it is undesirable to place a multitude of communication protocol handlers in the STB processor or STB stack. Thus, there is a need for a common communication interface that can address all the STBs and application servers. The present invention, the service gateway provides a communication protocol proxy that requires light processor utilization, well-suited for a typical STB possessing limited processing power and the SP. The service gateway enables the use of DATP which requires relatively few processing cycles compared to typical Internet communication protocols. DATP reduces the overhead of the communication protocol handler at the STB and makes the communication protocol handler common for all STBs. The DATP protocol is portable for all STBs since it is written in O-code, a STB independent byte code that interfaces with the operating system of the STB.

In the present invention, a SGW performs as a DATP server communication proxy and asymmetrical router. SGW enables SP clients at STBs to connect to application servers using DATP protocol. An HTML to native code proxy, H2O is provided that can be considered in this context as an SP application server. H2O performs specific content translation, such as HTML to SP O-codes. O-codes are the STB independent byte code of the virtual machine running on the SP. In a preferred embodiment, an O-code implementation of the DATP protocol stack exists in the client, typically a STB. The client communicates using DATP protocol to a DATP server, SGW. The H2O proxy exists on the other side of the SGW performing content translation such as HTML to O-code. An O-code implementation of a DATP stack in the

client/STB issues communication requests and communicates with SGW using DATP protocol. Content translated by H2O is passed through the SGW to the client where content is displayed.

SGW provides a DATP server function, which creates execution threads to handle each individual STB and process each related content. The SGW server stack communicates with the client/STB using DATP protocol. SGW also applies the appropriate protocol needed to enable the STB to communicate back and forth between the STB and different application servers via the SGW. Interactive television applications typically utilize well known Internet based protocols (HTML, etc.) to communicate back and forth between the client/STB and application servers. The present invention, SGW provides a generic and well-suited asymmetrical communication protocol between the client/STB and application servers via the SGW. The present invention accommodates the minimal processing and memory available at the client/STB.

The SGW provides an asymmetrical solution to data compression. The band width of the bi-directional path from the client/STB to the network operator is relatively small, typically , however, a regular telephone line or a return channel in a cable and usually connected to a low speed modem. Thus, to increase the bandwidth available over the low speed modem, the content down loaded from the server to the client/STB is compressed. At the client/STB, however, data compression is preferably not performed. The client/STB data returned is relatively small and not in need of data compression by the STB processor which typically does not have the processing power to perform data compression. In an alternative embodiment, there are, however, instances where data compression from the client/STB is desired and in this case data compression is performed at the SGW. Data compression, with respect to the client/STB is asymmetric in that data is compressed going down stream to the client/STB and is not compressed coming upstream from the STB. Thus, the architecture of the present invention is asymmetric, unlike typical Internet-based protocols where both entities communicating are assumed to be symmetrically powered.

The SGW and client/STB communicate with application servers utilizing session identifiers for clients rather than user identifiers so that the client users remain anonymous. The present invention also provides multicasting to clients. A multicast message can be sent to multiple clients via a broadcast link, when broadcast bandwidth and a tuner is in the STB and broadcast messages are available and sensed by a particular filter setup in the STB. SGW via DATP requests that the STB receive a message from a specific entry on the broadcast. If no tuner is available to receive the broadcast in the STB, message fragments are also sent on each point to point individual link to the STBs without a tuner. If the STBs are on a LAN, messages are sent to a well known address on the LAN to the STBs.

The present invention also provides a novel structure and method for handling cookies from Internet applications and provides a "light" HTTP protocol, LHTTP which encapsulates HTTP requests within DATP messages. LHTTP is a simplified version of HTTP that runs on top of DATP, which the SGW converts into HTTP for communication with service providers. The novel LHTTP runs on top of DATP and does not implement any part of the TCP/IP specification.

SGW establishes a link or a socket connection with a STB. To implement User Datagram Protocol (UDP), however, UDP is not performed directly. For a STB that can output UDP, encapsulates DATP on top of UDP. The DATP-encapsulated UDP is sent to the SGW. In the case of UDP, a socket in the SGW and a socket in the STB are effectively bound together in a simulated connection on top of UDP. Through this SGW-simulated connection, DATP packets are sent from the STB to the SGW server and from the SGW server to the STB.

Many STB modems do not provide data compression, possess minimal processing capability and cannot afford the processing cost to perform data compression in the STB. Thus in a preferred embodiment, the SGW provides asymmetrical data compression is performed at the SGW. The STB does not compress data. STB receives compressed data and decompresses it, however, the STB does not perform data compression which is

peformed by the SGW. Data decompression, however, is less compute intensive than data compression, thus, the STB preferably performs decompression.. Compressed data is sent to the DATP stack at the STB but uncompressed data is sent from the STB to the SGW. SGW performs data compression on the uncompressed data sent from the STB and SGW returns the compressed data to application servers. Thus, the preferred DATP/SGW asymmetric compression increases the bandwidth of the return path from the STB through the SGW to the application servers.

The present invention, SGW provides asymmetrical routing. In asymmetrical routing a portion of the bandwidth is allocated to SGW to send data to the broadcast stream for broadcast. SGW has the ability to decide whether to send data to one or more STBs over the broadcast stream or a point to point (PTP) connection between the SGW and the STB(s). SGW routes data via broadcast or PTP, based on the amount of data, the speed of the point to point link to the STB(s) and the current communication links loading conditions. Thus, SGW may decide not to send a data set over the point to point link because the data set is too large and instead send it over the broadcast stream. The data can be compressed by SGW before sending it to the recipient stream or point to point link to increase the bandwidth of the link between SGW and the link or stream and to accommodate memory limitations in the STB.

SGW enables DATP to be computationally light weight because it is designed so that all STB stack operations require a minimum of processing power. For example, in the DATP encryption scheme, when using Rivest, Shamir and Alderman (RSA) public key encryption, the key that comes from the server is chosen so that the its exponent is small (3 or greater) so that exponentiation phase takes a minimal amount of time and processing power. Thus the heavy computation is reserved for the SGW and thus, the STB or client processor requires minimum processing capability. Likewise the LHTTP layer on top of DATP in the STB does not have to perform any heavy parsing or other processing intensive operations. Instead, HTTP data is encapsulated in DATP messages by LHTTP and the HTTP compute intensive functions, such as conversion to HTTP protocol are handled by SGW.

DATP performs more than transactions. Rather, DATP is a message-based protocol rather than a transaction oriented protocol, thus, when a user sends a message from a STB to an application server, the application server does not have to respond. That is, there is not a one-to-one correspondence between STB and service provider messages. All DATP messages, except the class of unreliable DATP messages are processed through a DATP reliably layer. All DATP messages have unique identifiers which can be used as the basis of a transaction.

In a transaction using DATP, for example a HTTP request, the STB sends a DATP message to the SGW requesting a Web page. SGW converts LHTTP to HTTP and sends it to the Internet via H2O. Once the response containing the Web page returns from the Internet to SGW via H2O, which converts the content, SGW sends a LHTTP DATP message to the STB returning the content of the requested Web page to the STB. Another example of a transaction is a Fetchmail request sent from a STB. The Fetchmail request is encapsulated in a DATP message. DAML is used on top of the DATP message. DAML is a domain specific instance of XML.

Thus, the STB sends a DATP message to Fetchmail containing a DAML (XML) request. Fetchmail reads the DATP message and extracts the content from the message, passes the content to the application server which processes the transaction and returns a message to Fetchmail. Fetchmail then sends a DATP message containing requested content to the STB.

SGW also provides a store and forward function to handle peaks in numbers of orders sent in from multiple users, while rapidly reacting to the user order request. SGW quickly sends an "order acknowledge" to the user in response to user's order and stores the order for transmission later to the application server, which will actually process the order transaction. By sending the order later, a large number of orders can be spread out over time and not have to be sent all at once to the application server. Thus, bandwidth is

efficiently utilized. DATP/SGW also provides a sliding rejection window based on message fragment sequence numbers versus time. DATP/SGW are discussed in detail below.

The Service Platform

Turning now to **Figure 1**, the SP in which the SGW of the present invention resides is presented. The SP **50** comprises a group of applications roughly divided into three categories, Content Conversion **204**, Transaction Control/Business Functions **106** and Transport Conversion **108**. The SP enables services **200** to interact with a client **212**. The services **200** communicate through a communication link **102** to the SP **50**. The SP **50** in turn communicates with a client **212**. The client **212** may be a STB, a digital assistant, a cellular phone, or any other communication device capable of communicating with the SP through communication link **210**. The content conversion **204** and transport conversion **108** services provide the transport and communication function, and the business function services provide the business control functions.

Figure 2 illustrates an example of a preferred implementation of Service Platform **50**. Services **200** provide shopping, chat, and other services either over the Internet or over another network or communication channel accessible to the network operator. Using the SP, the network operator accesses those services. Business functions **206**, comprising service manager **238**, interact with carousel manager **254** to retrieve content from a service **200**. The carousel comprises a repeating stream of audio/video/interactive data broad cast to clients from the SP **50**. Carousel manager **254**, transaction manager **242** and service manager **238** control the content insertion and deletion from the broadcast carousel. Service content is retrieved and converted into a SP suitable format by H2O **248**. H2O **248** is a possible implementation of content conversion **204**. H2O converts HTML content into SP/client readable content. The converted content is formatted into a data carousel and multiplexed by the Open Streamer **256** for broadcast to the client **212**. Client **212** interacts with the services and if necessary communicates with the SP and the services **200**. PTP communication goes through SGW **246**. SGW **246** performs transport conversion to convert the STB DATP protocol into a standare communication protocol

which the Platform Business Agents **226** and H2O **248** understand. Load balancer **236** interacts with business functions **206**, carousel manager **254**, and SGW **246** to determine the optimal load between the broadcast link **241** and the PTP communication link **210**. Business functions **206**, interact with the platform business agents **226** to control access and information exchange between the services **200** and client **212**.

Services **200** negotiate with a network operator to offer services to subscribers via the operator's Service Platform. As shown in **Figure 3**, the network operator uses the Service Manger **238** to register the services and the negotiated business rules **222** (e.g. schedule, bandwidth requirements, service access to viewer information) associated with the service. The Service Manager **238** stores Service data **216** (e.g. URL address, content). Based on the business rules **222** and Service Data **216**, Service Manager **238** communicates with the Broadcast Communication **234** function to retrieve the content from the content provider.

When the content is retrieved from the Service **200**, it may be processed by the Content Conversion **204** and Content Filters **224** to convert the content into a form suitable for the client device **212**. The Broadcast **234** function converts the content into a form suitable for the broadcast **234** network. The converted content is received by the client **212** over broadcast link **241**. Client **212** and Service **200** interact via Point-to-Point link **210** and Point-to-Point function **232**, which are part of Transport Conversion **207**. The service **200** may comprise shopping, audio/video, gaming, voting, advertisement, messaging, or any other service.

Client **212** communicates through Point-to-Point **232** communication link to the Service Platform **50** and Service **200**. Load Balancer **236** interacts with the Business Functions **206** to determine the optimal load distribution between the Broadcast **234** Communication link **241** and the Point-to-Point **232** Communication link **210**. The Platform Business Agents **226** use business rules **222** to control the interaction and exchange of information between the Service **200** and the client **212**. For example, the network operator may choose to prevent Service **200** access to user information. Service

200 must pay a fee based on the Business Rules **222** and Service data **216** to access the user information.

Viewer Manager **240** stores client/user information in User Data **220**. Platform Business Agents **226** control the flow of viewer information to the Service **200**. Transaction Manager **242** records transactional information exchanged between the service **200** and Client **212**. Based on the Business Rules **222** and the User Data **220**, Advertising Manager **244** determines which advertisements and which type of advertisements will be presented to the client via Broadcast **234** link **241** and Point-to-Point **232** link **210**. The Service Platform Transaction Manager records all transactions in the Transaction Database to ensure accurate operator revenue collections (even when the STB is turned off) and subscriber profiles in Viewer Profile **162** and Viewer Category **160** (viewer buying and viewing habits), which provides added value data to the network operator.

The transaction log is also useful for mining a user's viewing and transaction data for generating cumulative user profiles or used for more sophisticated profiling techniques such as collaborative filtering. Viewers or clients are placed in one or more categories (e.g., "sports fan", "chef-French") based on viewer user profile. Categories enhance the network operator's ability to perform adaptive targeted advertising and broadcasting based on long term and short-term viewing and buying trends of the viewer/client.

The Service Platform provides a Wallet function, which provide a checkout/purchase function. The wallet function is supported by the Service Platform, although selected shops or services may bypass the wallet function in favor of their own checkout procedure. The Wallet function records and accesses data regarding the viewer profile, viewer category, and transaction log. Thus, the Wallet minimizes typing of input data by the user. This function is particularly useful when user is ordering via a TV remote control with limited data entry capacity. Typically, content for the Wallet function is a credit card number stripped out of 4 last digits, so that the user only has to complete the last 4 digits or the shipping address for confirmation.

Optionally, all user information is placed in the Wallet Function, hidden from the service. Wallet information exists in both the Service Platform and the client. The Wallet information on the Service Platform preferably contains as much information as possible, including client shipping address, client full credit card number, etc. The Wallet information in the STB may only contain partial information such as the last 4 digits of the credit card to remind the viewer which credit card on which a purchase was made. Partial information is stored in the STB for protection from untrusted persons using the STB and untrusted applications, which can access the STB data.

As linking occurs, when a viewer click triggers the call of an E-Commerce application or service, the Service Platform determines the subscriber's navigation location and records it in the Transaction Database **158**. The Service Platform also determines and records when the viewer followed a link to a store, or which program the viewer was watching when he/she made the decision to purchase (referred to as an "impulse buy"). The Transaction Database **258** enables the operator to store and provide a detailed context and purchase history to subscriber. Such storage of context and purchase history is also useful to improve subscriber profile and category information and which can generate additional revenue and/or become part of a channel deal with an E-Commerce provider.

The Service Platform enables a network operator to facilitate E-Commerce deployment on its own network and to capture a share of the E-Commerce revenues. The Service Platform Personal Wallet function enables the network operator to manage credit, impose spending limits and enable micro-payments. Services may interact with the Service Manager and Viewer Manager to provide a group of services from which the Service Platform chooses to present to clients based on viewer profiles and categories. Operator may conversely tell a service it subscriber profile to request specialized viewer targeted offerings and advertisements to subscribers.

Viewer Manager **252** manages subscriber/user registration **264**, preference, and profile information **262**. Viewer Manager **252** enables users to register and record personal information in a database. The personal information comprises viewing patterns, promotional preferences, personal, wallet and demographic information, etc. Based on this recorded information and the user's activities, the Viewer Manager **252**

generates profile information to categorize the user and produce targeted services, content and advertisements to suit the user's profile **262** and expected preferences and needs. The Viewer Manager **252** also performs centralized updating of service and viewer parameters.

The Service Platform, via the SGW and supporting functions, enables network operators to control access to the viewer database and allow only those service providers who have a prior contract or agreement with the network operator to access privileged information (e.g., credit card numbers, viewer's actual name, home address, telephone number, social security number, etc.). For distributed functions, that is, when the client has sufficient processing power and storage, the Viewer Manager **252** enables access to personal and profile information stored on the client devices and enables the client devices to select user-preferred content. Clients select user-preferred content via business filters in the client device (e.g., STB).

The Viewer Manager **252** provides Household/Subscriber/STB (or other client device) identification and authentication in support of the Service Gateway and Parental Control functions. The Viewer Manager **252** supports Multiple Viewer identification and Registration authentication at a single STB through nicknames and personal identification numbers. The viewer identifier preferably is derived from the client device identifier number(s). The Viewer Manager **252** provides household and individual viewer profiling through logging, generation, and matchmaking linked to observed cumulative TV viewing and purchasing habits in support of SGW. The Viewer Manager supports Distributed data capture and storage between the Service Platform and the STB, and supports bi-directional synchronisation. The Viewer Manager **252** enables Distributed profile usage between all Service Platform applications and provides synchronisation with an external SMS/CRM.

The Viewer Manager **252** enables multiple viewer registrations for a single STB or client device using abstract viewer identifiers comprising nickname, full name and PIN Storage in the STB or other client device. Business Agents **226** enforce transactional business rules for interaction between service providers and viewers. Based on business rules, which are defined by the network operators and based on agreements with the

service providers, the Business Agents **226** control transactions and service provider access to user information. Business Agents **226** insert, replace and delete viewer information during a transaction.

Business Agents **226** in conjunction with SGW **246** create sessions between subscribers and service providers. SGW/Business Agents **226** can control access to viewer information details and manipulate viewer information by inclusion, replacement and removal of viewer information presented to Service Providers. SGW/Business Agents provide default values and control access to user information. SGW/Business Agents also perform Transaction Logging, Messaging Logging, Load/Transaction Monitoring.

Advertising Campaign management makes use of viewer data mining and analytic systems in order to propose the best selection of products, advertisements and timing for broadcast. The Service Platform provides rule based systems to create 'smart'advertising campaigns. The campaigns are adaptive based on user preferences, profiles, buying and viewing habits, and demographics. Based on information coming from the Ad Content database, Campaign Rules database, Service Manager, and Carousel Manager, the Ad Manager decides the best products to present to the viewer. It triggers the Carousel Manager to rebuild the broadcast catalog. The Ad Manager also interfaces with the Business Agents to propose advertising contents presented to the viewer while the viewer is on line.

Open Streamer packages advertisements as N Service Platform carousels, one per transport stream, optimizing the bandwidth usage. STB client applications are broadcast with an advertisement library. This library performs Campaign acquisition, Matchmaking, Tracking and Reporting. The campaign acquisition client component runs in parallel with the client application, watching for the campaign carousel, caching information, and pre-fetching assets. The matchmaking client components evaluate each advertising campaign with local parameters (type of page displayed, user information, number of times campaign was ran, etc.) and accesses the best advertisements for display.

The SP in which the present invention resides, provides a system architecture that provides a comprehensive revenue solution for regulation of content, advertising,

messaging services, E-Commerce and television commerce (T-Commerce) in an interactive television environment. The revenue solution of the SP in which the present present resides, provides network operator control and optimal revenue participation by merchants, service providers, network operators and the Service Platform provider. The Service Platform provides a centralizing structure that enables creation of new revenue streams for network operators, solution providers and service providers.

The SGW enables the SP to hide the head-end operator's valuable subscriber profile database by requiring viewer information be given to a service exclusively by the network operator, and under the network operator's control. To protect the subscriber's identity, an abstracted user identifier (i.e., session identifier) is transmitted to the service during the session that the service transmits transaction details to the SP. The user identifier is session specific. There can be more than one user identifier associated with a client, as when different family members use the same STB. Each family member and the household STB can be individually assigned a viewer identifier, category, tracked as to transactions for purchases/movie requests/viewing habits/etc., and profiled by the SP Viewer Manager. The view identifiers are made available to the SGW. The service provider only knows the client or STB identifier through a session identifier. Only the network or head-end operator, by way of the SGW can resolve a session identifier into viewer information details (name, address, shipping information, etc.) needed for fulfilling an order. An exception can be made for a credit card number or other information, when the operator does not wish to perform credit card collections or other transactions.

