

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2009-278364

(P2009-278364A)

(43) 公開日 平成21年11月26日(2009.11.26)

(51) Int.Cl.

H04L 12/56

(2006.01)

F 1

H04L 12/56

3000D

テーマコード(参考)

5K030

審査請求 未請求 請求項の数 9 O L (全 12 頁)

(21) 出願番号

特願2008-127381 (P2008-127381)

(22) 出願日

平成20年5月14日 (2008.5.14)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(74) 代理人 100076428

弁理士 大塚 康徳

(74) 代理人 100112508

弁理士 高柳 司郎

(74) 代理人 100115071

弁理士 大塚 康弘

(74) 代理人 100116894

弁理士 木村 秀二

(74) 代理人 100130409

弁理士 下山 治

(74) 代理人 100134175

弁理士 永川 行光

最終頁に続く

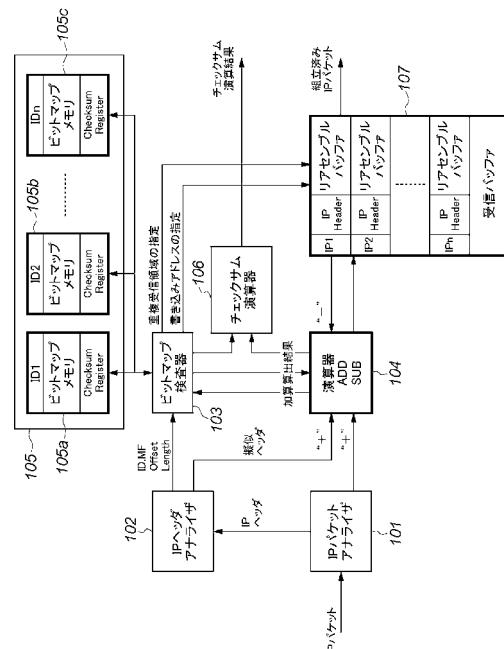
(54) 【発明の名称】パケット受信装置及びその処理方法

(57) 【要約】

【課題】 フラグメント(断片化)パケットの受信からリアセンブル(再組立)パケットの検証が完了するまでの処理を高速に実行可能とする。

【解決手段】 受信したパケットのデータを記憶するバッファを有するパケット受信装置であって、パケットのヘッダ部に含まれる情報に基づいて、バッファに記憶されるデータが先に記憶されたデータと重複するか否かを判断する。その結果、重複すると判断された場合には、重複するデータのチェックサムの減算を行い、重複しないと判断された場合には、バッファに記憶するデータのチェックサムの加算を行う。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

受信したパケットのデータを記憶するバッファを有するパケット受信装置であって、前記パケットのヘッダ部に含まれる情報に基づいて、前記バッファに記憶されるデータが先に記憶されたデータと重複するか否かを判断する判断手段と、前記判断手段によって重複すると判断された場合、前記重複するデータのチェックサムの減算を行い、前記判断手段によって重複しないと判断された場合、前記バッファに記憶するデータのチェックサムの加算を行う演算手段と、を有することを特徴とするパケット受信装置。

【請求項 2】

断片化されたパケットの受信状況を管理するビットマップメモリに基づいて前記断片化されたパケットを再組立する再組立手段を更に有することを特徴とする請求項 1 に記載のパケット受信装置。

【請求項 3】

前記パケットのヘッダ部に含まれる情報は、少なくともオフセット位置及びパケット長であることを特徴とする請求項 1 又は 2 に記載のパケット受信装置。

【請求項 4】

前記演算手段は、予め指定された領域におけるデータのチェックサムを算出しないことを特徴とする請求項 1 乃至 3 の何れか 1 項に記載のパケット受信装置。

【請求項 5】

受信したパケットから IP ヘッダを抽出し、疑似ヘッダを組み立てた後に前記演算手段によってチェックサムを算出することを特徴とする請求項 1 乃至 4 の何れか 1 項に記載のパケット受信装置。

【請求項 6】

断片化されたパケットから、再組立パケットの全長又はペイロード長を算出し、算出した再組立パケットの全長又はペイロード長を、前記バッファの中の IP ヘッダの全長記憶領域又はペイロード長記憶領域に書き込むことを特徴とする請求項 1 乃至 5 の何れか 1 項に記載のパケット受信装置。

【請求項 7】

断片化されたパケットの受信を完了すると、前記断片化されたパケットを再組立した受信パケットを出力することを特徴とする請求項 1 乃至 6 の何れか 1 項に記載のパケット受信装置。

