

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5442755号
(P5442755)

(45) 発行日 平成26年3月12日 (2014. 3. 12)

(24) 登録日 平成25年12月27日 (2013. 12. 27)

(51) Int.Cl.

F I

G 0 6 F 13/12 (2006.01)

G 0 6 F 13/12 3 4 0 C

請求項の数 15 (全 33 頁)

(21) 出願番号 特願2011-538616 (P2011-538616)
 (86) (22) 出願日 平成21年10月30日 (2009. 10. 30)
 (65) 公表番号 特表2012-510126 (P2012-510126A)
 (43) 公表日 平成24年4月26日 (2012. 4. 26)
 (86) 国際出願番号 PCT/US2009/062746
 (87) 国際公開番号 W02010/062679
 (87) 国際公開日 平成22年6月3日 (2010. 6. 3)
 審査請求日 平成24年10月23日 (2012. 10. 23)
 (31) 優先権主張番号 12/324, 305
 (32) 優先日 平成20年11月26日 (2008. 11. 26)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100107766
 弁理士 伊東 忠重
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (74) 代理人 100091214
 弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】 リモートデスクトッププロトコルのためのハードウェアアクセラレーション

(57) 【特許請求の範囲】

【請求項 1】

ターミナルサーバまたは仮想マシンへのリモートアクセスを提供するコンピューティングシステムにおいて、リモートデスクトップグラフィックス処理タスクをネットワーク周辺機器にオフロードする方法であって、前記タスクはグラフィックスレンダリング及び圧縮タスクを含み、前記方法は、

前記ネットワーク周辺機器が、指定された一または複数のリモートデスクトップグラフィックス処理タスクを実行するためのタスクオフロード機能を含むことを判定することと

、
 後続のデータユニットと共に使用されるコンテキスト情報を含む、前記ネットワーク周辺機器が前記一または複数のリモートデスクトップグラフィックス処理タスクを実行するという指示を、前記ネットワーク周辺機器に送信することと、

前記一または複数のリモートデスクトップグラフィックス処理タスクを前記ネットワーク周辺機器に実行させることと

を備え、

前記リモートデスクトップグラフィックス処理タスクは、さもなければ前記コンピューティングデバイスに実行されるタスクである、
 方法。

【請求項 2】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、部分的なりモ

10

20

ートアクセス処理を提供する請求項 1 に記載の方法。

【請求項 3】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、完全なリモートアクセス処理を提供する請求項 1 に記載の方法。

【請求項 4】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、ビットマップ圧縮処理を行う請求項 1 に記載の方法。

【請求項 5】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、バルク圧縮処理を行う請求項 1 に記載の方法。

【請求項 6】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、キャッシング処理を行う請求項 1 に記載の方法。

【請求項 7】

前記一または複数のリモートデスクトップグラフィックス処理タスクは、前記コンピューティングシステム、ネットワーク接続タイプ、およびリモートクライアントのうちの少なくとも 1 つの進行中の要求に応じて、前記ネットワーク周辺機器に選択的にオフロードされる請求項 1 に記載の方法。

【請求項 8】

前記ネットワーク周辺機器に関連付けられたタスクオフロードバッファ内の少なくとも 1 つのフラグインジケータを設定することにより前記タスクオフロード機能をイネーブルするステップをさらに備えた請求項 1 に記載の方法。

【請求項 9】

前記ネットワーク周辺機器は、ネットワークインタフェースカード (NIC) である請求項 1 に記載の方法。

【請求項 10】

前記データユニットは、階層化ネットワークモデルを介して転送され、前記データユニットは、ネットワークデータおよびパケット拡張データを含むことを特徴とする請求項 1 に記載の方法。

【請求項 11】

前記データユニットは、前記ネットワーク周辺機器が前記リモートデスクトップグラフィックス処理タスクのバッチを実行することを指示し、前記パケット拡張データは、前記ネットワーク周辺機器によって実行されるために、少なくとも 1 つの前記リモートデスクトップグラフィックス処理タスクを示す少なくとも 1 つのデータフィールドを含む請求項 10 に記載の方法。

【請求項 12】

ターミナルサーバまたは仮想マシンへのリモートアクセスを提供するコンピューティングシステムであって、前記コンピューティングシステムは、リモートデスクトップグラフィックス処理タスクをネットワーク周辺機器にオフロードするように適合され、前記コンピューティングシステムは、

少なくとも 1 つのプロセッサと、

前記少なくとも 1 つのプロセッサに通信できるように結合された少なくとも 1 つのメモリとを含み、前記メモリは、前記プロセッサに、

前記ネットワーク周辺機器によってビットマップ圧縮、バルク圧縮、またはキャッシングのうちの 1 つの処理を行うリモートデスクトップグラフィックス処理タスクを実行させることを指示するデータパケットを、前記ネットワーク周辺機器に送信させ、

前記ネットワーク周辺機器に前記リモートデスクトップグラフィックス処理タスクを実行するためのタスクオフロード機能が含まれないと判定されたことに応答して、前記処理タスクを実行させる

コンピュータ実行可能命令を格納していることを特徴とするコンピューティングシステム

10

20

30

40

50

ム。

【請求項 1 3】

ターミナルサーバまたは仮想マシンへのリモートアクセスを提供するコンピューティングシステムにより実行されると、前記コンピューティングシステムに、請求項 1 ないし 1 1 いずれか一項に記載の方法のステップを実行させるコンピュータプログラム。

【請求項 1 4】

前記タスクオフロード機能は、多重化／フレーム化、暗号化、およびコマンド命令符号化のうちの 1 つをさらに備える請求項 1 3 に記載のコンピュータプログラム。

【請求項 1 5】

請求項 1 3 または 1 4 に記載のコンピュータプログラムを記憶したコンピュータ可読記憶媒体。

10

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、リモートデスクトップのアクセス機能およびタスクを周辺機器にオフロードすることにより、ハードウェア高速化を図るシステムおよび方法に関する。

【背景技術】

【0 0 0 2】

コンピューティングシステムにおけるネットワークアプリケーションは、7 階層 I S O モデルまたは W i n d o w s (登録商標) (商標) オペレーティングシステムによって使用される階層化モデルなどの階層化アーキテクチャが原因で、相当なホストプロセッサリソースを必要とし得る。データパケットがレイヤ間を進むようにデータパケットに対して実行される機能は、ソフトウェア集約型とすることができ、相当な量のプロセッサリソースおよびメモリリソースを要求し得る。さらに、ネットワークインタフェースカード (N I C) などの多くのコンピュータハードウェア周辺機器の能力、効率、およびスループットは向上している。このようなコンピュータシステム周辺機器は、しばしば専用プロセッサおよび専用メモリを備えており、そうした備えがなければコンピュータシステムプロセッサによってソフトウェアで実行される、高度で複雑なコンピューティングタスクを実行することが一般に可能である。例えば、多くの N I C は、チェックサム計算 / 検証、データ暗号化 / 復号化、メッセージダイジェスト計算、T C P セグメンテーションまたは U D P セグメンテーション、受信側パケット分類、およびサービス妨害 (D o s) 攻撃に対する防御としてのパケットフィルタリングなど、N I C が実行しなければ適切なネットワークレイヤにおいて、C P U によって、ソフトウェアで実行されるタスクを独自に実行することが可能である。そのため、このような C P U 集約型タスクを周辺ハードウェア機器にオフロードすること (offloading) には利点がある。これにより、ホストコンピュータにおける C P U 使用率およびメモリ帯域幅使用量が低減され、それによって、システム全体の効率、速度、およびスループットが向上する。

20

30

【0 0 0 3】

リモートコンピューティングシステムは、リモートコンピューティングシステムによってホストリソースにアクセスするユーザを有効にすることができる。リモートコンピューティングシステム上のサーバは、プログラムを実行し、T C P / I P プロトコルなどの通信プロトコルに準拠するネットワークを介して信号を送信することによって接続できるクライアントに、ユーザインタフェースを示す信号を伝送することができる。各接続クライアントは、セッション、すなわち、リソース一式を含む実行環境を提供され得る。各クライアントは、ユーザ入力を示す信号をサーバに伝送することができ、サーバは、ユーザ入力を適切なセッションに適用することができる。クライアントは、リモートデスクトッププロトコル (R D P) などのプロトコルを使用して、サーバリソースに接続することができる。R D P などのプロトコルは、一般に、グラフィックス、ならびに U S B、プリンタ、キーボード、およびマウスなどのデバイストラフィック、加えて、サーバとクライアントの間のアプリケーションのための仮想チャネルも扱う。ターミナルサーバは、一般的な

40

50

サーバ構成では数百存在し得るクライアントセッションを提供する。

【 0 0 0 4 】

RDPなどのプロトコルは、様々なこのようなりモートクライアントマシン機能をサポートする。一般に、サーバからのグラフィックスデータは、グラフィックスプリミティブ形式で符号化され、暗号化されるか、またはサーバ上でレンダリングされる必要があり、結果として現れるビットマップは、圧縮/暗号化され、ネットワークを介して伝送される必要がある。符号化操作、レンダリング操作、および暗号化操作は、本来は、非常にコンピュータを利用しており、高いCPU使用率を必要とする。したがって、上述の技法を使用して、このようなCPU集約型タスクを周辺ハードウェア機器にオフロードすることが有利である。

10

【発明の概要】

【 0 0 0 5 】

本開示は、これまではソフトウェアのプロセッサレベルにおいて実行されたリモート端末アクセス機能およびタスクを、コンピュータシステムに結合された適切なハードウェア周辺機器にオフロードするためのシステムおよび方法に関する。一実施形態では、タスクは、ネットワークインタフェースカード(NIC)周辺機器にオフロードすることができ、周辺機器は、オフロードされなければコンピュータCPUによって、ソフトウェアで、実行されるタスクの一部または全部を実行することができる。

【 0 0 0 6 】

一実施形態では、OS(オペレーティングシステム)は、コンピュータシステムに接続された(NICなどの)ハードウェア周辺機器のデバイスドライバに「問い合わせ(query)」を行うことができる。様々なデバイスドライバは、本明細書において「タスクオフロード機能(task offload capabilities)」と呼ばれる、それぞれのハードウェア周辺機器の処理能力を識別することによって応答することができる。一実施形態では、各特定の周辺機器のタスクオフロード機能が識別されると、OSは、その後、選択された周辺機器が一定のタスクを実行できるようにすることができる。OSは、その後、既に実行できるようにされたタスクを周辺機器が実行するよう、動的に必要なに応じて要求することができる。

20

【 0 0 0 7 】

様々な実施形態では、このようなタスクは、RDPプロトコル処理中にレンダリングおよび圧縮機能が必要とされる場合は、このようなレンダリングおよび/または圧縮を含むことができる。したがって、プロトコルデータフローとともに埋め込まれ、TCP/IPなどであり、オフロードされた他のネットワークオフロードタスク上のRDPレベルの処理のための、オフロードパスを使用することによって、ネットワークカード上のハードウェア機能を利用することができる。

30

【 0 0 0 8 】

上述に加えて、本開示の一部を形成する特許請求の範囲、図面、および本文では、他の態様も説明される。本開示の1つまたは複数の様々な態様が、限定することなく、本明細書で言及される本開示の態様を達成するための回路構成および/またはプログラミングを含むことができ、回路構成および/またはプログラミングは実質的に、システム設計者の設計選択に応じて、本明細書で言及される態様を達成するように構成されたハードウェア、および/またはソフトウェア、および/またはファームウェアの任意の組み合わせとすることができる。ことが当業者には理解できよう。

40

【 0 0 0 9 】

上述されたことは要約であり、したがって、簡略化、一般化、および細部の省略を必然的に含んでいる。要約はもっぱら一例に過ぎず、いかなる方法でも限定的であることを意図していないことが当業者には理解されよう。

【図面の簡単な説明】

【 0 0 1 0 】

【図1】本開示の態様を実施できるコンピュータシステム例を示す図である。

50

【図 1 a】複数の仮想プロセッサおよび対応するゲストオペレーティングシステムを含む複数の仮想マシンを備える仮想マシン環境であって、仮想マシンは、スケジューラおよび他のコンポーネントを含み得る仮想化レイヤによって維持され、仮想化レイヤは、複数の仮想マシンのためにハードウェアを仮想化する、仮想マシン環境を示す図である。

【図 2】本開示の態様を実施するための動作環境を示す図である。

【図 3】本開示の態様を実施するための動作環境を示す図である。

【図 4】本開示の態様を実施するための動作環境を示す図である。

【図 5】ネットワークスタックの機能レイヤおよび本明細書で開示されるバイパス経路を示す図である。

【図 6】NDIS 経路の機能レイヤおよび本明細書で開示されるバイパス経路を示す図である。

【図 7】本明細書で開示されるオフロード機構を示すラダー図である。

【図 8】ホストコンピュータと周辺機器の間の同期を説明するブロック図である。

【図 9】階層化ネットワークアーキテクチャ内に存在する機能コンポーネントのいくつかを示す図である。

【図 10】本開示にしたがって、プログラムコンポーネントを通過するデータパケットのフローを説明する機能ブロック図である。

【図 11】データパケットおよびパケット拡張の実施形態を示す図である。

【図 12】リモートターミナルサービス処理を説明するアーキテクチャ例を示す図である。

【図 13】リモートターミナルサービス処理の態様を説明する機能ブロック図である。

【図 14】本開示の態様を実施するための操作手順例を示す図である。

【図 15】本開示の態様を実施するための操作手順例を示す図である。

【図 16】本開示の態様を実施するための操作手順例を示す図である。

【図 17】上の図 1 ~ 図 16 に関して説明されるコンピュータ実行可能命令を持つコンピュータ可読媒体を示す図である。

【発明を実施するための形態】

【0011】

一般論としてのコンピューティング環境

本開示の様々な実施形態についての完全な理解を提供するために、以下の説明および図において、ある具体的な詳細が説明される。本開示の主題の様々な実施形態を不必要に曖昧にすることを避けるために、しばしばコンピューティングおよびソフトウェア技術に関連するあるよく知られた詳細は、以下の開示では説明されない。さらに、以下で説明される詳細の 1 つまたは複数を用いずに、本開示の主題の他の実施形態を実施できることが当業者には理解されよう。最後に、以下の開示では、様々な方法が手順および順序を参照して説明されるが、このような説明は、本開示の主題の実施形態の明瞭な実施を可能にするためのものであり、その手順および順序が、本開示を実施するために必須であると理解されるべきではない。

【0012】

本明細書で説明される様々な技法は、ハードウェア、ソフトウェア、または必要に応じて両方の組み合わせ、に関連して実施できることを理解されたい。したがって、本開示の方法および本開示の一連の装置、またはそれらのある態様もしくは部分は、磁気ディスク、CD-ROM、ハードドライブ、または他の任意の機械可読記憶媒体など、有形な媒体内に具現化されるプログラムコード（すなわち命令）形式を取ることができ、プログラムコードがコンピュータなどの機械にロードされ、機械によって実行された場合、機械は開示される主題を実施するための装置になる。プログラム可能なコンピュータ上でのプログラムコード実行の場合、コンピュータデバイスは一般に、プロセッサ、プロセッサによって可読な（揮発性メモリおよび不揮発性メモリおよび / または記憶素子を含む）記憶媒体、少なくとも 1 つの入力デバイス、および少なくとも 1 つの出力デバイスを含む。1 つまたは複数のプログラムは、例えば、アプリケーションプログラミングインタフェース（A