The present invention enables network operators to control access to the viewer information database and allow only those service providers who have an agreement with the network operator to access privileged information (e.g., credit card numbers, viewer actual name, home address, telephone number, social security number, etc.). Viewer manager **252** enables access to personal and profile information stored on the client devices and enables the client devices or SP to select user-preferred content and purchasing habits based on viewing stored in the viewer profile. Clients, SGW or the SP

select user-preferred content based on viewer profiling via business filters activated in the client device by the client, SGW or another SP component.

The viewer manager **252** provides household/subscriber/STB (or other client device) identification and authentication in support of the SGW and parental control functions. The viewer manager **252** supports multiple viewer identification and registration authentication at a single STB through nicknames and/or personal identification numbers (PINs) plus, the viewer identifier derived from the client device identifier number(s),transaction history, viewer profiles, nicknames and personal identification numbers. The viewer manager **252** performs household and individual viewer profiling through logging, generation, and matchmaking linked to observed cumulative TV viewing and purchasing habits. The viewer manager supports distributed data capture and storage between the SP and the STB, and supports bi-directional synchronisation.

The viewer manager **252** enables distributed profile usage between all SP applications and provides synchronisation with an external SMS/CRM. The viewer manager **252** enables multiple viewer registrations for a single STB or client device using abstract viewer identifiers comprising pseudonyms or nicknames, full names and PIN storage in the STB or other client device. Business agents **226** enforce transactional business rules for interaction between service providers and viewers. Based on business rules, which are defined by the network operators and based on agreements with the service providers, the business agents **226** control transactions and service provider access to user information. Business agents **226** in support of SGW, supplement, add, replace and delete viewer information during a transaction based on the service provider agreements and abstract session identifiers.

Business agents **226** create sessions between client subscribers and service providers. Business agents **226** control access to viewer information details and manipulate viewer information by inclusion, replacement and removal of viewer information presented to service providers. The business agents **226** provide default

values and control access to user information. The business agents **226** also perform transaction logging, messaging logging, and load/transaction monitoring.

Advertising manager **244** provides an interface with both the broadcast and PTP links, which enables complimentary advertising interaction between the two delivery channels. For example, a broadcast (push) advertisement can trigger a PTP connection to the advertising service via the SP so that the user can buy the product or get more information related to the product. A broadcast advertisement can also be placed in the PTP content to inform a user of the availability of broadcast services (e.g., an infomercial).

In some instances, several products or advertising segments are pushed or broadcast to the client without the client requesting the information. Business filters associated with the client, preferably located in a STB are used to select the best advertisement for the viewer based on user profiles. For example, during a cooking show, the SP may schedule a group of cooking advertisements for broadcast to viewers. This group of advertisements may comprise cooking ads on Italian, French, Indian and German cuisine. The SGW will set up the business filter associated with or located in the STB or client to select which type of cuisine advertisement to present to the client, based on a viewer profile One viewer may see a French cooking advertisement while another viewer may see the Indian cooking advertisement depending on the STB filter set by the SGW, client or SP based on viewer profiles, user preferences and/or client profiles.

The SP enables reuse of Web Commerce infrastructure. The SGW residing in the SP replaces the 'usual' HTML templates with an SP friendly format. The business agents receive the order requests from the STB or client through the SGW. SGW queues messages (to manage peaks), some orders are received by the business agents with a delay (preferably orders that do not require any form of confirmation would use this scheme). The SGW business agents add viewer information to orders. The amount and type of the viewer information provided in a order/message is guided by business rules depending on the service/retail agreement.

As communications between services and viewers/clients the information are sent to either separate carousels with a single carousel per transport stream or merged into the existing application carousels. Orders then may proceed, if desired through a 'credit card

clearance' function provided by the SP. As confirmations are sent back from the retailers, the orders are sent real-time back to the user sent via email to the user or made available on-demand through the SGW.

The SP, via SGW also provides offline viewer identification (OVI), which enables a viewer to be identified or authenticated without an online viewer connection established. This ensures that the connection delay (e.g., 10 – 40 seconds) can be placed at the most appropriate place within the purchase process. This also enables viewer identification along with the store and forward function. OVI enables communications and completion of orders/operations with a client device that is intermittently on and off.

An offline order form function is provided which enables the SP/SGW to provide T-Commerce services for a viewer to add items to an order form (shopping cart) without being online. The store and forward function is enables greater scalability. Store and forward may be either forwarding in off peak hours or simply spreading the load over a given time period after a transaction has been initiated. The full store and forward solution is integrated with the so that responses can be forwarded from any channel at any time. Store and forward can be used for enhanced E-Commerce, T-Commerce transactions. The offline viewer authentication enables offline payment selection. Offline payment selection is provided by the SP/SGW to improve the purchase process and to enable use of the store and forward function with T-Commerce/E-Commerce.

The SP/SGW uses standard Web transport where applicable, i.e., it uses HTTP for real-time requests, and SMTP for asynchronous communication where applicable (e.g. merchant reporting, store and forward). Even when going online, the SP provides the ability to connect for a short period of time to access data (e.g., email) then uses the data locally. The SP/SGW provides session-based identifiers instead of the typical Web cookies to protect the operator viewer database. Instead of Web cookies, the SP/SGW provides a session-based identifier that cannot be used by the service to identify the user, only the session. The service must request the viewer information from the SGW (and be charged for it by the network operator).

The SP/SGW optionally informs the viewer when a connection takes place, and also can optionally ask for the viewer's approval to maintain the connection. The SP also displays a "Connection ON" status on the viewer's screen. The SP uses broadcast bandwidth for PTP communication when it is more efficient. A load balancer is provided that determines which information goes over the broadcast and which information goes over the PTP connection. Load balancing decisions are based on the urgency of the data, the delivery latency of the broadcast versus PTP transmission links, the comparative load on the broadcast and PTP paths and the number of viewers receiving the data. Generally, data going to a large number of viewers is broadcast, and small data quantities that need to be sent immediately are sent over the PTP link. STBs without a broadband tuner will receive PTP messages sent out along with broadband.

Referring to Figure **2**, the invention is a method for optimally controlling the choice of broadcast streams used for pulling content (e.g. programs, advertisement) into a client device **212** (e.g. a set top box). "Content" is meant to imply television content not normally obtained from an internet connection. In an interactive television environment, the head end server **50** and set top box **212** have several broadcast channels available for transferring content. For instance, the head end server **50** can place content into a broadcast carousel stream to be transferred over a choice of channels **261** (i.e., terrestrial, cable. satellite dish), and the set top box **212** can pull this content from this stream. Over a return channel **210** (e.g. PTP), the set top box **212** communicates with the head end server **50** on a client-server basis, whereas the set top box **212** requests content from the head end server **50**.

The invention generates a record of latencies in fulfilling requests for all sizes of files and times of request. It enables a small portion of file requests, preferably less than 5%, to be randomly distributed over all available channels. Since file size may be a contributing factor to latency, the method preferably pulls similar sized file from alternate channels. As an example, a 200kB file may be pulled from both the return (Point-to-Point) channel **210** and the broadcast stream **261**. At a given time of day, say 5:00 PM, the average latency on the Point-to-Point connection **210** may be 1.5 seconds while the average latency on the broadcast stream **261** is 0.5 seconds. Some time later, say at 9:00 PM, a file of the same size is pulled across the channels. For a 200kB file, the average

latency on the Point-to-Point connection **210** may now be 1.0 second, while the average latency on the broadcast stream **261** may be 5 seconds. These results would lead the set top box **212** to obtain more of its content from the broadcast stream **261** at 5:00 PM and to obtain more of its content from the Point-to-Point connection **210** at 9:00 PM.

The generated latency data is held for a specific amount of time and then is released at a time when it is determined to be old. A time frame is chosen to be long enough to allow an acceptable level of statistical sample points, as well as short enough to ensure that the sample points being used are recent and relevant to the current time. Sample points that are too old are dropped from the sample group before the next statistical iteration.

The latency in the broadcast stream **261** is primarily determined by repetition rate and can be affected by network traffic. On the broadcast side, the head end server **50** implements a feedback mechanism to determine whether it should broadcast the data resources or whether the receiver should instead obtain that data resource over the return channel **210** from a data server. Each client device **212** and data server keeps track of the demand for the different data resources they use or provide. As the demand for a data resource goes down, the head end server **50** automatically decreases the resource's frequency of appearance in the carousel (off loading the demand onto the IP return channel connection and associated data server). As the demand for a resource goes up, the head end server **50** automatically increases the resource's frequency of appearance in the carousel (decreasing the load on the return channel **210** and associated server). The present invention also provides a message flow rate controller for controlling message flow rate by controlling the bit rate of transmission.

SP provides STBs and/or clients with filters which selectively receive information in the broadcast path based on viewer profiling, so that only selected viewers having a particular filter set up in their STB captures content (advertising, information or A/V programming, etc.) in the broadcast stream. These filters enhance the adaptive and selective delivery aspects of the SP. The Carousel Manager provides a data carousel for Open Streamer. The Carousel Manager manages a carousel of data in real-time. The Carousel Manager complements Open Streamer. Carousel Manager provides a server component and an STB client OCOD library. The Carousel Server receives requests

from applications to add to, remove from or otherwise change the carousels contents. As Carousel Manager receives a request, it treats it as a single transaction, and obtains all necessary data (usually through HTTP). The Carousel Manager generates new carousel index or carousel directory file as needed. Carousel Manager publishes the updated carousel directory to Open Streamer, thereby controlling Open Streamer's broadcast priorities and tracks.

Open Streamer is a software/hardware product that enables network operators to broadcast SP applications and data in their network broadcast. Open Streamer enables SP data and applications to be transmitted simultaneously with the network operator A/V programs. Open Streamer enables a data stream to be updated in real time to match the A/V contents. For example, a network operator can broadcast an interactive sports application along with the live broadcast of a sporting event. Open Streamer comprises two components, a common server DLL and a broadcast streamer. An application server (e.g., a weather application server) or the Carousel Builder in the SP calls the common server DLL to send the carousel data to the broadcast streamer. The broadcast streamer then performs multiplexing (according to code/data ratio and bit rate requirements) of the applications and A/V data and sends the multiplexed data to the broadcast equipment for broadcast.

DAP/DATP Protocol Scheme Overview

The present invention enables communication between STBs using DATP and service providers using standard protocol via the SGW. DATP protocol is a message-based protocol where an entity sends a message to another entity with a delivery guarantee. Any time the STB sends a message to the SGW, STB receives an acknowledgement message once the message has reached its final destination (SGW provides the function of an application server). When the message has been processed by an application server, a response message may be sent to the STB provided that the STB session with SGW is still open. The DATP message transmission phase will be preceded with a DATP login phase and followed by a DATP logout phase needed to establish a

DATP session. DATP is a session oriented protocol. Figure 10 illustrates a simple example of DATP session.

DATP supports multiple sessions on top on the same STB Transport layer connection. STB clients can send in the middle of an open session with the SGW login packets to start a new session on the same STB transport link used for the first session. Both DATP session management modules in the STB client and in the SGW multiplexes the various session messages over the same link.

DATP Packet content overview

The DATP Protocol packet comprises a fixed size header, a variable size data payload (DAML messages) and a trailer. The Header comprises the following elements: Protocol Version Number; Packet type (Login/Logout Handshake, Ping, Data, Acknowledge, etc.); Actual Transport Info (Raw, TCP/IP, UDP, etc.); Message Sequence Number (DATP message number generated by STB or SG); Service Identifier (ID of the service to receive the data). The service id is an 8 bit identifier defined in the DATP protocol. Session ID (Session ID is provided by SGW at handshake); Encryption Flags for encrypted sessions; and Payload Data Size.

The Payload Data may contain the following depending on the packet type: Login/Logout info for Handshake packets; Acknowledge Info for Acknowledge packet; Data Payload for data packet. The trailer will contain at least the 32 bits CRC checksum of the DATP packet. The DATP protocol byte ordering is BIG ENDIAN.

Packet Fields Specification

The Protocol Version field is the version of the DATP protocol used by the transmitting entity. It's the first byte of a DATP packet. The DATP packet format may vary based on the DATP protocol version number. When new versions of the DATP protocol are specified, this version number is increased to reflect the changes. DATP

communications between two entities will use the highest version of DATP available on both entities. The version negotiation will be part of the login process.

The Packet Type Info field is the second byte of a DATP packet. It indicates what type of DATP packet is being sent. STB transport Info field is the third byte of a DATP packet. It provides information on the transport used on the STB side. It is divided in 3 sub-fields: STB_transport_info[7..4]: The four MSB bits of the field represent the STB native transport protocol type. STB_transport_info[3]: This bit indicates if the underlying transport is reliable. Note that this bit has is set to the correct value even if the native transport protocol type value can provide a good indication of the protocol reliability. STB_transport_info[2..1]: This bit indicates the speed class of the native STB transport.

The Service ID is the forth byte in a DATP packet and indicates the id of the destination (STB to SGW packets) or transmitting (SGW to STB packets) host of a DATP packet. The session ID is the second quadlet (double word) of a DATP packet. It indicates the session id of the DATP packet. Session id values are generated by the SGW during the login process. Login packets have their session id field set to 0.

In DATP, a sequence number is the first word of the third quadlet of a DATP packet. It indicates the DATP message sequence number. This number identifies a DATP "transaction" from a packet sent to its corresponding acknowledge. Message sequence numbers are generated by the transmitting entity and are unique only across the messages sent on one leg of a DATP connection. This means that a DATP message sent from the STB client to the SGW and a message sent from the SGW to the STB client can have the same sequence number but still correspond to two separate "transactions".

In DATP data size is the second double word of the third quadlet of a DATP packet. It indicates the size of the payload data of the packet in bytes. By construction this size is limited to 64KB to accommodate various common factor on low end STBs

such as slow modem links, extremely noisy communication channels, limited RAM memory resources, etc. In DATP, encryption flags constitute the first byte of the fourth quadlet of a DATP packet. The DATP data payload starts from the first byte after the 16 bytes fixed size header up to the size of the Data payload as indicated in the header data size field. In DATP, CRC is the first quadlet after the data payload. It contains the value of the 32 CRC checksum of the whole DATP packet (header included).

The Login packet is sent by the STB client to initiate a DATP session with the SGW. It represents the first phase of the login process negotiation where the STB introduces itself to the SGW. The SGW answers to a login request with an acknowledge packet in case of success. It will decide on the negotiable attributes of the DATP connection and it will assign a session id to the newly created session.

The SGW will answer to a login request with a negative acknowledge packet in case of failure. This packet is sent by the STB to close a DATP session with the SGW. The SGW will answer to a logout request with Logout Acknowledge packet in case of success.

The SGW answers a logout request with Logout Negative Acknowledge packet in case of failure. Cases of failure include errors like unknown session id, bad crc, etc. A data packet can be sent by any entity of a DATP connection. A STB client application can send DATP data packets to Application Severs and Application Servers can respond back to a STB forcing the transmission of a data packet from the SGW to the Client STB. An entity that received a Data Packet will answer with Data Acknowledge Packet on successful reception. An entity that received a Data Packet will answer with Data Negative Acknowledge Packet on unsuccessful reception. If no packet has been received from a remote DATP entity for a configurable period of time, the other remote entity could test the DATP link by sending a DATP ping packet and waiting for a reply. A remote entity that received a ping packet must send a Ping Acknowledge packet to its remote peer if the ping packet was successfully received. A remote entity that received a ping packet must send a Ping Negative Acknowledge packet to its remote peer in case of

unsuccessful reception of a ping packet. Cases of failure include errors like unknown session id, bad CRC, etc.

Turning now to **Figure 4**, the following summarizes the architecture for DATP/SGW as shown in **Figure 4**. A large number of SP and STB client applications have common needs that are more transport specific than application specific that are addressed in the DATP/SGW architecture. DATP/SGW performs encryption, data compression, HTTP routing, and many other functions discussed below. The architecture for the DATP/SGW application backend framework is illustrated in **Figure 4**. DATP/SGW provides Lightweight HTTP (LHTTP) at the O-code application level, store and forward function, STB Identification (using the OpenTV Central Registry [OCR]), and many other functions. These functions are provided as part of or on top of the DATP/SGW protocol.

As shown in **Figure 4**, the SGW **1018** provides a robust communication link between the STB **1008** and a variety of application servers **1026**, **1028**, **1030** and **1032**, including the FetchMail server **1026**. SGW **1018** routes requests back and forth from the STB to application services. SGW receives DATP packets from the client/STB **1018**, contacts the appropriate application server, and sends/receives data to the application server via TCP/IP connection. SGW enables a Third-Party server, or SP specific servers such as the FetchMail server **1026**, to send messages to the STB.

As shown in **Figure 5**, the STB/client stack architecture features a plurality of modules as well as an extra layer, message manager **1104** between the application and the native STB/client transport. APIs are provided to STB applications such as an LHTTP API **1106** and a store and forward API **1108**. SGW provides an asynchronous version of the PAL layer, implements pools of threads and process isolation techniques.

In a preferred embodiment, DATP/SGW provides increased message sizes while guaranteeing delivery reliability and addressing complex memory issues due to constrained embedded environments in the STB. In order to increase DATP message

size, large messages are divided into smaller sections, transmitted, reordered and delivered in a reconstructed DATP message. On a non-reliable link with a binary error rate (BER) of 10^{-64}, the probability of having an error on a 64KB message is roughly 7% (1 message out of 14). Knowing that transferring 64KB takes a bit more than five minutes over a 2400 bits/s modem, DATP avoids retransmitting the same message for another five minutes just because one of its bits is corrupted. To avoid retransmission, the following implementation guidelines for DATP are preferably as follows.

In a preferred embodiment, large messages, that is messages over 64Kb are fragmented into smaller DATP packets. Smaller fragment thresholds less than 64kb may be used. Each DATP fragment is acknowledged separately. As shown in **Figure 9**, DATP/SGW keeps track of message sequence numbers and the time at which the sequence number was last used. DATP messages with a "recently" used sequence number are rejected as "already received." To implement this policy DATP/SGW hosts maintain a sliding window of recently used sequence numbers with a timestamp on each sequence number. Older sequence numbers are removed from the window of the remote host if they are older than (host_max_retry+1)*host_timeout. In a preferred embodiment the time out value is programmable and can be set to any value desired.

