

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4612820号  
(P4612820)

(45) 発行日 平成23年1月12日(2011.1.12)

(24) 登録日 平成22年10月22日(2010.10.22)

(51) Int.Cl.

F I

H O 4 L 29/10 (2006.01)

H O 4 L 13/00 3 O 9 C

請求項の数 11 (全 15 頁)

(21) 出願番号 特願2004-264644 (P2004-264644)  
 (22) 出願日 平成16年9月10日(2004.9.10)  
 (65) 公開番号 特開2006-81032 (P2006-81032A)  
 (43) 公開日 平成18年3月23日(2006.3.23)  
 審査請求日 平成19年9月10日(2007.9.10)

(73) 特許権者 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 100090273  
 弁理士 國分 孝悦  
 (72) 発明者 森川 健一  
 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  
 (72) 発明者 川島 淳  
 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  
 (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

最終頁に続く

(54) 【発明の名称】 通信制御装置及び方法

(57) 【特許請求の範囲】

【請求項 1】

複数レイヤのプロトコルのヘッダ生成に必要なヘッダ生成情報を収集し、ヘッダ生成指示及びヘッダ生成情報をフレーム毎に出力するプロトコル制御手段と、

前記プロトコル制御手段から前記ヘッダ生成指示及び前記ヘッダ生成情報が入力されると、複数レイヤのプロトコルのヘッダを並列して1ヘッダ毎に生成し、ヘッダ生成完了通知を出力するヘッダ生成手段と、

前記ヘッダ生成手段から前記ヘッダ生成完了通知が入力されると、前記生成された複数レイヤのプロトコルのヘッダ及びフレーム単位のペイロードをフレームとして合成するヘッダ/ペイロード合成手段と

を有することを特徴とする通信制御装置。

【請求項 2】

前記ヘッダ生成手段は、

T C Pヘッダを1ヘッダ毎に生成するT C Pヘッダ生成手段と、

U D Pヘッダを1ヘッダ毎に生成するU D Pヘッダ生成手段と、

I Pヘッダを1ヘッダ毎に生成するI Pヘッダ生成手段と、

M A Cヘッダを1ヘッダ毎に生成するM A Cヘッダ生成手段とを有し、

前記プロトコル制御手段は、前記T C Pヘッダ生成手段、前記U D Pヘッダ生成手段、前記I Pヘッダ生成手段及び前記M A Cヘッダ生成手段にヘッダの生成指示を出力することができることを特徴とする請求項1記載の通信制御装置。

## 【請求項 3】

前記プロトコル制御手段は、前記ヘッダ/ペイロード合成手段が1つ前のフレームとして合成したヘッダを出力した後に、次のフレームのヘッダ生成指示を前記ヘッダ生成手段に出力することを特徴とする請求項1又は2記載の通信制御装置。

## 【請求項 4】

さらに、フレーム単位のペイロードを記憶するためのメモリを有し、

前記プロトコル制御手段は、ペイロードが前記メモリに書き込まれた後に、そのペイロードに対応するヘッダ生成指示を前記ヘッダ生成手段に出力することを特徴とする請求項1又は2記載の通信制御装置。

## 【請求項 5】

さらに、フレーム単位のペイロードを記憶するためのメモリを有し、

前記プロトコル制御手段は、前記ヘッダ/ペイロード合成手段が1つ前のフレームとして合成したヘッダを出力した後、かつ次のペイロードが前記メモリに書き込まれた後に、次のフレームのヘッダ生成指示を前記ヘッダ生成手段に出力することを特徴とする請求項1又は2記載の通信制御装置。

## 【請求項 6】

さらに、前記ヘッダ/ペイロード合成手段により合成されたフレームを送信するフレーム送信手段を有することを特徴とする請求項1～5のいずれか1項に記載の通信制御装置。

## 【請求項 7】

さらに、送信データをフレーム単位のペイロードに分割するペイロード分割手段を有することを特徴とする請求項1～6のいずれか1項に記載の通信制御装置。

## 【請求項 8】

さらに、前記ペイロード分割手段により分割されたフレーム単位のペイロードを記憶するためのメモリを有することを特徴とする請求項7記載の通信制御装置。

## 【請求項 9】

さらに、前記メモリからフレーム単位のペイロードを読み出して前記ヘッダ/ペイロード合成手段に出力するペイロード読み出し手段を有することを特徴とする請求項8記載の通信制御装置。

## 【請求項 10】

さらに、送信データ長を基にフレーム個数及びフレーム長を算出するフレーム個数フレーム長算出手段を有し、

前記ペイロード分割手段は、前記算出されたフレーム個数及びフレーム長を基にペイロードの分割を行うことを特徴とする請求項7～9のいずれか1項に記載の通信制御装置。

## 【請求項 11】

通信制御装置が行う通信制御方法であって、

複数レイヤのプロトコルのヘッダ生成に必要なヘッダ生成情報を収集し、ヘッダ生成指示及びヘッダ生成情報をフレーム毎に出力するプロトコル制御ステップと、

前記ヘッダ生成指示及び前記ヘッダ生成情報が入力されると、複数レイヤのプロトコルのヘッダを並列して1ヘッダ毎に生成し、ヘッダ生成完了通知を出力するヘッダ生成ステップと、

前記ヘッダ生成完了通知が入力されると、前記生成された複数レイヤのプロトコルのヘッダ及びフレーム単位のペイロードをフレームとして合成するヘッダ/ペイロード合成ステップと

を有することを特徴とする通信制御方法。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、通信制御装置及び方法に関するものである。

## 【背景技術】

## 【 0 0 0 2 】

パーソナルコンピュータ等でなくとも、機器を直接ネットワークに接続する機会や需要が高まっている。例として携帯端末、プリンタ、カメラ、コピー機、ディスプレイ、ビデオ機器、音響装置などが挙げられる。これらの機器は比較的低価格・低処理能力のCPUを搭載、又は全く搭載していない場合もあり、その様な機器資源を利用しネットワークにおいて標準的に使用されているTCP/UDP/IPといったインターネット標準プロトコルを実現しようとすれば、TCP/UDP/IPのプロトコルスタックを低処理能力CPUで低ビットレート（帯域）ではあるが実現するか、又はTOE（TCP/IPオフロードエンジン）といったプロトコル処理に特化した補助的デバイスを利用する事で比較的広帯域なネットワーク通信を実現するか、の方法しかない。とはいったものの機器が扱う情報の量は写真、動画、音楽、いずれにおいても大きくなる一方で、機器のコストを低価格に抑える必要がある、といった背景においてはTOEによるネットワーク通信機能実現のアプローチが今後増加してくることは想像に難くない。