10

20

30

40

50

P I)、または再利用可能コントロールなどの使用を通して、本開示の主題に関連して説明されるプロセスを実施または利用することができる。このようなプログラムは、望ましくは、高水準手続き型プログラミング言語またはオブジェクト指向プログラミング言語で実装されて、コンピュータシステムと情報伝達を行う。しかし、プログラムは、必要に応じて、アセンブリ言語または機械語で実装することもできる。いずれの場合も、言語は、コンパイル言語またはインタプリタ言語とすることができ、ハードウェア実装と組み合わせることができる。

【 0 0 1 3 】

リモートデスクトップシステムは、クライアントコンピュータシステムによって遠隔地で実行できるアプリケーションを維持するコンピュータシステムである。入力、クライアントコンピュータシステムにおいて入力され、（例えば、リモートデスクトッププロトコル（RDP）など、国際電気通信連合（ITU）T. 120 ファミリのプロトコルに基づいたプロトコルを使用して）ネットワークを介してターミナルサーバ上のアプリケーションに転送される。アプリケーションは、入力がターミナルサーバにおいて入力されたかのように、入力を処理する。アプリケーションは、受信した入力にตอบสนองして出力を生成し、出力は、ネットワークを介してクライアントコンピュータシステムに転送される。クライアントコンピュータシステムは、出力データを渡す。したがって、入力が受け取られ、出力が渡されるのは、クライアントコンピュータシステムにおいてであるが、処理が実際に行われるのは、ターミナルサーバにおいてである。セッションは、デスクトップ、デスクトップ内でのマウスの動きを追跡するサブシステム、アイコン上でのマウスクリックをプログラムのインスタンスを生成させるコマンドに変換するサブシステムなどの、シェルおよびユーザインタフェースを含むことができる。別の実施形態例では、セッションは、アプリケーションを含むことができる。この例では、アプリケーションは描画されているが、デスクトップ環境はまだ生成されており、ユーザから隠されていることがある。上述の説明は、例示的なものであること、本開示の主題は、様々なクライアント/サーバ環境で実施することができ、特定のターミナルサービス製品に限定されないことを理解されたい。

【 0 0 1 4 】

必ずしもすべてのリモートデスクトップ環境ではないが、ほとんどの場合、（クライアントコンピュータシステムにおいて入力される）入力データは一般に、アプリケーションに対するコマンドを表すマウスおよびキーボードデータを含み、（ターミナルサーバのアプリケーションによって生成される）出力データは一般に、ビデオ出力デバイス上に表示するためのビデオデータを含む。多くのリモートデスクトップ環境は、他のタイプのデータを転送できるように拡張する機能も含む。

【 0 0 1 5 】

プラグインがRDP接続を介してデータを転送できるようにすることによって、通信チャネルを使用して、RDPプロトコルを拡張することができる。このような多くの拡張が存在する。プリンタリダイレクション、クリップボードリダイレクション、ポートリダイレクションなどの機能は、通信チャネル技術を使用する。したがって、入力データおよび出力データに加えて、データを転送する必要がある多くの通信チャネルが存在することがある。したがって、出力データを転送するための不定期な要求と、利用可能なネットワーク帯域幅を求めて競合する他のデータを転送するための1つまたは複数のチャネル要求が存在することがある。

【 0 0 1 6 】

図2は、ターミナルサービスを可能にする実施200を示している。ターミナルサーバクライアントマシン202とターミナルサーバ204は、RDPを使用して通信する。ターミナルサーバクライアントマシン202は、ターミナルサーバクライアントプロセス206を実行し、ターミナルサーバクライアントプロセス206は、例えば、キーボードデータおよびマウスクリックデータなどのRDP入力デバイスデータ208を、ターミナルサーバ上で生成されたターミナルサーバセッション210に送信し、ユーザインタフェ

ースグラフィックスデータなどのRDP表示データ212を受信する。一般に、ターミナルサーバクライアントプロセス206は、シンクライアントプロセスであり、ほとんどの処理が、ターミナルサーバ204上で提供される。

【0017】

図3は、ファイアウォール302を通してターミナルサービスを可能にする実施300を示している。リモートターミナルサーバクライアント304は、ネットワーク308を介してターミナルサービスゲートウェイ(TSG)306に接続する。ターミナルサーバクライアント上のハイパーテキスト転送プロトコル(HTTP)転送プロセス310とTSG 306上のHTTPプロセス312は、ファイアウォール302を通過する通信を容易にする。HTTP転送プロセス310は、リモートプロシージャコール(RPC)データ、またはRDPデータなどのデータを、TSG 306のためのHTTPSヘッダ内にラップする。TSG 306は、ソケット出力プロセス316によってソケット接続318を介してターミナルサーバ 314に接続することができる。ターミナルサーバクライアント304が認証され、接続が確立されると、RDPデータ320は、ターミナルサーバクライアント304とターミナルサーバ 314の間で双方向に通過することができる。

10

【0018】

図4は、一般化された実施例400を示しており、既存のリモートプロシージャコール/ハイパーテキスト転送プロトコル(RPC/HTTP)プロキシを利用し、それによって、ファイアウォール402を通過するRPC/HTTP接続を介して、RDPなどのターミナルサービスプロトコルを提供する。実施のアーキテクチャは、RDPプロトコルをRPCコール内にラップすることによって、既存のRPCベースのプロキシを有利に利用できることを示している。特に、ターミナルサーバクライアント406上のRPC転送プラグイン404は、ターミナルサーバクライアント406とターミナルサーバ408の間の通信を提供するRDPストリームをRPCプロトコル内にラップする。これは、RPCベースのプロキシの利用を容易にし、それによって、ファイアウォールナビゲーションを可能にする。ターミナルサーバ上で、ユーザモードで動作できるRPCベースのプロキシ410は、受信したデータを、ターミナルサーバ上で、カーネルモードで動作できるソケットリスナ412に転送することができる。

20

【0019】

上述したように、クライアントは、ターミナルサービスを使用してリソースに接続するために、リモートデスクトッププロトコル(RDP)などのリモートプロトコルを使用することができる。リモートデスクトップクライアントが、ターミナルサービスゲートウェイを介してターミナルサーバに接続する場合、ゲートウェイは、ターミナルサーバとのソケット接続を開き、クライアントトラフィックをRDPポート、またはリモートアクセスサービス専用のポートにリダイレクトすることができる。ゲートウェイはまた、HTTPSを介して伝送されるターミナルサービスゲートウェイプロトコルを使用して、クライアントとのあるゲートウェイ固有のやり取りを実行することができる。

30

【0020】

ハイパーバイザなどの仮想マシンモニタは、仮想マシンを生成するプログラムであり、各仮想マシンは、下層の物理的なハードウェアリソースによって支援できる仮想ハードウェアリソースを有する。図1aは、複数の仮想マシン120、121を有する仮想マシン環境100を示しており、仮想マシン120、121は、複数の仮想プロセッサ110、112、114、116と、対応するゲストオペレーティングシステム130、132とを備える。仮想マシン120、121は、スケジューラ142および他のコンポーネント(図示されず)から成ることができる仮想化レイヤ140によって維持され、仮想化レイヤ140は、複数の仮想マシン120、121のためのハードウェア150を仮想化する。複数の仮想プロセッサ110、112、114、116は、下層のハードウェアの物理的なプロセッサ160、162に相当する仮想のものとするすることができる。

40

【0021】

50

上記区分を実施するためのバリエーションのすべては、単なる例示的な実施にすぎず、本明細書において本開示を任意の特定の仮想化態様に限定すると解釈すべきではない。

【0022】

オフローディングを使用するハードウェアアクセラレーション

本開示の実施形態は、一般に、処理ユニット21の処理オーバーヘッドおよびメモリ使用量を低減する能力を提供することに関する。これは、例えば、処理ユニット/ＣＰＵ 21上で動作しているオペレーティングシステム、および/またはアプリケーションプログラム、および/または他のプログラムモジュールを用いて達成される特定のコンピューティングタスクを、コンピュータシステム20に接続された適切な周辺ハードウェア機器にオフロードすることによって達成される。このような周辺機器の多くは、専用プロセッサおよび専用メモリをいっそう装備するようになっており、一般にはもっぱらＣＰＵ 21によって達成される同じタスクの多くを十分に実行することが可能である。このようなデバイス例は、例えば、ネットワークインタフェースカード（図1の53）、ディスクドライブインタフェースカード（例えば、図1の32、33、34）、小型コンピュータシステムインタフェース（ＳＣＳＩ）デバイス、インテリジェントシリアルインタフェースカード、またはデータの暗号化/復号化のためのデバイスなどのアプリケーション固有の周辺機器を含むことができる。

10

【0023】

本明細書で説明する本発明の一般的な概念は、上記の周辺ハードウェア機器のいずれかと接続して、コンピューティングタスクをオフロードするために使用できるが、開示された主題は、現在推奨の一実施形態例に関して説明され、コンピューティングタスクは、図1に示されるNIC 53などのネットワーク通信デバイスにオフロードされる。より具体的には、実例となる実施形態は、マイクロソフト（登録商標）コーポレーションから入手可能なWindows（登録商標）オペレーティングシステムのネットワーク環境およびアーキテクチャにおいて実施されるものとして説明される。しかし、Windows（登録商標）の概念およびWindows（登録商標）の専門用語を具体的に参照しはするが、ほとんどではないものの多くのオペレーティングシステムおよびネットワークアーキテクチャが、本開示の環境に関連する類似性を共有することを当業者が認識することが理解されよう。

20

【0024】

図5は、本開示のネットワークモデルおよび本開示のコンポーネントを構成するコンポーネントの一部の相互関係を示している。通常の動作中、ネットワーク伝送されるメッセージが、アプリケーション500によって、ネットワークスタック502を介して周辺機器504に送られ、周辺機器504では、メッセージは、ネットワーク上の他のデバイスおよびアプリケーションに送信され、また他のデバイスおよびアプリケーションから受信される。ネットワークスタック502は、1つまたは複数の中間ソフトウェアレイヤ506を含む。アプリケーション500から送られたデータは、中間ソフトウェアレイヤ506を通過し、中間ソフトウェアレイヤ506では、データのパッケージ化、信頼性の高いデータ伝送、データ暗号化、およびメッセージダイジェストの計算など、特定の操作をデータに対して実行することができる。

30

40

【0025】

スイッチ508は、中間ソフトウェアレイヤ506のためのネットワークスタック操作を実行することから、処理ユニット150をオフロードするために使用される。スイッチ508は、別個に示されているが、ネットワークスタック502の最上位の中間レイヤに統合できることに留意されたい。データが、周辺機器504のためのチャムニ510を通して周辺機器504に送られ、ネットワークスタック操作を実行する。この階層構造では、中間ソフトウェアレイヤは、ホストまたは周辺機器に排他的に存在する必要はなく、中間レイヤのいずれについても、完全にオフロードすること、ホストに残すこと、または両方を組み合わせる（例えば、1つもしくは複数の特定の接続をオフロードする）ことが可能である。加えて、チャムニは、チャムニ上にレイヤを形成することができる（例えば、IPs

50

e c チムニは、T C P チムニの上にレイヤを形成することができる)。接続は、信頼性の高いデータ転送および信頼性の低いデータ転送、ならびにユニキャストデータ転送またはマルチキャストデータ転送の任意の組み合わせとすることができる。中間レイヤをホストに残す場合、ホストは、周辺機器 5 0 4 内のキャッシュされた変数を(以下で説明されるように)更新する。例えば、接続のための転送制御ブロック(transport control block)(T C B)状態エントリは、周辺機器 5 0 4 にオフロードされるネットワーク層のためのルートキャッシュエントリ(route cache entry)(R C E)とともに、トランスポート層のためにオフロードすることができる。スイッチ 5 0 8 は、同じ R C E を共有するネットワークスタック 5 0 2 を介して異なる T C B のためのトラフィックを送り続ける一方、スイッチ 5 0 8 は、オフロードされた T C B のためのチムニ 5 1 0 を通してトラフィックを送る。

10

【 0 0 2 6 】

スイッチ 5 0 8 は、オフロード要求を中間レイヤ 5 0 6 に送信することによって、オフロードを開始する。オフロード要求は、正常に接続をオフロードできるかどうかを決定する周辺機器 5 0 4 を支援するリソース情報を含む。各中間レイヤ 5 0 6 は、オフロード要求を拒否するか、またはリソース情報をオフロード要求に追加して、ネットワークスタック 5 0 2 内の隣接したソフトウェアレイヤにオフロード要求を送る。周辺機器 5 0 4 は、オフロード要求を受け取ったとき、接続をオフロードするために利用可能なリソースを有するかどうかを計算する。周辺機器 5 0 4 は、オフロードが可能でない場合はオフロード要求を拒否する。オフロードが可能である場合、周辺機器 5 0 4 は、オフロード要求を受け入れ、接続のためのリソースを割り当てる。周辺機器 5 0 4 は、パラメータの連結リストを有する完了メッセージを中間ソフトウェアレイヤ 5 0 6 に送ることによって、オフロード要求を完了する。パラメータの連結リストは、中間ソフトウェアレイヤ 5 0 6 およびスイッチ 5 0 8 が周辺機器と情報伝達することを可能にするために、中間ソフトウェアレイヤ 5 0 6 およびスイッチ 5 0 8 に情報を提供する。各中間ソフトウェアレイヤ 5 0 6 は、自身のための情報をパラメータの連結リストから削除する。

20

【 0 0 2 7 】

中間レイヤ 5 0 6 は、オフロードについての完了メッセージを受け取ったとき、その状態を周辺機器 5 0 4 に渡す。各状態は、3つのタイプの変数、すなわち C O N S T、C A C H E D、および D E L E G A T E D を有することができる。状態は、3つのタイプの変数のすべてまたは一部を有することができる。C O N S T 変数は、オフロードされた接続の存続期間中に変化しない定数である。C O N S T 変数は、接続がアップロードされるとき、レイヤにリードバック(read back)されない。ホスト処理ユニット 2 1 は、C A C H E D 変数の所有権を維持し、ホスト処理ユニット 2 1 における C A C H E D 変数へのいかなる変更も、周辺機器 5 0 4 において更新されることを保証する。C A C H E D 状態を変更する制御メッセージは、ネットワークスタック 5 0 2 によって処理される。結果として、接続がアップロードされるとき、ホストは、C A C H E D 変数を書き込むが、リードバックする必要はない。ホスト処理ユニット 2 1 は、D E L E G A T E D 変数の所有権を周辺機器 5 0 4 に転送する。D E L E G A T E D 変数は、オフロードが発生したときに一度書き込まれ、オフロードが終了したときにリードバックされる。D E L E G A T E D 変数を転送し戻すことだけなので、接続をホストに転送し戻すことのオーバーヘッドは最小限に抑えられる。ネットワークスタック 5 0 2 と、様々な性能上の理由でオフロード(すなわち権限委譲)されている周辺機器 5 0 4 との間で共有されなければならない状態(例えば制御)は、ネットワークスタック 5 0 2 および周辺機器 5 0 4 の両方が、状態の排他的な部分を互いに所有するように、ネットワークスタック 5 0 2 とチムニ 5 1 0 の間できれいに分割される(例えば、T C P オフロードにおける I P I D)。ホスト処理ユニット 2 1 は、(例えば統計のために)必要とされる場合、D E L E G A T E D 変数について周辺機器 5 0 4 に問い合わせる。ホスト処理ユニット 2 1 は、診断のために C O N S T または C A C H E D 変数について問い合わせることもできる。状態を3つのカテゴリに分割することで、ネットワークスタック 5 0 2 は、チムニ 5 1 0 ときれいに共存することができる