The rejection window keeps track of the sequence numbers of packets received in a certain time frame starting from current time. When a packet is received by the DATP core layer, its sequence number is looked up in the rejection window. If the sequence number is found in the window, it is discarded, that is, the packet or fragment associated with that sequence number is ignored. If the sequence number of the packet is not found in the window, the new sequence number is added to the window. The window or "rejection window" is periodically cleaned to get rid of packet numbers older than a certain date depending on the time used on the communication link. The packet rejection window algorithm provides an efficient protection against multiple reception of identical packets which can occur frequently with retransmission/timeout based reliable message oriented transport protocols.

DATP protocol is a message based protocol where an entity send a message to the other entity with a delivery guarantee. Any time the STB sends a message to the service gateway it will receive an Acknowledge message once the message has reached its final destination (The Service Gateway itself or an Application Server). When the message has been processed by an Application Server, a response message may be sent to the STB provided that the STB session with the Service Gateway is still open. The DATP message transmission phase will be preceded with a DATP login phase and followed by a DATP logout phase needed to establish a DATP session. It is important to note that messages sent through DATP are fragmented into DATP packets of at most MTU (Medium Transmission Unit) bytes that are transmitted and acknowledged independently. This allows DATP messages to be as large as physically manageable by DATP entities. Figure 10 presents a simple example of DATP session.

DATP supports multiple sessions on top on the same STB Transport layer connection. STB clients can send in the middle of an open session with the Service Gateway login packets to start a new session on the same exact STB transport link they are using for the first session. Both DATP session management modules in the STB client and in the Service Gateway will be in charge of multiplexing the various session messages on the same link.

To support large DATP message transmission, DATP relies on a packet fragmentation / reconstruction scheme. Large messages are fragmented into small DATP packets of at most MTU size. Each host has a MTU size and each DATP entity can have a different one. Each fragment (DATP packet) of a DATP message is acknowledged separately.

DATP message with "recently" used sequence number will be rejected to avoid "multiple reception of identical fragments" type of race conditions. To implement this policy DATP hosts maintain a sliding window of recently used (sequence number, fragment id) with a timestamp on each entry in the window. Old (sequence number, fragment id) entries will be removed from the window of a DATP host if they are older than (host_max_retry+1)*host_timeout.

A Default DATP fragment size (i.e. MTU size) is limited to 4KB to accommodate constrained STB environment where memory fragmentation is an issue. Fragment size can be increased to a maximum 64KB at the application discretion.

DATP supports up to 65536 fragments per DATP message. This gives a maximum theoretical message size of 4GB. A DATP message's first fragment provides a marker indicating that the fragment is a new message first fragment and its fragment identification (id) field is set to the number of fragments composing this DATP message. Incomplete DATP messages should be discarded by remote entities after (host_max_retry+1)*host_timeout.

DATP messages are sent based on remote host memory conditions. Each acknowledged packet of a DATP message contains a memory available field that indicates current memory condition of the receiving entity. Active entities send messages to SGW as to available memory which is stored at SGW. SGW checks to see if memory is available before forwarding a message to a receiving entity. A sending entity may check with SGW to see if sufficient memory is available at a receiving entity. Utilizing DATP, to send a message to a peer, a remote entity first checks at SGW or at the receiving entity to see if the size of the DATP message is smaller than the memory available in the receiving entity. If there is sufficient memory available at the receiving entity to receive the message, the fragments of the DATP message are sent to the receiving host. Upon receipt of the message, the receiving host acknowledges receipt of the message. Otherwise the transmitting host sends control packets to the receiving host to query for the remote or receiving host's memory availability. Partial delivery based on available memory holding only a portion of a message may also be implemented if desired. In this case, partial messages are cached until completed. The control packets are sent until sufficient memory is available in the remote entity or until the maximum number of message send retries is exceeded. If the maximum number of retries is exceeded and there is still not enough memory available at the receiving host to complete

message transmission, then the message transmission fails (unless partial message delivery is authorized).

The DATP protocol stack reserves memory for DAML messages when the first fragment of a given message is received. Default DATP fragment size is preferably limited to 4KB to accommodate the constrained capacity of the typical STB or client environment, where memory fragmentation is an issue. Fragment size can be increased preferably to a maximum of 64KB by the application sending the message or by the application receiving the message. DATP preferably supports up to 65,536 fragments per DATP message. This provides a maximum theoretical message size of 4GB for a single message. Larger message sizes can be configured if desired.

In a preferred embodiment, the first DATP message fragment contains a marker indicating that the fragment is a new message first fragment, a fragment identification field is set to the number of fragments composing this DATP message. Incomplete DATP messages are discarded by remote entities after (host_max_retry+1)*host_timeout unless additionally messaging time is requested.

SGW/DATP provides encryption to enable applications to send sensitive data back to their respective application servers. Providing encryption at the transport level addresses the challenge of providing encryption in STB or client low processing capacity environment. Thus, encryption is addressed with a carefully designed encryption scheme and a preferred DATP secure API. Security/encryption is provided at a session level. Applications open a secure session using DATP secure API. DATP encryption parameters are negotiated at session login. Secure session negotiation is provided in at least two phases: during a standard DATP login phase, and during a key negotiation phase.

A brief description of the main steps of the key negotiation phase follows. SGWsends its public key server_epk to a client or STB. DATP preferably uses Rivest, Shamir, & Adleman (public key encryption technology) RSA (others may be used). DATP chooses the RSA exponent server_epk = (e, n) so that e=3 while maintaining a

robust level of security (security depends only on n). Since to encrypt a message with RSA the STB needs to compute (m^e) mod n. A small "e" means that the exponentiation phase will be small, leading to a faster computation of the encrypted message. The STB or client initializes its random number generator with the system time plus any random source available to the O-code layer (example: current video frame, etc.). The STB/client picks a STB/client secret key (stb_sk). The STB encrypts the secret key, stb_sk with server_epk using RSA. The STB sends encrypted secret key, stb_sk to the SGW. SGW decrypts encrypted stb_sk with its private key server_dpk.

SGW initializes its random generator and picks a server secret key server_sk. SGW encrypts server_sk with stb_sk using a secret key encryption scheme. SGW sends encrypted server_sk to the STB. The STB decrypts encrypted server_sk with its secret key stb_sk. Once the keys have been successfully exchanged, secret encrypted data can be exchanged between the two entities via DATP using each other's secret keys. In a preferred embodiment, a DATP/SGW server authentication step is added to the protocol, to enhance key exchange scheme and make it robust against "man in the middle" attacks. Thus, signing DATP stacks and managing authentication certificates is provided in the DATP protocol.

To minimize communication time with SGW, the public key of the server is preferably embedded in the stack so that encryption of the STB private key can be performed off-line. This introduces a new key management issue since the SGW should know the server public key used by the STB or client. Messages sent over a secure session will preferably be encrypted at the fragment level. This means that each individual fragment of a DATP message will be encrypted independently.

A DATP Secure API is provided with the ability to send unencrypted messages over a secure DATP session, which provides SP applications the option of saving CPU cycles by not encrypting non-sensitive data sent over a secure session. This is useful for clients or STBs with limited processing power, such as the Motorola DCT 2000.

Once a secure session is established between a SGW and a DATP client or STB, messages sent by the client/STB to any application server are first decrypted in the SGW and then forwarded to application servers using asecure socket layer (SSL) connection. The encryption layer is based on a cryptographic library available to O code developers as well as application server developers. This library can be used by applications to manage encryption at the application level. This ability might be useful to manage end-to-end encryption for security in critical applications such as banking applications.

Data compression on slow links such as the ones available on most STBs and clients (2400 to 33600 bps) it is desirable to send compressed data to increase the total throughput of the line. In some cases modem data compression is available at OSI link level. Higher-level protocols do not gain appreciably by compressing their payload. A large number of client/STB modems do not offer compression at the link level so compression is provided by higher-level protocols. The present invention provides data compression.

The challenge is that STBs or client processors lack capacity to perform efficient pattern searches (or other CPU-intensive operations) needed by most compression algorithms. Decompression, however, is a relatively easy task and decompression APIs are provided to the client/STB at the O code level. Based on these considerations DATP support for compression is asymmetric, that is, only the downlink from the SGW to the STB or client is preferably compressed using standard SP compression tools.

Compressed DATP packets possess a "data compressed" flag in the packet header indicating that the payload data is compressed. Packet headers are not compressed. Compression and decompression will use standard provided SP compression and decompression tools and APIs. DATP packet size indicates the size of the compressed payload. The decompressed size of the payload will be indicated in the payload's compression header. Compression of DATP messages is performed at the fragment level. Each individual DATP packet of a DATP message is compressed independently.

This is preferred since DATP message fragments are not necessarily stored contiguously when received, thus, it is preferred that DATP decompress each fragment separately. Independent decompression is possible since each DATP fragment is compressed independently. The DATP STB stack and the DATP application server API can disable or enable data compression on the downlink. This feature provides application servers at least two important capabilities, the ability to transfer large amounts of data to clients or STBs using the high-speed broadcast channel and the ability to send multicast data to a collection of clients or STBs through the broadcast channel saving overall SP bandwidth.

SGW provides an OpenStreamer application server module that manages a configurable number of broadcast streams. These streams are used to send large chunks of data as well as multicast data to clients and/or STBs. Multicasting is provided as a feature important as routing over broadcast since it enables application servers to send data to a group of STBs without addressing each STB individually. Multicast support in DATP provides unreliable DATP packets. The SP maintains multicast group's list of session identifiers and handles cases where an STB or client with no broadcast tuner available is a member of a multicast group.

DATP Name Service (DNS) provides a mapping between application server names and service identifiers. Though well known services have reserved service identifiers, a large number of user-defined service identifiers are available and can be used by various applications. To avoid hard coding of service identifiers in STB or O-code applications, applications have the ability to refer to services by name after a name resolution stage. This way the application is less dependent upon the SGW configuration file.

The following is a description of how DNS capabilities are provided to DATP clients. DNS is viewed as just another service from a DATP protocol standpoint. A specific service identifier is reserved for the DNS service. The DNS service is hosted within the SGW or may be hosted elsewhere in the SP or in a STB or other client. The

DATP client provides a simple API to resolve names of application servers. Preferably, the main call (datp_get_asid_by_name (as_name)) returns a request number synchronously. An asynchronous notification returns the status of the name resolution including the application server identifier in successful cases. Concurrent name resolutions are possible with no significant detrimental consequences on performance. Users are able to dispatch name server notifications based on a request identifier tagged to each request. The Application Servers' name parameter is added to the current DNS configuration file. The same name is not be used for different service identifiers. To achieve redundancy or satisfy scalability issues registering several machines per service identifier are supported.

In the preferred implementation, DNS is considered as an instance of a yet to be defined directory service. DNS request packet format comprises the following fields: Query Type (indicating the type of query (0 for DNS query for instance)), Query Tag (user provided tag to be matched against directory service responses), Query Data (data used to perform the query operation (typically the name of the service for DNS)). The DNS response packet format comprises the following fields: answer type (indicating the type of answer (0 for DNS resolve OK)), answer tag (same as the query tag that generated the answer) and answer data (query's response data (typically the id of the service for DNS)).

In an alternative embodiment of DATP, the assumption is all DATP clients are behind a modem rack and for each connected client the modem rack terminal server opens a dedicated TCP/IP connection with the SGW and forwards whatever it receives from a given STB to this TCP connection. With the possible deployment in older generation cable boxes with no TCP/IP support, but with User Datagram Protocol (UDP) the DATP server (e.g., SGW) provides the ability to listen on a UDP port. UDP is supported as follows. SGW creates a new datp_socket_listener class to handle UDP connections. A socket type abstraction layer is created to accommodate UDP sockets (PAL_udp_socket).

UDP connections are processed as follows. UDP_listener reads the new connection request datagram and creates a new AL_udp_socket. UDP_listener replies to the connection, sending the datagram using newly created PAL_udp_socket. UDP_listener creates a new Session Manager thread passing the newly created PAL_udp_socket as an attribute. The new session manager talks back directly to DATP client using pal_udp_socket_send with the provided PAL_udp_socket. Note that the remote address of the datagram need not be specified. It is already set by the UDP_listener while answering the connection request.

On the client side a UDP stb_transport module is created that implements the already specified stb_transport API on top of whatever UDP API is available in the targeted STB or client. This UDP stb_transport preferably sends a connection request datagram to the SGW UDP listener port and waits until it receives a reply from the SGW before notifying the DATP core that the STB transport link is up. Subsequent datagrams are sent using the port specified in the connection request reply from the SGW.

HTTP routing is provided to provide an interface for the SGW with standard application servers that use Web servers as their front-end. In this case, DATP preferably does not use the standard DATP application server API that is provided to application server developers, but instead interfaces directly with these application servers by forwarding DATP messages to their Web server front-end using the HTTP POST (HTTPP) mechanism. In this scheme, client and/or STB applications use the DATP API unaware that they are talking to an HTTP server.

In order to support HTTPP, a DATP application server type function is provided by SGW. All servers of this type are provided with an extra entry in the name server configuration file to specify their post URL. The application server communication module provides the ability to post DATP messages to HTTP servers depending on the targeted server type. Preferably, this module is divided into an application server (AS) communication manager and two AS data senders. One AS data sender sends data to the

DATP AS API compatible application servers and another one sends data to HTTP based application servers. HTTP cookies received from the HTTP server are stored in the SGW and are resent to the HTTP server as needed. DATP messages received on a secure DATP session are forwarded to HTTP servers using HTTPS. DATP login and logout are preferably not anonymous, to enable the SGW to control access to SP interactive services and to offer a way for application servers access to the identity of a connected client.

The following further describes STB or client identification as part of DATP. DATP stacks contain a STB or client dependent unique hardware identifier (HID). In the case of an STB this hardware identifier is retrieved from the STB/Network dependent STB transport layer. The HID format is a variable length character string. The HIDs for a given network are stored in a HID list. The network operator, via SP updates the HID list from its customer database using APIs. In the case in which one cannot interface directly with the network operator subscriber database, the SP imports the subscriber information (including their HID) from a flat file.

To establish DATP sessions, STB or client DATP stacks include their HID within the DATP login packet. The SGW checks the validity of the HID using a central repository. Once the HID is cleared by the central repository, access is granted to the STB stack. The HID enables the SGW to determine the identity of a connected STB or client. Similar to HTTP cookies, HID does not "strongly" authenticate a remote STB or client. Thus, formal authentication of remote users preferably will be performed by SGW when applications require more robust authentication of their remote peers.

DATP/SGW provides LHTTP of HTTP functions to O code application developers that enables them to interact with remote HTTP servers. LHTTP is provided to enable development of Web-like HTTP based applications. LHTTP completes the H2O strategy by offering an OS independent simplified HTTP interface for back channel communications between the client, the network operator and services. The LHTTP interface is based on the DATP stack, encapsulating HTTP requests into DATP messages. A special DATP service identifier is assigned to the LHTTP layer and DATP

messages received on this service identifier, which are routed to the destination HTTP server using a specific LHTTP data sender module in SGW.

Preferably, a limited set of HTTP commands is supported, comprising GET and POST commands. Additional HTTP commands can be added to LHTTP. LHTTP requests are transformed into standard HTTP requests at SGW. HTTP requests are generated by the SGW on the behalf of LHTTP clients. Cookies are forwarded to LHTTP clients. SGW caches the cookies and maintains a cookie to session ID translation table. DNS answers HID resolve requests from HTTP servers using this translation table. HTTP servers preferably use the HID to extract user information from the central registry server. LHTTP also provides a secure API, LHTTPS. This API is based on the DATP encryption layer. LHTTPS requests are automatically translated to HTTPS requests at SGW.

Simple Mail Transfer Protocol (SMTP) routing or simply forwarding messages by email is provided to the interface between the SGW and application servers. This interface can be used for non real-time transactions where an application sends DATP messages to SMTP-based application servers and these messages are forwarded by e-mail to the targeted application servers.

In order to support SMTP routing, a DATP application server type is created for SMTP application servers. Servers of this type have extra entries in the name server configuration file to specify their email address as well as the email subject of forwarded messages. The application server communication module posts DATP messages to SMTP based application servers depending on the targeted server type. A SMTP application server data sender module is provided to support this type of transaction. DATP messages sent to SMTP application servers are attached to multipart Multipurpose Internet Mail Extensions (MIME) encoded emails. The first part of the message contains the hardware identifiers of the senders as well as the DATP message ID of the messages

being forwarded. The second part of the message contains MIME encoded DATP messages.

DATP messages sent to an SMTP application server are acknowledged once the message is decoded by a session manager and is ready to be sent by email to the targeted application server. Subsequent SMTP related errors might occur once the SGW makes an email delivery attempt of the DATP message to the targeted application server. Messages sent using the DATP encryption layer will be forwarded unencrypted to the final host. PGP encryption is also supported to securely route DATP messages over SMTP.

The DATP/SGW store and forward service provides functionality for applications to send non real-time messages to a specific application server. A store and forward library is provided on top of DATP. Application uses the store and forward module to send messages with different timing constraints depending on their needs. Timing constraints vary from "as soon as possible", "a specified time", "a specified occurrence, event or message" to "whenever we get connected" including "after a random period of time".

The store and forward module stores undelivered DATP messages into the file system along with some specific attributes (time stamp, timing constraints, targeted AS identifier, etc.). The file system storage path is configurable at least at compile time to accommodate a specific network. Messages not forwarded while a given DATP store and forward enabled application is running are not forwarded until another store and forward enabled application starts running. The store and forward module does not alter the content of the forwarded DATP message. The message is forwarded without alteration to the targeted application server.

Turning now to **Figure 5**, the DATP architecture of the client stack comprising a plurality of modules is illustrated. Modules below line **1121** are written in native client

code while modules above line **1121** are written in O-code. The lightweight HTTP module **1106** provides lightweight HTTP capabilities to O code applications. It is implemented on top of the DATP API. The store and forward module **1108** provides store and forward capabilities to O code applications. It is implemented on top of the DATP API. The DNS module **1110** utilizes the DATP message manager module **1104** to provide DATP name resolution services. The DATP message manager module **1104** provides the front end of DATP. All DATP message-related API calls go through the DATP message manager module. This module divides messages into DATP packets and reconstructs DATP packets into messages. The DATP transport core module **1102** manages DATP sessions, sends and receives DATP packets, and manages DATP module reception from broadcast. The DATP secure transport extension module **1120** handles secure DATP sessions. The DATP packet library **1134** provides the functionality for reading (parsing) and writing (composing) DATP packets to the DATP STB transport module **1132** based on the DATP packet format specification. Upon reading a complete DATP packet, this module will notify the DATP Transport core with the parsed DATP packet.