10

## 【 0 0 0 3 】

TOE実現（TCP/IPプロトコルスタックのハードウェアによる実現）において、最も簡単にこれを実現する為には、ただ単に今迄ソフトウェアで実現していた方法をそのままハードウェア化すればよいと言う考え方がある。ソフトウェアで実現されているTCP/UDP/IPプロトコルスタックは十分に枯れた方法により実現されており、この考え方は1つの妥当な考え方ではある。しかしソフトウェア（CPU向け）に考案された方法をそのままハードウェア化したときに、それがゲートやレジスタ（フリップフロップ）やメモリといったプリミティブなデバイスで実現される事を考えるとソフトウェアの実現方法がそのままハードウェアの適切な実現方法であるかといえ、違ふと言わざるを得ない。ハードウェアは複雑なポインタ処理や、変数の動的生成や消滅、再帰処理、多くのプロセス生成、といった処理方法は苦手であり、ソフトウェアで実現されたTCP/IPプロトコルスタックは複雑なポインタ処理と変数やメモリ領域の動的生成や消滅といった手順が到るところに存在している。

20

## 【 0 0 0 4 】

従来手順をそのままハードウェアで実現すると大きく以下の流れになる。（1）アプリケーションデバイスから送信指示された一連の送信データを一時的に連続した共有メモリ領域に保存する。（2）共有メモリに保存された一連の送信データを、最も上位のプロトコル制御ブロックが参照、分割、ヘッダ追加を行い、次のプロトコル制御ブロックに共有メモリのアクセス権利と共に制御を移す。各プロトコル制御ブロックは順次、各々共有メモリの参照、分割、ヘッダ追加を行う。（3）ヘッダとペイロードを連続したメモリ空間に隣接して配置する。ペイロードの全てを連続したメモリ空間に配置できない場合は別のメモリ空間にペイロードのみを配置する。（4）ヘッダと隣接したペイロードや隣接しないペイロード、フレームの連続関係はメモリ制御情報として必要な数だけメモリ内に保存される（一般にmbufと呼ばれる）。

30

## 【 0 0 0 5 】

即ち、複雑なペイロード分割とペイロードやヘッダの管理情報（mbuf）の生成後にヘッダが生成され、この処理は各プロトコルスタックで各々順次実行される。

40

## 【 0 0 0 6 】

また、下記の特許文献1には、TCP/IPをハードウェア的に処理する装置及びその動作方法が記載されている。

## 【 0 0 0 7 】

【特許文献1】特開2001-268159号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【 0 0 0 8 】

しかし、上記のような手順を踏むと、アプリケーションの転送単位によるTCPセグメント化やIPフラグメント化が必要な場合は、1つ1つのフレームに対する全てのヘッダ

50

をメモリに書き込まなければならない。すなわちリンク層に依存するMTUに対して、アプリケーションの転送単位が大きくなるにつれて、分割後に発生するフレームが増加し、そのフレームに付加しなければならないヘッダを書き込むために確保しなければならないメモリ領域が増大するという課題がある。

【0009】

一般的には、メモリ内でヘッダとペイロードをフレーム化して出力する。しかし、アプリケーションの転送単位によりTCPのセグメント化やIPのフラグメント化が必要な場合は、一つ一つのフレームに対するヘッダ全てをメモリに書き込まなければならない。

【0010】

本発明の目的は、アプリケーションの転送単位が大きくなることにより送信するフレーム数が増加する場合であっても、ヘッダを書き込むために確保しなければならないメモリ領域の増大を抑止することである。

【課題を解決するための手段】

【0011】

本発明の通信制御装置は、複数レイヤのプロトコルのヘッダ生成に必要なヘッダ生成情報を収集し、ヘッダ生成指示及びヘッダ生成情報をフレーム毎に出力するプロトコル制御手段と、前記プロトコル制御手段から前記ヘッダ生成指示及び前記ヘッダ生成情報が入力されると、複数レイヤのプロトコルのヘッダを並列して1ヘッダ毎に生成し、ヘッダ生成完了通知を出力するヘッダ生成手段と、前記ヘッダ生成手段から前記ヘッダ生成完了通知が入力されると、前記生成された複数レイヤのプロトコルのヘッダ及びフレーム単位のペイロードをフレームとして合成するヘッダ/ペイロード合成手段とを有することを特徴とする。

【0012】

また、本発明の通信制御方法は、複数レイヤのプロトコルのヘッダ生成に必要なヘッダ生成情報を収集し、ヘッダ生成指示及びヘッダ生成情報をフレーム毎に出力するプロトコル制御ステップと、前記ヘッダ生成指示及び前記ヘッダ生成情報が入力されると、複数レイヤのプロトコルのヘッダを並列して1ヘッダ毎に生成し、ヘッダ生成完了通知を出力するヘッダ生成ステップと、前記ヘッダ生成完了通知が入力されると、前記生成された複数レイヤのプロトコルのヘッダ及びフレーム単位のペイロードをフレームとして合成するヘッダ/ペイロード合成ステップとを有することを特徴とする。

【発明の効果】

【0013】

本発明によれば、フレーム単位で複数レイヤのプロトコルのヘッダを並列して生成し、フレーム単位のペイロードと合成するので、アプリケーションの転送単位が大きくなることにより送信フレーム数が増加する場合であっても、フレームに付加しなければならないヘッダを書き込むために確保しなければならないメモリ領域の増大を抑止することができる。

【発明を実施するための最良の形態】

【0014】

図1は、本発明の実施形態による通信制御装置の構成例を示すブロック図である。以下、図1を用いてその構成を説明する。

【0015】

[アプリケーションデバイス]