【請求項 8】

受信したパケットのデータを記憶するバッファを有するパケット受信装置の処理方法であって、

前記パケットのヘッダ部に含まれる情報に基づいて、前記バッファに記憶されるデータが先に記憶されたデータと重複するか否かを判断する判断工程と、

前記判断工程において重複すると判断された場合、前記重複するデータのチェックサムの減算を行い、前記判断工程において重複しないと判断された場合、前記バッファに記憶するデータのチェックサムの加算を行う演算工程と、

を有することを特徴とするパケット受信装置の処理方法。

【請求項 9】

請求項 1 乃至 7 の何れか 1 項に記載のパケット受信装置としてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、パケット受信装置及びその処理方法に関するものである。

【背景技術】**【0002】**

10

20

30

40

50

通信ネットワークのプロトコルとしてTCP/IPやUDP/IPなどのプロトコルが用いられている。TCP/IPはTransmission Control Protocol/Internet Protocolの略で、UDP/IPはUser Datagram Protocol/Internet Protocolの略である。

【0003】

このようなプロトコルを用いるネットワークでは、IP層における1パケットの全長を65535 Octets(1 Octet=8bit)まで定義できる仕様となっている。しかし、IP層の下のデータリンク層、例えばIEEE802.3では、1パケットとして送出できる最大送信ユニットは1492 Octetsと制限されている。以下、最大送信ユニットをMTU(Maximum Transfer Unit)と称す。

【0004】

また、Ethernet(登録商標)では1500 Octet、電話回線によるダイアルアップ接続では576 Octet、FibreChannelでは4352 Octetなど、MTUはデータリンク層によって異なる長さが採用されている。

【0005】

IP層で取り扱うパケット長がデータリンク層のMTUよりも長い場合や、パケットが今のネットワークのMTUより小さいMTUをもつネットワークに転送される場合など、元のIPパケットを分割してからデータリンク層に渡す必要がある。この分割の仕組みはIPフラグメントとして知られている。

【0006】

また、フラグメントパケットの格納位置情報を示すパケットディスクリプタを転送回路に指示することによってパケットの重複がないように所定の転送先へ転送する技術が開示されている(例えば、特許文献1参照)。この特許文献1では、図6に示すように、受信パケットをパケット解析回路601がパケットバッファメモリ602とパケットディスクリプタメモリ603に取り込む。そして、各フラグメントパケットの格納位置情報を示すパケットディスクリプタによって全てのリアセンブルデータが揃うと、制御回路609がパケット組立回路604に指示し、転送回路605を介して重複がないように所定の転送先へと転送する。

【0007】

また、フラグメントパケットが重複なく順番通りに連続して到達するか否かに応じて、チェックサム演算方法を切り換える技術が開示されている(例えば、特許文献2参照)。この特許文献2では、重複のない、順番通りに連続した領域が到達しているフラグメントパケットに対してはチェックサム演算装置を用いて高速に演算を行う。一方、フラグメントパケットが順番通りに連続しておらず、重複した領域があった場合にはプロセッサなどでチェックサム演算を行う。

【特許文献1】特開平05-327771号公報

【特許文献2】特開2007-215013号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

しかしながら、上記従来のIPフラグメント処理方法では、次のような問題があった。フラグメントされる前のIPパケットの上位プロトコルにTCP/UDPやICMPなどが用いられ、これらのプロトコルにはヘッダ情報としてチェックサムフィールドを有し、チェックサムの整合性によって当該パケットの有効性の判定を行っている。

【0009】

このチェックサムを算出するためにはIPヘッダの一部から構成されるIP擬似ヘッダと、TCP(UDP)ヘッダ、ペイロードを2 Octets単位での1の補数和、即ちRFC1071で定義されるインターネットチェックサムを算出する必要があった。この算出のために、再組立を行ったメモリ上からヘッダ、ペイロード情報を再度読み出して算出していた。

【0010】

また、フラグメントを再組立するために、一旦全ての受信フラグメントパケットを蓄積

10

20

30

40

50

し、並べ替えを行った後、データグラムを再組立するという工程を経ていた。そのため、IP フラグメントパケットを生じるネットワークにおいては受信性能が低下してしまうという問題もあった。