The DATP broadcast library **1126** listens on selected SP streams based on the DATP transport core **1102** specifications, waiting for modules intended for a given STB or client and notifying the DATP transport core **1102** with the parsed DATP modules. The DATP STB transport module **1132** provides a link-level packet interface on top of whatever native transport or data link is available on the DATP host. The event-loop stub **1116** provides a stubbed version of the message API specified in the DATP portability later. This stub is based on the common library event-loop. The role of the portability layer **1114** is to abstract the DATP stack from application dependent issues such as message dispatching mechanism, encryption APIs, etc. The cryptographic library stub **1118** is a stubbed version of the cryptographic API specified in the DATP portability layer. This stub is based on the common library cryptography package. The module lib stub **1124** is a stubbed version of the multi-track module download API specified in the DATP portability layer. This stub is based on the common library's multi-track module download package.

Turning now to **Figure 7**, DATP is a subset of the Digital TV Application Protocol (DAP). DAP/DATP is depicted in **Figure 7**. DAP is used to standardize back channel communications between SP applications and SGW. DATP and SGW provide a generic virtual transport mechanism to SP applications, since not all SP enabled STBs provide a TCP/IP stack extension. Moreover, some of the STBs run their own proprietary stack or provide no communication stack at all.

DAP is a simple lightweight application protocol suite. DAP's main purpose is to provide a simple and effective way to leverage existing application protocols, such as POP3, SMTP, internet message access protocol (IMAP) and others onto low-end STB's. STBs often possess low capacity processing resources and/or proprietary communications protocols. DAP is designed to abstract communications complexity from the application providers, which in turn leverages existing network infrastructure for today's application standards.

As shown in **Figure 7**, DAP is divided into two parts: DAML **1610** – digital TV application meta language and DATP **1620** - digital TV application transport protocol. DAML **1610** is a meta language that spans many SP applications. Each SP application has its own domain of DAML. The client application responds to and requests messages encapsulated in an DAML domain. These request messages are translated by application servers into the appropriate protocol for existing applications, such as SMTP or IMAP.

DATP **1620** is a lightweight, simple transport protocol designed for low bandwidth applications when TCP/IP or another known protocol is not available. DATP is designed to interface with existing communication protocols in current STBs. DAP comprises: DATP, DAML-Mail (XML domain for mail); DAML- Regi (XML domain for account registration); and DAML-Acct (XML domain for accessing SP VMS/AMS system).

Typical STBs are based on a thin client architecture, that is, possessing minimal processing capability. The services provided by today's STBs are often low-end,

"dumb" applications. Today's resource intensive applications such as email, chat and Internet browsers require more powerful processing appliances. Today's STB cannot provide this type of processing power, hence the need for a low-end, lightweight application protocol. DAP is simple enough to hide or abstract the client/server network complexity from the application developer.

DAP is modular, flexible and adaptable to today's emerging software architectures. Which could be either a Common Object Request Broker Architecture (Object Management Group) (CORBA) based model or Common Object Module (COM)/ Distributed Component Object Module (DCOM) model. DAP is flexible enough to accommodate and integrate with existing third party legacy systems. DAP provides an interface to various open and proprietary protocols. These protocols exist for service systems where the PC is the main client, for example, IMAP or POP3 services. DAP leverages the SP middle ware technology. DAP server ware translates DAP protocol to existing application specific protocols.

DAP and its subset DAML **1610** are designed to be lightweight and capable of supporting low-end bandwidth sensitive STBs. DAML tags are preferably no larger then 4 characters and when possible limited to 2 or 3 characters. DAML incorporates binary XML to facilitate DAML tags. DAP is used as a communication protocol between applications running on the STB and service sub-systems. DATP **1620** controls the communication handshaking, routing and transport specific authentication, where as DAML manages the application specific requirements. DAML requests and responses are communicated between a STB client and a service provider over an existing communication protocol, for example, TCP, UDP, DATP or a proprietary communications protocol.

The DAP protocol and its subset DAML can be a session oriented or "sessionless" protocol suite. DAML domains are application dependent. New domains of the DAP protocol can be used for new types of applications. The addition of new DAP domains have little effect on existing DAP domains. Thus, DAP provides a unique and simplistic

SP for network operators to add additional services without impacting existing services. Each DAML domain can be based on either a simplistic human readable tag or a cryptic abbreviated tag for increasing protocol performance by decreasing the packet size when performance is a critical factor.

The following outlines the role of DAML in the DAP architecture. DAML is an application level communication protocol, utilized to specify communication behavior and communication data for interactive TV services. The service level communication protocol is above the transport level protocol. It defines how the application specific content is encapsulated between client/server communications.

DAML is a collection of domain specific protocols that enables a modular design of the SP. For example, DAML-Mail is a subset of DAP. DAML-Mail is a mail domain specific protocol. New domain-specific protocols can be added as a subset of DAP simply by creating new DTD's. DAP specifies communication behaviors through the sending and receiving of DAP messages. The application specific data is encapsulated in an XML format. The syntax of each XML application domain specifies the actions for the application servers to perform. This enables design of very lightweight simplistic protocols that today's STBs can utilize to interface with existing infrastructure such as SMTP services and IMAP services.

DATP is a transport/service level protocol that provides a communication platform between SGW and multiple STBs or clients. DAML is encapsulated in a DATP packet. In general service level protocols are above transportation protocols, but DATP is unique in that it can sit in a typical network model either at the service level, data link level or transport level. This makes DATP very flexible. DATP interfaces with the underlying transportation protocols, such as TCP, UDP, X.25, raw sockets, or other protocols.

SGW provides routing and SGW technology for low-end STBs to connect to a network backend infrastructure. SGW provides transport level protocol support between

the STB/clients and SGW, for example, a sequential-stream protocol over raw sockets. DAML leverages this feature.

DAML-Mail is a protocol subset of DAP. DAML-Mail is a mail domain specific protocol. This protocol is used to link STBs with IMAP, POP3 and SMTP services. DAML-Regi is a DAP service domain protocol that specifies a generic method for the registration of accounts for multiple services. DAML-Regi is a simple protocol between a STB and the registration server. This enables complex interaction between an STB and a variety of different application systems, with only a single point of integration, the registration server.

DAML-Acct is a DAP service domain protocol that communicates with the SP VMS/AMS database. DAML-Acct enables the STB/client to query and return user specific data from the VMS/AMS system. All the DAML domains are defined using XML document type definition (DTD) syntax. DTDs describe the message syntax but not the logic for the exchanges of requests and responses. XML is useful in defining the markup of a block of text. Specific DAML requests and responses are interactions that are related to each other. The rules for their interaction are modularized in the STB and application server components.

The messaging manager provides various types of message communications among the users and with outsiders (those that are not network service subscribers). For example, it enables users to send and receive email, to chat with other non-subscribers and to receive instant messages from non-subscribers. The email portion of the messaging manager contains a Fetchmail component connected to an Internet based email server such as IMAP, POP3 and other Webmail messages for the appropriate mail-hosting server.

Fetchmail manages all SP server-side mail management. Fetchmail translates DAP messages to IMAP, POP3 or Webmail messages for the appropriate mail hosting server. SGW routes DAP mail messages to "Fetchmail" for processing. Fetchmail

responds with the appropriate response to the request. Fetchmail interfaces with IMAP servers. An email application is provided by the SP. All SP applications can 'send' email via the email service offered by SGW.

The chat SP service interfaces to a chat server or alternative includes a chat server. Chat service is accessible through a dedicated chat application, but also from any SP application linking with the SP chat client DLL. By providing an interface between a chat and a program listing, a chat room can be created dynamically with a broadcast show. Applications and other services can use the SP "alert" service to trigger STB resident mini-applications. Alert utilizes the SP OMM extension and functionality of Open Streamer. The Email service uses alert triggers to inform the viewer of an incoming message.

Turning now to **Figure 6**, SGW incorporates a plurality of modules to support DATP features. The SGW architecture is a multi-process based architecture providing pools of threads. The entire server runs on an asynchronous version of a platform abstraction layer (PAL). The PAL implements a message queue process. PAL communicates using message passing techniques. SGW uses three types of processes, as shown in **Figure 6**.

As shown in **Figure 6**, application servers or services communicate with multiple clients/STBs through the SGW using a domain-specific DAP protocol. In certain cases, clients/STBs can directly connect to the application services. For example, if the transport protocol between the STB and the network is TCP/IP, the STB is TCP/IP enabled, and there is no requirement to perform complex common services provided by SGW, faster network performance can be improved via the client/STB communicating directly to a service via TCP/IP.

Turning now to **Figure 8**, the DATP Server, SGW main process is the main DATP server process described above. SGW hosts several key modules. The TCP socket listener module **1204** is a simple TCP socket listener thread that waits for connections on the DATP TCP listen port, accepts them and requests the creation of new

session managers to handle new connections. The UDP socket listener **1202** waits on a well-known port for UDP connections. Once a connection request is received, the UDP socket listener **1202** creates a new socket and sends a connection request acknowledge to the remote host. UDP socket listener **1202** then requests the creation of a new session manager to handle the connection.

The session manager monitor **1206** module is part of the main thread. The primary role of this component is to monitor session manager (SM) processors **1214** population (creating and deleting SM Processors based on load), and to forward session manager creation requests to the least busy SM Processor **1215**. Each SM processor (0-n) **1215** comprises a DATP application server communication module (ASCM) **1217** and a separate application server data sender (ASDS) for DATP, HTTPP, LHTTP and SMTP.

The DNS name server **1212** thread maintains a matching table between application server identifiers and their attributes (hostname, port, type, etc.) as well a matching table between session identifiers and session manager message queue identifiers. The name server module, DNS answers name resolve queries posted to its message queue. The application server socket listener thread **1208** is in charge of waiting for message post requests coming from application servers. The name server **1212** then forwards the post requests to the targeted session managers based on the post request session identifier.

The session manager processor process **1214**, **1216** hosts a pool of session manager threads **1215**. New session manager threads are created based on requests from the session manager monitor **1206** to the session manager processor thread. The session manager processor **1214**, **1216** thread honors requests from the session manager processors **1214**, **1216** and creates or deletes session managers based on requests from the SM monitor and notifies the session manager processor with the result of its requests. Session manager threads **1215** manage DATP sessions and forward DATP messages from STBs or clients to application servers and vice-versa. There will be one thread per STB or client. These threads utilize several key modules to handle DATP sessions (Packet library; Application Server Communication Module; DATP Application Server Data Sender; HTTPP Application Server Data Sender; LHTTP Application Server Data Sender; and SMTP Application Server Data Sender).

The broadcast manager process **1210** is the main component of DATP routing over broadcast. This process is an Openstreamer application server that manages DATP server carousels. Broadcast Manager Process updates these carousels dynamically depending on requests it receives from other components of the DATP Server.

The SP and SGW are preferably supported on Sun Solaris 7 data processing system with memory, monitor, GUI, mouse, keyboard and processor, which is well known in the art and available from Sun Microsystems. SGW runs as a UNIX daemon, is configured using a configuration file, and is started from the command line. Once a connection has been made between the SGW and the STB/client over a network, TCP/IP handles all communications between the other services. Besides handling different transport protocols, the SGW also routes messages to different service sub-systems depending on the configuration of SGW.

SGW performs its functions at the point of entrance to the application servers. This enables features to be easily configured and/or added since network and messaging functionality is isolated on SGW. This frees the service sub-systems to function on core application functionality and leaves network connectivity issues to SGW. This also enables greater scalability by isolating specific functionality to separate hosts: email message delivery and receiving (using the FetchMail server) from network routing and security using SGW.

SGW is sized to support hundreds of simultaneous connections on a single server. SGW is configurable to handle more connections depending upon the processing power of the processor hosting SGW. This limit is based on the number of modems (typically several hundred) per POP (Point of Presence) for major ISPs. In the case of a WAN architecture where the SGW is located at one central point, a hardware network address translation (NAT) based load balancing device is provided to connect several SGWs in parallel to distribute the load.

The following is an overview of the H2O Proxy environment using a logical view of H2O architecture and sample transactions. Requests for URI may come either from different H2O components – for example: STB/SGW and Carousel. The following

context overview focuses on the STB/SGW issuing the requests – but the general flow of information stays the same.

A viewer chooses to interact with its TV Web page, thus issuing a request from the STB to the H2O system and waiting for a reply. STB requests are sent to the SGW, using lightweight HTTP requests (LHTTP) encapsulated in DATP messages as transport protocol. The requested object is returned through the same channel and protocol. The SGW converts the LHTTP protocol to standard HTTP over TCP/IP and issues the request to a Web Cache.

The Compiled Object Cache (COC) uses its internal disk space to service any request it can serve (following an heuristic taking into account the time-to live of objects). Its role is to service all static objects (standard URLs without queries, no posted form) without querying the H2O proxy, thus reducing it's processing load. In this architecture, the COC will only store compiled objects (H2O modules). The COC machine is I/O driven.

Now turning to **Figure 11**, H2O proxy **248** provides a scalable environment for the different H2O compilers (or filters) to run. It is processing HTTP request and responses "on the fly" and thus the H2O Proxy machine is process driven. The H2O HTML Compiler **1420** is in charge of HTML to SP resource compilation. To compute the TV Layout to render **1422**, this component issues HTTP requests by itself based on the size of the embedded images. This compiler rearranges the Web based image to fit onto the client display device, e.g. a TV.

The MPEG Compiler **1426** charge of the conversion from regular web images format to SP H2O MPEG resources. Source format includes JPEG and GIF and may include PNG. The conversion process may be driven by passing arguments through the URL. The PIXMAP Compiler is in charge of the conversion from regular web images to SP H2O resources. Source format comprises GIF and may include PNG.

The Request Patcher **1424** charge of completing or modifying the request or responses to incorporate data coming from another system (e.g. credit card number…). It communicateswith an external process or database to fetch customer information. The SP component provides a central repository of user information. The Request Patcher

communicates with this component to get the data necessary to patch the requests / responses.

The Not Compiled Object Cache **1430** will use its internal disk space to service any request he can serve (following an heuristic taking into account the time-to live of objects). The objects cached comprise static HTML, GIF images, JPEG images and all standard web formats files. Its role is to service all static objects (standard URLs without queries, no posted form) without querying the Internet, thus reducing latency time to get an object and giving a kind of fault-tolerance to the system. The Customer web site holds the web site to publish through the H2O system.

Figure 12 illustrates a request for a static page, already cached The STB user issues a request to load an HTML page **1520**. This request is sent to the SGW **248** using LHTTP over DATP. The SGW converts the request to HTTP over TCP/IP and forwards it **1522** to the Compiled Object Cache **1410**. The Compiled Object Cache **1410** has the requested (compiled to a module) HTML page stored in its internal hard disk space; if the object's time to live has not expired and the compiled object cache service the request with the compiled HTML page. It transmits the HTTP response **1424** to the SGW, using HTTP over TCP/IP. The SGW translates the protocol from HTTP over TCP/IP to LHTTP over DATP. The STB loads the requested page **1526** (compiled) in its memory and gives it to the H2O browser engine for interpretation. The H2O browser engine requests **1528** the SGW to get the images necessary to render the screen on TV, with conversion options (mpeg or pixmap, width, height...) on the url. The SGW transmits the HTTP request **1530** to the Compiled Object Cache. The Compiled Object Cache has the requested (compiled to a module) image stored in its internal hard disk space; the objects time to live has not expired and the compiled object cache service **1532** and **1534** the request with the compiled image. In this scenario, the H2O Proxy was spared of the request and thus can process other requests.

As shown in **Figure 13**, the STB **212** user issues a request **1610** to load an HTML page (home.asp), the host and user info of the request header hold [STB Model+ STB Serial number] and [Access Card Id] of the user. This request **1610** is sent to the SGW using LHTTP over DATP. The SGW converts the request to HTTP over TCP/IP and

forwards it **1612** to the Compiled Object Cache. The requested object is not available in the disk space of the Web cache. The Web Cache then forwards the request **1614** to the H2O Proxy. The H2O Proxy asks **1616** the SP to return **1620** the name of the user (for the amazon.com service). The H2O Proxy patches the request with the name of the user, and issues this request **1622** to the "Not Compiled Object Cache". The "Not Compiled Object Cache" does not hold the requested HTML page in its disk space and then issues the request **1624** to the targeted web server here amazon.com. The targeted web server computes the HTML page, given the user information and returns **1626** it to the "Not Compiled Object Cache". The "Not Compiled Object Cache" returns the HTML page **1628** to the H2O Proxy.

The H2O Proxy sends the HTTP request **1630** to the "Not Compiled Object Cache" to get the images **1632, 1634, 1636** necessary for it's layout computations (gif, jpeg...). The H2O Proxy compiles the HTML page; computes the layout, patches the embedded images urls and sends back to the "Compiled Object Cache" the resulting OpenTV resource **1646** (with an SP resource mime-type). The Compiled Object Cache stores the object in its internal disk space and sends back the compiled HTML **1648** page to the SGW. The SGW converts the response to LHTTP over DATP and sends it back to the **1650** STB. The STB loads the requested object in its memory and gives it to the H2O browser engine for interpretation.

The H2O browser engine issues requests **1652** to the SGW to get the images necessary for the rendering (through the patched urls: here the logo.gif url include a directive for pixmap resource format): pix/logo.gif. The SGW converts the request **1652** to HTTP over TCP/IP and forwards it to the Compiled Object Cache. The "Compiled Object Cache" already has the requested gif image, in the correct resource format – because a user already requested this image at a previous time – and the image is directly returned **1654** to the SGW. The SGW converts the response to LHTTP over DATP and sends it **1656** back to the STB. The H2O browser engine issues requests **1658** to the SGW to get the images necessary for the rendering: mpg/banner.jpg. The "Compiled Object Cache" does not hold the requested image in its disk space and therefore issues the request **1660** to the H2O Proxy. The H2O Proxy sends the HTTP request **1662** to the "Not Compiled Object Cache" to get the /banner.jpg image.

The "Not Compiled Object Cache" holds the image in its cache and returns it **1664** immediately to the H2O Proxy. The H2O Proxy converts the image, using the parameters given in the url (mpg format, width, height...) and returns the result to the Compiled Object Cache **1668**. The Compiled Object Cache stores the object in its internal disk space and sends back **1668** the converted mpeg image to the SGW. The SGW converts the response to LHTTP over DATP and sends it back **1670** to the STB. The STB renders the HTML page to screen.

The H2O Proxy component provides to other H2O components or compilers, a robust and scalable architecture and an interface for "compilers" configuration Other service provided are: the ability to log errors; the ability to alert an administrator on defined events; and the ability to debug-trace the "compilers". From the provided H2O Proxy environment and APIs the compilers "patch" HTTP requests and responses on the fly, eventually accessing an external database, file or process to do so. The compilers patch HTTP requests by removing specific HTTP Header (STB identifier, Access Card Identifier...); by adding specific HTTP Header (Username, Credit Card Number...); by adding HTML Form fields to incoming post request (Visa Card number...); and by performing string substitution (UID -> User Identifier) the compilers convert web objects formats and mime types "on the fly" in HTTP responses and issue HTTP requests by themselves and get a response object in return.