アプリケーションデバイス(101)は、送信要求(106)と送信データ長(107)をフレーム個数フレーム長算出部(102)に出力し、送信データ(110)をペイロード分割部(103)に出力する。アプリケーションデバイス(101)には、2つの種類が想定される。1つ目はコンスタントビットレート(単位時間あたりの情報量が一定)のデバイスである。2つ目はバリエブルビットレートのデバイス(単位時間あたりの情報量が一定ではないが、単位時間辺りの情報量は明示される)である。その他にバリエブルビットレートのデバイスからの情報を一時的にバッファし定期的に一定量の情報を出力す

10

20

30

40

50

る様シェーピングするデバイスである場合も想定されるがこれは1つ目の種類のデバイスと同じと見なせる。また、サンプリング周波数で予め規定された量子化ビットレートでRAW情報を出力してくるデバイスも想定できるが、これは1つ目の種類のデバイスと同様に見なすことが出来る。これら全てのアプリケーションデバイス(101)に接続される本実施形態による通信制御装置は、アプリケーションデバイス(101)とのインタフェースにおいて、送信要求(106)、送信データ長(107)、送信データ(110)、を受ける事とする。

#### 【0016】

##### [フレーム個数フレーム長算出部]

フレーム個数フレーム長算出部(102)は、アプリケーションデバイス(101)から入力された送信要求(106)をトリガに、同じく入力された送信データ長(107)を基に送信データを幾つのフレームに分割して出力すべきかを算出する。

#### 【0017】

図2のフローチャートを用いて、ペイロード長とヘッダ数算出の流れについて詳説する。フレーム個数フレーム長算出部(102)は、アプリケーションデバイス(101)からの送信要求(106)を受信すると併せて、その一度の送信要求で送信したいデータの長さを送信データ長(107)として受け取る(ステップ301、302)。するとフレーム個数フレーム算出部(以後FFCと呼ぶ)はフレーム数(FLM\_COUNT)、先頭フレームペイロード長(HEAD\_PAY\_LEN)、中間フレームペイロード長(MID\_PAY\_LEN)、最終フレームペイロード長(FIN\_PAY\_LEN)を算出する必要があるが、これらをまず初期化する(ステップ303)。フレームをどのような長さの単位で送信するかについてはTCP/UDPを想定した場合、インタフェースのMTU又はTCPのMSSが想定されるが、ここではまずインタフェースのMTUを取得する(ステップ304)。また、TCP/UDP/IPから形成されるフレームを扱う場合にフレームのペイロード長を算出する為には各々のプロトコルヘッダの長さを鑑みる必要があるのでIPHL(IP Header Length)及びUDPHL(UDP Header Length)及びTCPLH(TCP Header Length)を規定する(ステップ305)。このステップ305では、それぞれ基本のヘッダ長を示してあるが各プロトコルにおいてオプションヘッダを想定する場合はこの場でオプションヘッダ長まで加味したヘッダ長を規定する。次にコネクションの種別についてTCPかUDPか又はRAWであるか、を識別する。コネクション種別は予めFFC(102)にて保持しておく方法とコネクション識別子を基に導出する方法が考えられる(ステップ306)。

#### 【0018】

UDPの場合は送信データ長がインタフェース層MTU-IPHL-UDPHL以下であるかを確認する(ステップ311)。そうである場合は1つのフレームに与えられた送信データが全て格納可能であるのでフレーム数1、先頭フレームペイロード長=送信データ長である、とする(ステップ308)。そうでない場合は送信データを複数のフレームに分割する必要がある。この分割数や各ペイロード長を算出する作業は仮にフレームを送信していった場合にどうなるかという事を想定しながら分割作業を進めていく。まず仮の送信済みデータ長(DONE\_LEN)を0とする(ステップ312)。次に未送信データ長がフラグメントペイロード長以下であるかを確認する。これは最終フラグメントであるかという判断をおこなっており、未送信データ長は送信データ長から仮の送信済みデータ長(DONE\_LEN)を減算して求め、これとMTU-IPHLの長さを比較する等という方法が想定できる(ステップ313)。ここまでの事前チェックの流れにおいては、初めての比較では未送信データ長がフラグメントペイロード長以下であるということは起き得ない。次にフレーム数(FLM\_COUNT)をインクリメントする(ステップ314)。送信済みデータ長が0であれば(ステップ315)、先頭フレームであると判断されるので先頭フレームのペイロード長はMTU-IPHL-UDPHLであり、仮の送信済みデータ長はMTU-IPHL-UDPHLであると決定される(ステップ317)。

10

20

30

40

50

。先頭フレームではない場合は中間フレームであるから中間フレーム長を  $MTU - IPH L$ 、仮の送信済みデータ長 ( $DONE\_LEN$ ) を  $DONE\_LEN + (MTU - IPH L)$  とする。これらの後再び未送信データ長がフラグメントペイロード長以下であるか、即ち最終フラグメントであるかを判断し (ステップ 313)、もし最終フラグメントであれば最終フレームは送信データ長から  $DONE\_LEN$  を減算する事で求められる (ステップ 318)。その後最終フラグメントフレームの分のフレーム数をカウントアップ ( $FLM\_COUNT++$ ) し (ステップ 319)、フレーム数、先頭フレームペイロード長、中間フレームペイロード長、最終フレームペイロード長が決定出力される (ステップ 310)。

#### 【0019】

コネクション種別が  $RAW$  の場合はフレーム数は1つで、フレームペイロード長は  $MTU - IPH L$  の範囲内に収まるべきであり、もしそうでないとしてもその様に送信データを切り落として送信する事を想定している。又はアプリケーションデバイス (101) にエラーとして応答を送るという方法も考えられる (ステップ 308、309)。

#### 【0020】

コネクション種別が  $TCP$  の場合はフレーム長は  $MSS$  (マキシマムセグメントサイズ) が基本となるので取得する (ステップ 320)。次に仮の送信済みデータサイズ ( $DONE\_LEN$ ) をゼロに初期化し (ステップ 321)、(仮の) 未送信データ長が  $MSS$  以下であるかをチェックする (ステップ 322)。もし  $MSS$  以下であれば与えられた送信データは次の1セグメントで送信完了できる事が判断できる。もし  $MSS$  より大きければフレーム数をインクリメント (ステップ 323) し、先頭セグメントであれば (この判断は  $DONE\_LEN$  が0であるか否かで行う) (ステップ 324)、先頭フレーム長は  $MSS$  であり、 $DONE\_LEN$  は  $MSS$  となる (ステップ 326)。再び仮の未送信データ長が  $MSS$  以下であるかを確認し、まだ  $MSS$  より大きい場合はフレーム数をカウントアップし今度は中間フレームペイロード長を  $MSS$  とし送信済みデータ長  $DONE\_LEN = DONE\_LEN + MSS$  とする (ステップ 325)。再び未送信データ長が  $MSS$  以下であるかを確認し、もし最終セグメントであると判断されれば最終フレームとして最終フレーム長を送信データ長から  $DONE\_LEN$  を減算した値とし (ステップ 327)、最終フレームを送信フレーム数としてカウントする (ステップ 328)。