30

40

50

。状態をオフロード要求に含めることができることに留意されたい。これは、状態が、委譲された状態変数を含まない場合、または最初のオフロード要求からオフロード要求が完了するまでの間に変化しない委譲された状態変数を含む場合に行うことができる。

【 0 0 2 8 】

周辺機器 5 0 4 またはホストは、オフロードされた接続をいつアップロードすべきかを決定する。アップロードは、周辺機器 5 0 4 またはスイッチ 5 0 8 のいずれかによって開始される。アップロードが開始されると、周辺機器 5 0 4 は、すべての未解決の要求を適切な状態で完了し、最上位の中間レイヤの委譲された状態をスイッチ 5 0 8 に渡す。スイッチ 5 0 8 は、さらなる送信要求をすべて待ち行列に入れ、受信バッファの提示を停止する。スイッチ 5 0 8 は、委譲された状態の制御権を取得するように最上位の中間レイヤに命令する。最上位の中間レイヤは、委譲された状態の制御権を取得し、スイッチ 5 0 8 に完了メッセージを送る。スイッチ 5 0 8 は、完了メッセージを受け取った後、周辺機器 5 0 4 にアップロードを確認し、それによって、周辺機器 5 0 4 は、もはや使用されていないリソースを解放することができる。

【 0 0 2 9 】

最上位の中間レイヤは、委譲された状態の制御権を取得するまで、オフロードされた接続についての到着データパケットを、処理のために周辺機器 5 0 4 に転送することに留意されたい。周辺機器 5 0 4 が委譲された状態をスイッチ 5 0 8 に渡すときと、最上位の中間レイヤが委譲された状態の制御権を取得するときの間に、データパケットが到着することがある。周辺機器 5 0 4 は、委譲された状態をスイッチ 5 0 8 に渡した後は、もはや到着データパケットを処理しない。周辺機器 5 0 4 は、到着データを受け取った場合、アップロードが進行中であることを通知するエラーメッセージを最上位の中間レイヤに送る。エラーメッセージは、最上位の中間レイヤが委譲された状態を受け取るまで、到着データの転送を停止し、さらなるデータをバッファリングするように最上位の中間レイヤに通知する。あるいは、周辺機器 5 0 4 上の増設バッファメモリのコストにおいて、到着データを周辺機器 5 0 4 に転送し、周辺機器 5 0 4 にデータをバッファリングさせることができる。

【 0 0 3 0 】

中間ソフトウェアレイヤ 5 0 6 によって、複数の接続を周辺機器 5 0 4 にオフロードすることができる。オフロードのために中間ソフトウェアレイヤの状態オブジェクトを参照する、上位レイヤ状態オブジェクト（すなわち、中間ソフトウェアレイヤ 5 0 6 の上位のレイヤの状態オブジェクト）の数についての参照カウンタが、中間ソフトウェアレイヤ 5 0 6 によって維持される。本明細書で使用される状態オブジェクトは、本明細書で使用されるような C O N S T、C A C H E D、または D E L E G A T E D として分類される、特定のレイヤのための状態変数の集まりである。中間レイヤのオフロードされた状態オブジェクトが、中間レイヤの上位のレイヤによる状態オブジェクトへの参照を有さない場合、中間レイヤ 5 0 6 は、中間レイヤのための状態オブジェクトをアップロードし、委譲された状態変数を中間レイヤ 5 0 6 に送るように、メッセージを周辺機器 5 0 4 に送る。周辺機器 5 0 4 は、中間レイヤ 5 0 6 のための状態オブジェクトを削除し、中間レイヤ 5 0 6 は、完了メッセージをスイッチ 5 0 8 に送る。

【 0 0 3 1 】

図 9 は、W i n d o w s（登録商標）環境におけるネットワークモデルを構成する一定のコンポーネント群からなる概略図を示している。説明の目的で、様々な W i n d o w s（登録商標）コンポーネントに対応する O S I レイヤも示されている。O S I モデルの物理層に対応する下部レイヤには、（ネットワークカードまたはネットワークアダプタと呼ばれることがある）実際の N I C 9 0 0 ~ 9 0 4 が存在する。N I C は、物理的な媒体（ネットワークケーブル）との物理的な相互接続と、特定のネットワークプロトコリに従った、すべてのより高位のレベルのレイヤによって生成されたデータを運ぶ信号の伝送とを提供するハードウェアデバイスである。上述したように、多くの N I C は、専用プロセッサおよびメモリを装備し、N I C が実行しなければホストプロセッサ C P U によって処理

10

20

30

40

50

できるタスクを含む、追加の高度なコンピューティングタスクを実行することが可能である。NICは、コンピュータのスロット内に配置されるプリント回路基板カードとして、PCMCIA対応のスロット内に配置されるPCMCIAタイプカードとして、マザーボード上のコンピュータ筐体内に配置される専用チップとして、または他の任意の適切なもので、物理的に実装することができる。

【0032】

各NICは、双方向ライン908～912によって概略的に表されるように、対応するネットワークドライバ916～920を介して、Windows（登録商標）ネットワークモデルと論理的に相互接続される。ネットワークドライバは、ネットワークモデルのMACサブレイヤ内に常駐し、対応するNICを介してWindows（登録商標）を物理ネットワークチャンネルにリンクする。対応するNICのベンダによって提供されるソフトウェアコンポーネントとして一般に実装される各ドライバは、対応するネットワーク接続を介してパケットを送信および受信すること、ならびにオペレーティングシステムの代わりにNICを管理することを担う。各ドライバはまた、対応するNIC上でI/Oを開始し、対応するNICから割り込みを受け取り、上位のプロトコルドライバを呼び出すことで送信データ転送の完了を通知する。また、デバイスドライバは、対応するNICの追加の処理機能のいずれかを起動し、制御し、および/または監視することも担う。

【0033】

ある環境では、ドライバコンポーネントは、TCP/IPまたはXNSなどの、単一の特定のネットワークプロトコルを実装するために書かれる。本開示の主題は、このような環境に適用可能である。しかし、説明目的で、本開示は、Windows（登録商標）ネットワークアーキテクチャに関連して説明され、Windows（登録商標）ネットワークアーキテクチャでは、network driver interface specification (NDIS) と呼ばれるインタフェースおよび環境が提供される。NDISインタフェースは、図9では926として機能的に示されている。NDISは、（その例が928～934として示される）様々なトランスポートプロトコルの一つ一つからネットワークドライバ916～920の各々を、またネットワークドライバの各々から様々なトランスポートプロトコルの一つ一つを隠蔽する。より詳細には、NDISは、1つまたは複数のNICドライバ（916～920）が、1つまたは複数の下位のNIC（900～904）、（図9で928～934として表される）1つまたは複数の上位のトランスポートプロトコルドライバまたはトランスポート、およびオペレーティングシステムと情報伝達を行うインタフェースを表す。

【0034】

本質的に、NDISは、NICドライバ開発のための完全に抽象化された環境を定義する。したがって、NICハードウェア割り込みの登録および捕捉から、トランスポートプロトコルドライバとの情報伝達、レジスタ操作およびポートI/Oを介した下位のNICとの情報伝達まで、NICドライバが実行する必要があるあらゆる外部機能について、NICドライバは、NDIS APIを頼りに機能を実行することができる。このレベルの抽象化およびその結果の移植性を提供するため、NDISは、NDISインタフェースライブラリラップ(NDIS Interface Library Wrapper)と呼ばれるエクスポートライブラリ（図示されず）を使用する。NICドライバとプロトコルドライバ、NICドライバとオペレーティングシステム、およびNICドライバとNICの間の対話は、ラップ機能へのコールを介して実行される。したがって、Windows（登録商標）NTのためのトランスポート固有のドライバを書く代わりに、ネットワークベンダは、単一のネットワークドライバの最上位レイヤとしてNDISインタフェースを提供する。そうすることで、いずれのプロトコルドライバも、このインタフェースをコールすることによって、そのネットワーク要求をネットワークカードに送ることができる。したがって、ユーザは、1つのネットワークカードおよび単一のネットワークドライバを使用して、TCP/IPネットワークおよびDLLC（もしくはNWLINK、またはDECnet、VINES、およびNetBEUIなど）ネットワークを介して通信することができる。

【 0 0 3 5 】

ネットワーク層およびデータリンク層には、図 9 の例では 9 2 8 ~ 9 3 4 として示される、トランスポート、プロトコル、および関連ドライバが存在する。Windows（登録商標）では、トランスポートプロトコルドライバは、ネットワークのユーザにサービスを提供するために、トランスポートドライバインタフェース（TDI）を、または場合によっては TDI の上位で別のアプリケーション固有のインタフェースを実装する、ソフトウェアコンポーネントである。Windows（登録商標）では、TDI は、機能ブロック 9 3 8 として示されるリダイレクタ機能およびサーバ機能など、セッション層において通信するネットワークコンポーネントに共通インタフェースを提供する。よく知られているように、トランスポートプロトコルは、ネットワークのためのデータオーガナイザとして動作し、基本的に、次の受信レイヤにデータをどのように提示すべきかを定義し、それに沿ってデータをパッケージ化する。トランスポートプロトコルは、対応する NIC を介してネットワーク上にデータを送信できるように、（Windows（登録商標）環境では NDIS パケットと呼ばれることがある）パケットを割り当て、送信元アプリケーションからのデータをパケットにコピーし、NDIS をコールすることによって、より下位のレベルのデバイスドライバにパケットを送る。

10

【 0 0 3 6 】

データパケットが様々なネットワークレイヤを、一般にネットワークモデルのレイヤ 3 およびレイヤ 4 を通過するとき、データパケットに対して追加の機能またはタスクも実行できることが理解されよう。本開示によれば、このような追加の機能またはタスクは、NIC ハードウェアによって、その代わりに実行することもできる。

20

【 0 0 3 7 】

例えば、従来はトランスポートプロトコルドライバによって実行していた 1 つのタスクは、チェックサム値を計算し、次にチェックサム値をパケットに追加することができる。これは、データがネットワークリンクを行き来するときに、データの整合性を保証する助けとなる。一般に、この操作は、パケットを構成するデータ要素を合計することによって計算された数をネットワークパケットに追加するために、ネットワークパケットの送信側に対応するトランスポートプロトコルを必要とする。次に、パケットの受信側は、追加されたチェックサム数をデータと比較し、それによって、データが送信中に変更されなかったことを確認する。このチェックサム計算および比較は、代わりに NIC ハードウェアにオフロードすることができる。

30

【 0 0 3 8 】

NIC ハードウェアにおいて最適に実行できる別の関連タスクは、データパケットのためのメッセージダイジェストの計算である。チェックサムと同様に、メッセージダイジェストは、パケット内のデータの整合性を保証するために使用される。加えて、メッセージダイジェストは、メッセージを送った当事者が、その主張通り当事者本人であることを保証することによって、データの信頼性を保証するために使用することができる。メッセージダイジェストの計算は、非常に CPU 集約型であり、ソフトウェアで実行するにはコストが高い機能である。

【 0 0 3 9 】

別の望ましい機能は、パケット内のデータの暗号化である。暗号化とは、パケットの読み込み権限のない者が、事前に暗号鍵の情報を得られないまま、メッセージの内容を実際に見ることが不可能になるように、パケット内のメッセージを変換する暗号プロセスのことを指す。もちろん、暗号アルゴリズムも、非常に CPU 集約型およびメモリ集約型となる傾向があり、ソフトウェアで実行した場合には非常にコストが高くなり得る。そのような暗号化の例は、セキュアソケットレイヤ（SSL）プロトコル暗号化、およびインターネットセキュリティプロトコル暗号化、すなわち「IPSec」を含む。よく知られているように、SSL 暗号化プロトコルおよび IPSec 暗号化プロトコルはどちらも、非常に CPU 集約型およびメモリ集約型である。

40

【 0 0 4 0 】

50

データパケットに対して実行できる別のタスクは、TCPセグメンテーションまたはUDPセグメンテーションである。よく知られているように、TCPプロトコルおよびUDPプロトコルは、大きなデータパケットを、下位のネットワークによって許容される最大データサイズに合ったセグメントに分割する。例えば、イーサネット（登録商標）は、最大で1514バイトのパケットをネットワーク上で許容する。したがって、TCPまたはUDPは、例えば64Kバイトを送信しなければならない場合、データを解析、1514バイトのセグメントに分割しなければならない。

【0041】

加えて、データパケットがネットワークから受信されるとき、データパケットに対してしばしばパケット分類が実行される。パケット分類は、サービス品質（QoS）についてのデータパケットの識別を含む。言い換えれば、各データパケットは、最適性能を保証するためにデータパケットに対して実行されるべきサービスの形態を定義するフィールドを含む。例えば、ビデオおよびオーディオを含むデータパケットは、ビデオおよびオーディオを提示する際の高忠実性を保証するために、データパケットに対して実行される一定の機能を必要とすることがある。サービスの形態を識別するためのデータパケットの分類は、本開示に従って、NICハードウェアにオフロードすることができる。

10

【0042】

パケットフィルタリングも、CPUの代わりに、NICハードウェアによって実行することができる。特に、データパケットは、それらがサービス妨害攻撃の一部である特徴を有するかどうかを特定するために評価することができる。しばしば、サーバソフトウェアは、サービスを要求するクライアントには悪意がないという仮定の下で作成される。しかし、クライアントが十分な教育を受けており、しかも悪意を有する場合、ユーザはしばしば、他のユーザに対するサーバのサービスを妨害するように設計された要求をサーバに向けて発行することができる。サービス妨害攻撃フィルタリングとは、データパケットがサービス妨害攻撃の特徴を有するかどうか特定するためにデータパケットを評価する能力に関係する。

20

【0043】

サービス妨害攻撃の一例は、「SYNフラッド攻撃」であり、SYNフラッド攻撃では、1つまたは別の理由で、クライアントは、ハンドシェイクシーケンスにおいて、サーバのSYNchronize-ACKnowledge（SYN-ACK）応答に対していかなる最終的な肯定応答も送信しない。これによって、サーバが最終的にタイムアウトになるまで、サーバはシグナリングを続けることになる。

30

【0044】

別のタイプのサービス妨害攻撃は、フラグメンテーションまたは「ティアドロップ（teardrop）」攻撃と呼ばれている。インターネットプロトコルは、パケットが次のルータで処理するには大き過ぎる場合、パケットを断片化する必要がある。後続の断片化されたパケットは、受信側においてパケット全体を再構築することを可能にする、最初のパケットの先頭のオフセットを識別する。フラグメンテーション攻撃では、ユーザは、しばしば受信側を機能不能にする偽のオフセット値を後続のフラグメント内に配置する。NICハードウェアは、データパケットを評価して、データパケットが特定の種類のサービス妨害攻撃の特徴を有するかどうかを特定することによって、これらのタイプのサービス妨害攻撃のためのフィルタリングを処理するように構成することができる。