As shown in **Figure 14**, in a preferred embodiment, the H2O Proxy is implemented by developing an extension of enclosing software (Web Proxy, Firewall, web server or other...). This host software provides H2O threading and scheduling of the H2O tasks as well as some of the needed functionalities to implement H2O "compilers" and patching components.

Using the API provided by the Proxy Host Software, a set of API (the H2O Proxy API) is provided to implement the functionalities needed by the H2O compilers missing from the H2O Proxy Host Software Services, and provide a higher abstraction level for the services available from the H2O Proxy Host Software. The request patcher **1424** component reads incoming HTTP requests for HTML pages and completes them with information from another process or file or database. The HTML2RES Compiler **1420**

compiles returned HTML pages into SP resources files and change the mime type of the HTTP response header to suit new format: Mime-Type: text/otvres.The GIF2PIX Compiler **1422** converts a returned GIF image into an SP resource file and changes the mime type of the HTTP response header to suit new format: Mime-Type: image/otvpix. The 2MPEG Compiler **1426** converts a returned GIF or JPEG image into an SP resource file and change the mime-type of the HTTP response header to suit new format: Mime-Type: image/otvmpg.

Turning now to **Figure 15**, a dynamic request for an HTML page sequence diagram is illustrated. The Object Caches are not displayed in the Sequence diagram, being "passive" components in this interaction. The User STB **212** issues a request **1810** for a page (home.asp) through HTTP request. The Request Patcher **1424** accesses an external process/file/database/url **1812**, **1814** to get user name, patches the request and sends **1816** it to the HTML2RES Compiler. The HTML2RES Compiler sends the request **1818** to the target web site (amazon.com). The Web site computes the request and sends back **1820** the resulting HTML page to the HTML2RES Compiler. The HTML2RES Compiler parses the file to get the image links URL and issue the requests **1822** to the web site to get **1824** the image files (logo.gif, banner.jpg). The HTML2RES Compiler computes the TV layout for the page, compiles it into SP resource file, and sends it **1830** back to the STB. The STB renders the HTML page on TV.

Turning now to **Figure 16**, a request for an image file, sequence diagram is illustrated. An HTML page being loaded in the User STB needs a image to render its screen. It issues an HTTP request **1910** for the image (URL embedded conversion options) to the 2MPG Compiler. The 2MPG Compiler requests the image **1912** from the target site (amazon.com). The target site, returns the banner.jpg image file **1914** to the 2MPG Compiler. The 2MPG Compiler converts the banner.jpg file, using the options given on the URL and returns the result **1916**, with a image/otvmpg mime-type, to the STB. The STB renders the image on screen.

The different identified H2O compilers inherit from the class H2OCompiler and provide an implementation for the different pure virtual entry points of the class.

Memory functions are given to compilers to allocate and free the requests/responses buffers. The size of the allocated buffer is given to a FreeBuffer function so that different schemes can be used to free the buffer (for a certain size, the buffer might be implemented as a temporary file mapped in memory whereas for smaller sizes, it is preferably implemented as in memory buffer).

The buffer is passed to an Execute function containing the full HTTP request / response the compiler parses the request headers, mime-types and takes the appropriate actions. This buffer is preferably read-only. The buffer can be writeable as well to enable augmentation by the compiler or other functions after. The buffer returned by the Execute functions contains a valid HTTP request / response, the memory will be freed by the H2O proxy using the appropriate FreeBuffer function and has to be allocated by the provided AllocBuffer function. Debug member is provided for compiler implementers to be able to get a debug trace from within the H2O Proxy environment.

The parameters functions are used to get the names of the parameters, get the current values (string) of the parameters, set a new value for a parameter, and validate a parameter set. The URL functions are provided for the HTML compiler to fetch images. Those functions are available to the other compilers to provide extra services to the components whenever needed.

For example, a network with 1 Million STBs, with an average 20,000 users connected, generates 2,000 requests per second for HTML pages to the SGW and "Compiled Object Cache" (unless part of the requested pages come from broadband). Assuming of those pages should be static and should be served immediately from the "Compiled Object cache", the H2O Proxy will have to serve 200 requests per second. Assuming that a typical HTML page embeds 10 images, 8 out of 10 being JPEG H2O Proxy issues 10 outgoing requests for each incoming request. The "Not Compiled Object cache" serves 2,000 requests per second.

The MPG conversion is preferably performed in advance whenever possible. A Web crawler may request the HTML pages and images at night to have them converted

in advance, leveraging this issue. The compilers thus interact with H2O. H2O **248** is the preferred client/server solution provided in the SP that enables Internet content developers to create interactive TV content, applications and services for network operators running on the SP. Thus, via H2O enables the larger pool of Internet talent and content is made available to the vast growing worldwide market of interactive TV applications. The H20 server converts Internet content (HTML pages, ECMA scripts, and HTML page formatting) into SP assets. The H2O client, H2OC renders the assets and interacts with the clients. In the T-Commerce/E-Commerce case scenario, H2O enables E/T-Commerce shops to utilize existing Web tools to create shopping services and to interface with the preferred SP (operator), using standard Web protocols. Thus the present invention is user friendly providing APIs through known methodologies.

H2O acts as a proxy to the SGW and the broadcasting tools to convert Web content to SP content. Thus, Web sites developers can use their current HTTP servers and application servers to generate interactive TV content inexpensively. In a preferred embodiment, H2O converts HTML, JavaScript, and Internet graphics, however, any other known or developed Internet or other content or protocol can also be added to the proxy functionality of H2O. H2O enables the SP to display Web pages on STBs that are not fully browser capable and to create original user interfaces. H2O enables SP connection to any commerce engine that uses only HTML. H20 is responsible for converting all now or future broadband and Web content such as HTML pages, JPG pictures, wav audio files, etc. into SP resources.

The server side of H20, H2OS is an HTTP proxy. For other purposes, it can be packaged as a dynamic link library (DLL) or batch tool. The client side of H20, H2OC is an STB O-Code application. H2OC is built on top of other SP client components, such as the SGW library or the Carousel Load library. H2O enables URLs to be used to address documents and services. H2O also enables tracking in the broadcast and online environments. H2OS provides HTTP proxy functionality. SP applications request a document through H2O, upon which H2O retrieves the document, parses it, compiles it, and returns the document to the requester. This H2O functionality enables use of the

same engine for different uses, online and broadcast, facilitates scalability, and enables flexible use of H20. The parsing depends on the type of document, parsing can be HTML parsing, a GIF picture, or JPEG images, etc. To make it expandable, H2O is able to "plug-in" and run new third party filters.

H2O enables scripting using different languages. Preferably, all SP server components standardize around monitoring, especially the ability to remotely manage the different processes. SNMP is used to handle basic functions ("process OK", and traps on major problems). A command-line interpreter is provided on socket for inspecting status. Setting parameters enables remote management and provides an interface with the Web through Web scripts. In a preferred embodiment, standardized warnings and error logs are provided.

HTML/JS does not allow sharing of information from one page to another for the Web, as the server takes care of the context. In broadcast mode, this arrangement is insufficient. The present invention provides a broadcast mode, preferably, wherein a global permanent object is provided, that is not cleared when starting a new page. The permanent object maintains context between pages. Other base objects provided by the SP are also made permanent on transition (e.g., station control, OSD). Gadgets are defined through an interface definition language to enable creation of new gadgets, modification of gadgets and to enable adding methods without modifying the compiler.

The H2O Carousel feature provides real-time updating of catalogs, maintaining consistency of catalogs during updates, and providing safe transactional models. H2O carousel enables updating a single page, or an entire set of pages in a single transaction. Carousel management provides management of a carousel index or directory. The index contains information for accessing and fetching data from the carousel. That is, for a given page, Carousel Management contains a list of all modules necessary, so that H2OC requests all necessary modules at once to expedite the process.

Carousel provides data compression, meta data on pages (e.g., page relative priority, how often the page is sent) and page tracking (elementary stream). The carousel client is a STB OCOD library, handling the loading of resources. Carousel client manages dynamics of resources, i.e., new resources, deleted resources, and changed resources. Dynamic resource management enables the client (H2OC) of this library to present dynamic content. The carousel client manages memory allocation, pre-fetching and caching of resources, and decompression of modules. The carousel client manages sub-index/directories and manages a 'set' of resources instead of a 'tree' of resources, which facilitates sharing of assets. Subsets of a single tree of resources can be assigned to separate processes thereby enabling shared resources.

H2O monitors TV triggers performance and bandwidth, e.g. shared resources in shared modules. H2O optimizes bandwidth utilization. H2O provides multi-tracks, multi-priorities, and management of bid volume of data. H2O provides run-time pre-fetching and caching at the module level. H2O supports compressed modules. H2O supports arrow and direct key navigation (e.g. digit or color), handling international (Chinese), meta data on pages, (e.g., page relative priority, how often it is sent) and page tracking (elementary stream). Global GUI is shared, that is, a direct key linking is provided so that any information page can be shared by every other page.

H2O manages pages and sub-pages to handle cases where pages are too large to fit on one screen without the need for scrolling. H2O enables use of HTML to present content, online, point-to-point, and broadcast. H2O enables composition of a page with a mixture of broadcast and online components. For example, a page can come from an online server, while its background is broadcast. H2O enables merger of content in the STB. For example, a bank application can send a viewer's last 20 credit card transactions from its server while the HTML page is broadcast. Preferably a Java Script function request (HTTP like) the server some XML, using the result and some DOM functions patches a table with the result.

Preferably, security is provided for secured authentication of the viewer, which is performed at SGW rather than at H20. However, H2O can alternatively comprise authentication functionality. H2O also sends encrypted data to (e.g., sending a credit card number) and from a STB to an online server. For some services, it is appropriate to go through a security proxy near the HTML to SP conversion. SP can utilize HTTPS from the proxy to the service provider, and an SSL like OCOD library from the STB to the proxy. In other cases (e.g., banks), security is provided from end to end, in which case H20 does not usually perform translation. That scenario is preferably reserved for data the STB is able to process without translation through H2O. Encryption can alternatively be performed at SGW or STB.

The present invention has been described in interactive television in a preferred embodiment, however, the present invention may also be embodied in a distributed computer system comprising a server and a client device. In another embodiment, the present invention is implemented as a set of instructions on a computer readable medium, comprising ROM, RAM, CD ROM, Flash or any other computer readable medium, now known or unknown that when executed cause a computer to implement the method of the present invention.

While a preferred embodiment of the invention has been shown by the above invention, it is for purposes of example only and not intended to limit the scope of the invention, which is defined by the following claims.

Claims

1. A computer readable medium containing instructions that when executed cause a computer to:
receive a first message containing at least one of application code, control, data, and audio/visual data at a server in a service provider compatible protocol;
translate the first message into a client device compatible protocol, that is different from the service provider compatible protocol;
compress the first message at the server; and
send the compressed first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connection.

2. The medium of claim 1 further comprising instructions that cause the computer to:
send a uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server;
translate the uncompressed second message into the service provider compatible protocol; and
send the translated uncompressed second message from the server to the service provider.

3. The medium of claim 2 further comprising instructions that cause the computer to:
prior to sending the first message to the client device,
encrypt the first message at the server and encapsulating the encrypted data into the client device compatible protocol message; and
set a flag in the encrypted first message to indicate that the encrypted first message is encyrpted;
send the encrypted first message to the client device; and
decrypt the encrypted first message at the client device.

4. The medium of claim 3 further comprising instructions that cause the computer to:
individually encrypt at the server a fragment of the first message into the client device compatible protocol.

5. The medium of claim 1 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

6. The medium of claim 1 further comprising instructions that cause the computer to:
control message flow rate by controlling the bit rate of transmission.

7. A computer readable medium containing instructions that cause a computer to:
retrieve a client device hardware identifier for the client device from the client device dependant transport layer;
store the client device hardware identifier in a network operator hardware identifier list;
authenticate the client device hardware identifier before establishing a communication session between the server and the client device;
receive a first message containing at least one of application code, control, data, and audio/visual data at a server in a service provider compatible protocol;
translate the first message into a client device compatible protocol;
send the first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connnection;
send a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server;
receive the second message at the server;
generate a session identifier from the client device hardware identifier;
inserting the session identifier in place of the client device hardware identifier in the second message;

translate the second message into the service provider compatible protocol; and
send the translated message from the server to the service provider.

8. The medium of claim 7 further comprising instructions that cause the computer to:
determine whether to send the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

9. The medium of claim 7 further comprising instructions that cause the computer to:
store the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

10. The medium of claim 9 wherein the first message and second messsage comprise an ecommerce transaction.

11. The medium of claim 7 further comprising instructions that cause the computer to:
encapsulate HTTP in the device compatible protocol in the second message at the client device prior to sending the second messsage to the server;
convert the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
receive a cookie from a service provider via a HTTP server at the server in response to the converted HTTP message;
caching the cookie at the server;
generate a cookie to session identifier translation table;

use the cookie to session identifier table to answer a client device hardware identifier name request from the HTTP server; and
use the client device hardware indentifer to extract user information from a central registry.

12. The medium of claim 11 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

13. The medium of claim 7 wherein the server sends a business filter to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

14. The medium of claim 7 further comprising instructions that cause the computer to:
complete an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

15. The medium of claim 7 further comprising instructions that cause the computer to:
send a third message to the client device requesting a quantity of memory available at the client device; and
checking a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

16. The medium of claim 7 further comprising instructions that cause the computer to:
generate a sequence number in the first message at the server before sending the first message to the client device;
store the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and

reject the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

17. The medium of claim 7 further comprising instructions that cause the computer to:
resolve a request to a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

18. The medium of claim 7 further comprising instructions that cause the computer to:
provide a socket type abstraction layer to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

19. The medium of claim 7 further comprising instructions that cause the computer to:
authenticate multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

20. The medium of claim 7 further comprising instructions that cause the computer to:
control , in a business agent, access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

21. A computer readable medium containing instructions that when executed cause a computer to:
receive a first message from a server at a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at the server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

22. The medium of claim 21 further comprising instructions that cause the computer to:
send an uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for translation of the uncompressed second message into the service provider compatible protocol for transmission to the service provider.

23. The medium of claim 22 wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to receipt of the first message at the client device, further comprising instruciton that cause the computer to:
decrypt the encrypted first message at the client device.

24. The medium of claim 23 wherein the first message has been divided in to fragments which have been individually encrypted at the server prior to receipt of the first message at the client device.

25. The medium of claim 21 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

26. The medium of claim 21 further comprising instructions that cause the computer to:
control message flow rate by controlling the transmission bit rate.

27. The medium of claim 21 further comprising instructions that cause the computer to:
retrieve a client device hardware identifier for the client device from the client device dependant transport layer;
send the client device hardware indentifier to an identifier list for storage for authentication of the client device hardware identifier before establishing a communication session between the client device and a server;
receive a first message containing at least one of application code, control, data, and audio/visual data at the client device from a server, wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol;
receive the first message at the client device over at least one of a broadcast carrier wave, local area network and point to point connnection;
send a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

28. The medium of claim 27 further comprising instructions that cause the computer to:
determine whether to receive the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

29. The medium of claim 27 further comprising instructions that cause the computer to:
store the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

30. The medium of claim 29 wherein the first message and second messsage comprise an ecommerce transaction.

31. The medium of claim 27 further comprising instructions that cause the computer to:
encapsulate HTTP in the device compatible protocol in the second message at the client device prior to sending the second messsage to the server for converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
send the session identifer to the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device, wherein the cookie is identified in a cookie to session identifier translation table, wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

32. The medium of claim 31 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

33. The medium of claim 27 wherein the client device receives a business filter from the server to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

34. The medium of claim 27 further comprising instructions that cause the computer to:
completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

35. The medium of claim 27 further comprising instructions that cause the computer to:
receive a third message at the client device requesting a quantity of memory available at the client device; and
check a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

36. The medium of claim 27 further comprising instructions that cause the computer to:
generate a sequence number in the first message at the server before sending the first message to the client device;
store the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and
reject the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

37. The medium of claim 27 further comprising instructions that cause the computer to:
resolve request to a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

38. The medium of claim 27 further comprising instructions that cause the computer to:

provide a socket type abstraction layer to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

39. The medium of claim 27 further comprising instructions that cause the computer to:
authenticate multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

40. The medium of claim 27 further comprising instructions that cause the computer to:
control, in a business agent, access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

41. A computer readable medium containing instructions that when executed cause a computer to:
send a first message from a server to a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at the server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

42. The medium of claim 41 further comprising instructions that cause the computer to:
receive an uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol from the client device for translation of the

uncompressed second message into the service provider compatible protocol for transmission to the service provider.

43. The medium of claim 42 wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to sending the first message to the client device for and decrypting of the encrypted first message at the client device.

44. The medium of claim 43 wherein the first message is divided in to fragments and each fragment individually encrypted at the server prior to sending the first message to the client device.

45. The medium of claim 41 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

46. The medium of claim 41 further comprising instructions that cause the computer to:
control message flow rate by controlling the transmission bit rate at the server.

47. A computer readable medium containing instructions that when executed cause a computer to:
send a client device hardware indentifier retrieved from the client device dependent transport layer, to a server identifier list for storage and authentication of the client device hardware identifier before establishing a communication session between the client device and a server;
send a first message containing at least one of application code, control, data, and audio/visual data at the client device from the server over at least one of a broadcast carrier wave, local area network and point to point connnection, wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol;

receive a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

48. The medium of claim 47 further comprising instructions that cause the computer to:
determine whether to send the first message from the server to the client device over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

49. The medium of claim 47 further comprising instructions that cause the computer to:
store, at the server the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

50. The medium of claim 49 wherein the first message and second messsage comprise an ecommerce transaction.

51. The medium of claim 47 further comprising instructions that cause the computer to:
receive at the server, HTTP encapsulated in the device compatible protocol in the second message;

convert the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;

receive the session identifer at the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device, wherein the cookie is identified in a cookie to session identifier translation table, wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

52. The medium of claim 51 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

53. The medium of claim 47 further comprising instructions that cause the computer to:

send a business filter from the server to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

54. The medium of claim 47 further comprising instructions that cause the computer to:

complete an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

55. The medium of claim 47 further comprising instructions that cause the computer to:

send a third message to the client device requesting a quantity of memory available at the client device; and

check a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

56. The medium of claim 47 further comprising instructions that cause the computer to:
generate a sequence number in the first message at the server before sending the first message to the client device for storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device for rejection of the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

57. The medium of claim 47 further comprising instructions that cause the computer to:
resolve a request to a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

58. The medium of claim 47 further comprising instructions that cause the computer to:
provide a socket type abstraction layer in the server to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

59. The medium of claim 47 further comprising instructions that cause the computer to:
uthenticate at the server, multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

60. The medium of claim 47 further comprising instructions that cause the computer to:
control, in a business agent in the server, access to client device user information in an ecommerce transaction between the service provider and the client device user,

wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

61. A method for asymmetrical communication in an interactive television system comprising:
receiving a first message containing at least one of application code, control, data, and audio/visual data at a server in a service provider compatible protocol;
translating the first message into a client device compatible protocol, that is different from the service provider compatible protocol;
compressing the first message at the server; and
sending the compressed first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connnection.