#### 【0021】

上述の様にフレーム数、先頭フレームペイロード長、中間フレームペイロード長、最終フレームペイロード長が決定出力される (ステップ 310)。これら4つの値をフレーム個数とフレーム長に関する値とし、ペイロード分割部 (103) とプロトコル制御部 (115) 及びヘッダ/ペイロード合成部 (120) に出力する。

#### 【0022】

##### [ペイロード分割部の分割]

ペイロード分割部 (103) ではフレーム個数フレーム長算出部 (102) から与えられたフレーム個数とフレーム長に関する入力 (109) を基にアプリケーションデバイス (101) から与えられた送信データ (110) を分割しながらフレームペイロードとしてペイロード  $TX$  メモリ (104) に書き込み保存を行う (111)。ペイロード分割部 (103) は、フレームペイロード書き込みが完了したらプロトコル制御部 (115) に対しフレームペイロードの送信準備が出来た事を通知する (136)。

#### 【0023】

##### [ペイロード分割部のチェックサム]

ペイロード分割部 (103) ではフレーム個数フレーム長算出部 (102) から与えられたフレーム個数とフレーム長に関する入力 (109) を基にアプリケーションデバイス (101) から与えられた送信データ (110) を分割しながらフレームペイロードとしてペイロード  $TX$  メモリ (104) に書き込み保存を行う (111) 際、フレームペイロードのチェックサムをフレーム毎に求め、 $UDP$  ヘッダ生成部 (117) 及び、 $TCP$  ヘッダ生成部 (116) にチェックサム値を伝える (135)。

10

20

30

40

50

## 【 0 0 2 4 】

## [ T X メモリ ]

T X メモリ ( 1 0 4 ) では 1 つの送信要求に伴う一連の送信データを、送信要求毎、フレームペイロード F I F O にフレームペイロードの列として保存している。図 3 に示す様に T X メモリ ( 1 0 4 ) 内部には 1 つの送信要求に伴い 1 つの F I F O が割り当てられ ( 3 4 0 )、例えばここでは 1 つの送信要求による送信データが 5 つのフレームペイロードから成ることを示している。本図ではフレームペイロード 1 ( 先頭フレーム ) からフレームペイロード 5 ( 最後尾フレーム ) が F I F O に蓄えられているがフレーム単位での送信が行われる場合はこの様に 5 つのフレームが溜まることはない。各々のフレームペイロードは F I F O に挿入された直後にペイロード読み出し部 ( 1 0 5 ) により抜き出され送信される事も考えられる。

10

## 【 0 0 2 5 】

## [ ペイロード読み出し部 ]

ペイロード読み出し部 ( 1 0 5 ) ではヘッダ / ペイロード合成部 ( 1 2 0 ) からの、1 つのペイロード読み出し指示 ( 1 1 4 ) に対し、送信すべきフレームペイロードを 1 つ取り出し、ヘッダ / ペイロード合成部 ( 1 2 0 ) へ出力する ( 1 1 3 )。

## 【 0 0 2 6 】

## [ プロトコル制御部 ]

プロトコル制御部 ( 1 1 5 ) ではフレーム個数フレーム長算出部 ( 1 0 2 ) からの指示により、フレームごとに必要なヘッダを特定してヘッダに必要な情報を収集し、各プロトコルヘッダ生成部 ( 1 1 6 ~ 1 1 9 ) にヘッダ生成指示及びヘッダ生成情報を通知する ( 1 2 2 ~ 1 2 5 )。図 4 にプロトコル制御部 ( 1 1 5 ) のフローチャートを示す。ステップ S 1 2 0 1 においてフレーム個数フレーム長算出部 ( 1 0 2 ) よりフレーム個数、先頭フレームのペイロード長、中間フレームのペイロード長、最終フレームのペイロード長の 4 つの値 ( 1 0 8 ) を受信したかの判断を行う。もし受信すれば ( ステップ S 1 2 0 1 : Y e s )、ステップ S 1 2 0 2 において送信フレームが T C P なのか U D P なのかを判断し、フレーム送信に必要なプロトコルヘッダを生成するための情報を収集する ( ステップ S 1 2 0 3 )。この時、収集する情報は 1 つのフレームに付加されるヘッダについてのみ行われる。つまり最初は先頭フレームに付加されるヘッダのみの情報収集を行う。また、収集する各種情報は静的に設定されている場合と動的にテーブル検索を行う場合がある。動的なテーブルとしては、コネクションに付随する宛先 I P アドレス、送信元 I P アドレス、宛先ポート、送信元ポートを割り出すアドレステーブルや、I P のルーティングテーブルなどが挙げられる。次にペイロード分割部 ( 1 0 3 ) からフレームペイロード送信準備完了通知を受信したかの判断を行う ( ステップ S 1 2 0 4 : Y e s )。もし、フレームペイロード送信準備完了通知を受信したら、ステップ S 1 2 0 5 にてヘッダ情報の収集が終了しているかの判断を行う。ヘッダ情報の収集が終了した場合 ( ステップ S 1 2 0 5 : Y e s ) は、ステップ S 1 2 0 6 にて送信フレームに必要なヘッダを生成するヘッダ生成部にヘッダ生成指示及びヘッダ生成情報を出力する。ヘッダ生成指示を出力後はステップ S 1 2 0 7 においてフレーム個数分のヘッダ生成指示を行ったかの判断を行う。もし、フレーム個数分のヘッダ生成指示を行った ( ステップ S 1 2 0 7 : Y e s ) のなら、ステップ S 1 2 0 8 においてステップ S 1 2 0 6 で作成を指示したヘッダのヘッダ送信完了通知をヘッダ / ペイロード合成部 ( 1 2 0 ) から受信するのを待ち、ヘッダ送信完了通知を受信したら 1 つのデータグラムに対する全てのフレームのヘッダが送信されたとしてプロトコル制御を終了する。ステップ S 1 2 0 7 においてもしフレーム個数分のヘッダ生成指示を行っていないのならば ( S 1 2 0 7 : N o )、まだ送信すべきフレームを生成しなければならぬとして、ステップ S 1 2 0 9 においてステップ S 1 2 0 6 で作成を指示したヘッダのヘッダ送信完了通知をヘッダ / ペイロード合成部 ( 1 2 0 ) から受信するのを待ち、ヘッダ送信完了通知を受信したら次のフレームのヘッダ作成を開始する ( ステップ S 1 2 1 0 )。そしてステップ S 1 2 0 3 に戻る。