40

【0045】

上記および他の機能は一般に、様々なネットワークレイヤに存在するソフトウェアコンポーネントにおいてコンピュータCPU 20によって実行され、その結果、相当のコンピュータリソースを利用することがあり、コンピュータシステム性能の全般的な低下をもたらす。したがって、これらまたは他の類似のタスクを、代わりに対応するNICハードウェアにおいて実行できるようにオフロードすることによって、コンピュータシステムの全体的な速度および効率を大幅に向上させることができる。

【0046】

50

ここで図 6 を参照すると、周辺機器 5 0 4 が N I C 5 6 であり、スイッチ 5 0 8 は、トランスポート層インタフェース (T L I) スイッチ 6 0 6 である一実施形態が示されており、ネットワークスタック 5 0 2 は、トランスポート層 6 0 0、ネットワーク層 6 0 2、およびフレーム化層 (framing layer) 6 0 4 を含む。ネットワーク層 6 0 2 は、経路層 (path layer) としても知られており、フレーム化層 6 0 4 は、近傍層 (neighbor layer) としても知られている。

【 0 0 4 7 】

ネットワークメッセージは、動作中に、アプリケーション 5 0 0 によって、ネットワークスタック 5 0 2 を介して N I C 5 6 に送られる。アプリケーション 5 0 0 から送られたデータは、 T L I スイッチ 6 0 6 を通過し、 T L I スイッチ 6 0 6 は、データが、ホストベースのネットワークスタック 5 0 2 を下るのか、それともチムニ 6 0 8 を下るのかを制御する。 T L I スイッチ 6 0 6 をネットワークスタック 5 0 2 の最上位レイヤに組み込むことができることに留意されたい。ネットワークスタック 5 0 2 内のソフトウェアレイヤは、アプリケーション 5 0 0 からデータを受け取り、そのデータをパケット形式にパッケージ化し、 N D I S ミニドライバ 6 1 0 を介して、そのデータを周辺ハードウェア 6 1 4 に送る。データパケットがスタック 5 0 2 を通過するときに、ネットワークスタック 5 0 2 が実行できる他のタスクは、データ暗号化、信頼性の高いデータ伝送、およびメッセージダイジェスト (例えば、データパケットのためのチェックサムまたは巡回冗長検査 (C R C)) の計算を含む。これらのタスクの多くは、処理ユニット 2 1 によって実行され、プロセッサ集約型である。

【 0 0 4 8 】

接続のためのデータを、チムニ 6 0 8 (およびチムニドライバ 6 1 2) を介して N I C 5 6 に送ることによって、スタック操作を実行することから処理ユニット 2 1 をオフロードするために、 T L I スイッチ 6 0 6 が使用される。 N D I S ミニドライバ 6 1 0 とチムニドライバ 6 1 2 の上縁が、マイクロソフト (登録商標) の製品版オペレーティングシステムの N D I S A P I であることを当業者は理解されよう。説明目的で、伝送制御プロトコル (T C P) ベースのプロトコルスタックを使用して、開示される主題について説明する。しかし、本開示の教示を使用して、多くの種類の周辺機器を使用できること、また他のネットワークスタックをオフロードできることを当業者が認識することが理解されよう。例えば、ストリーム制御伝送プロトコル (S C T P) またはユーザデータグラムプロトコル (U D P) ベースのプロトコルスタックをオフロードすることができる。加えて、本開示は、インターネット小型コンピュータシステムインタフェース (i S C S I)、ネットワークファイルシステム (N F S)、または共通インタフェースファイルシステム (C I F S) など、より高位の機能プロトコルをオフロードするために使用することもできる。

【 0 0 4 9 】

オフロードが発生する多くの理由が存在する。限定ではなく、例として、理由のいくつかを以下で提供する。システム管理者は、オフロードされる特定のサービスを選択することができる。特定の接続上のトラフィック (バイト数またはパケット数の点で) が相当な量のリソースを消費している場合、その接続をオフロードすることができる。サービスの種類をオフロードすることができる。例えば、 I P s e c などのセキュリティプロトコルをオフロードすることができる。オフロードは、ポリシーによってさせることができる。例えば、管理者は、ある組織内からのすべての接続が最初にオフロードされるというポリシーをもつことができる。使用されているシステムリソース (例えば、 C P U 利用、データキャッシュ使用、ページテーブルキャッシュ使用、メモリ帯域幅) は、接続をオフロードするようにホストプロセッサを誘導することができる。

【 0 0 5 0 】

図 7 は、 T C P 接続をオフロードするために取られる手順を示している。 3 ステージのプロセスが使用される。一般に、 3 ステージのプロセスは、 T C P 接続をオフロードするのに必要とされるリソースを割り当てること、層 5 0 0、5 0 2、5 0 4、5 0 6 の各々

にハンドルを提供すること、層 500、502、504、506 の各々についての状態を NIC 53 にオフロードすることである。オフロード遷移中、TLISwitch 506 は、アプリケーション 200 から送られたすべてのメッセージをバッファリングする。あるいは、トランスポート層 500 が、データをバッファリングする。オフロードが完了したとき、バッファリングされたデータが、オフロードデータ伝送と同じ機構を使用して、NIC 53 に転送される。オフロード遷移中に到着パケットを受信した場合、NIC 53 は、トランスポート層委譲状態が NIC 53 に渡されるまで、層 500、502、504、506 を介してデータを上方に移動させ続ける。

【0051】

TLISwitch 506 は、トランスポート層 500 にオフロード要求を送ることによって(ライン 700)、オフロードを開始する。オフロード要求は、次層の局所的状態を示すポインタ(例えば、トランスポート層 500 の場合の TC B ポインタ、ネットワーク層 502 の場合の RC E ポインタ、フレーム化層 504 の場合の ARP テーブルポインタ、または NDIS ミニドライバ 510 の場合の NDIS ミニポートポインタ)、オフロードタイプ(例えば、TLISwitch 506 の場合の TCP、ネットワーク層 502 の場合の IPv 6 など)、および正常に TCP 接続をオフロードできるかどうかを NIC 53 が決定する助けとなるリソース情報を含む。また、TLISwitch 506 は、ディスパッチテーブルも NIC 53 に提供することができる。トランスポート層 500 は、オフロード要求を拒否するか、または TLISwitch リソース情報に TCP リソース情報を追加して、オフロード要求をネットワーク層 502 に送る(ライン 702)。

【0052】

ネットワーク層 502 は、オフロード要求を受け取り、接続をオフロードすることを拒否するか、または TCP リソース情報および TLISwitch リソース情報にネットワークリソース要件を追加して、オフロード要求をフレーム化層 504 に送る(ライン 704)。ネットワーク層 502 は、ディスパッチテーブルも NIC 53 に提供することができる。フレーム化層 504 は、接続をオフロードすることを拒否するか、またはネットワークリソース要件、TCP リソース情報、および TLISwitch リソース情報にフレーム化リソース要件を追加して、オフロード要求を NIC 53 に送る(ライン 506)。

【0053】

NIC 53 は、オフロード要求を受け取り、TCP 接続をオフロードするのに利用可能なリソースを有するかどうかを計算する。NIC は、オフロードが可能でないと判定した場合、オフロード要求を拒否する。NIC は、オフロードが可能であると判定した場合、オフロード要求を受け入れ、接続のためのリソース(例えば、TC B、ルートキャッシュエントリ(RC E)、アドレス解決プロトコル(ARP)テーブルエントリ(ATE))を割り当てる。NIC 53 は、パラメータの連結リストおよびディスパッチテーブルを生成し、層 500、502、504、506 に渡し、パラメータの連結リストを有する完了メッセージをフレーム化層 504 に送ることによって(ライン 408)、オフロード要求を完了する。パラメータは、層 500、502、504、506 の各々のためのオフロードハンドルおよびディスパッチテーブルを含む。本明細書で使用されるオフロードハンドルとは、ソフトウェアレイヤが周辺機器と情報伝達することを可能にする機構を意味する。限定ではなく、例として、オフロードハンドルは、ポインタベースのハンドル、配列への参照として使用される整数値、ハッシュテーブル(例えばハッシュ関数)、ソフトウェアレイヤ(もしくはネットワークスタック)と周辺機器との間の通信チャネル、または状態オブジェクトを参照するために周辺機器が使用する、ソフトウェアレイヤによって渡されるパラメーター式になり得る。

【0054】

ディスパッチテーブルは、データを直接 NIC 53 に送るため、またはデータを直接 NIC 53 から受け取るために使用される。ディスパッチテーブルは、診断を提供するために使用することもできる。例えば、ソフトウェアレイヤを追加して、システムを監視すること、またシステムが適切に機能していることを確認するために誤りを投入すること

10

20

30

40

50

ができる。加えて、必要に応じて、追加の機能を追加できるソフトウェアレイヤによって、ディスパッチテーブルにパッチを適用することができる。例えば、ソフトウェアレイヤを追加して、フィルタドライバの機能を提供することができる。パッチは一般に、追加機能が挿入される元の機能を指示するポインタを獲得し、追加機能にポインタをリダイレクトする（すなわちポインタを向ける）ことによって適用される。パッチが挿入された後、元の機能が呼び出されたときは常に、追加機能がその機能を実行し、その後、元の機能を呼び出す。

【 0 0 5 5 】

フレーム化層 5 0 4 は、宛先 M A C アドレスが変化した場合またはカプセル化タイプが変化した場合の容易な更新のために、A R P テーブルエントリ内にフレーム化層のためのオフロードハンドルおよびディスパッチテーブルを格納する。その後、フレーム化層 5 0 4 は、A T E に関連する N I C 5 3 状態を更新する（ライン 7 1 0）。フレーム化層 5 0 4 は、その状態を連結リストから除去し、連結リスト内の残りの情報をネットワーク層 5 0 2 に転送する（ライン 7 1 2）。

10

【 0 0 5 6 】

ネットワーク層 5 0 2 は、ネットワーク層 5 0 2 のためのオフロードハンドルおよびディスパッチテーブルを格納する。ネットワーク層 5 0 2 はまた、その状態を N I C 5 3 に送る（ライン 7 1 4）。ネットワーク層 5 0 2 は、連結リストからネットワーク層情報を除去し、パラメータの連結リストおよびディスパッチテーブルを有する完了メッセージをトランスポート層 5 0 0 に送る（ライン 7 1 6）。ネットワーク層 5 0 2 は、オフロードされた状態のために受け取る I P フラグメントを、処理のために N I C 5 3 に転送することができ、また、ネットワーク層 5 0 2 が I P フラグメントをネットワーク層において処理し、トランスポート層 5 0 0 に転送することができる。

20

【 0 0 5 7 】

一代替実施形態では、層の状態オブジェクトが、オフロード要求と共に送られる。例えば、フレーム化層状態オブジェクトおよびネットワーク層状態オブジェクトは、オフロード要求と共に送られ、オフロード要求と完了イベントとの間に、キャッシュされた状態が変化した場合にのみ、状態が更新される。層状態オブジェクト全体は、委譲された状態が存在しないか、またはオフロード要求とオフロード要求の完了との間に変化できない場合にのみ、オフロード要求と共に送ることができる。しかし、委譲された状態が存在し、オフロード要求とオフロード要求の完了との間に変化できる場合であっても、C O N S Tとして分類された変数は、オフロード要求と共に送ることができる。

30

【 0 0 5 8 】

トランスポート層 5 0 0 は、トランスポート層のためのオフロードハンドルを格納し、その状態を N I C 5 3 に送る（ライン 7 1 8）。任意の未解決送信バッファまたは任意の未解決受信バッファが保留状態で存在する場合、トランスポート層 5 0 0 は、バッファを T L I スイッチ 5 0 6 に返す。トランスポート層 5 0 0 がバッファを T L I スイッチ 5 0 6 に返し始めると、T L I スイッチ 5 0 6 は、トランスポート層 5 0 0 にバッファを送るのを停止し、それらを待ち行列に入れ、トランスポート層 5 0 0 がパラメータの連結リストおよびディスパッチテーブルを有する完了メッセージを T L I スイッチ 2 0 4 に送るのを待つ。トランスポート層 5 0 0 は、すべてのバッファを返し、その後、完了メッセージを送る（ライン 7 2 0）。T L I スイッチ 5 0 6 は、完了メッセージを受け取ると、送信バッファおよび受信バッファを N I C 5 3 に転送する（ライン 7 2 2）。T L I スイッチ 5 0 6 は、ディスパッチテーブルを使用して、すべての未解決受信バッファをおよびすべての将来の受信バッファをポストし、処理のために N I C 5 3 に送る。オフロード要求が完了するのにかかる時間の間、各層 5 0 0、5 0 2、5 0 4 は、オフロードされる状態オブジェクト（すなわち、層に関連する状態オブジェクト）についての新しいオフロード要求を拒否するか、またはオフロードが完了するまで、それらを待ち行列に入れる。

40

【 0 0 5 9 】

トランスポート層 5 0 0 は、トランスポート状態が N I C 5 3 にオフロードされてい

50

ない場合、到着TCBデータを処理して、TLISwitch506にデータを渡す能力を依然として有する。TCBデータがオフロードの最中に到着した場合、トランスポート層500は、データを保持し続けるか、またはデータを処理して、それをTLISwitch506に渡すことができる。トランスポート層500がその状態をNIC53に送ったとき(ライン718)とTLISwitchがバッファをNIC53に転送したとき(ライン722)の間は、ネットワークスタック202を通して来る到着TCBデータが、NIC53に送られる。

【0060】

後のオフロード要求のとき、ネットワーク層502およびフレーム化層504は、NIC53から受け取ったオフロードハンドルを、先のオフロードからNIC53に渡す。これは、ネットワーク層502およびフレーム化層504のためのリソースがすでに割り当てられていることをNIC53に伝え、それによって、NICリソースを節約し、オフロードをスピードアップする。

【0061】

先に示したように、層500、502、504は、それらの状態をNIC53に渡す。各状態は、3つのタイプの変数を、すなわち、CONST、CACHED、およびDELEGATEDを有する。CONST変数は、オフロードされた接続の存続期間中に変化しない定数である。CONST変数は、接続が終了したとき、その層にリードバックされない。ホスト処理ユニット21は、CACHED変数の所有権を維持し、ホスト処理ユニット21におけるCACHED変数へのいかなる変更も、NIC53において更新されることを保証する。結果として、ホストは、CACHED変数を書き込むが、(システム診断がそれを要求しない限り)リードバックはしない。ホスト処理ユニット21は、DELEGATED変数の所有権をNIC53に転送する。DELEGATED変数は、オフロードが発生したときに一度書き込まれ、オフロードが終了したときにリードバックされる。DELEGATED変数を転送し戻すことだけなので、接続をホストに転送し戻すことのオーバーヘッドは最低限に抑えられる。ホスト処理ユニット21は、(例えば統計のために)必要とされる場合、DELEGATED変数についてNIC53に問い合わせる。