【0011】

本発明は、フラグメント(断片化)パケットの受信からリアセンブル(再組立)パケットの検証が完了するまでの処理を高速に実行可能とすることを目的とする。

【課題を解決するための手段】

【0012】

本発明は、受信したパケットのデータを記憶するバッファを有するパケット受信装置であって、前記パケットのヘッダ部に含まれる情報に基づいて、前記バッファに記憶されるデータが先に記憶されたデータと重複するか否かを判断する判断手段と、前記判断手段によって重複すると判断された場合、前記重複するデータのチェックサムの減算を行い、前記判断手段によって重複しないと判断された場合、前記バッファに記憶するデータのチェックサムの加算を行う演算手段とを有することを特徴とする。

10

【0013】

また、本発明は、受信したパケットのデータを記憶するバッファを有するパケット受信装置の処理方法であって、前記パケットのヘッダ部に含まれる情報に基づいて、前記バッファに記憶されるデータが先に記憶されたデータと重複するか否かを判断する判断工程と、前記判断工程において重複すると判断された場合、前記重複するデータのチェックサムの減算を行い、前記判断工程において重複しないと判断された場合、前記バッファに記憶するデータのチェックサムの加算を行う演算工程とを有することを特徴とする。

20

【発明の効果】

【0014】

本発明によれば、フラグメント(断片化)パケットの受信からリアセンブル(再組立)パケットの検証が完了するまでの処理を高速に実行することが可能となる。

【発明を実施するための最良の形態】

【0015】

以下、図面を参照しながら発明を実施するための最良の形態について詳細に説明する。

【0016】

図1は、プロトコル処理装置における受信処理部の構成の一例を示す図である。図1において、101はIPパケットアナライザであり、受信されたIPパケットのヘッダ部とペイロード部とを分離する。102はIPヘッダアナライザであり、IPパケットアナライザ101で分離されたIPパケットのヘッダ部を解析する。103はビットマップ検査器であり、フラグメントパケットの受信状況を管理し、IPパケットアナライザ101で分離されたペイロード部を後述するバッファへ書き込む位置を管理する。

30

【0017】

104は演算器であり、チェックサム演算を行うために、ペイロード部のデータを加算し、或いはリアセンブルバッファからのデータを減算する。105はビットマップメモリであり、複数のリアセンブル中のIPパケットの受信状況を管理すると同時に、演算中のチェックサム値を記憶する。

40

【0018】

106はチェックサム演算器であり、演算器104での演算結果とビットマップメモリ105に記憶された演算途中のチェックサム演算結果とから最終的なチェックサムを算出して出力する。107はリアセンブルバッファ・受信バッファであり、リアセンブル中のIPパケットやフラグメントされていないIPパケットなどが一時記憶される。

【0019】

以上の構成において、断片化(フラグメント)されていないIPパケットを受信した際の動作について説明する。PHY Device(不図示)で受信したフレームはMAC Device(不図示)に入力され、フレーム全体に渡ってFCS(Frame Check Sequence)によりフレームの検証を行う。MAC Deviceでは算出したFCSとパケットの末尾2 OctetsのFCSを比較

50

し、同じ値であればIPパケットをIPパケットアナライザ101に入力し、異なる値であれば受信パケットはエラーが含まれているパケットとして廃棄する。

【0020】

ここで、IPパケットアナライザ101に入力されるIPパケットのフォーマットを、図2を用いて説明する。

【0021】

図2は、IP層におけるIPv4パケットのフォーマットを示す図である。図2に示すように、IPv4ヘッダはverフィールドからDest IP Addrまでの最小で20 Octetsで構成される。IPヘッダの先頭にはバージョンを示す4ビットのverフィールドがあり、IPv4の場合は“0100=4”が示される。続いて4ビットのヘッダ長がHlenとして4 Octets単位で示される。そのため、オプションが4 Octets単位にならなければPADDING DATAが加えられてヘッダサイズが4 Octets単位になるように揃えられる。例えば、オプションが無い場合、Hlenは“0101”で、ヘッダ長は20 Octets (5×4 Octets)となる。そして、ヘッダ長が判明すれば、入力したIPパケットをIPヘッダ部とIPデータグラム部とに分割することができる。

10

【0022】

Total Lengthフィールドは自身のIPヘッダを含むIPパケットの全長を示す全長記憶領域である。これは、16bitで表現されていることからIPパケットの最大長は65535 Octetsということになる。そして、IPパケット長(Total Length)により、ペイロードの終端が判るため、IPヘッダ、ペイロードの分離と同時に、パケット長、ペイロード長のチェックを行うことができる。