20

30

40

## 【 0 0 2 7 】

50

## [ TCPヘッダ生成部 ]

TCPヘッダ生成部(116)ではプロトコル制御部(115)からのヘッダ生成指示及びヘッダ生成情報に基づきヘッダ生成を開始し、ヘッダ生成完了後にヘッダ/ペイロード合成部(120)にTCPヘッダ生成完了通知(126)を行う。TCPヘッダ生成指示をプロトコル制御部(115)から受信したら、プロトコル制御部(115)からTCPヘッダ情報を取得する。TCPヘッダ情報は送信元ポート番号、宛先ポート番号、シーケンス番号、確認応答番号、ヘッダ長、フラグビット、ウィンドウサイズ、緊急ポインタ及び擬似ヘッダ用の宛先IPアドレス、送信元IPアドレス、セグメント長(TCPヘッダ長+ペイロード長)である。取得したヘッダ生成情報から擬似ヘッダとチェックサム以外のTCPヘッダを作成する。次にフレームごとのペイロードサムをペイロード分割部(103)より受信し、擬似ヘッダとチェックサム以外のTCPヘッダとペイロードサムからTCPのチェックサムを算出し、チェックサムをTCPヘッダのチェックサムフィールドに設定し、TCPヘッダの生成を完了する。TCPヘッダの生成が完了すると、ヘッダ/ペイロード合成部(120)にTCPヘッダ生成完了通知を行い、TCPヘッダ生成を終了し、プロトコル制御部(115)より次のヘッダ生成指示があるまで待機する。TCPヘッダ生成部(116)で作成するヘッダのフォーマットを図6に、擬似ヘッダフォーマットを図7に示す。

10

## 【0028】

また、TCPヘッダ生成部(116)はヘッダ/ペイロード合成部(120)からの出力要求(126)を受信すると、生成したTCPヘッダをヘッダ/ペイロード合成部(120)に出力(130)する。

20

## 【0029】

## [ UDPヘッダ生成部 ]

UDPヘッダ生成部(117)ではプロトコル制御部(115)からのヘッダ生成指示及びヘッダ生成情報に基づきヘッダ生成を開始し、ヘッダ生成完了後にヘッダ/ペイロード合成部(120)にUDPヘッダ生成完了通知(127)を行う。UDPヘッダ生成指示をプロトコル制御部(115)から受信したら、プロトコル制御部(115)からUDPヘッダ情報を取得する。UDPヘッダ情報は、送信元ポート番号、宛先ポート番号、UDPデータ長である。取得したヘッダ生成情報から擬似ヘッダとチェックサム以外のUDPヘッダを作成する。次にペイロードサムをペイロード分割部(103)より受信したかの判断を行い、もし受信していれば擬似ヘッダとチェックサム以外のUDPヘッダとペイロードサムの合計からUDPチェックサムを算出し、チェックサムをUDPヘッダのチェックサムフィールドに設定し、UDPヘッダの生成を完了する。UDPヘッダの生成が完了すると、ヘッダ/ペイロード合成部(120)にUDPヘッダ生成完了通知を行い、UDPヘッダ生成を終了する。

30

## 【0030】

ただし、UDPヘッダのチェックサムはオプションでの対応となっているため、UDPチェックサムを行わない設定にした場合は、プロトコル制御部(115)よりUDPヘッダ情報(送信元ポート番号、宛先ポート番号、UDPデータ長)を取得した時点で、UDPチェックサムフィールドに0を設定し、UDPヘッダ生成完了通知をヘッダ/ペイロード合成部(120)に出力する。

40

## 【0031】

UDPヘッダ生成部で作成するヘッダのフォーマットを図8に、擬似ヘッダフォーマットを図9に示す。

## 【0032】

また、UDPヘッダ生成部(117)はヘッダ/ペイロード合成部(120)からの出力要求(127)を受信すると、生成したUDPヘッダをヘッダ/ペイロード合成部(120)に出力(131)する。

## 【0033】

## [ IPヘッダ生成部 ]

50



IPヘッダ生成部(118)ではプロトコル制御部(115)からのヘッダ生成指示及びヘッダ生成情報に基づきヘッダ生成を開始し、ヘッダ生成完了後にヘッダ/ペイロード合成部(120)にIPヘッダ生成完了通知(128)を行う。IPヘッダ生成指示をプロトコル制御部(115)から受信したら、プロトコル制御部(115)からIPヘッダ情報を取得する。IPヘッダ情報はバージョン、ヘッダ長、TOS、全長、識別子、フラグ、フラグメントオフセット、TTL、プロトコル、送信元IPアドレス、宛先IPアドレスである。これらの情報からIPヘッダのヘッダチェックサムを算出してIPヘッダを完成させ、ヘッダ/ペイロード合成部(120)にIPヘッダ生成完了通知を行う。IPヘッダ生成部で作成するヘッダのフォーマットを図10に示す。

【0034】

10

また、IPヘッダ生成部(118)はヘッダ/ペイロード合成部(120)からの出力要求(128)を受信すると、生成した順番にIPヘッダをヘッダ/ペイロード合成部(120)に出力(132)する。

【0035】

[MACヘッダ生成部]