【0062】

トランスポート層500のためのCONST変数は、宛先ポート、送信元ポート、「気付(care-of)」アドレスが変化する可能性がある場合にモバイルIPが存在することを示すフラグ、SENDウィンドウスケールファクタおよびRECVウィンドウスケールファクタ、ならびにネットワーク層502のためのNICハンドルを含む。トランスポート層500のためのCACHED変数は、TCP変数およびIP変数である。TCP変数は、有効MSS(Effective MSS)、NIC53によって受信指示にコピーされるバイトの数、ネーグリング(Nagling)をオフにするフラグ、キープアライブ(Keep-Alive)が必要であることを示すフラグ、およびキープアライブ設定(すなわち、間隔、プローブの数、デルタ)を含む。IP変数は、TOSおよびTTLを含む。DELEGATED変数は、現在のTCP状態、次のRECV(すなわちRCV.NEXT)のシーケンス番号、受信ウィンドウサイズ(RCV.WND)、最初の送達確認なしのデータ(First Un-Acked Data)のシーケンス番号(SND.UNA)、次のSENDのシーケンス番号(SND.NEXT)、これまで送信された最大のシーケンス番号(SND.MAX)、最大の送信ウィンドウ(MAX__WIN)、現在の輻輳ウィンドウ(CWIN)、スロースタート閾値(SSTHRESH)、平滑化RTT(8*A)、デルタ(8*D)、現在の再送カウント、次の再送についての残存時間、およびエコーされるタイムスタンプを含む。

【0063】

ネットワーク層502のためのCONST変数は、(IPv4またはIPv6のための)宛先IPアドレスおよび(IPv4またはIPv6のための)送信元宛先IPアドレスを含む。ネットワーク層502のためのCACHED変数は、フレーム化層504のためのNICハンドルを含む。ネットワーク層502のためのDELEGATED変数は、I

10

20

30

40

50

P パケット ID 開始値を含む。フレーム化層 504 のための CACHED 変数は、ARP アドレス、およびヘッダのフォーマット（例えば、LLC / SNAP [Logical Link Control / Sub - Network Access Protocol] または DIX [Digital, Intel, Xerox] ）を示すフラグを含む。

【0064】

ネットワーク層状態は、複数の接続の間で共有でき、フレーム化層状態は、複数の経路（例えば IP エイリアス）の間で共有できるので、トランスポート層状態は、ネットワーク層のためのハンドルを含み、ネットワーク層状態は、フレーム化状態へのハンドルを含む。この階層構造は、いくつかの理由のために維持される。IP ID 名前空間は、すべてのオフロードされた接続にわたって、経路毎に管理されなければならないので、接続は、ネットワーク層のための NIC ハンドルを必要とする。ルート更新は次のホップアドレスを変更することができ、その結果、新しい MAC アドレスをポイントすることができるので、経路は、フレーム化層のための NIC ハンドルを必要とする。階層はまた、NIC によって維持される必要がある状態の量を縮約する。例えば、IPv4 のための ARP 更新は、IP アドレスから MAC アドレスへのマッピングを変更することができる（例えば、サーバ上でフェイルオーバーされるインタフェース）。ホストは、キャッシュされた変数として MAC アドレスを維持し、その結果、ホストは、キャッシュされた状態を 1 回だけ更新すればよく、すべての接続は、新しいインタフェースに引き継がれる。

【0065】

TCP 接続がオフロードされると、NIC 53 は、NIC 53 が送信するパケットのパケット識別子（例えば IP ID）の割り当てを担う。IP ID は、インタフェース毎または層状態オブジェクト毎にオフロードされる。NIC 53 には、IP ID 名前空間の一部が割り当てられる。一実施形態では、NIC 53 には、IP ID 名前空間全体の半分が割り当てられ、ネットワーク状態が NIC 53 に渡されるときに使用する、IP パケット ID 開始値が与えられる。NIC 53 は、以下の公式を使用して、NIC 53 が送信する IP パケット上で IP ID を生成する。
$$Cur.sub.-IPID = (Start.sub.-IPID_For_This_Path) + (Counter_For_This_Path) mod 64K$$
$$Counter_For_This_Path = Counter_For_This_Path + 1$$

【0066】

オフロードされた接続がアップロードされたか、または無効化された場合、NIC 53 は、NIC 53 が使用する次の IP ID 値をネットワーク層に転送し、次に発生するオフロードのために格納し、ホスト処理ユニット 21 は、割り当てられた IP ID 名前空間の一部を使用し続ける。ホスト処理ユニット 21 は、最大限の IP ID 名前空間を使用することができるが、オフロードが発生するたびに、カウンタが設定されなければならない。

【0067】

NIC 53 は、データが受信された順番で、データを受信バッファ内に配置し、オフロードされた接続についてポストされた順番で、アプリケーションバッファを満たす。多くのアプリケーションは、受信バッファをポストする前に、受信指示を待つ。一実施形態では、NIC 53 は、グローバルなバッファプールを有し、接続のためにデータが到着したが、アプリケーション受信バッファがポストされていない場合に使用する。グローバルなバッファプールは、オフロードされた接続にわたって使用され、1) 順序が狂った TCP 伝送に対する対処、2) IP データグラムのデフラグメント、3) アプリケーションがゼロコピーアルゴリズムには小さ過ぎるバッファをポストしている場合に、ゼロコピーアルゴリズムで以外のバッファコピーアルゴリズムを実行するために使用することができる。あるいは、リソースの効率的な使用が問題とならない場合、接続毎のバッファプールを使用することもできる。しかし、NIC が接続毎のバッファプールをサポートしない場合、またはシステムリソースが不足している（例えば、メモリ内にアプリケーションバッ

10

20

30

40

50

ファを確保するにはリソースが十分でない)場合は、グローバルなバッファプールが使用される。

【0068】

ここで図8を参照すると、接続がNIC 53にオフロードされると、NIC 53への2つの経路が存在することとなる。第1の経路は、NDISミニドライバ510を通して、フレーム化層504、ネットワーク層502、およびトランスポート層500に到る。第2の経路は、オフロードされた接続808を通り、これはチムニと呼ばれる。ホストコンピュータの観点からすれば、2つの経路について、通信の点ではすべてが同じである。キャッシュされた状態変数は、先に示したように、NIC 53のキャッシュされた状態変数を更新する処理ユニット21を用いて、2つの経路を同期させる。キャッシュされた変数の更新は、矢印802、804、806によって示されている。

10

【0069】

到着データパケットが到着したとき、NIC 53は、到着データパケットがオフロードされた経路を通過するか、それともオフロードされていない経路を通過する(すなわち、NDISミニドライバ510およびレイヤ504、502、500のNDIS経路を通る)かを判定する。一実施形態では、NIC 53は、送信元TCPポート番号および宛先TCPポート番号、送信元IPアドレスおよび宛先IPアドレス、ならびにプロトコルタイプに対してハッシュ関数を実行することによって、どちらの経路に到着データパケットを送るべきかが判定する。ハッシュがオフロードされた接続パラメータと一致する場合(すなわち、ハッシュパケットチェーンが探索され、接続のすべてのタプルについて正確な一致が生じる場合)、チムニ808が使用される。ハッシュがハッシュインデックスと一致しない場合、ネットワークスタック502を通るオフロードされていない経路が使用される。キャッシュされた状態を更新する制御メッセージが、ホストによって処理される。これによって、NIC 53は、ICMP、DNS、およびRIPメッセージなど、オフロードされた接続を除いて、どのような制御メッセージも処理する必要がなくなる。

20

【0070】

先に述べたように、Windows(登録商標)または類似の階層化ネットワークモデルにおけるデータ伝送の基本単位は、データパケットである。Windows(登録商標)環境では、データパケットは、NDISパケットと呼ばれる。各パケットは、スタックの最上位(すなわちISOスタックのレイヤ5)から最低位のソフトウェアレイヤ(すなわちISOスタックのレイヤ2)まで移動する。したがって、パケットは、データの送信中およびデータの受信中にレイヤを通過するとき、各レベルにわたって共通のデータ構造を定義する。例として、図10は、パケットがそれぞれのレイヤを通過して、イーサネット(登録商標)NICとして900で示されるNICまで下るときに、パケットによってたどられる経路を示している。上で述べたように、トランスポートドライバ928は、送信元アプリケーションからデータを受け取り、それを下位のプロトコルに準拠するパケット形式でパッケージ化し、その後、パッケージ化したパケットを、NDISインタフェース926を介してより下位のレベルのデバイスドライバ916に転送する。加えて、トランスポートプロトコルは、パケットに対して他の機能(例えばチェックサム計算など)を実行することができる。あるいは、図10に示されるIPセキュリティ機能1044(例えば、暗号化および/またはメッセージダイジェスト計算)などの追加機能をパケットに対して実行する他の機能コンポーネントが、ネットワーク層またはデータリンク層に存在することができる。

30

40

【0071】

一実施形態では、データパケット1042は、コンピューティングタスクをNICハードウェア900などの周辺機器にオフロードする手段となる。例えば、図10では、アプリケーションデータ1040は、ネットワークモデルの上位レイヤから、TCP/IP 928などの適切なトランスポートプロトコルドライバに渡される。ドライバは、データを適切なデータパケット1042に再パッケージ化する。その後、この特定のデータパケット1042に対してどのような追加機能が実行されようとも、それに応じて、パケット

50

拡張と呼ばれる事前定義されたデータ構造をデータパケットに追加する機能コンポーネントが含まれる。以下でさらに詳細に説明されるように、パケット拡張の内容は、データパケットがNIC 53に到達したときに、1つのタスクまたは複数のタスクのどれを、データパケットに対して実行するか指示する。データパケット1042がネットワークドライバ916に到達したとき、NIC 53によってどのタスクを実行するかを確認するために、このパケット拡張の内容が、ドライバ916によって問い合わせられる。その後、ドライバ916は、パケット拡張の内容を介して要求された機能タスクが何であろうとも実行できるように、NIC上でハードウェアを制御/操作する。

【0072】

例えば、図10では、データパケット1042は、ソフトウェアコンポーネント1044に渡され、ソフトウェアコンポーネント1044は、別途実装することができ、またはパケット拡張をパケット1042に付加するトランスポートプロトコルドライバ自身の一部として実装することができる。オフロードされる特定のタスクに応じて、データがパケット拡張内に含まれる。例えば、IPセキュリティ機能が実装される場合、指定された暗号鍵に従ってNIC 53がデータパケットを暗号化すべきであることを示すデータが含まれるだろう。もちろん、ソフトウェアコンポーネント1044は、上で説明したような多くの機能のいずれか1つが、ネットワーク層に存在するソフトウェアコンポーネントによる代わりに、ハードウェアレベルで実行されるように、事前定義されたデータを付加することができる。デバイスドライバ916は、パケット拡張から情報を取り出し、その後、NIC 53において指定されたタスクを起動する。

【0073】

図11は、データパケット1042の一般的な構造の一実施形態を示している。パケット1042は、使用されるネットワーク環境そのものに応じて、任意のフォーマットをとることができるが、Windows（登録商標）環境では、パケットは、NDISに従ってフォーマットされ、協同するデバイスドライバおよび協同するプロトコルドライバによって目的が定義されるフラグ、パケットに関連するOut-of-band data（OOB）の格納のための領域、パケットの長さに関する情報、およびパケットのデータ内容に関するメモリ記憶場所を指示するポインタなどの情報を含む。

【0074】

さらに、図11は、タスクオフロードを識別するためにNDISデータパケットに付加される追加のデータ構造フィールドである、パケット拡張1150を示している。説明されるように、宛先NICにオフロードされる特定の1つのタスクの識別、または特定の複数のタスクの識別のために必要な情報を含むデータ構造を定義するのは、このパケット拡張1150である。好ましい実施形態では、タスクオフロードタイプ（例えば、チェックサム、暗号化/復号化など）毎に、事前定義されたデータフィールドが、パケット拡張1150内に含まれる。このデータフィールドは、（チェックサムなどの）特定の機能が実行されることを指示するだけの、1つの制御フラグの型または複数の制御フラグの型を容易に取ることができ、または、情報は、タスクをどのように実行すべきかをさらに定義するデータ構造を指示するポインタの型を取ることができる。

【0075】

NIC 53は、添付したパケットのみに適用するように、パケット拡張1150内のタスクオフロード制御フィールドを認識するように構成することができる。したがって、例えば、NIC 53がチェックサム操作を実行することを示すフラグを特定のパケットが含む場合、NIC 53は、添付したパケットのみにチェックサム操作を実行する。しかし、与えられたパケットにおいてそのようなフラグが存在しない場合、NIC 53は、そのパケットに対してチェックサム操作を実行しない。あるいは、NIC 53が別途命令されるまで、NICはネットワーク上に送信されたこのパケットおよびすべての後続のパケットに対して、オフロードされたタスクを実行することを、パケット拡張1150内のタスクオフロード制御フィールドは指示できる。

【0076】

パケット拡張 1 1 5 0 は、どのタスクを実行するかを、受信機構の N I C 機構に命令することができる。例えば、パケット拡張は、復号化、チェックサム操作、パケット分類、サービス妨害攻撃に対する防御としてのパケットフィルタリング、パケットの再構築、およびパケット拡張を実行するように N I C を構成できる他の任意の受信機能などの、ネットワークからパケットを受信したときに適切な一定の機能を、受信機構 N I C が実行することを、N I C に命令することができる。もちろん、送信機構は、受信機構 N I C のタスクオフロード機能を知らない可能性がある。受信機構 N I C が要求された機能を実行できない場合、受信機構 C P U が、代わりにその機能を果たすだろう。したがって、送信機構は、受信機構において N I C が特定のパケットに対して何を行うかについて、一部の制御を有することができる。

10

【 0 0 7 7 】

加えて、送信機構は、パケットを受信するために N I C 5 3 が機能するとき、N I C 5 3 を制御するために、パケット拡張を使用することもできる。例えば、ネットワーク上に送信される特定のパケットは、N I C 5 3 がパケットを受信したときに、N I C 5 3 がパケットの復号化を実行するという、N I C 5 3 への命令を含む、パケット拡張を含むことができる。この場合、前記命令はネットワークから受信したパケットに対して機能を実行するためのものであるので、前記命令はネットワーク上に送信されるパケットには適用されない。したがって、この場合、ネットワーク上に送信されるパケットのためのパケット拡張は、パケットを受信するときに、どの受信機能が N I C 5 3 にオフロードされるかを制御するための機構としての機能を果たす。送信機構は、パケットを送信するときは常に、パケット毎に N I C 5 3 の受信機能を制御する機会を有する。

20

【 0 0 7 8 】

図 1 1 に示された例をさらに参照すると、パケット拡張 1 1 5 0 は、N I C (例えば、N I C 5 3) にオフロードされた送信機能を制御する多くのデータフィールド 1 1 5 1 から 1 1 5 3、および N I C (例えば、N I C 5 3 または受信機構の N I C) にオフロードされた受信機能を制御する多くのデータフィールド 1 1 5 4 から 1 1 5 8 を含む。

【 0 0 7 9 】

例えば、パケット拡張 1 1 5 0 は、N I C 5 3 がチェックサム操作を実行することを示すデータフィールド 1 1 5 1 を含む。これは、送信 N I C 5 3 自身が、付加されたパケットに対してチェックサム操作を実行することを、送信 N I C 5 3 に指示する。