20

【0023】

IPIDはIPデータグラムに与えられるユニークな識別子(ID)である。ネットワークの途中でIPデータグラムがフラグメントされた場合や、IPパケットの複製が発生した場合など、同一のIPIDを持つIPデータグラムが別個に処理されるのを防ぐことができる。

【0024】

Flag及びOffsetは経路上でIPフラグメントする場合に用いられるフィールドである。IPパケットアナライザ101により、IPヘッダがIPヘッダアナライザ102に入力され、IPヘッダ中のFlagフィールドに含まれるMFビットとOffsetフィールドの値を確認する。フラグメントされていないIPパケットであれば後続のフラグメントパケットの存在を示すMFビットと、フラグメントパケットのオフセット位置を示すOffsetフィールドの値が共に“0”である。

30

【0025】

MFビットが“1”であれば、後続のフラグメントされたIPパケットの存在を示し、Offsetフィールドに値が与えられていれば、IPフラグメントパケットの再組立後のオフセット位置を表す。ここで確認されたMFビットの値とOffsetフィールドの値とIPIDの値はビットマップ検査器103に渡される。

【0026】

PTCLフィールドはIPパケットの上層に位置するプロトコル種別である。Header Check sumフィールドはIPヘッダ部分のみのRFC1071で定義されているインターネットチェックサムが与えられる。送出元IPアドレスであるSource IP Addr、宛先IPアドレスであるDest IP Addrが示されることでIPヘッダが構成される。

40

【0027】

一方、IPヘッダアナライザ102はチェックサム算出のための疑似ヘッダを組み立てる。疑似ヘッダはL4プロトコルであるTCP/UDPヘッダや、ICMPヘッダにあるチェックサム値を算出する時に使用されるIPヘッダのうちの送信元IPアドレス、宛先IPアドレス、プロトコル種別、パケット長が含まれる。

【0028】

パケット長に関しては、MFが“0”となるフラグメントパケットのヘッダに記述して

50

いるOffset+Total Lengthによって再組立後のIPパケット長を算出する手法がある。また、Offsetが“0”のパケット中のL4ヘッダを解析することによって事前に再組立後のIPパケット長を類推する手法がある。

【0029】

後者の手法はUDP、ICMPに対しては有効であるが、TCPにはパケット長を示すフィールドがないため、その場合はIPヘッダから全長を算出する。このようにして抽出された疑似ヘッダは演算器104に加えられ、算出すべきチェックサムの一部として加算される。

【0030】

ビットマップ検査器103ではOffsetとMFが“0”であるため、フラグメントされていないIPパケットとして処理される。フラグメントされていないIPパケットの場合、ビットマップメモリ105を用いず、ビットマップ検査器103はリアセンブルバッファ・受信バッファ107の書き込み領域として受信バッファ領域を指定する。リアセンブルはバッファを1つのID当たり最大で64KB用意しておく必要があるが、受信バッファ領域はMTU分にMAXヘッダを加えた容量、例えばイーサネット（登録商標）であれば、1500+14Byteあれば良い。

【0031】

このようにして指定された書き込みアドレスに演算器104を経由して受信パケットがリアセンブルバッファ・受信バッファ107の受信バッファに書き込まれる。IPヘッダ中に記述されているTotal Length分の転送が終了した段階で演算器104からの演算結果を受けてチェックサム演算結果を出力すると共に、リアセンブルバッファ・受信バッファ107からIPパケットが出力される。

【0032】

従って、フラグメントされていないIPパケットの受信に関してはチェックサムによるIPパケットの確認と、パケットの出力をほぼ同時に行うことができる。

【0033】

次に、IPフラグメントされていて且つ重複した受信領域のある複数のIPパケットを受信した場合について説明する。

【0034】

図3は、IPフラグメントが発生するネットワーク構成の一例を示す図である。図3に示すように、ホストA、BはネットワークA、Bにそれぞれ接続され、ネットワークA、BはMTUが1500 Octetsである。そして、ルータA、Bを経由してMTUが576 OctetsのネットワークCにより相互に通信できる環境を構築している。