MACヘッダ生成部(119)ではプロトコル制御部(115)からのヘッダ生成指示及びヘッダ生成情報に基づきヘッダ生成を開始し、ヘッダ生成完了後にヘッダ/ペイロード合成部(120)にMACヘッダ生成完了通知(129)を行う。MACヘッダ生成部(119)はプロトコル制御部(115)からMACヘッダ生成指示を受信したら、MACヘッダ情報を取得する。MACヘッダ生成情報は、ヘッダ生成数、送信元MACアドレス、宛先MACアドレス、タイプレングスである。これらの情報からMACヘッダを完成させ、ヘッダ/ペイロード合成部(120)にMACヘッダ生成完了通知を行う。MACヘッダ生成部で作成するMACヘッダのフォーマットを図11に示す。

20

【0036】

また、MACヘッダ生成部(119)はヘッダ/ペイロード合成部(120)からの出力要求(129)を受信すると、生成したMACヘッダをヘッダ/ペイロード合成部(120)に出力(133)する。

【0037】

[ヘッダ/ペイロード合成部]

ヘッダ/ペイロード合成部(120)はフレーム個数フレーム長算出部(102)からのフレーム個数から生成すべきヘッダの種類とその個数を割り出す。次に、各プロトコルヘッダ生成部からのヘッダ完了通知により、送信フレームに必要なヘッダが完了していることを認識し、TCP/IPの場合はMACヘッダ生成部(119)、IPヘッダ生成部(118)、TCPヘッダ生成部(116)、ペイロード読み出し部(105)の順に出力要求を行い、ヘッダ及びペイロードを読み出してフレームとして合成し、フレーム送出部(121)に出力する。UDP/IPの場合は先頭フレームはMACヘッダ生成部(119)、IPヘッダ生成部(118)、UDPヘッダ生成部(117)、ペイロード読み出し部(105)の順に出力要求を行い、中間、最終フレームはMACヘッダ生成部(119)、IPヘッダ生成部(118)、ペイロード読み出し部(105)の順に出力要求を行い、ヘッダ及びペイロードを読み出してフレームとして合成しフレーム送出部(121)に出力する。ペイロードを持たないフレームを送信する場合はヘッダのみ読み出して、フレームとして合成し、フレーム送出部(121)に出力する。ヘッダ/ペイロード合成部(120)は、ヘッダをフレーム送出部(121)に出力した時点で、プロトコル制御部(115)にヘッダ送信完了(137)を出力する。

30

40

【0038】

[フレーム送出部]

フレーム送出(送信)部(121)では、ヘッダ/ペイロード合成部(120)から出力されるフレームを通信制御装置外に送信する機能を有する。IEEE802.3に準拠し、プリアンプルの付加、フレームパディング、CRCの生成等を行う。

【0039】

50

## &lt; T C Pセグメントの送信例 &gt;

T C P / I Pのフレーム送信時に送信データグラムが3つのセグメントに分割された場合の例を図5に示す。アプリケーションデバイス(101)からの送信指示及び送信データ長(1301)を受けてフレーム個数フレーム長算出部(102)は、フレーム個数(3個)、先頭フレームのペイロード長(PL0の長さ)、中間フレームのペイロード長(PL1の長さ)、最終フレームのペイロード長(PL2の長さ)をプロトコル制御部(115)及びヘッダ/ペイロード合成部(120)に出力(1302)する。プロトコル制御部(115)ではプロトコルタイプをT C Pであると判断し、T C P、I P、M A Cのヘッダ情報の収集を行う。またペイロード分割部(103)においてペイロードをT Xメモリ(104)に書き込んでいき、1フレーム分(1 T C Pセグメント分)書き込みが完了すると、プロトコル制御部(115)に対してフレームペイロード送信準備完了(1303)を出力する。プロトコル制御部(115)はフレームペイロード送信準備完了(1303)を受信するとデータグラムをセグメント化した先頭フレームのT C P、I P、M A Cのヘッダを生成するためのヘッダ情報収集が終了しているかを判断し、終了していればT C Pヘッダ生成部(116)にヘッダ生成指示及びヘッダ情報(1304)を、I Pヘッダ生成部(118)にヘッダ生成指示及びヘッダ情報(1305)を、M A Cヘッダ生成部(119)にヘッダ生成指示及びヘッダ情報(1306)を同時に出力する。ヘッダ生成指示を受信したT C Pヘッダ生成部(116)はプロトコル制御部(115)からのヘッダ情報とペイロード分割部(103)からのペイロードサム(1307)よりT C Pヘッダを生成し、ヘッダ/ペイロード合成部(120)にT C Pヘッダ生成完了(1308)を出力する。ヘッダ生成指示を受信したI Pヘッダ生成部はプロトコル制御部(115)からのヘッダ情報からI Pヘッダを生成し、ヘッダ/ペイロード合成部(120)にI Pヘッダ生成完了(1309)を出力する。ヘッダ生成指示を受信したM A Cヘッダ生成部(119)はプロトコル制御部(115)からのヘッダ情報からM A Cヘッダを生成し、ヘッダ/ペイロード合成部(120)にM A Cヘッダ生成完了(1310)をフレーム個数分出力する。ヘッダ/ペイロード合成部(120)はフレーム個数フレーム長算出部(102)から受信したフレーム個数(1302)から3つにセグメント化されたフレームの先頭フレームにはT C Pヘッダ、I Pヘッダ、M A Cヘッダが必要であることを認識しており、T C Pヘッダ生成完了(1308)、I Pヘッダ生成完了(1309)、M A Cヘッダ生成完了(1310)の全てを受信するとフレームを送信可能な状態であると判断する。ヘッダ/ペイロード合成部(120)はM A Cヘッダ、I Pヘッダ、T C Pヘッダ、ペイロードと読み出していきフレーム送出部(121)に先頭フレームを出力する。また、M A C、I P、T C Pヘッダをフレーム送出部(121)に出力した時点でプロトコル制御部(115)にヘッダ送信完了(1311)を出力する。