30

【 0 0 8 0 】

パケット拡張 1 1 5 0 はまた、パケットデータの S S L 暗号化もしくはパケットデータの I P S e c 暗号化、および / またはメッセージダイジェストの計算に関連して実行されるようなセキュリティ機能を、N I C 5 3 が実行すべきであることを指示するために、セキュリティ機能データフィールド 1 1 5 2 も含むことができる。このタイプのセキュリティタスクのため、フィールド 1 1 5 2 は、データ構造 (例えばデータ構造 1 1 6 0) を含むメモリ記憶場所を指示するポインタを好ましくは含み、同様に、そのデータ構造が、暗号化および / またはメッセージダイジェスト機能の実行に関する情報を含む。ある状況下では、関連するデータを有するメモリ記憶場所を指示するポインタを含むことは、パケット拡張自身の中に実際のデータを格納することに勝る利点を有する。

40

【 0 0 8 1 】

図 1 1 を参照すると、チェックサムデータフィールド 1 1 5 4 は、チェックサム計算が送信機構で実行されたすべての受信パケットについて、データが N I C 5 3 へ行く途中で変更されていないことを確認するために、N I C 5 3 がチェックサム計算を実行することを指示する。後のチェックサムフィールドによってはっきりと違反されない限り、このチェックサムフィールド 1 1 5 4 は、無期限に、または与えられた期間の間、すべてのパケットのためのチェックサム受信機能を制御することができる。

【 0 0 8 2 】

パケット拡張は、パケットデータの S S L 復号化もしくはパケットデータの I P S c c 復号化、および / またはメッセージダイジェストの計算に関連して実行されるようなセキ

50

セキュリティ機能を、パケットが受信されたときに、NIC 53が実行することを指示する、セキュリティフラグ1155も含むことができる。このタイプのセキュリティタスクのため、フィールド1155は、データ構造（例えばデータ構造1160）を含むメモリ記憶場所を指示するポインタを好ましくは含み、同様に、そのデータ構造が、暗号化および/またはメッセージダイジェスト機能の実行に関する情報を含む。

【0083】

パケット拡張はまた、受信したパケットを大量データ(bulk data)に構築するようNIC 53に命令するアセンブリデータフィールド1156、上述したように、サービス品質のために各パケットを分類するようNIC 53に命令する分類フィールド1157、および上述したように、DOS攻撃の特徴について到着パケットをフィルタリングするためのDenial of Service(DOS)攻撃フィルタフィールド1158も含むことができる。パケット拡張1150はまた、受信機構NICが、攻撃されたパケットに対して実行することが意図された、フラグ1154~1158に類似した受信機能データフィールドを含むことができる。

【0084】

一実施形態では、パケット拡張1150内に含まれる情報は、パケット1142が送信された特定のデバイスドライバによって問い合わせされる。例示的な実施形態において説明されるWindows(登録商標)環境では、このタイプの機能は、適切なNDISファンクション呼び出しを行うことによって、好んで実行される。例えば、パケットのためのパケット拡張1150のメモリ記憶場所を指示するポインタを返す、事前定義されたNDIS機能への呼び出しを実行することができる。その後、デバイスドライバソフトウェアは、どのタスクを実行すべきかを識別することができ、オフロードされたタスクに応じて、適切な方法で、ドライバの対応するNICハードウェアを実行/操作することができる。

【0085】

コンピュータプロセッサからハードウェア周辺機器にコンピューティングタスクをオフロードするために実際のデータパケットを利用することは、多くの理由で有利である。例えば、トランスポートドライバは、パケット毎に周辺機器の機能を利用することができる。これによって、タスクを動的にダウンロードすることが可能になり、周辺機器の機能を必要に応じて使用することができる。したがって、コンピュータシステムについての処理オーバーヘッドが特定の時点において低い場合、あるタスクを、従来の方式によりコンピュータプロセッサ上で実行するほうが望ましいことがある。あるいは、他のコンピューティングタスクによってCPUに重い負荷がかかっている場合、CPUは、必要なパケット拡張をデータパケットに付加するだけで、周辺機器にタスクをオフロードすることができる。

【0086】

別の利点は、単一のパケットによって複数のタスクをオフロードし、実質的に多くの操作を一度に「バッチ処理」する能力である。例えば、コンピュータプロセッサが、チェックサム操作または暗号化操作を実行する場合、データフィールド全体をメモリ記憶場所にロードしなければならず、その後で、操作を、すなわち、チェックサムの計算またはパケットデータの暗号化を完了することができる。さらに、階層化ネットワークモデルのため、一度にただ1つの操作しか実行できず、その結果、複数回にわたってデータをメモリにコピーする必要がある。しかし、パケット単位の手法は、1つのパケットで複数のタスクをオフロードすることを可能にする。したがって、ハードウェア周辺機器は、ハードウェアの能力に応じて、データに対して単一パスで2つ以上の操作を実行することができ、それによって、コンピュータシステムのスループットおよび効率を大幅に高めることができる。

【0087】

上記の方式は、特定のNICにオフロードされるタスクを指定する能力に関連して特に役立つが、他の方法では、パケット単位の情報転送も使用できることが理解されよう。例

10

20

30

40

50

えば、特定のNICが所定の時間にパケットの配信をスケジュールすることが可能である場合、NICハードウェアによってパケットをどのように、および/または、いつ送信すべきかを識別する情報を渡すために、パケット拡張データ構造を使用することができる。

【0088】

一実施形態では、トランスポートプロトコルドライバが、特定のタスクをNICにオフロードするために、パケット拡張をデータパケットに付加する前に、2つの追加機能が最初に実行される。様々な処理能力を各々が有する、多くの異なるタイプのハードウェア周辺機器が存在する場合において、本開示の実施形態は、トランスポートドライバが、コンピュータシステムに接続された周辺機器のタスクオフロード機能について最初に問い合わせることができる手段を望ましくは提供する。これらの機能が確認されると、その後、トランスポートプロトコルドライバは、関心のあるこれらのタスクを設定し、または有効にすることができる。有効にされると、以降は、上述した方法で、パケット単位に、指定されたタスクを利用することができる。

10

【0089】

リモートアクセスプロトコルのためのハードウェアアクセラレーション

言及したように、リモートアクセスおよびターミナルサービスは一般に、リモートデスクトッププロトコル(RDP)などのプロトコルを使用するデスクトップ仮想化を必要とする。本開示はRDPプロトコルに関して説明されるが、開示される原理は、仮想ネットワークコンピューティング(VNC)およびCitrix(登録商標) XenApp(登録商標)など、リモートアクセスサービスを提供する任意のシステムに容易に適用できることを当業者は理解されよう。

20

【0090】

RDPなどのプロトコルは、グラフィック表示情報をリモートコンピュータからユーザに転送することによって、また入力をユーザからリモートコンピュータに転送し、リモートコンピュータにおいて入力を局所的に投入できるようにすることによって、リモートコンピュータシステムとのユーザ対話を円滑化するように設計される。このプロトコルはまた、ユーザコンピュータ上のコンポーネントとリモートコンピュータ上で動作するコンポーネントの間で専用の通信を行なうことを可能にする、拡張可能なトランスポート機構を提供する。そのようなプロトコルは、仮想デスクトップを提示し、一般にグラフィックスを扱い、USB、プリンタ、キーボード、およびマウスなどのデバイストラフィックを処理する。ターミナルサーバは、一般に複数のリモートクライアントセッションを提供する。ターミナルサーバからのグラフィックスデータは、グラフィックスプリミティブ形式で符号化され、暗号化されるか、またはサーバでレンダリングされる必要があり、結果のビットマップは、圧縮および暗号化され、クライアントに伝送される必要がある。

30

【0091】

符号化操作、レンダリング操作、および暗号化操作は、本来はきわめてコンピュータ集約型であり、一般に高いCPU使用率を必要とする。豊富なデスクトップ仮想化技術を提供するため、対応するグラフィックス忠実度は、豊富であるべきである。符号を最適化するよりも、RDP処理中の必要なときにレンダリングタスクおよび/または圧縮タスクを実行できる追加のハードウェアリソースを利用することによって、レンダリングおよび暗号化に関連するホストCPUサイクルおよびネットワーク使用率を低減することができる。加えて、データが伝送される前に、CPU集約型リモートデスクトップ操作をネットワークプロセッサまたは他の周辺機器にオフロードすることによって、ホストCPU処理を低減することができる。上述したように、チムニオフロードとも呼ばれるタスクオフロードは、チェックサム、大きな送信セグメンテーション、IPsec、およびTCB状態処理などのTCP/IP処理を、インテリジェントネットワークインタフェースにオフロードするために使用することができる。したがって、上で説明されたチムニオフロードの原理を利用して、RDPなどのプロトコルを使用するリモートアクセスサービスの提供に関連するタスクの一部または全部を扱うほうが有利である。

40

【0092】

50

図12は、リモートデスクトップ仮想化を提供するための例示的アーキテクチャを示している。リモートデスクトッププロセスは一般に、リモートデスクトップデータパケットを送信および受信するためにトランスポート層インタフェース(TLI)を利用する、一般にはTCPである、トランスポートプロバイダ上に積み重ねられる。例えば、1つまたは複数のアプリケーション1200は、グラフィックスAPIおよび媒体APIを呼び出すことができ、それを受けて、グラフィックスAPIおよび媒体APIは、リモートアクセス処理スタック1210によってさらに処理されるグラフィックス出力を生成する。その後、ネットワーク移送ドライバ1220が、データパケットの伝送を処理することができる。

【0093】

10

図13は、リモートデスクトップ処理機能のさらなる例示的な図を提供する。1つまたは複数のアプリケーションは、サーバ上で動作し、クライアントディスプレイに表示するためのグラフィックスを生成することができる。例えば、アプリケーション11310およびアプリケーション21312は、グラフィックススタック1318によって処理されるグラフィックスを生成することができる。グラフィックススタック1318は、グラフィカルオブジェクトを表現し、それらを出力デバイスに伝送することを担う、様々なグラフィックスコンポーネントを含むことができる。このようなコンポーネント例は、グラフィックスデバイスインタフェース(GDI)1314およびDirectX(DX)1316を含む。

【0094】

20

このようなアプリケーションのグラフィックス出力は、リモートアクセスプロトコルスタック1320によって捕捉され、コマンド命令1322およびビットマップ1324としてさらに処理される。コマンド命令または描画命令は、グラフィック画像を生成するのに必要な操作を符号化するため、または特定のキャッシュを操作するために使用される。プライマリ描画命令は一般に、描画操作を符号化するために使用される。各プライマリ命令は、フィールド圧縮アルゴリズムが適用される1組のフィールドに体系化することができる。これらのアルゴリズムは、命令が最後に送られて以降、フィールドが変更されていない場合は、そのフィールドの送信を排除するように、またあるフィールドタイプがより小さなサイズのデータによって表すことができる場合は、そのフィールドタイプのためのフィールド符号化のサイズを削減するように設計することができる。プライマリ命令例は、長方形および線などのグラフィックオブジェクトを描画すること、ならびにテキストの断片を表示することを含む。セカンダリ描画命令は、キャッシュを管理するために使用することができる。

30

【0095】

ビットマップ処理1324は、キャッシング1326および圧縮1328をさらに含む。キャッシングプロセスは、タイリング機能1330およびハッシュ計算機能1332をさらに含むことができる。リモートアクセスプロトコルは、ビットマップ、カラーテーブル、および文字などの描画プリミティブを格納するために、キャッシュを使用することができる。キャッシング技法の使用は、複数の描画操作で使用されるアイテムをサーバからクライアントに送信するのが一度だけであることを保証することによって、データトラフィックを低減することができる。

40

【0096】

ビットマップ処理1324およびコマンド命令処理1322の出力は、符号化された命令1334として組み合わせることができる。リモートアクセスプロトコルは、データを圧縮するために大量圧縮1336を使用することができる。

【0097】

データのための大量圧縮を利用することに加えて、リモートアクセスプロトコルは、サーバからクライアントに送信されるビットマップデータの圧縮を実行するために、ランレングス符号化(RLE)規則の変形を使用することもできる。この時点で、データをフレーム化し(1338)、データを暗号化し(1340)、ネットワーク(1342)を介

50

して、データを伝送することができる。

【 0 0 9 8 】

上述したように、TCPなどの機能は、上述したチムニオフロードの技法を利用して、周辺機器にオフロードすることができる。上述したように、RDPなどのプロトコルに関連するグラフィックス画像のリモート処理は、更新されたビットマップ画像をサーバからクライアントに継続的に送信することによって達成され、これはコンピュータ集約的なタスクである。したがって、本開示は、上述のリモートアクセス処理タスクを包含するように、チムニオフロードの概念を拡張する。様々な実施形態では、リモートデスクトップ仮想化に関連する1つまたは複数の処理タスクは、チムニオフロードすることができる。一部の実施形態では、リモートアクセス処理タスクのサブセットをオフロードすることができる。例えば、一部の実施形態では、ビットマップ圧縮1328、キャッシング1326、および大量圧縮1336の1つまたは複数の、開示したオフロードの原理を使用して処理するために、NICなどの周辺機器にオフロードすることができる。他の実施形態では、リモートアクセス処理の大部分または全部を、例えば、図13において説明したタスクを、周辺機器にオフロードすることができ、ホストプロセッサは、制御機能および管理機能のみを保持する。

10

【 0 0 9 9 】

図14および図15は、処理1400、処理1402、処理1404、処理1406、処理1408、処理1410、処理1412、処理1414、および処理1416を含む、ターミナルサーバまたは仮想マシンへのリモートアクセスを提供するための例示的な操作手順を示しており、リモートアクセスデータユニットを送信および受信するために、トランスポート層インタフェースが使用される。図14を参照すると、処理1400は、操作手順を開始し、処理1402は、周辺機器が1つまたは複数の指定されたリモートアクセス処理タスクを実行するためのタスクオフロード機能を含むことを判定することを示している。処理1404は、後続のデータユニットと共に使用されるコンテキスト情報を含む、周辺機器が1つまたは複数の処理タスクを実行するという指示を、周辺機器に送ることを示している。処理1406は、1つまたは複数のリモートアクセス処理タスクを、周辺機器に実行させることを示している。

20

【 0 1 0 0 】

処理1408は、一実施形態では、リモートアクセスがリモートデスクトッププロトコル(RDP)を使用して実行されることを示している。処理1410は、処理タスクが部分的なRDP処理を提供できることを示している。あるいは、処理1412は、処理タスクが完全なRDP処理を提供できることを示している。処理1414は、例示的な処理タスクはRDPビットマップ圧縮を含むことを示している。処理1416は、例示的な処理タスクはRDP大量圧縮を含むことを示している。

30

【 0 1 0 1 】

図15に進むと、処理1502は、処理タスクがRDPキャッシングを含むことを示している。処理1504は、処理タスクが、コンピュータシステムの進行中の要求に応じて、選択的に周辺機器にオフロードされることを示している。処理1504は、タスクオフロードバッファ内の少なくとも1つのフラグインジケータを設定することによって、タスクオフロード機能を使用可能にすることを示している。一実施形態では、各周辺機器ドライバ(例えば図9を参照)は、そのデバイスドライバおよびそれに対応する周辺機器のためのタスクオフロード機能を各々が含む、事前定義されたタスクオフロードバッファロケーションを、それに関連付けておくことができる。タスクオフロードバッファは、周辺機器およびそのデバイスドライバによってサポートされる特定のタスクを識別することができ、サポートされる個々のタスクに固有の任意の情報も含む。一実施形態では、デバイスドライバのタスクオフロードバッファの内容は、NDISファンクション呼び出しによって検索することができる。処理1508は、一実施形態では、周辺機器がネットワークインタフェースカード(NIC)であることを示している。