62. The method of claim 61 further comprising:
sending a uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server;
translating the uncompressed second message into the service provider compatible protocol; and
sending the translated uncompressed second message from the server to the service provider.

63. The method of claim 62 further comprising:
prior to sending the first message to the client device,
encrypting the first message at the server and encapsulating the encrypted data into the client device compatible protocol message; and
setting a flag in the encrypted first message to indicate that the encrypted first message is encyrpted;
sending the encrypted first message to the client device; and
decrypting the encrypted first message at the client device.

64. The method of claim 63 further comprising:
individually encrypting at the server a fragment of the first message into the client device compatible protocol.

65. The method of claim 61 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

66. The method of claim 61 further comrpising:
controlling message flow rate by controlling the bit rate of transmission.

67. A method for communication in a distributed computing system comprising:
retrieving a client device hardware identifier for the client device from the client device dependant transport layer;
storing the client device hardware identifier in a network operator hardware identifier list;
authenticating the client device hardware identifier before establishing a communication session between the server and the client device;
receiving a first message containing at least one of application code, control, data, and audio/visual data at a server in a service provider compatible protocol;
translating the first message into a client device compatible protocol;
sending the first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connnection;
sending a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server;
receiving the second message at the server;
generating a session identifier from the client device hardware identifier;
inserting the session identifier in place of the client device hardware identifier in the second message;
translating the second message into the service provider compatible protocol; and
sending the translated message from the server to the service provider.

68. The method of claim 67 further comprising:
determining whether to send the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

69. The method of claim 67 further comprising:
storing the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

70. The method of claim 69 wherein the first message and second messsage comprise an ecommerce transaction.

71. The method of claim 67 further comprising:
encapsulating HTTP in the device compatible protocol in the second message at the client device prior to sendint the second messsage to the server;
converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
receiving a cookie from a service provider via a HTTP server at the server in response to the converted HTTP message;
caching the cookie at the server;
generating a cookie to session identifier translation table;
using the cookie to session identifier table to answer a client device hardware identifier name request from the HTTP server; and
using the client device hardware indentifer to extract user information from a central registry.

72. The methof claim 71 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

73. The method claim 67 wherein the the server sends a business filter to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

74. The method of claim 67 further comprising:
completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

75. The method of claim 67 further comprising:
sending a third message to the client device requesting a quantity of memory available at the client device; and
checking a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

76. The method of claim 67 further comprising:
generating a sequence number in the first message at the server before sending the first message to the client device;
storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and
rejecting the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

77. The method of claim 67 further comprising:
a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

78. The method of claim 67 further comprising:
providing a socket type abstraction layer to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

79. The method of claim 67 further comprising:
authenticating multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

80. The method of claim 67 further comprising:
controlling, in a business agent, access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

81. A method for asymmetrical communication comprising:
receiving a first message from a server at a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at the server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

82. The method of claim 81 further comprising:
sending an uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for translation of the uncompressed second message into the service provider compatible protocol for transmission to the service provider.

83. The method of claim 82 further comprising:
wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to receipt of the first message at the client device; and
decrypting the encrypted first message at the client device.

84. The method of claim 83 further comprising:
wherein the first message has been divided in to fragments which have been individually encrypted at the server prior to receipt of the first message at the client device.

85. The method of claim 81 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

86. The method of claim 81 further comrpising:
controlling message flow rate by controlling the transmission bit rate.

87. A method for communication in a distributed computing system comprising:
retrieving a client device hardware identifier for the client device from the client device dependant transport layer;
sending the client device hardware indentifier to an identifier list for storage for authentication of the client device hardware identifier before establishing a communication session between the client device and a server;
receiving a first message containing at least one of application code, control, data, and audio/visual data at the client device from a server, wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol;
receiving the first message at the client device over at least one of a broadcast carrier wave, local area network and point to point connection;

sending a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

88. The method of claim 87 further comprising:
determining whether to receive the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

89. The method of claim 87 further comprising:
storing the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

90. The method of claim 89 wherein the first message and second messsage comprise an ecommerce transaction.

91. The method of claim 87 further comprising:
encapsulating HTTP in the device compatible protocol in the second message at the client device prior to sending the second messsage to the server for converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
sending the session identifer to the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device, wherein the cookie is identified in a

cookie to session identifier translation table, wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

92. The methof claim 91 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

913. The method claim 87 further comprising:
wherein the client device receives a business filter from the server to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

94. The method of claim 87 further comprising:
completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

95. The method of claim 87 further comprising:
receiving a third message at the client device requesting a quantity of memory available at the client device; and
checking a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

96. The method of claim 87 further comprising:
generating a sequence number in the first message at the server before sending the first message to the client device;
storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and
rejecting the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

97. The method of claim 87 further comprising:
a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

98. The method of claim 87 further comprising:
providing a socket type abstraction layer to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

99. The method of claim 87 further comprising:
authenticating multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

100. The method of claim 87 further comprising:
controlling, in a business agent, access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

101. A method for asymmetrical communication comprising:
sending a first message from a server to a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at the server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

102. The method of claim 101 further comprising:

receiving an uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol from the client device for translation of the uncompressed second message into the service provider compatible protocol for transmission to the service provider.

103. The method of claim 102 further comprising:
wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to sending the first message to the client device for and decrypting of the encrypted first message at the client device.

104. The method of claim 103 further comprising:
wherein the first message is divided in to fragments and each fragment individually encrypted at the server prior to sending the first message to the client device.

105. The method of claim 101 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

106. The method of claim 101 further comrpising:
controlling message flow rate by controlling the transmission bit rate at the server.

107. A method for communication in a distributed computing system comprising:
sending a client device hardware indentifier retrieved from the client device dependent transport layer, to a server identifier list for storage and authentication of the client device hardware identifier before establishing a communication session between the client device and a server;
sending a first message containing at least one of application code, control, data, and audio/visual data at the client device from the server over at least one of a broadcast carrier wave, local area network and point to point connection,

wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol;
receiving a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

108. The method of claim 107 further comprising:
determining whether to send the first message from the server to the client device over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

109. The method of claim 107 further comprising:
storing, at the server the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

110. The method of claim 109 wherein the first message and second messsage comprise an ecommerce transaction.

111. The method of claim 107 further comprising:
receiving at the server, HTTP encapsulated in the device compatible protocol in the second message;

converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
receiving the session identifer at the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device, wherein the cookie is identified in a cookie to session identifier translation table, wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

112. The methof claim 111 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

113. The method claim 107 further comprising:
sending a business filter from the server to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

114. The method of claim 107 further comprising:
completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

115. The method of claim 107 further comprising:
sending a third message to the client device requesting a quantity of memory available at the client device; and
checking a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

116. The method of claim 107 further comprising:

generating a sequence number in the first message at the server before sending the first message to the client device for storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device for rejection of the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

117. The method of claim 107 further comprising:
a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

118. The method of claim 107 further comprising:
providing a socket type abstraction layer in the server to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

119. The method of claim 107 further comprising:
authenticating at the server, multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

120. The method of claim 107 further comprising:
controlling, in a business agent in the server, access to client device user information in an ecommerce transaction between the service provider and the client device user,
wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

121. An apparatus for asymmetrical communication in an interactive television system comprising:

a server for receiving a first message containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol;
translating the first message into a client device compatible protocol, that is different from the service provider compatible protocol;
a compressor for compressing the first message at the server; and
a message transmission component for sending the compressed first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connection.

122. The apparatus of claim 121 further comprising:
a communication link for sending a uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server;
a translater component for translating the uncompressed second message into the service provider compatible protocol; and
a message transmission component for sending the translated uncompressed second message from the server to the service provider.

123. The apparatus of claim 122 further comprising:
an encryption component for, prior to sending the first message to the client device, encrypting the first message at the server and encapsulating the encrypted data into the client device compatible protocol message, setting a flag in the encrypted first message to indicate that the encrypted first message is encyrpted and sending the encrypted first message to the client device; and
a decryption component for decrypting the encrypted first message at the client device.

124. The apparatus of claim 123 further comprising:
a fragment encyption component for individually encrypting at the server a fragment of the first message into the client device compatible protocol.

125. The apparatus of claim 121 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

126. The apparatus of claim 121 further comprising:
a message flow rate controller for controlling message flow rate by controlling the bit rate of transmission.

127. An apparatus for communication in a distributed computing system comprising:
a client device hardware identifier for the client device retrieved from the client device dependant transport layer;
computer memory for storing the client device hardware identifier in a network operator hardware identifier list;
a authentication component for authenticating the client device hardware identifier before establishing a communication session between the server and the client device;
a server message handling component for receiving a first message containing at least one of application code, control, data, and audio/visual data at a server in a service provider compatible protocol and sending the first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connnection, the server message handling component further comprising translating a second message received from a client device into the service provider compatible protocol and sending the translated second message from the server to the service provider;
a server translation component for translating the first message into a client device compatible protocol;
a client device message handling component for sending the second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server; and
a server component for generating a session identifier from the client device hardware identifier and inserting the session identifier in place of the client device

hardware identifier in the second message prior to sending the second message to the service provider.

128. The apparatus of claim 127 further comprising:
a server component for determining whether to send the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

129. The apparatus of claim 127 further comprising:
computer memory for storing the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

130. The apparatus of claim 129 wherein the first message and second messsage comprise an ecommerce transaction.

131. The apparatus of claim 127 further comprising:
a client device component for encapsulating HTTP in the device compatible protocol in the second message at the client device prior to sending the second messsage to the server;
a HTTP conversion component for converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
computer memory for caching a cookie received from a service provider via a HTTP server at the server in response to the converted HTTP message;
a server component for generating a cookie to session identifier translation table and using the cookie to session identifier table to answer a client device hardware

identifier name request from the HTTP server and using the client device hardware indentifer to extract user information from a central registry.

132. The apparatus of claim 131 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

133. The apparatus of claim 127 wherein the the server sends a business filter to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

134. The apparatus of claim 127 further comprising:
a server component for completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

135. The apparatus of claim 127 further comprising:
a third message sent to the client device requesting a quantity of memory available at the client device; and
a component for checking a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

136. The apparatus of claim 127 further comprising:
a sequence number generated in the first message at the server before sending the first message to the client device;
computer memory for storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and
a rejection component for rejecting the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

137. The apparatus of claim 127 further comprising:
a data name service component for resolving a service identifier identifying a service provider in a transport communication protocol message.

138. The apparatus of claim 127 further comprising:
a socket type abstraction layer component to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

139. The apparatus of claim 127 further comprising:
an authentication component for authenticating multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

140. The apparatus of claim 127 further comprising:
a business agent for controlling access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

141. An apparatus for asymmetrical communication in an interactive television system comprising:
a communication link for receiving a first message from a server at a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at the server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

142. The apparatus of claim 141 the further comprising:

a client message handler component sending an uncompressed second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for translation of the uncompressed second message into the service provider compatible protocol for transmission to the service provider.

143. The apparatus of claim 142 wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to receipt of the first message at the client device; and
a decryption component for decrypting the encrypted first message at the client device.

144. The apparatus of claim 143 wherein the first message has been divided in to fragments which have been individually encrypted at the server prior to receipt of the first message at the client device.

145. The apparatus of claim 141 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

146. The apparatus of claim 141 further comrpising:
a message flow rate controller for controlling message flow rate by controlling the transmission bit rate.

147. An apparatus for communication in a distributed computing system comprising:
a first message containing at least one of application code, control, data, and audio/visual data at the client device from a server, wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol and sent the first message to the client device over at least one of a broadcast carrier wave, local area network and point to point connnection;

a client device hardware identifier for the client device retrieved from the client device dependant transport layer;
a client device message handler for sending the client device hardware indentifier to an identifier list for storage for authentication of the client device hardware identifier before establishing a communication session between the client device and a server, and sending a second message containing at least one of application code, control, data, and audio/visual data from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

148. The apparatus of claim 147 further comprising:
a client component for determining whether to receive the first message over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

149. The apparatus of claim 147 further comprising:
computer memory for storing the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

150. The apparatus of claim 149 wherein the first message and second messsage comprise an ecommerce transaction.

151. The apparatus of claim 147 wherein the second message encapsulates HTTP in the device compatible protocol at the client device prior to sending the second

messsage to the server for converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
a session identifier component for sending the session identifer to the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device, wherein the cookie is identified in a cookie to session identifier translation table, wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

152. The apparatus of claim 151 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

153. The apparatus claim 147 wherein the client device receives a business filter from the server to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

154. The apparatus of claim 147 further comprising:
a client component for completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

155. The apparatus of claim 147 further comprising:
a third message at the client device requesting a quantity of memory available at the client device; and
a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

156. The apparatus of claim 147 further comprising:

a sequence number in the first message at the server before sending the first message to the client device;
computer memory for storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device; and
a rejection component for rejecting the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

157. The apparatus of claim 147 further comprising:
a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

158. The apparatus of claim 147 further comprising:
a socket type abstraction layer to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

159. The apparatus of claim 147 further comprising:
multiple users identifiers authenticated at the client device using nicknames, personal identifiers and the client device hardware identifier.

160. The apparatus of claim 147 further comprising:
a business agent for controlling access to client device user information in an ecommerce transaction between a service provider and a client device user, wherein the amount and type of client business information provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

161. An apparatus for asymmetrical communication in an interactive television system comprising:

a first message from a server to a client device, containing at least one of application code, control, data, and audio/visual data in a service provider compatible protocol, wherein the first message has been compressed and translated into the client device compatible protocol at a server from the service provider compatible protocol, the client device compatible protocol being different from the service provider compatible protocol.

162. The apparatus of claim 161 further comprising:
an uncompressed second message containing at least one of application code, control, data, and audio/visual data sent from the client device in the client device compatible protocol to the server for translation of the uncompressed second message into the service provider compatible protocol for transmission to the service provider.

163. The apparatus of claim 162 wherein the first message is encrypted and encapsulated into the client device compatible protocol and a flag set in the first message indicating that the message is encrypted at the server prior to sending the first message to the client device for and decrypting of the encrypted first message at the client device.

164. The apparatus of claim 163 wherein the first message is divided into fragments and each fragment individually encrypted at the server prior to sending the first message to the client device.

165. The apparatus of claim 161 wherein the client device compatible protocol is compatible with a native transport layer in the client device.

166. The apparatus of claim 161 further comrpising:
a message flow rate controller for controlling message flow rate by controlling the transmission bit rate at the server.

167. An apparatus for communication in a distributed computing system comprising:
a client device hardware indentifier retrieved from the client device dependent transport layer, to a server identifier list for storage and authentication of the client device hardware identifier before establishing a communication session between the client device and a server;
a first message sent from the server to the client device containing at least one of application code, control, data, and audio/visual data sent over at least one of a broadcast carrier wave, local area network and point to point connnection, wherein the server translated the message from a service provider compatible protocol into the client device compatible protocol;
a second message containing at least one of application code, control, data, and audio/visual data received from the client device in the client device compatible protocol to the server for generating a session identifier from the client device hardware identifier, wherein the server inserts the session identifier in place of the client device hardware identifier in the second message and translates the second message into the service provider compatible protocol for sending the translated message to the service provider.

168. The apparatus of claim 167 further comprising:
a server routing component for determining whether to send the first message from the server to the client device over the broadcast carrier wave or the point to point connection based on broadcast carrier wave and point to point connection latency, broad cast stream and point to point connection loading conditions and message size.

169. The apparatus of claim 167 further comprising:
computer memory for storing at the server the second message for transmission upon satisfaction of a timing constraint comprising at a least one of as soon as possible, when connected, after a random period of time, after a set period of time, after the occurrence of an event, after occurrence of a message, and spread over available bandwidth.

170. The apparatus of claim 169 wherein the first message and second messsage comprise an ecommerce transaction.

171. The appartus of claim 167 further comprising
a converter for receiving the second message containing HTTP encapsulated in the device compatible protocol in the second message converting the second message into a standard HTTP communication protocol message at the server prior to sending the converted HTTP message to the service provider;
a session identifer component for receiving the session identifer at the server for association with a cached cookie received from a service provider via a HTTP server in response to the converted HTTP message sent by the client device,
wherein the cookie is identified in a cookie to session identifier translation table,
wherein the cookie to session identifier table is used to answer a client device hardware identifier name request from the HTTP server and to extract user information from a central registry.

172. The apparatus of claim 171 wherein the first message comprises HTTP over TCP/IP and the second message comprises LHTTP over DATP.

173. The apparatus claim 167 further comprising:
a business filter from the server to the client device to select information to be captured from input to the client device based on at least one of client preferences, viewer profiles and transaction history.

174. The apparatus of claim 167 further comprising:
a server component for completing an ecommerce transaction between a user at the client device and the service provider when the client device is offline.

175. The apparatus of claim 167 further comprising:
a third message to the client device requesting a quantity of memory available at the client device; and

a message size of a fourth message directed to the client device to verify that the quantity of available memory at the client device is sufficient to receive the fourth message before forwarding the fourth message to the client device.

176. The apparatus of claim 167 further comprising:
a sequence number in the first message at the server before sending the first message to the client device for storing the sequence number along with a time stamp in the client device upon receipt of the first message at the client device for rejection of the first message at the client device if the sequence number appears within a sliding time rejection window to avoid duplicate receipt of the first message.

177. The apparatus of claim 167 further comprising:
a data name service for resolving a service identifier identifying a service provider in a transport communication protocol message.

178. The apparatus of claim 167 further comprising:
a socket type abstraction layer in the server to accommodate User Datagram Protocol (UDP) data, wherein the socket type abstraction layer runs on top of UDP and encapsulates UDP into transport level protocol messages.

179. The apparatus of claim 167 further comprising:
an authentication component authenticating at the server, multiple users at the client device using nicknames, personal identifiers and the client device hardware identifier.

180. The apparatus of claim 167 further comprising:
a business agent in the server, for controlling access to client device user information in an ecommerce transaction between the service provider and the client device user, wherein the amount and type of client business information

provided to the service provider is guided by a business rule depending on an agreement between the service provider and a network operator.