## 【0040】

プロトコル制御部(115)はヘッダ送信完了(1311)を受信すると次のフレーム、つまりここでは3つにセグメント化されたフレームの中間フレームのヘッダ情報収集を行う。中間フレームに必要なヘッダは先頭と同様にT C P、I P、M A Cヘッダの情報収集を行う。ペイロード分割部(103)からの次のフレームペイロード送信準備完了(1312)を受信すると、中間フレームのT C P、I P、M A Cのヘッダを生成するためのヘッダ情報収集が終了しているかを判断し、終了していればT C Pヘッダ生成部(116)にヘッダ生成指示及びヘッダ情報(1313)を、I Pヘッダ生成部(118)にヘッダ生成指示及びヘッダ情報(1314)を同時に出力する。ヘッダ生成指示を受信したT C Pヘッダ生成部(116)はプロトコル制御部(115)からのヘッダ情報とペイロード分割部(103)からのペイロードサム(1315)よりT C Pヘッダを生成し、ヘッダ/ペイロード合成部(120)にT C Pヘッダ生成完了(1316)を出力する。ヘッダ生成指示を受信したI Pヘッダ生成部(118)はプロトコル制御部(115)からのヘッダ情報からI Pヘッダを生成し、ヘッダ/ペイロード合成部(120)にI Pヘッダ生成完了(1317)を出力する。

## 【0041】

ヘッダ/ペイロード合成部(120)はフレーム個数フレーム長算出部(102)から受信したフレーム個数(1302)から3つにセグメント化されたフレームの中間フレームにはTCPヘッダ、IPヘッダ、MACヘッダが必要であることを認識しており、TCPヘッダ生成完了(1316)、IPヘッダ生成完了(1317)を受信するとフレームを送信可能な状態であると判断する。MACヘッダ生成完了については先頭フレームの時にフレーム個数分出力されているため、ヘッダ/ペイロード合成部(120)はこの時点で中間フレームのMACヘッダ生成が完了していると判断している。フレームが送信可能な状態であると判断したヘッダ/ペイロード合成部(120)はMACヘッダ、IPヘッダ、TCPヘッダ、ペイロードと読み出していきフレーム送出部(121)に中間フレームを出力する。また、MAC、IP、TCPヘッダをフレーム送出部に出力した時点でプロトコル制御部(115)にヘッダ送信完了(1320)を出力する。

10

**【0042】**

プロトコル制御部(115)はヘッダ送信完了(1320)を受信すると次のフレーム、つまりここでは3つにセグメント化されたフレームの最終フレームのヘッダ情報収集を行う。最終フレームに必要なヘッダは先頭、中間と同様にTCP、IP、MACヘッダの情報収集を行う。ペイロード分割部(103)からの次のフレームペイロード送信準備完了(1318)をすでに受信しているため、TCP、IP、MACのヘッダを生成するためのヘッダ情報収集が終了し次第、TCPヘッダ生成部(116)にヘッダ生成指示及びヘッダ情報(1321)を、IPヘッダ生成部(118)にヘッダ生成指示及びヘッダ情報(1322)を同時に出力する。ヘッダ生成指示を受信したTCPヘッダ生成部(116)はプロトコル制御部(115)からのヘッダ情報とペイロード分割部(103)からのペイロードサム(1319)よりTCPヘッダを生成し、ヘッダ/ペイロード合成部(120)にTCPヘッダ生成完了(1323)を出力する。ヘッダ生成指示を受信したIPヘッダ生成部(118)はプロトコル制御部(115)からのヘッダ情報からIPヘッダを生成し、ヘッダ/ペイロード合成部(120)にIPヘッダ生成完了(1324)を出力する。

20

**【0043】**

ヘッダ/ペイロード合成部(120)はフレーム個数フレーム長算出部(102)から受信したフレーム個数(1302)から3つにセグメント化されたフレームの最終フレームにはTCPヘッダ、IPヘッダ、MACヘッダが必要であることを認識しており、TCPヘッダ生成完了(1323)、IPヘッダ生成完了(1324)を受信するとフレームを送信可能な状態であると判断する。MACヘッダ生成完了については先頭フレームの時にフレーム個数分出力されているため、ヘッダ/ペイロード合成部(120)はこの時点で中間フレームのMACヘッダ生成が完了していると判断している。フレームが送信可能な状態であると判断したヘッダ/ペイロード合成部(120)はMACヘッダ、IPヘッダ、TCPヘッダ、ペイロードと読み出していきフレーム送出部(121)に最終フレームを出力する。また、MAC、IP、TCPヘッダをフレーム送出部に出力した時点でプロトコル制御部(115)にヘッダ送信完了(1325)を出力する。

30

**【0044】**

以上のように、本実施形態によれば、アプリケーションデバイス(101)から任意の長さの送信データを受信し、サポートするデータリンク層に適切なフレームサイズでフレーム出力する通信制御装置が提供される。プロトコル制御部(101)は、フレーム生成に必要なヘッダ生成情報を収集し、ヘッダ生成指示及びヘッダ生成情報をフレーム毎に出力する。ヘッダ生成手段(116～119)は、プロトコル制御部(115)からヘッダ生成指示及びヘッダ生成情報を入力すると、ヘッダを1ヘッダ毎に生成し、ヘッダ生成完了通知を出力する。ヘッダ/ペイロード合成部(120)は、ヘッダ生成部(116～119)からヘッダ生成完了通知を入力すると、生成されたヘッダ及びフレーム単位のペイロードをフレームとして合成する。プロトコル制御部(115)は、所望のヘッダの生成指示をヘッダ生成部(116～119)に出力することができる。

40

**【0045】**

50

T Xメモリ(104)は、フレーム単位のペイロードを記憶するためのメモリである。プロトコル制御部(115)は、ヘッダ/ペイロード合成部(120)が1つ前のフレームとして合成したヘッダを出力した後、かつ次のペイロードがT Xメモリ(104)に書き込まれた後に、次のフレームのヘッダ生成指示をヘッダ生成部(116~119)に出力する。

【0046】

本実施形態によれば、リンク層に依存するMTUに対して、アプリケーションの転送単位が大きくなるにつれて分割後に発生するフレームが増加しても、フレーム単位でヘッダを生成し送信することで、フレームに付加しなければならないヘッダを書き込むためのメモリ領域を確保する必要がなくなる。

10

【0047】

なお、上記実施形態は、何れも本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその技術思想、またはその主要な特徴から逸脱することなく、様々な形で実施することができる。