【0035】

ここで、ホストAのアプリケーションが自身のIP層からIPデータグラム（2000 Octets）をホストBへ送信する場合を説明する。まず、ホストAのIP層はネットワークAのMTUが1500 Octetsであるので、IPパケットを1500 OctetsにするためにIPデータグラムを分割する。具体的には、IPヘッダ長の20 Octetsを除いた1480 Octetsのデータグラムで構成するIPパケットと、残りの520 Octetsのデータグラムを持つIPパケットに分割する。IPパケットは、ネットワークBに対して送出されるので、このパケットを受け取ったルータAがネットワークCに送出するために、MTUが576 Octetsに合わせたIPパケットの転送を行う。

【0036】

この場合、1480 OctetsのIPパケットは552 Octetsのデータグラムを持つ572 OctetsのIPパケットが2つと、376 Octetsのデータグラムを持つ396 OctetsのIPパケットにIPフラグメントされる。続く520 OctetsのIPパケットはそのまま転送される。一方、MTUが576 OctetsのネットワークからMTUが1500 Octetsのネットワークに転送する場合、特にIPフラグメント等の処理を行う必要はない。

【0037】

このように、ホストAのアプリケーションから送信された2000 Octetsのデータはホス

10

20

20

30

40

50

トBに到達するまでに、図4に示すように、4つのIPパケットに分割される。そして、ホストBのアプリケーションはホストAが送信した2000 Octetsのデータが到着すると、ホストBのIP層でリアセンブル(再組立)処理が行われる。

【0038】

ホストBに到着した4つのIPフラグメントされたパケットは、それぞれのIPヘッダには同一のIPIDが記述されているので、元は1つのIPパケット、つまり、リアセンブル処理を行うIPパケットであることが判る。更に、Total Length、Offset、Flagフィールドに記述されたMore Fragment(MF)ビットからIPフラグメントされる前のパケットのどの位置を構成していたかが判明する。従って、元のIPパケットにリアセンブルすることができる。

10

【0039】

また、フラグメントする前のIPパケットにUDPを用いていた場合は、UDPヘッダの先頭から終端までがIPデータグラムとしてフラグメント・リアセンブルされる。そのため、ホストBではリアセンブルが完了してIPデータグラムが完成した後に、UDPの処理を行うことができる。

【0040】

図3に示す例では、ホストAからホストBにIPパケットが到達するまでの経路は1つであるため、フラグメントが発生してもIPデータグラムの重複は発生しない。しかし、異なるMTUをもつネットワークインターフェースをリンクアグリゲーションした場合や、複数の経路をもつネットワークでは、異なるMTUでフラグメントされたパケットが到着する場合がある。また、ネットワークアドレスを学習していないスイッチングハブが大量の複製パケットを発生させ、同一パケットが別の時刻に到着することもある。

20

【0041】

図5は、時系列で受信したフラグメントパケットが復元中のIPデータグラムの受信済み領域に重複される場合を示す図である。まず、MFが“1”、Offsetが1480、Lengthが1500のパケット2を受け取る。IPヘッダアナライザ102は、MF、Offsetが共に零でないので、フラグメントされたIPパケットであると解析する。これにより、IPID、MF、Offset、Length情報が103ビットマップ検査器103に渡され、ビットマップメモリ105の1つのIDが割り当てられる。

30

【0042】

ビットマップメモリ105は、フラグメントパケットの受信済み領域を判断するメモリと、チェックサムの途中演算結果を保持するチェックサムレジスタとを有し、IDで管理される。受信済み領域を管理するメモリは、8 Octets受信単位当たり1Bitのメモリを割り振っており、64KBの最大IPパケットに対応するために、1つのID当たり8KBの容量を持っている。このIDはビットマップ検査器103によってIPIDに対応付けられ、複数のIPIDを持つIPフラグメントパケットの組立が可能である。更に、IDはリアセンブル・受信バッファのIDと対応付けられ、リアセンブル管理にも用いられている。

【0043】

ここで、受信パケットのLengthはIPヘッダ込みのLengthなので、1500から20 Octets分を減算し、再組立するペイロードは1480となる。結果、ビットマップメモリ105には1480～2960までが受信済みであることを示す“1”を立てると共にIPIDが記録される。

40

【0044】

次に、ビットマップ検査器103は、リアセンブルバッファ・受信バッファ107への書き込みアドレスを通知する。そして、演算器104からのペイロードが、そのアドレスに従ってリアセンブル・受信バッファ107のペイロード長記憶領域に記録される。このようにして、演算器104を通してリアセンブルバッファ・受信バッファ107にLengthで示されたペイロードの記録が完了すると、演算器104はビットマップ検査器103に加算結果を通知する。そして、当該IDのビットマップメモリ105のチェックサムレジスタに加算結果を保持する。