40

【 0 1 0 2 】

50

処理 1 5 1 0 は、データパケットを、階層型ネットワークモデルを介して転送できることを示している。処理 1 5 1 2 は、データパケットはネットワークデータおよびパケット拡張データを含むことができることを示している。処理 1 5 1 4 は、データパケットは、周辺機器が処理タスクのバッチを実行することを指示する。処理 1 5 1 6 は、パケット拡張データが、少なくとも 1 つの処理タスクが周辺機器によって実行されることを示す少なくとも 1 つのデータフィールドを含むことを示している。

【 0 1 0 3 】

図 1 6 は、処理 1 6 0 0、処理 1 6 0 2、および処理 1 6 0 4 を含む、リモートアクセス処理タスクを周辺機器にオフロードするための例示的な操作手順を示している。図 1 6 を参照すると、処理 1 6 0 0 は、操作手順を開始し、処理 1 6 0 2 は、デバイスがリモートアクセスプロトコル処理タスクを実行することを指示するデータパケットを、周辺機器に送ることを示している。処理 1 6 0 4 は、処理タスクを前記プロセッサに実行させることを示している。

【 0 1 0 4 】

上記で言及した態様のいずれも、方法、システム、コンピュータ可読媒体、または任意のタイプの製造物で実施することができる。例えば、図 1 7 によれば、コンピュータ可読媒体は、リモートアクセス処理タスクを周辺機器にオフロードためのコンピュータ実行可能命令を格納することができる。このような媒体は、周辺機器がタスクオフロード機能を含むことを判定するための第 1 の命令サブセット 1 7 1 0、周辺機器が前記処理タスクの 1 つを実行するという指示を周辺機器に送るための第 2 の命令サブセット 1 7 1 2、および 1 つまたは複数のリモートアクセス処理タスクを周辺機器に実行させるための第 3 の命令サブセット 1 7 1 4 を含むことができる。本明細書で開示される他の様々な態様を表現するために追加的な命令セットが使用できること、また現在開示された 3 つの命令サブセットは、本開示によれば細部において異なり得ることが当業者には理解されよう。

【 0 1 0 5 】

例えば、命令には、リモートデスクトッププロトコル (R D P) を使用して実行される前記リモートアクセスのための命令 1 7 1 6 をさらに含むことができる。命令には、キャッシングおよび大量圧縮のための命令 1 7 1 8、多重化 / フレーム化、暗号化、およびコマンド命令符号化のための命令 1 7 2 0、ならびにデータパケットを、階層型ネットワークモデルを介して転送するための命令 1 7 2 2 をさらに含むことができる。

【 0 1 0 6 】

上述したように、本開示の主題の態様は、プログラムされたコンピュータ上で実行することができる。図 1 および以下の説明は、これらの態様を実行できる適切なコンピューティング環境の簡単な説明を提供することを意図している。図 1 のコンピュータシステムが、一部の実施形態において、図 2 ~ 図 4 のサーバおよびクライアントを達成できることを当業者は理解できよう。これらの例示的な実施形態では、サーバおよびクライアントは、図 1 で説明されたコンポーネントの一部または全部を含むことができ、ある実施形態では、サーバおよびクライアントは各々、本開示の特定の態様を実例化するように構成された回路構成を含むことができる。

【 0 1 0 7 】

本開示において使用される回路構成という用語は、専用ハードウェアコンポーネントを含むことができる。同一の実施形態または他の実施形態では、回路構成は、ファームウェアまたはスイッチによって機能を実行するように構成されたマイクロプロセッサを含むことができる。同一の実施形態または他の例示的な実施形態では、回路構成は、機能を実行するように動作可能なロジックを具現化するソフトウェア命令がメモリに、例えば、 R A M および / または仮想メモリにロードされたときに構成することができる、 1 つまたは複数の汎用処理ユニットおよび / またはマルチコア処理ユニットなどを含むことができる。回路がハードウェアとソフトウェアの組み合わせを含む例示的な実施形態では、実装者は、ロジックを具現化するソースコードを書くことができ、ソースコードは、汎用処理ユニットによって処理できる機械可読コードにコンパイルすることができる。

【 0 1 0 8 】

図 1 は、本開示の態様と共に構成されるコンピューティングシステムの例を示している。コンピューティングシステムは、コンピュータ 20 など含むことができ、コンピュータ 20 は、処理ユニット 21、システムメモリ 22、およびシステムメモリを含む様々なシステムコンポーネントを処理ユニット 21 に結合するシステムバス 23 を含む。システムバス 23 は、様々なバスアーキテクチャのいずれかを使用する、メモリバスまたはメモリコントローラ、周辺機器のバス、およびローカルバスを含む、いくつかのタイプのバス構造のいずれかとすることができる。システムメモリは、リードオンリメモリ (ROM) 24 およびランダムアクセスメモリ (RAM) 25 を含む。基本入出力システム (BIOS) 26 は、スタートアップ中などにコンピュータ 20 内の要素間で情報を転送する助けとなる基本ルーチンを含み、ROM 24 内に格納される。コンピュータ 20 は、図示されていないハードディスクから読み取り、ハードディスクに書き込むためのハードディスクドライブ 27、リムーバブル磁気ディスク 29 から読み取り、または磁気ディスク 29 に書き込むための磁気ディスクドライブ 28、および CD ROM または他の光学媒体などのリムーバブル光ディスク 31 から読み取り、または光ディスク 31 に書き込むための光ディスクドライブ 30 をさらに含むことができる。一部の例示的な実施形態では、本開示の態様を具現するコンピュータ実行可能命令は、ROM 24、ハードディスク (図示されず)、RAM 25、リムーバブル磁気ディスク 29、光ディスク 31、および / または処理ユニット 21 のキャッシュ内に格納することができる。ハードディスクドライブ 27、磁気ディスクドライブ 28、および光ディスクドライブ 30 は、それぞれハードディスクドライブインタフェース 32、磁気ディスクドライブインタフェース 33、および光ドライブインタフェース 34 によってシステムバス 23 に接続される。ドライブおよび関連するコンピュータ可読媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、および他のデータの不揮発性ストレージをコンピュータ 20 に提供する。本明細書で説明した環境は、ハードディスク、リムーバブル磁気ディスク 29、およびリムーバブル光ディスク 31 を利用するが、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌイカートリッジ、ランダムアクセスメモリ (RAM)、およびリードオンリメモリ (ROM) など、コンピュータによってアクセス可能なデータを格納できる他のタイプのコンピュータ可読媒体も、動作環境において使用できることを当業者であれば理解されたい。

【 0 1 0 9 】

オペレーティングシステム 35、1 つまたは複数のアプリケーションプログラム 36、他のプログラムモジュール 37、およびプログラムデータ 38 を含む、多くのプログラムモジュールは、ハードディスク、磁気ディスク 29、光ディスク 31、ROM 24、または RAM 25 上に格納することができる。ユーザは、キーボード 40 およびポインティングデバイス 42 などの入力デバイスを介して、コマンドおよび情報をコンピュータ 20 に入力することができる。他の入力デバイス (図示されず) は、マイク、ジョイスティック、ゲームパッド、衛星用アンテナ、またはスキャナなどを含むことができる。上記および他の入力デバイスは、システムバスに結合されるシリアルポートインタフェース 46 を介して、しばしば処理ユニット 21 に接続されるが、パラレルポート、ゲームポート、またはユニバーサルシリアルバス (USB) などの他のインタフェースによって接続することもできる。ディスプレイ 47 または他のタイプの表示装置も、ビデオアダプタ 48 などのインタフェースを介して、システムバス 23 に接続することができる。ディスプレイ 47 に加えて、コンピュータは、スピーカおよびプリンタなどの他の周辺出力デバイス (図示されず) も一般に含む。図 1 のシステムは、ホストアダプタ 55、小型コンピュータシステムインタフェース (SCSI) バス 56、および SCSI バス 56 に接続される外部記憶デバイス 62 も含む。

【 0 1 1 0 】

コンピュータ 20 は、リモートコンピュータ 49 など、1 つまたは複数のリモートコンピュータへの論理接続を使用して、ネットワーク環境において動作することができる。リ

10

20

30

40

50

モートコンピュータ４９は、別のコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピアデバイスまたは他の共通ネットワークノード、仮想マシンとすることができ、図１にはメモリ記憶デバイス５０しか示されていないが、コンピュータ２０に関連して上述した要素の多くまたはすべてを一般に含むことができる。図１に示された論理接続は、ローカルエリアネットワーク（ＬＡＮ）５１およびワイドエリアネットワーク（ＷＡＮ）５２を含むことができる。このようなネットワーク環境は、オフィス、企業規模のコンピュータネットワーク、イントラネット、およびインターネットにおいて一般的である。

【０１１１】

ＬＡＮネットワーク環境において使用される場合、コンピュータ２０は、ネットワークインタフェースまたはアダプタ５３を介してＬＡＮ ５１に接続することができる。ＷＡ 10
Ｎネットワーク環境において使用される場合、コンピュータ２０は、インターネットなどのワイドエリアネットワーク５２上で通信を確立するためのモデム５４または他の手段を一般に含むことができる。モデム５４は、内蔵または外付けとすることができ、シリアルポートインタフェース４６を介してシステムバス２３に接続することができる。ネットワーク環境では、コンピュータ２０に関連して示されたプログラムモジュールまたはそれら 20
の一部は、リモートメモリ記憶デバイス内に格納することができる。示されるネットワーク接続は例であり、コンピュータ間の通信リンクを確立する他の手段が使用できることが理解されよう。さらに、本開示の主題についての多くの実施形態は、コンピュータシステムに非常によく適合すると想定されるが、本文書において、本開示をこのような実施形態に限定することは意図していない。

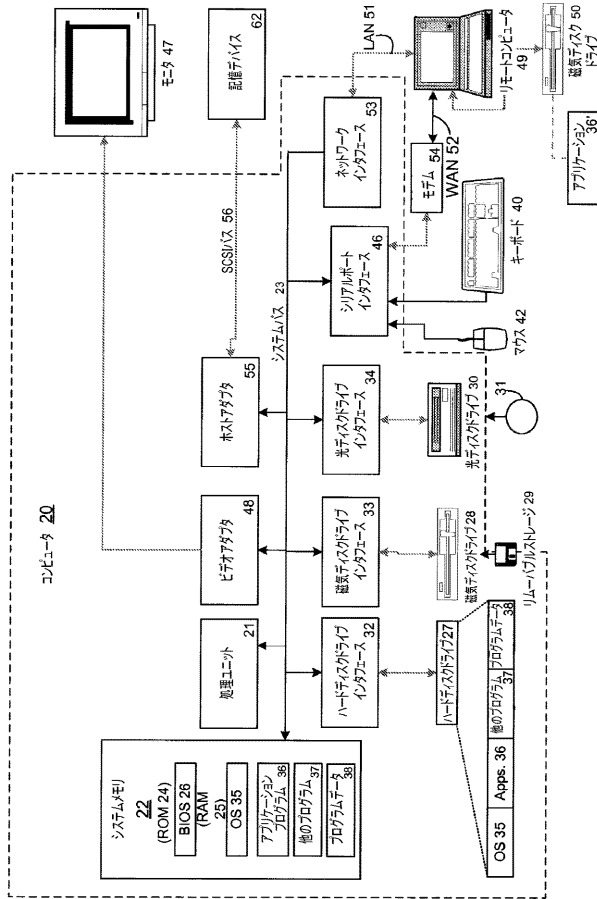
【０１１２】

上述の詳細な説明は、例および／または動作図によって、システムおよび／またはプロセスの様々な実施形態について説明した。このようなブロック図および／または例が、１つまたは複数の機能および／または動作を含む場合、このようなブロック図または例の中の各機能および／または動作は、幅広いハードウェア、ソフトウェア、ファームウェア、または実質的にそれらの任意の組み合わせによって、個別におよび／または一括して実装 30
できることが当業者には理解されよう。

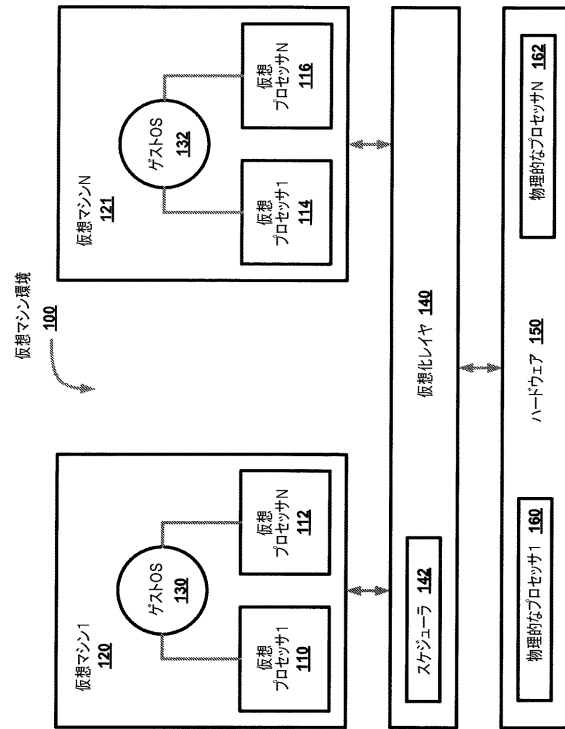
【０１１３】

本明細書で説明された主題についての特定の態様および実施形態が示され、説明されたが、本明細書の教示に基づいて、変更および修正を施すことができ、したがって、本明細 30
書で説明された主題の真の主旨および範囲内にあるかのように、添付の特許請求の範囲は、このような変更および修正のすべてを包含することが当業者には明らかであろう。