【図面の簡単な説明】

【0048】

【図1】本発明の実施形態による通信制御装置の構成図である。

【図2】ペイロード長とヘッダ数算出の流れを示すフローチャートである。

【図3】T Xメモリを示す図である。

20

【図4】プロトコル制御部のフローチャートである。

【図5】シーケンス1を示す図である。

【図6】T C Pヘッダフォーマットを示す図である。

【図7】T C P擬似ヘッダフォーマットを示す図である。

【図8】U D Pヘッダフォーマットを示す図である。

【図9】U D P擬似ヘッダフォーマットを示す図である。

【図10】I Pヘッダフォーマットを示す図である。

【図11】M A Cヘッダフォーマットを示す図である。

【符号の説明】

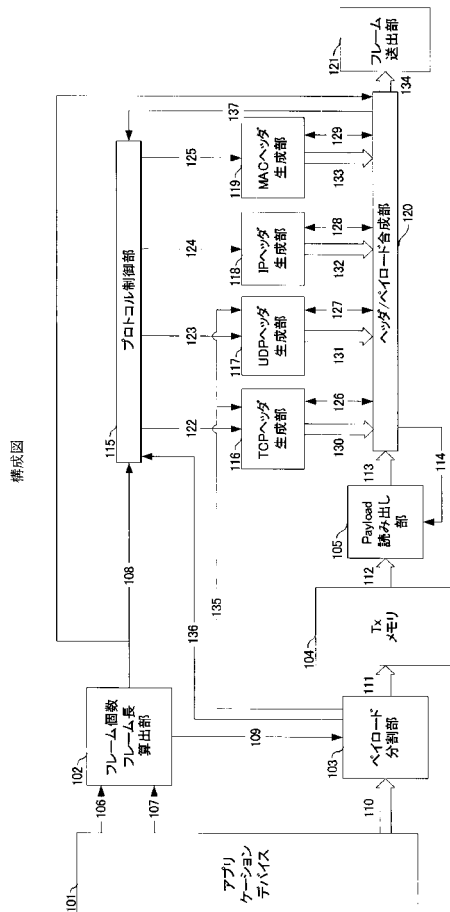
【0049】

30

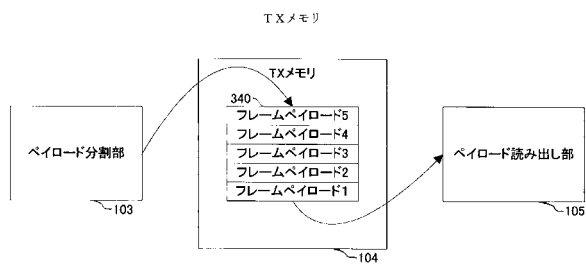
- 101 アプリケーションデバイス
- 102 フレーム個数フレーム長算出部
- 103 ペイロード分割部
- 104 T Xメモリ
- 105 ペイロード読み出し部
- 115 プロトコル制御部
- 116 T C Pヘッダ生成部
- 117 U D Pヘッダ生成部
- 118 I Pヘッダ生成部
- 119 M A Cヘッダ生成部
- 120 ヘッダ/ペイロード合成部
- 121 フレーム送出部

40

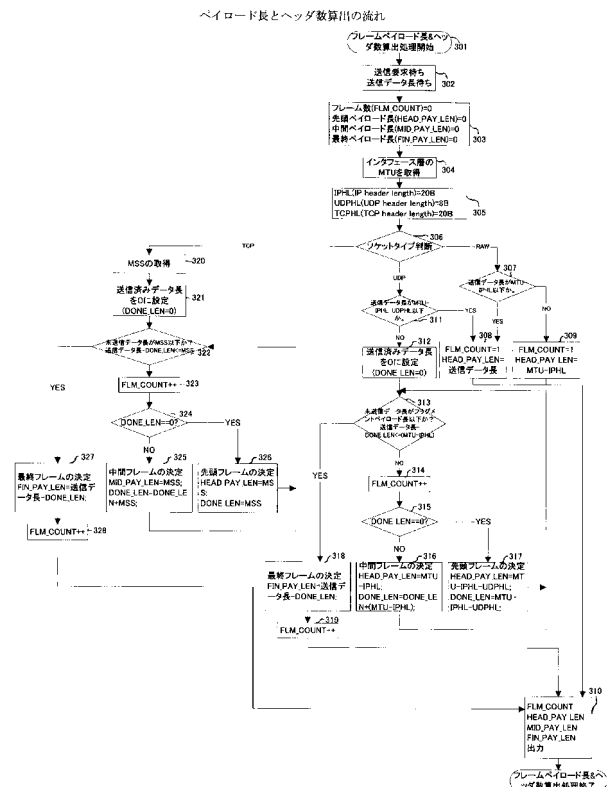
【 図 1 】



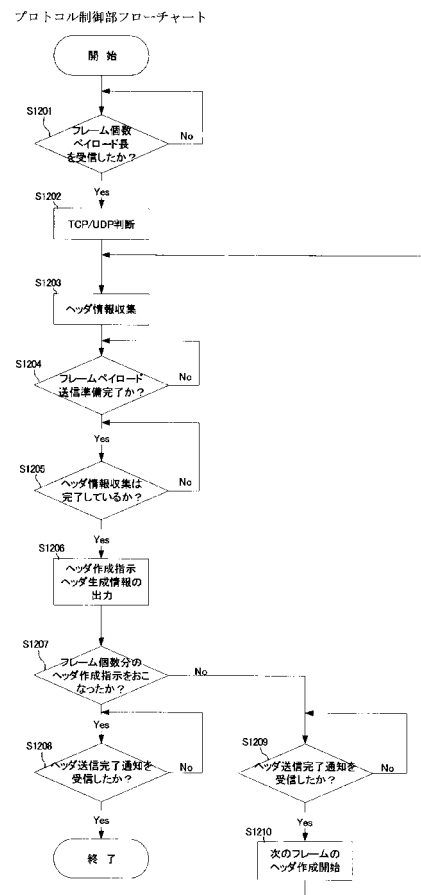
【圖 3】



【 図 2 】



【圖 4】





---

フロントページの続き

審査官 阿部 弘

- (56)参考文献 特開2000-253054(JP,A)  
特開2003-046566(JP,A)  
特開平06-069957(JP,A)  
特開2001-016226(JP,A)  
特開2004-215203(JP,A)  
特開平11-317762(JP,A)  
特開昭63-099651(JP,A)  
特開2004-241872(JP,A)  
特開2001-339462(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
H04L 29/10