【0045】

50

続いてIPIDが先のパケット2と同一であるパケット1を受信する。パケット1はOffsetが0であることからリアセンブル後に先頭のペイロードになるIPパケットである。また、Lengthが1500なので0~1480までが先のパケット2と同様に受信する。ここで、ビットマップ検査器103は重複受信領域を検査するために、ビットマップメモリ105からビットマップを読み出し、0~1480までの受信済み領域の有無を検査する。この場合、全ての受信領域が未受信であることが確認されるので、リアセンブルバッファ・受信バッファ107には何も通知せずに書き込みアドレスの指定のみを行う。

【0046】

パケット1のように、先頭のパケットを受信した場合、IPヘッダアナライザ102は疑似ヘッダを生成し、演算器104を介してリアセンブルバッファ・受信バッファ107に記録する。このようにして疑似ヘッダを含めたチェックサム演算が可能となる。

【0047】

次に、重複受信領域があった場合の演算方法について説明する。パケット3を受信した後に、パケット4を受信し、ビットマップ検査器103がビットマップメモリ105から読み出した既受信領域と、今回受信したパケット4の受信領域とを比較する。比較の結果、パケット3はOffsetが4024、Lengthが1500、パケット4はOffsetが2960、Lengthが1500なので、4024~4440までの416 Octetsが重複していると判断する。尚、ペイロードデータに関しては、後から受信したパケットが上書きされる。

【0048】

従って、ビットマップ検査器103は重複受信領域が4024~4440で、書き込みアドレスが2960~4440まであるという2つの情報をリアセンブルバッファ・受信バッファ107に出力する。重複受信領域があった場合、ペイロードを書き込む前にリアセンブルバッファ・受信バッファ107に記録している重複受信領域に関して演算器104にて予め減算を行っておく。減算を行った領域に関しては新規に受信したペイロードデータを上書きすることが可能となる。

【0049】

次に、パケット5、パケット6のように、完全に受信領域が重複している場合も同様の処理を行う。重複した7000~8480の領域はリアセンブルバッファ・受信バッファ107に通知される。ここで、パケット6のバッファ書き込みに先立ち、以前に読み込んだデータを減算した後に加算処理とリアセンブルバッファ・受信バッファ107への書き込み処理を行う。

【0050】

その後、MFが“0”的パケット7を受信した時点で、そのパケットのTotal LengthとOffsetによってリアセンブル後のIPパケットの終端が明らかになる。この位置はビットマップ検査器103によって記憶され、終端が判明した場合はフラグメントパケット受信毎に先頭から終端までのビットマップメモリの領域に渡って受信済みであるか否かの検査を行う。そして、先頭から終端までの領域が受信済みとなったことが判明した時点で再組立後のIPパケット、もしくはその上位プロトコルのパケットにおける長さ（全長又はペイロード長）が判明する。そして、リアセンブルバッファ・受信バッファ107中のIPヘッダの全長記憶領域、もしくはペイロード長記憶領域に再組立パケットの長さ（全長、もしくはペイロード長）を書き込む。

【0051】

全ての領域が受信済みとなった場合、リアセンブル完了となり、途中まで演算した結果であるチェックサムレジスタの値と、最終パケットの演算器104からのチェックサムの結果をチェックサム演算器106に通知する。そして、チェックサムの演算結果を出力すると同時に、リアセンブルバッファ・受信バッファ107から組立済みのIPパケットを出力することが可能となる。

【0052】

また、フラグメントパケットの組立中に、様々なIPIDを持つパケットを受信するため、複数のIPIDのリアセンブルを同時並列的に処理することが求められる。そのため、ビット

10

20

30

40

50

マップメモリ 105 はそれぞれのIPIDに対応した複数のIDに対して処理が可能な容量を持ち、それぞれのチェックサムレジスタを有している。即ち、図1に示す105a、105b、105cなどを備えている。更に、リアサンブルバッファ・受信バッファ107に関するビットマップメモリ105に対応した容量を持つ必要がある。

【0053】

そして、ビットマップ検査器103は、フラグメントパケットのIPIDに対応したビットマップメモリ105のID管理とリアサンブル・受信バッファ107のID管理を行い、再組立が完了するまで保持する。