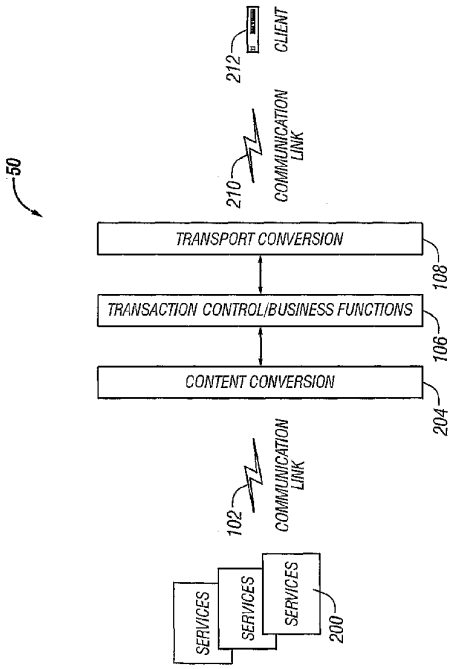

50
212
CLIENT
210
COMMUNICATION LINK
TRANSPORT CONVERSION
108
TRANSACTION CONTROL/BUSINESS FUNCTIONS
106
CONTENT CONVERSION
204
102
COMMUNICATION LINK
SERVICES
SERVICES
SERVICES
200

FIG. 1

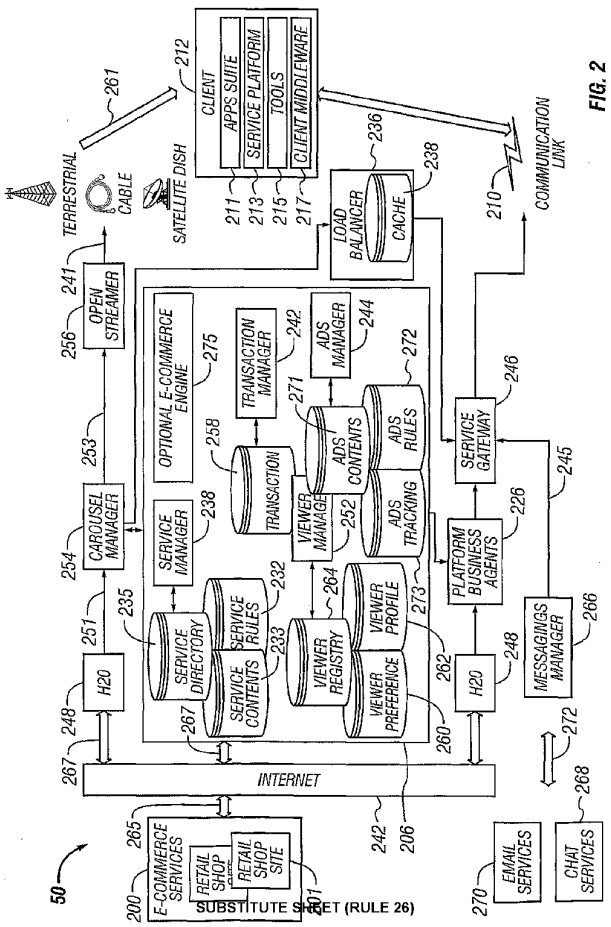

50
E-COMMERCE SERVICES
200
RETAIL SHOP SITE
RETAIL SHOP SITE
201
265
INTERNET
267
H20
248
SERVICE DIRECTORY
235
SERVICE CONTENTS
267
SERVICE RULES
232
233
CAROUSEL MANAGER
254
SERVICE MANAGER
238
OPTIONAL E-COMMERCE ENGINE
275
OPEN STREAMER
256
241
253
TRANSACTION
258
TRANSACTION MANAGER
242
VIEWER MANAGE
252
ADS CONTENTS
271
ADS TRACKING
ADS RULES
272
ADS MANAGER
244
VIEWER REGISTRY
264
VIEWER PREFERENCE
260
VIEWER PROFILE
273
262
206
242
TERRESTRIAL
261
CABLE
SATELLITE DISH
CLIENT
212
APPS SUITE
211
SERVICE PLATFORM
213
TOOLS
215
CLIENT MIDDLEWARE
217
LOAD BALANCER
CACHE
236
238
COMMUNICATION LINK
210
SERVICE GATEWAY
246
245
PLATFORM BUSINESS AGENTS
226
H20
248
MESSAGINGS MANAGER
266
272
268
270
EMAIL SERVICES
CHAT SERVICES

FIG. 2

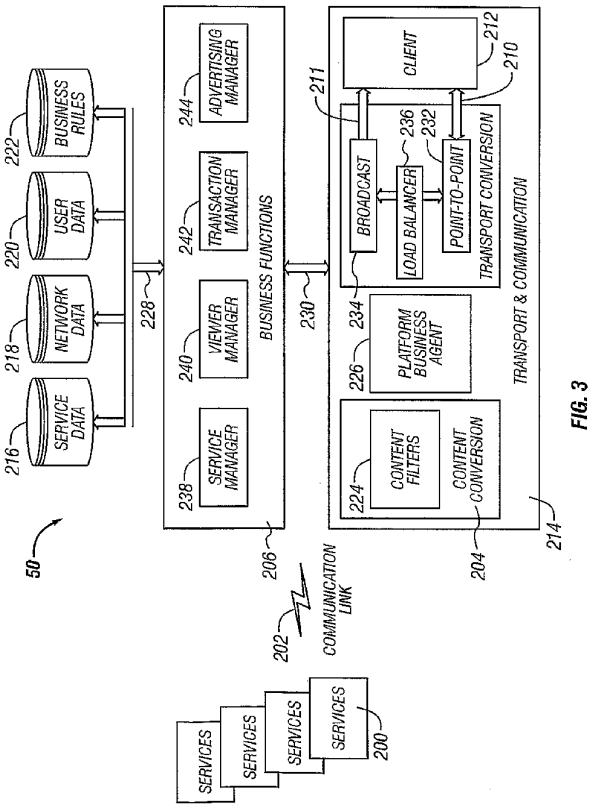

50
SERVICE DATA
216
NETWORK DATA
218
USER DATA
220
BUSINESS RULES
222
228
SERVICE MANAGER
238
VIEWER MANAGER
240
TRANSACTION MANAGER
242
ADVERTISING MANAGER
244
BUSINESS FUNCTIONS
206
230
211
CLIENT
212
210
BROADCAST
LOAD BALANCER
236
POINT-TO-POINT
232
TRANSPORT CONVERSION
PLATFORM BUSINESS AGENT
226
234
CONTENT FILTERS
224
CONTENT CONVERSION
204
TRANSPORT & COMMUNICATION
214
COMMUNICATION LINK
202
SERVICES
SERVICES
SERVICES
SERVICES
200

FIG. 3

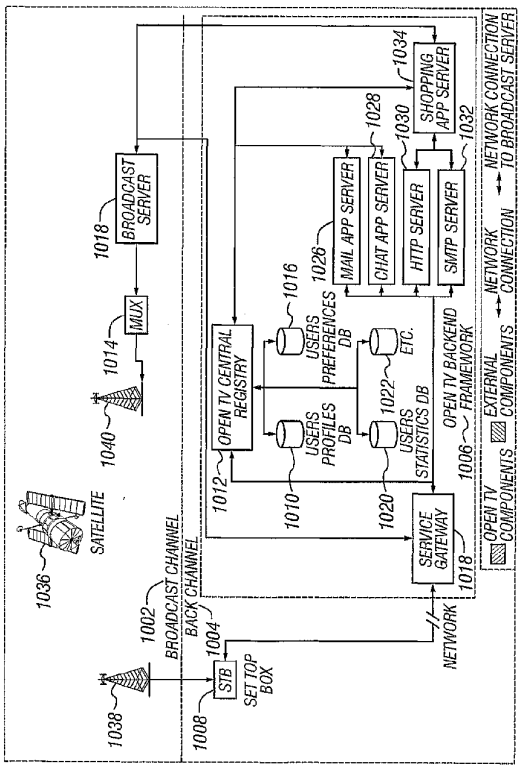

1018
BROADCAST SERVER
1014
MUX
1040
SATELLITE
1036
1002
BROADCAST CHANNEL
BACK CHANNEL
1004
STB
SET TOP BOX
1008
1038
1012
OPEN TV CENTRAL REGISTRY
1016
USERS PREFERENCES DB
USERS PROFILES DB
1010
1022
ETC.
USERS STATISTICS DB
1020
SERVICE GATEWAY
1018
NETWORK
1006
OPEN TV BACKEND FRAMEWORK
1026
MAIL APP SERVER
1028
CHAT APP SERVER
1030
HTTP SERVER
1032
SMTP SERVER
1034
SHOPPING APP SERVER
OPEN TV COMPONENTS
EXTERNAL COMPONENTS
NETWORK CONNECTION
NETWORK CONNECTION TO BROADCAST SERVER

FIG. 4

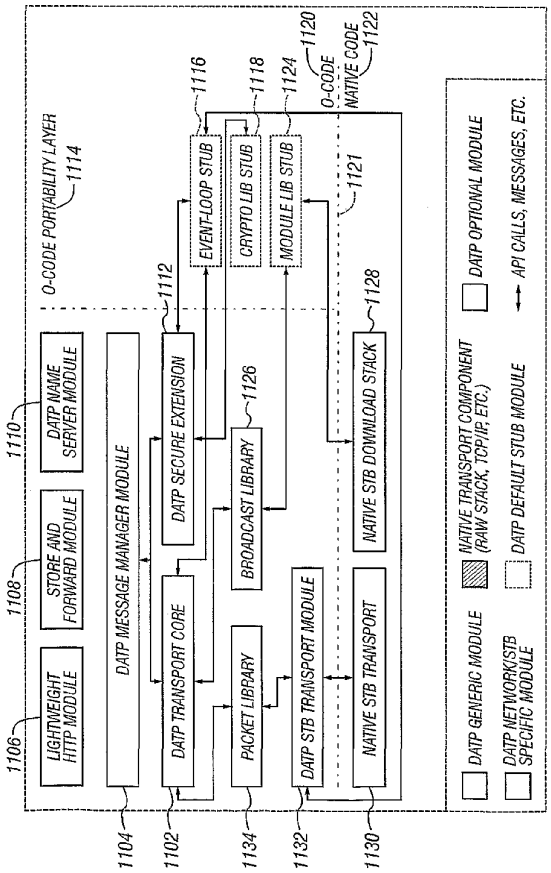

O-CODE PORTABILITY LAYER
1114
1110
DATP NAME SERVER MODULE
1108
STORE AND FORWARD MODULE
1106
LIGHTWEIGHT HTTP MODULE
DATP MESSAGE MANAGER MODULE
1112
DATP SECURE EXTENSION
DATP TRANSPORT CORE
PACKET LIBRARY
1126
BROADCAST LIBRARY
DATP STB TRANSPORT MODULE
NATIVE STB TRANSPORT
1128
NATIVE STB DOWNLOAD STACK
1116
EVENT-LOOP STUB
1118
CRYPTO LIB STUB
1124
MODULE LIB STUB
1120
O-CODE
1121
NATIVE CODE
1122
1104
1102
1134
1132
1130
DATP GENERIC MODULE
DATP NETWORK/STB SPECIFIC MODULE
NATIVE TRANSPORT COMPONENT (RAW STACK, TCP/IP, ETC.)
DATP DEFAULT STUB MODULE
DATP OPTIONAL MODULE
API CALLS, MESSAGES, ETC.

FIG. 5

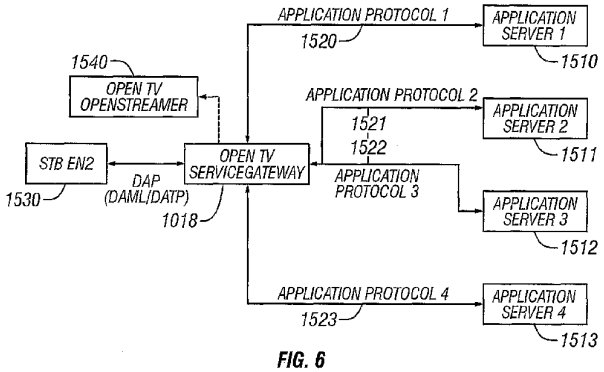

APPLICATION PROTOCOL 1
1520
APPLICATION SERVER 1
1510
1540
OPEN TV OPENSTREAMER
APPLICATION PROTOCOL 2
1521
1522
APPLICATION SERVER 2
1511
STB EN2
OPEN TV SERVICEGATEWAY
APPLICATION PROTOCOL 3
APPLICATION SERVER 3
1512
1530
DAP (DAML/DATP)
1018
APPLICATION PROTOCOL 4
1523
APPLICATION SERVER 4
1513

FIG. 6

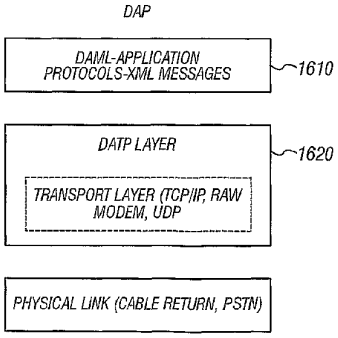

DAP
DAML-APPLICATION PROTOCOLS-XML MESSAGES
1610
DATP LAYER
1620
TRANSPORT LAYER (TCP/IP, RAW MODEM, UDP
PHYSICAL LINK (CABLE RETURN, PSTN)

FIG. 7

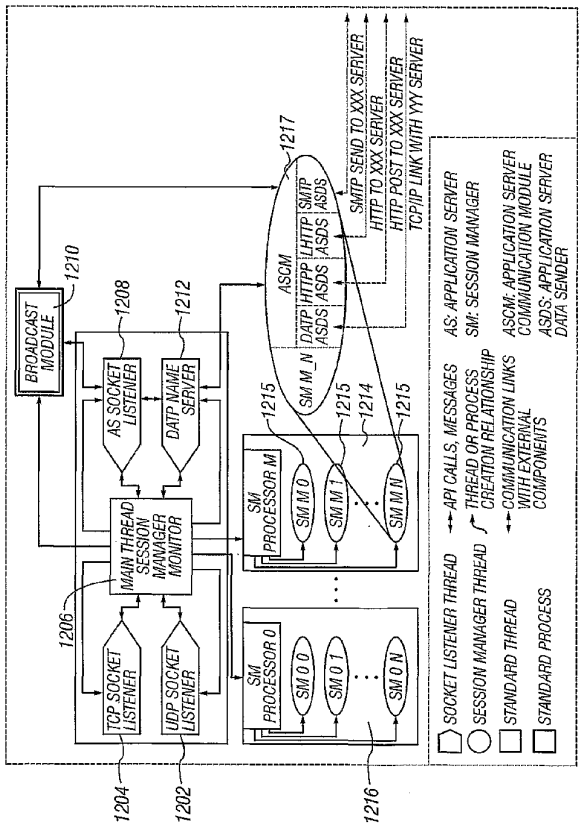

BROADCAST MODULE
1210
AS SOCKET LISTENER
1208
DATP NAME SERVER
1212
MAIN THREAD SESSION MANAGER MONITOR
1206
TCP SOCKET LISTENER
UDP SOCKET LISTENER
1204
1202
SM PROCESSOR M
SM M 0
SM M 1
SM M N
SM PROCESSOR 0
SM 0 0
SM 0 1
SM 0 N
1216
1215
1214
1217
ASGM
DATP ASDS
HTTP ASDS
LHTTP ASDS
SMTP ASDS
SM M_N
SMTP SEND TO XXX SERVER
HTTP TO XXX SERVER
HTTP POST TO XXX SERVER
TCP/IP LINK WITH YYY SERVER
SOCKET LISTENER THREAD
SESSION MANAGER THREAD
STANDARD THREAD
STANDARD PROCESS
API CALLS, MESSAGES
THREAD OR PROCESS CREATION RELATIONSHIP
COMMUNICATION LINKS WITH EXTERNAL COMPONENTS
AS: APPLICATION SERVER
SM: SESSION MANAGER
ASGM: APPLICATION SERVER COMMUNICATION MODULE
ASDS: APPLICATION SERVER DATA SENDER

FIG. 8

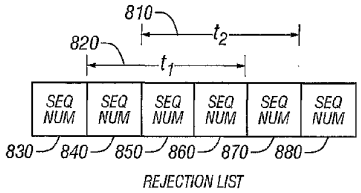

810
t_2
820
t_1
SEQ NUM
SEQ NUM
SEQ NUM
SEQ NUM
SEQ NUM
SEQ NUM
830
840
850
860
870
880
REJECTION LIST

FIG. 9

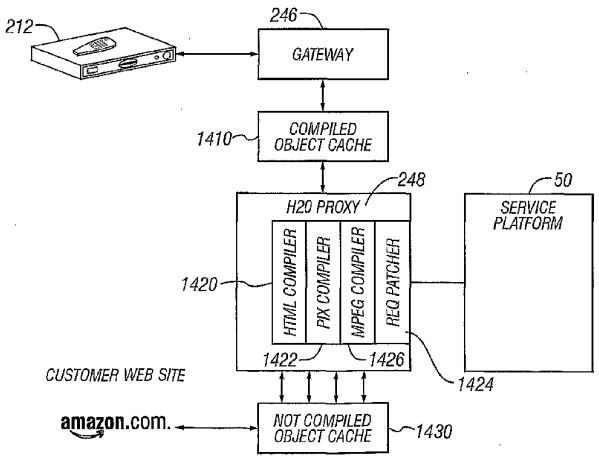

212
246
GATEWAY
1410
COMPILED OBJECT CACHE
248
H20 PROXY
50
SERVICE PLATFORM
1420
HTML COMPILER
PIX COMPILER
MPEG COMPILER
REQ PATCHER
1422
1426
1424
CUSTOMER WEB SITE
amazon.com.
NOT COMPILED OBJECT CACHE
1430