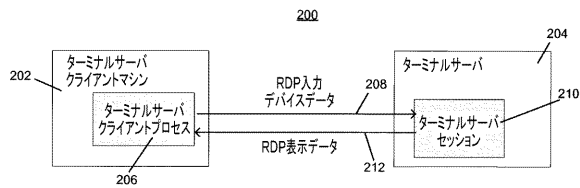
【図 1】



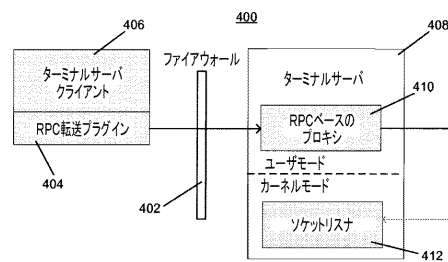
【図 1 a】



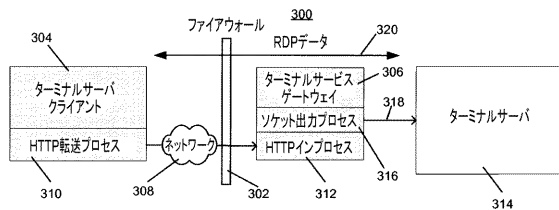
【図 2】



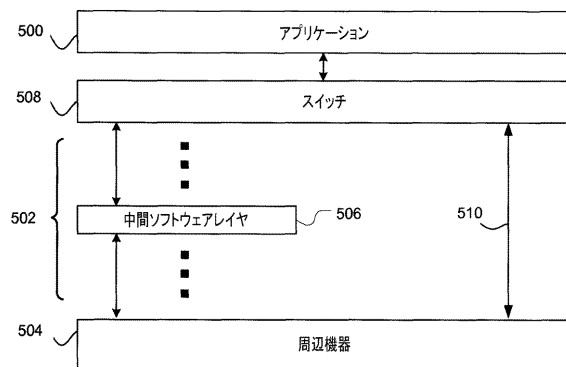
【図 4】



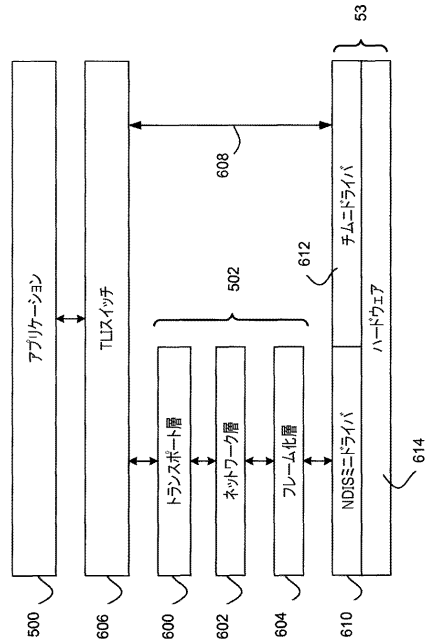
【図 3】



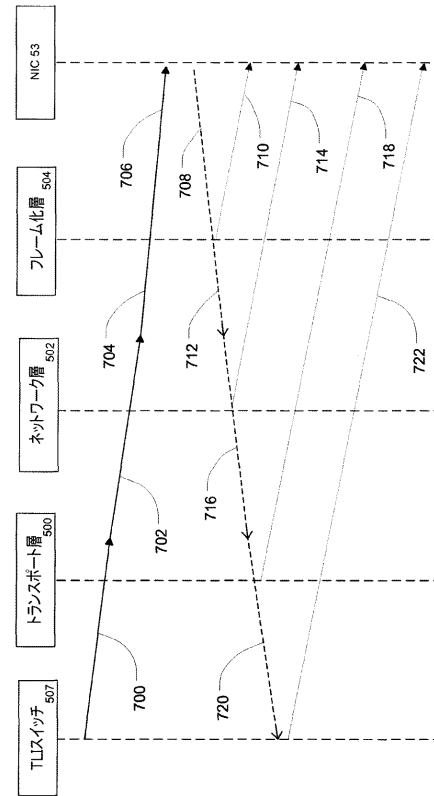
【図 5】



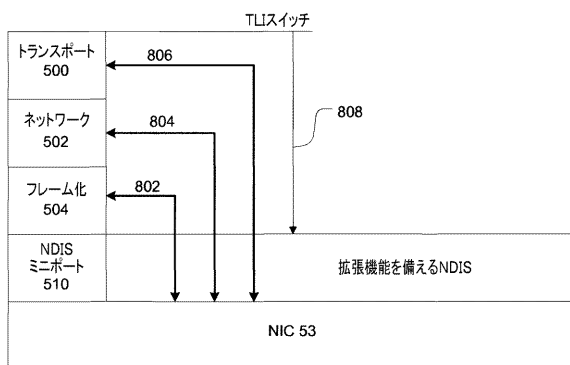
【図 6】



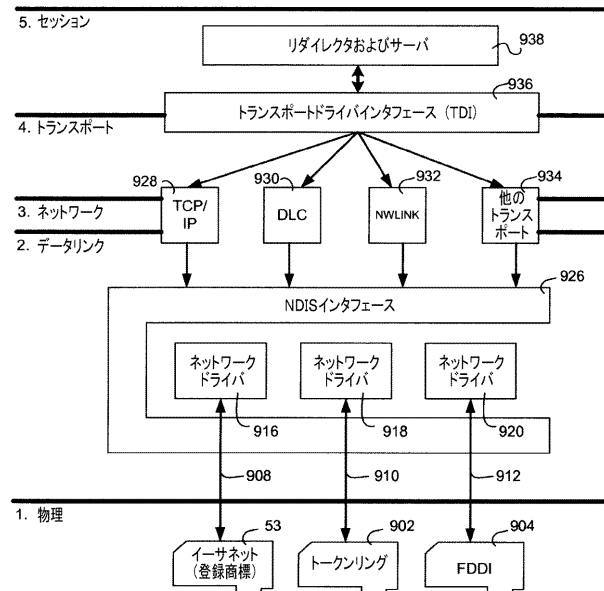
【図 7】



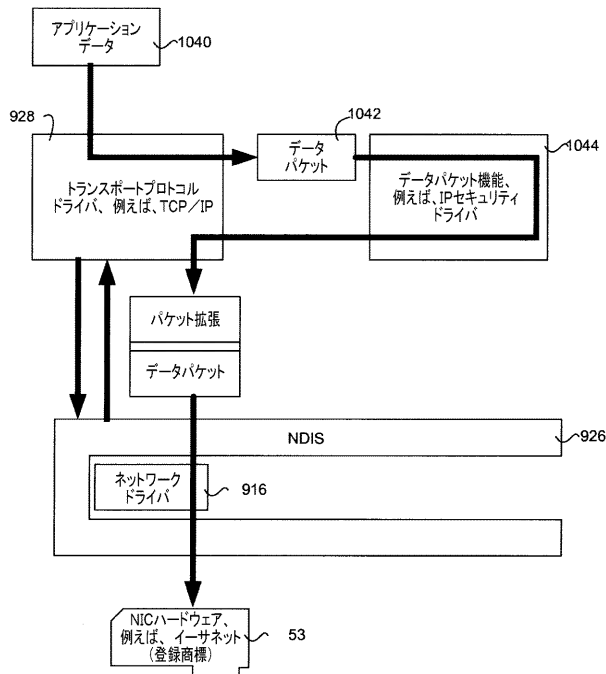
【図 8】



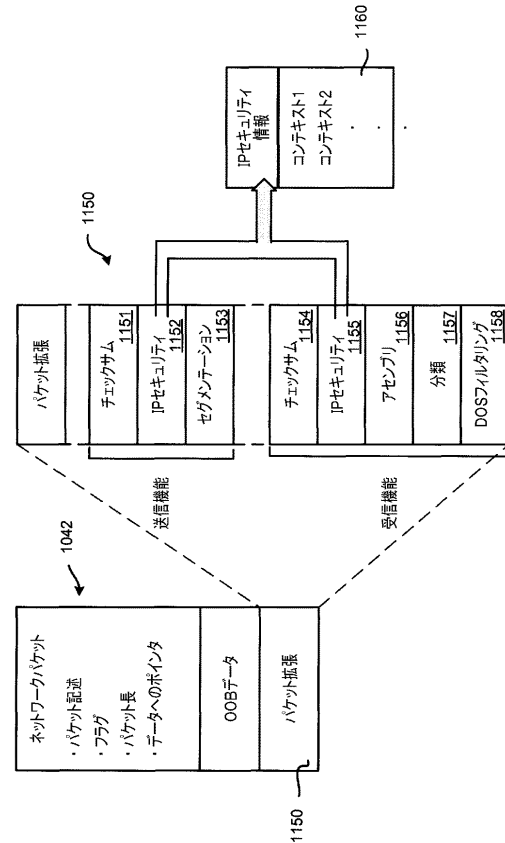
【図 9】



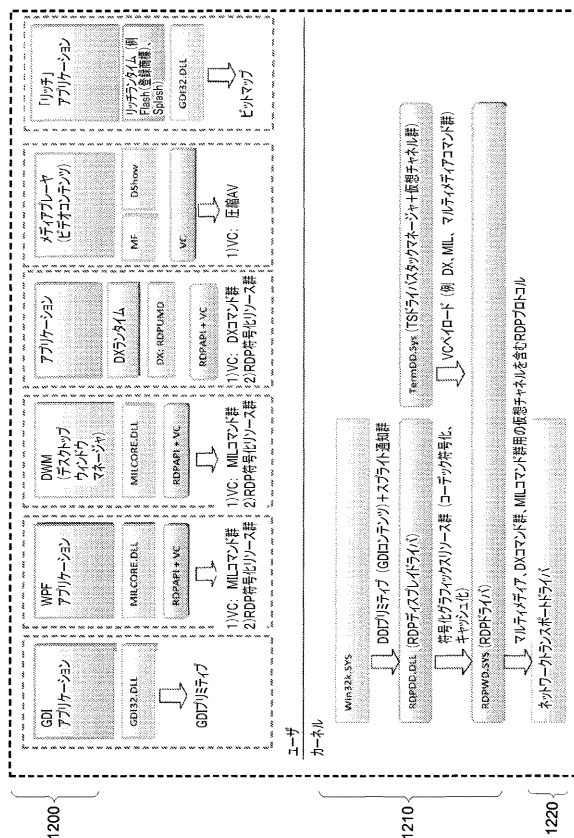
【 図 1 0 】



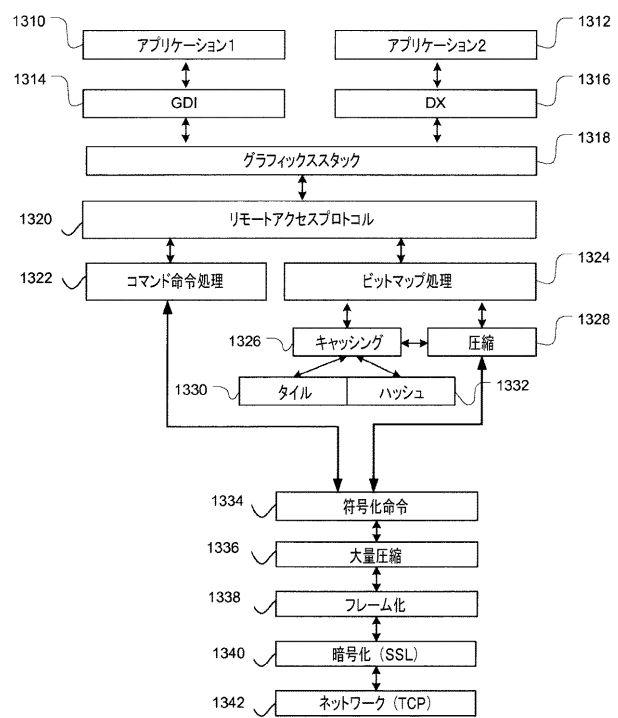
【 図 1 1 】



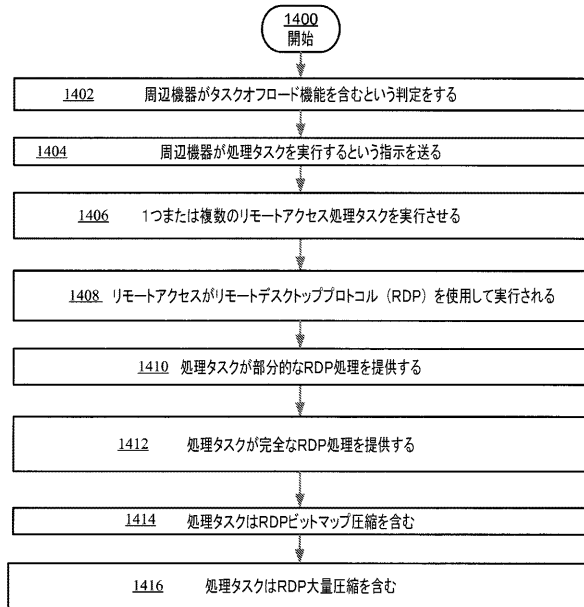
【 図 1 2 】



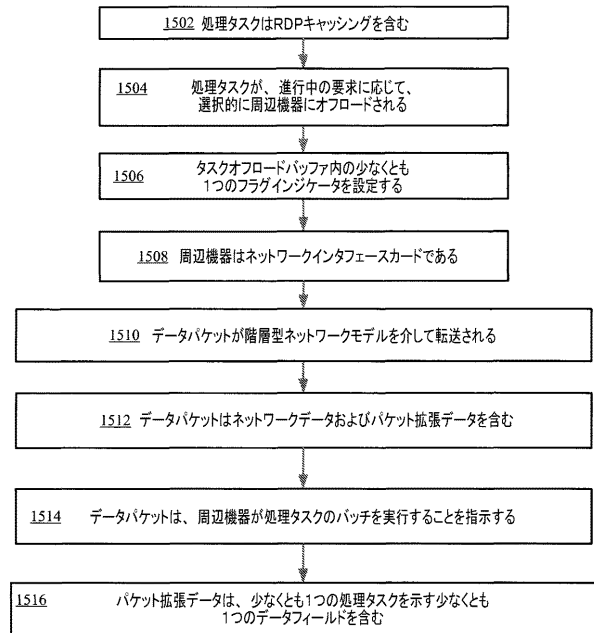
【 図 1 3 】



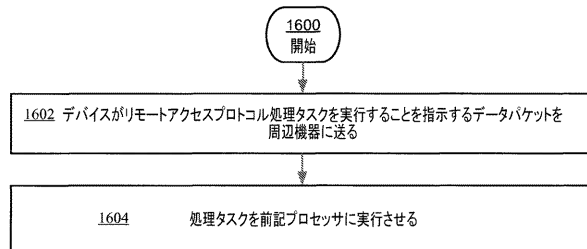
【図 14】



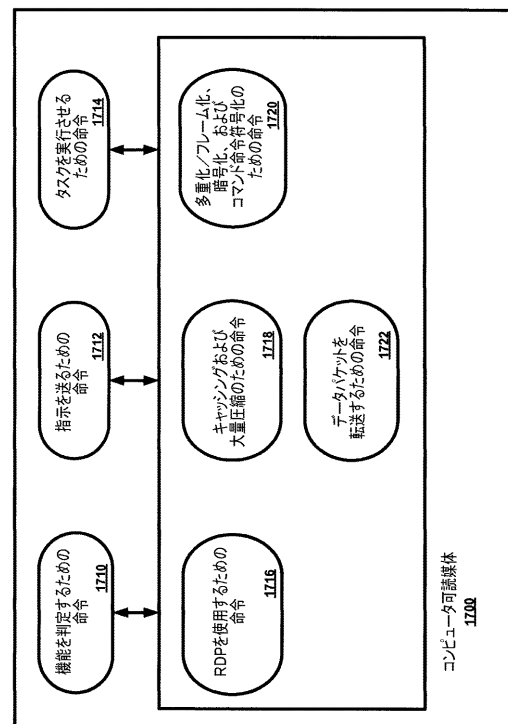
【図 15】



【図 16】



【図 17】



フロントページの続き

- (72)発明者 エヌケー スリニヴァス
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテント内
- (72)発明者 ロバート ウィルヘルム シュミーター
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテント内
- (72)発明者 ナディム アブド
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテント内

審査官 木村 貴俊

- (56)参考文献 特表2002-517855(JP, A)
特開2003-333076(JP, A)
特開2004-030612(JP, A)
国際公開第2007/053304(WO, A1)
Presentation and Hosted Desktop Virtualization Team, Remote Desktop Protocol Performance, MICROSOFT, 2008年10月13日, 全17ページ, URL, http://download.microsoft.com/download/4/d/9/4d9ae285-3431-4335-a86e-969e7a146d1b/RDP_Performance_WhitePaper.docx

(58)調査した分野(Int.Cl., DB名)

G06F 3/06 - 3/08
G06F 12/00 - 12/16
G06F 13/00 - 13/42