【0054】

実施形態では、IPパケットの受信、特に、フラグメントされたIPパケットの受信において、IPフラグメントパケットの受信と同時に、受信領域を確認しながらペイロード部のチェックサム及びIPパケットの再組立を行う。次に、全てのフラグメントパケット受信完了を検出するとIPヘッダによる疑似ヘッダを作成、疑似ヘッダのチェックサムを算出、先のペイロード部のチェックサムと疑似ヘッダのチェックサムからトランスポート層のチェックサムによる検証を行う。

【0055】

これにより、フラグメントパケット受信から再組立パケットの検証まで完了する時間が従来例と比較して短縮され、高速に処理することが可能となる。

【0056】

尚、上述の実施形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ(CPU若しくはMPU)が記録媒体に格納されたプログラムコードを読み出し実行する。これによっても、本発明の目的が達成されることは言うまでもない。

【0057】

この場合、コンピュータ読み取り可能な記録媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記録媒体は本発明を構成することになる。

【0058】

このプログラムコードを供給するための記録媒体として、例えばフレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0059】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、次の場合も含まれることは言うまでもない。即ち、プログラムコードの指示に基づき、コンピュータ上で稼働しているOS(オペレーティングシステム)などが実際の処理の一部又は全部を行い、その処理により前述した実施形態の機能が実現される場合である。

【0060】

更に、記録媒体から読み出されたプログラムコードがコンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込む。その後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行い、その処理により前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【0061】

【図1】プロトコル処理装置における受信処理部の構成の一例を示す図である。

【図2】IP層におけるIPv4パケットのフォーマットを示す図である。

【図3】IPフラグメントが発生するネットワーク構成の一例を示す図である。

【図4】IPパケットのフラグメント(断片化)処理を説明するための図である。

【図5】時系列で受信したフラグメントパケットが復元中のIPデータグラムの受信済み

10

20

30

40

50

領域に重複される場合を示す図である。

【図6】従来のプロトコル処理装置の構成を示すブロック図である。

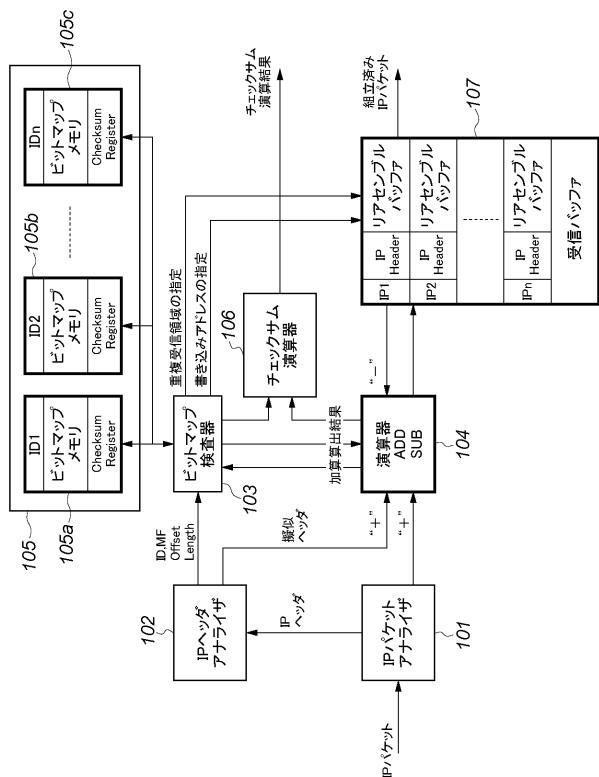
【 符号の説明 】

【 0 0 6 2 】

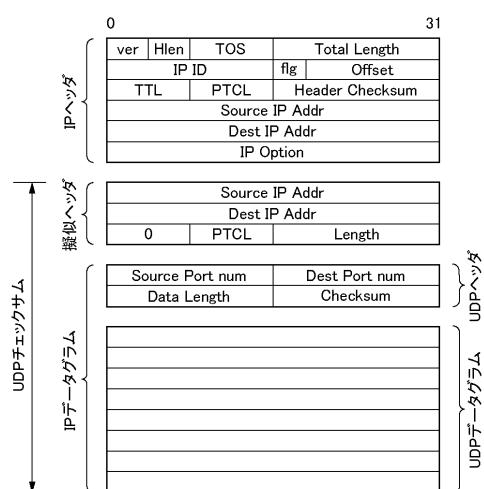
- 1 0 1 I P パケットアナライザ
 - 1 0 2 I P ヘッダアナライザ
 - 1 0 3 ビットマップ検査器
 - 1 0 4 演算器
 - 1 0 5 ビットマップメモリ
 - 1 0 6 チェックサム演算器
 - 1 0 7 リアセンブルバッファ・受信バッファ