FIG. 11

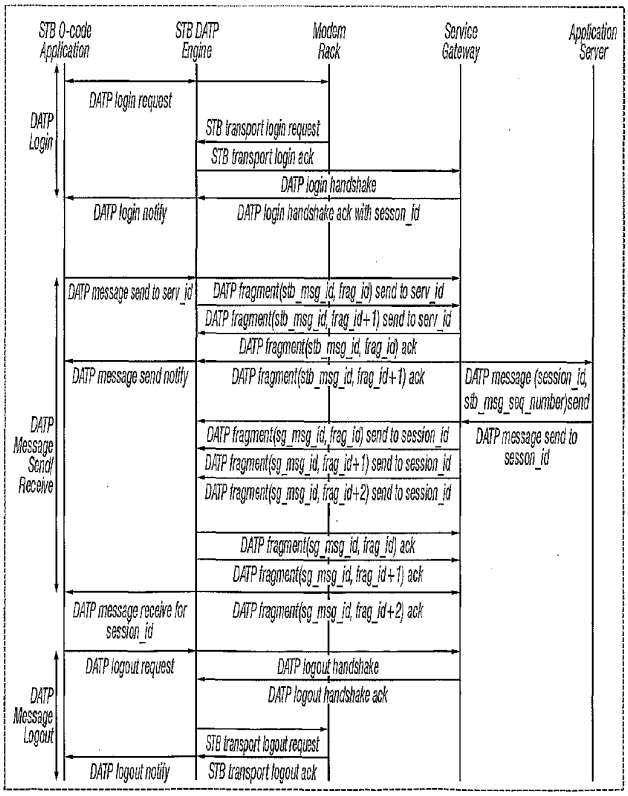

STB O-code Application
STB DATP Engine
Modem Rack
Service Gateway
Application Server
DATP Login
DATP login request
STB transport login request
STB transport login ack
DATP login handshake
DATP login notify
DATP login handshake ack with sesson_id
DATP Message Send/ Receive
DATP message send to serv_id
DATP fragment(stb_msg_id, frag_id) send to serv_id
DATP fragment(stb_msg_id, frag_id+1) send to serv_id
DATP fragment(stb_msg_id, frag_id) ack
DATP message send notify
DATP fragment(stb_msg_id, frag_id+1) ack
DATP message (session_id, stb_msg_seq_number)send
DATP fragment(sg_msg_id, frag_id) send to session_id
DATP message send to sesson_id
DATP fragment(sg_msg_id, frag_id+1) send to session_id
DATP fragment(sg_msg_id, frag_id+2) send to session_id
DATP fragment(sg_msg_id, frag_id) ack
DATP fragment(sg_msg_id, frag_id+1) ack
DATP message receive for session_id
DATP fragment(sg_msg_id, frag_id+2) ack
DATP Message Logout
DATP logout request
DATP logout handshake
DATP logout handshake ack
STB transport logout request
DATP logout notify
STB transport logout ack

FIG. 10

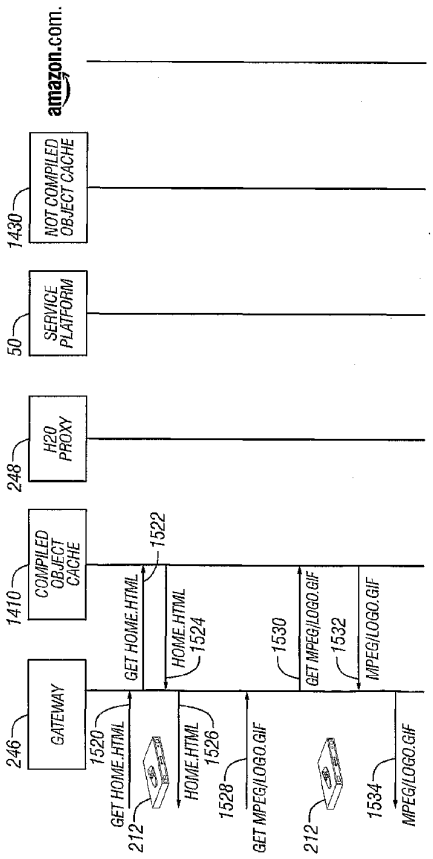

amazon.com.
1430
NOT COMPILED OBJECT CACHE
50
SERVICE PLATFORM
248
H20 PROXY
1410
COMPILED OBJECT CACHE
246
GATEWAY
1520
GET HOME.HTML
212
1522
GET HOME.HTML
HOME.HTML
1524
HOME.HTML
1526
1528
GET MPEG/LOGO.GIF
212
1530
GET MPEG/LOGO.GIF
1532
MPEG/LOGO.GIF
1534
MPEG/LOGO.GIF

FIG. 12

11/13

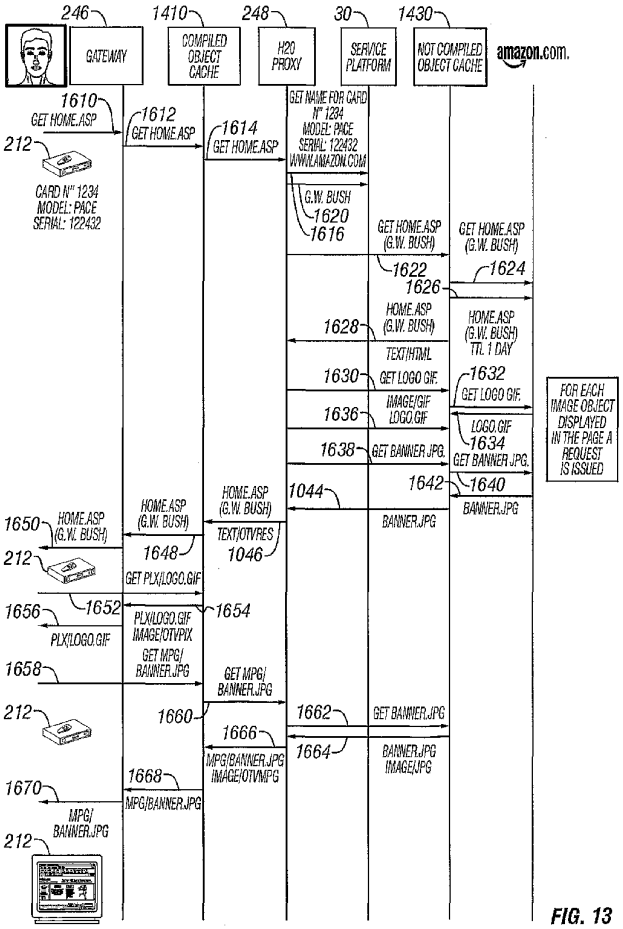

246
1410
248
30
1430
GATEWAY
COMPILED OBJECT CACHE
H20 PROXY
SERVICE PLATFORM
NOT COMPILED OBJECT CACHE
amazon.com.
1610
GET HOME.ASP
1612
GET HOME.ASP
1614
GET HOME.ASP
GET NAME FOR CARD N° 1234 MODEL: PACE SERIAL: 122432 WWW.AMAZON.COM
212
CARD N° 1234 MODEL: PACE SERIAL: 122432
G.W. BUSH
1620
1616
GET HOME.ASP (G.W. BUSH)
GET HOME.ASP (G.W. BUSH)
1622
1624
1626
HOME.ASP (G.W. BUSH)
1628
HOME.ASP (G.W. BUSH) TTL 1 DAY
TEXT/HTML
1630
GET LOGO GIF
1632
GET LOGO GIF
IMAGE/GIF LOGO.GIF
1636
LOGO.GIF
1634
1638
GET BANNER.JPG
GET BANNER.JPG
1642
1640
1044
BANNER.JPG
BANNER.JPG
FOR EACH IMAGE OBJECT DISPLAYED IN THE PAGE A REQUEST IS ISSUED
HOME.ASP (G.W. BUSH)
1650
HOME.ASP (G.W. BUSH)
HOME.ASP (G.W. BUSH)
TEXT/OTVRES
1046
1648
212
GET PLX/LOGO.GIF
1656
1652
1654
PLX/LOGO.GIF IMAGE/OTVPIX
PLX/LOGO.GIF
GET MPG/ BANNER.JPG
1658
GET MPG/ BANNER.JPG
212
1660
1662
GET BANNER.JPG
1666
1664
MPG/BANNER.JPG IMAGE/OTVMPG
BANNER.JPG IMAGE.JPG
1668
1670
MPG/BANNER.JPG
MPG/ BANNER.JPG
212

FIG. 13

12/13

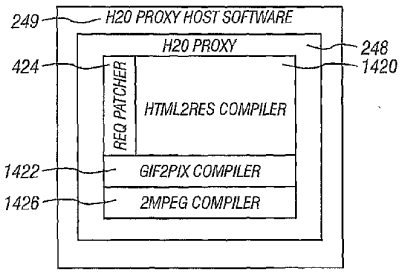


FIG. 14

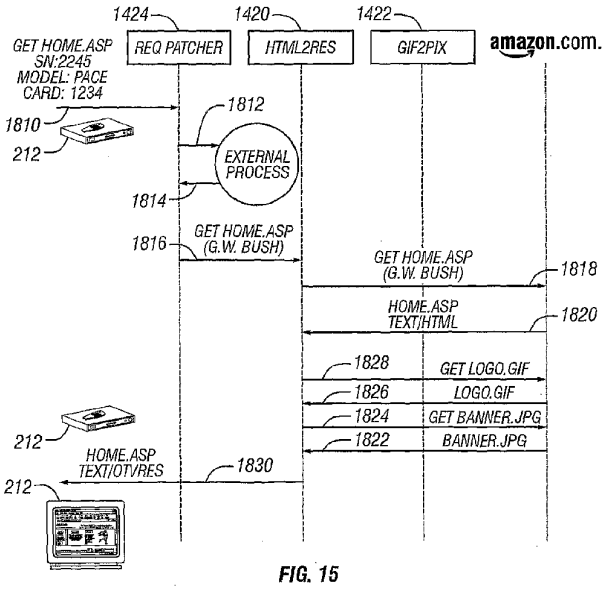


FIG. 15

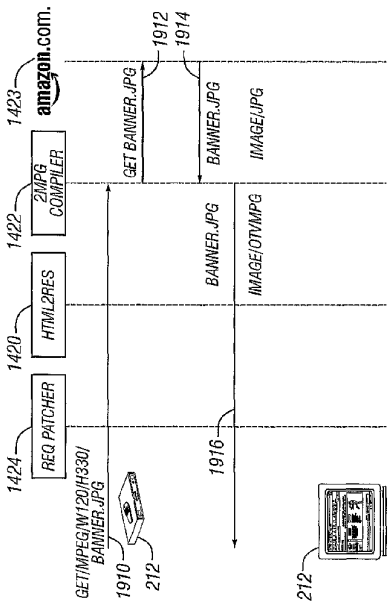

1423
amazon.com.
1422
2IMPG COMPILER
1420
HTML2RES
1424
REQ PATCHER
GET/IMPEG/W120/H330/ BANNER.JPG
1910
212
GET BANNER.JPG
1912
1914
BANNER.JPG
IMAGE/JPG
BANNER.JPG
IMAGE/OTVMPG
1916
212

FIG. 16

【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 August 2002 (15.08.2002)

PCT

(10) International Publication Number
WO 02/063879 A3

(51) International Patent Classification[7]: H04L 29/06, H04N 7/173

(21) International Application Number: PCT/US02/02725

(22) International Filing Date: 1 February 2002 (01.02.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/265,986	2 February 2001 (02.02.2001)	US
60/266,210	2 February 2001 (02.02.2001)	US
60/267,876	9 February 2001 (09.02.2001)	US
60/269,261	15 February 2001 (15.02.2001)	US
60/279,543	28 March 2001 (28.03.2001)	US
09/858,436	16 May 2001 (16.05.2001)	US

(71) Applicant: OPENTV, INC. [US/US]; 275 Sacramento Street, San Francisco, CA 94111 (US).

(72) Inventors: ALAO, Rachad; 330 Angel Avenue, Sunnyvale, CA 94086 (US). DELPUCH, Alain; 20, avenue André Prothin, F-92927 Paris la Défense (FR). DUREAU, Vincent; 3519 South Court, Palo Alto, CA 94306 (US). HENRARD, Jose; 14, rue de Liège, F-75005 Paris (FR). HUNTINGTON, Matthew; 23 Gordon Avenue, Twickenham TW1 1NII (GB). LAM, Waiman; 2137 Sunsprite Drive, Union City, CA 94587 (US). KIDD, Taylor; 977 Upland Road, Redwood City, CA 94062 (US).

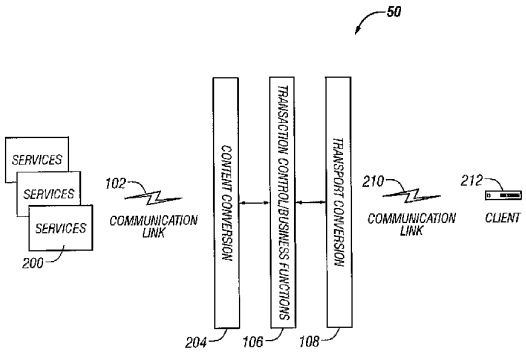
(74) Agent: RANKIN, Rory, D.; Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C., P.O. Box 398, Austin, TX 78767-0398 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: A SERVICE GATEWAY FOR INTERACTIVE TELEVISION



(57) Abstract: A service gateway provides a proxy between a client protocol and a plurality of standard communication protocols. The service gateway provides asymmetrical routing, data compression and encryption to optimize client processing power and communication link bandwidth. The service gateway enables content translation between clients and service providers. The service gateway keeps track of client available memory and sequence numbers in messages to generate error codes when applicable. A store and forward message capability is provided along with abstract session identifiers. The service gateway supports user datagram protocol.

WO 02/063879 A3

WO 02/063879 A3

European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
with international search report

(88) Date of publication of the international search report:
18 December 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Internal application No: PCT/US 02/02725

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04N7/173

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)
IPC 7 H04L H04N

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, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 53581 A (COACTIVE NETWORKS INC) 26 November 1998 (1998-11-26) abstract page 2, line 20 -page 2, line 30 page 3, line 13 -page 4, line 13 page 4, line 24 -page 6, line 30 page 7, line 29 -page 8, line 25 page 9, line 1 -page 11, line 9 page 12, line 5 -page 12, line 30 page 13, line 16 -page 14, line 15; claims 1-3,5,8,9,22; figures 1-6 --- -/--	1-6, 61-66, 121-126

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in 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.

& document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 April 2003	02/05/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Todorut, C

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Internatl: pplication No
PCT/US 02/02725

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 722 249 A (US WEST MARKETING RESOURCES) 17 July 1996 (1996-07-17) abstract column 1, line 49 -column 2, line 34 column 3, line 42 -column 6, line 19; claims 1-8; figures 1-5	1-6, 61-66, 121-126
X	WO 00 24192 A (GEN INSTRUMENT CORP ;MEANDZIJA BRANISLAV N (US)) 27 April 2000 (2000-04-27) abstract page 3, line 7 -page 4, line 28 page 5, line 15 -page 6, line 26 page 11, line 28 -page 13, line 30 page 15, line 1 -page 16, line 30 page 22, line 5 -page 22, line 30 page 26, line 5 -page 40, line 9 page 42, line 15 -page 52, line 26; claim 1; figures 1-10	1-6, 61-66, 121-126
A	WO 01 06784 A (UNITED VIDEO PROPERTIES INC) 25 January 2001 (2001-01-25) abstract page 8, line 23 -page 12, line 6 page 12, line 18 -page 13, line 30 page 16, line 19 -page 16, line 29 page 20, line 17 -page 21, line 5	1-6, 61-66, 121-126
A	DROITCOURT J L: "UNDERSTANDING HOW INTERACTIVE TELEVISION SET TOP BOX WORKS... AND WHAT IT WILL MEAN TO THE CUSTOMER" , INTERNATIONAL BROADCASTING CONVENTION, LONDON, GB, VOL. 413, PAGE(S) 382-394 XP000562377 the whole document	1-6, 61-66, 121-126
A	EVAIN J-P: "THE MULTIMEDIA HOME PLATFORM" , EBU REVIEW- TECHNICAL, EUROPEAN BROADCASTING UNION. BRUSSELS, BE, NR. 275, PAGE(S) 4-10 XP000767493 ISSN: 0251-0936 the whole document	1-6, 61-66, 121-126
A	COLAITIS F ET AL: "MHEG AND ITS PROFILE FOR ITV APPLICATIONS" , IEE COLLOQUIUM ON INTERACTIVE TELEVISION, IEE, LONDON, GB, NR. 1995/159, PAGE(S) 3-1-3-8 XP000646001 the whole document	1-6, 61-66, 121-126

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte al application No.
PCT/US 02/02725

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 7-60,67-120,127-180
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

International Application No. PCT/US 02 02725

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box I.2

Claims Nos.: 7-60,67-120,127-180

In view of the large number and also the wording of the claims presently on file, which render it difficult, if not impossible, to determine the matter for which protection is sought, the present application fails to comply with the clarity and conciseness requirements of Article 6 PCT (see also Rule 6.1(a) PCT) to such an extent that a meaningful search is impossible. Consequently, the search has been carried out for those parts of the application which do appear to be clear (and concise), namely claims 1-6,61-66, and 121-126 which define a computer readable medium (1-6), the corresponding method for asymetrical communication implemented in an interactive television system (61-66) and the apparatus for implementing such a method (121-126).

The applicant's attention is drawn to the fact that claims, or parts of claims, relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure.

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internati pplication No
PCT/US 02/02725

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9853581	A	26-11-1998	EP	1013047 A1	28-06-2000
			JP	2002512758 T	23-04-2002
			WO	9853581 A1	26-11-1998
EP 0722249	A	17-07-1996	US	5583563 A	10-12-1996
			EP	0722249 A2	17-07-1996
			JP	8235091 A	13-09-1996
WO 0024192	A	27-04-2000	AU	6158499 A	08-05-2000
			BR	9914604 A	11-12-2001
			CA	2346891 A1	27-04-2000
			CN	1326638 T	12-12-2001
			EP	1123620 A1	16-08-2001
			JP	2002528971 T	03-09-2002
			WO	0024192 A1	27-04-2000
WO 0106784	A	25-01-2001	AU	5926500 A	05-02-2001
			EP	1279285 A2	29-01-2003
			WO	0106784 A2	25-01-2001

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(31)優先権主張番号 60/269,261
(32)優先日 平成13年2月15日(2001.2.15)
(33)優先権主張国 米国(US)
(31)優先権主張番号 60/279,543
(32)優先日 平成13年3月28日(2001.3.28)
(33)優先権主張国 米国(US)
(31)優先権主張番号 09/858,436
(32)優先日 平成13年5月16日(2001.5.16)
(33)優先権主張国 米国(US)

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

(72)発明者 ドゥリュー，ヴィンセント
アメリカ合衆国・９４３０６・カリフォルニア州・パロ　アルト・サウス　コート・３５１９
(72)発明者 ヘンラード，ジョゼ
フランス国・エフ－７５００５　パリ・リュ　ド　リージュ・１４
(72)発明者 ハンティントン，マシュー
イギリス国・ティダブリュ１　１エヌエイチ・ツイッケンハム・ゴードン　アベニュ・２３
(72)発明者 ラム，ワイマン
アメリカ合衆国・９４５８７・カリフォルニア州・ユニオン　シティ・サンスプライト　ドライブ・２１３７
(72)発明者 キッド，テイラー
アメリカ合衆国・９４０６２・カリフォルニア州・レッドウッド　シティ・アップランド　ロード・９７７

Ｆターム(参考) 5C064 BB10 BC16 BC20
5J104 PA07 PA14