10

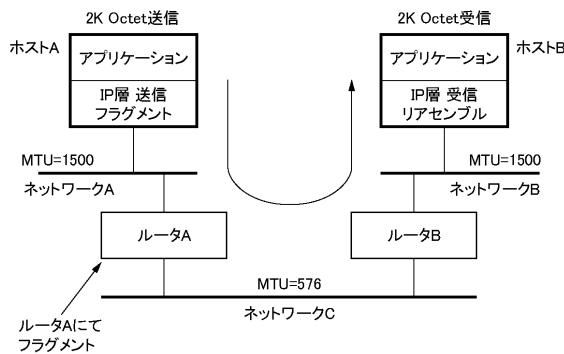
【 义 1 】



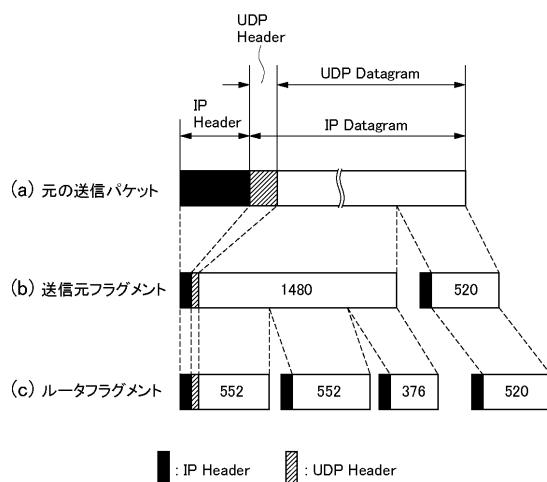
【 図 2 】



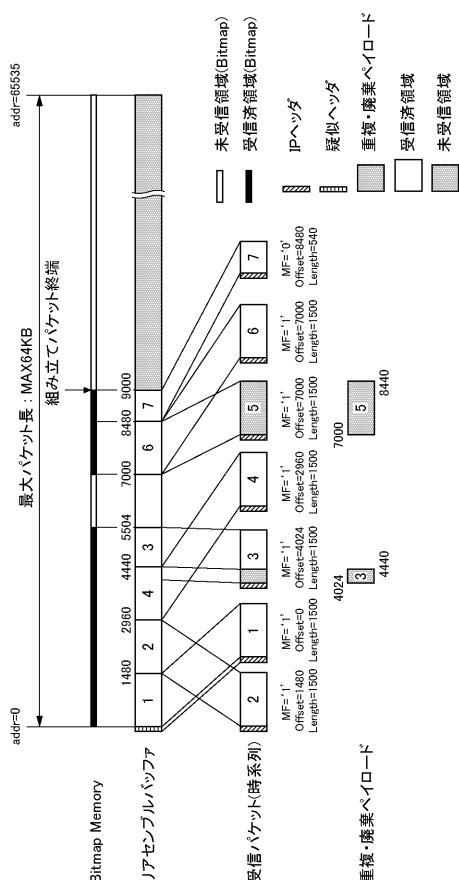
【図3】



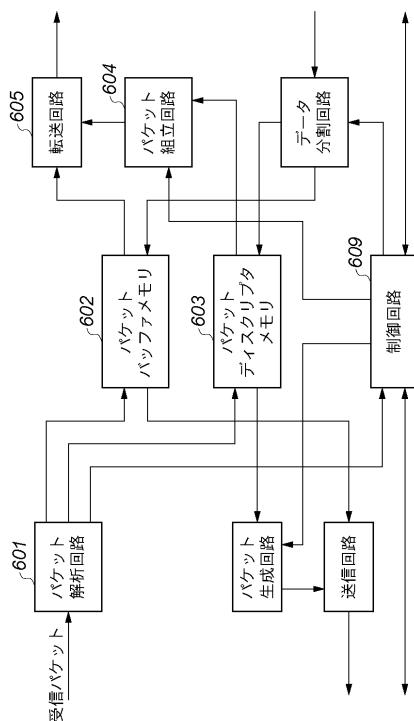
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 森村 和彦
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
F ターム(参考) 5K030 GA01 HA08 JA05 KA03 LA01