

ネットワークプロセッサを適用したパケットカプセル化方式の検討

日高 稔[†] 亀山 達也[†] 長坂 充[†] 中嶋 淳[‡] 太田 琢[‡]

[†](株)日立製作所 中央研究所 〒185-8601 東京都国分寺市東恋ヶ窪 1-280

[‡](株)日立コミュニケーションテクノロジー キャリアネットワーク事業部 〒244-8567 神奈川県横浜市戸塚区戸塚町 216

E-mail: † {hidaka, kameyama, nagasaka}@crl.hitachi.co.jp

‡ {jun_nakajima, migaku_ohta}@hitachi-com.co.jp

あらまし ネットワークプロセッサは、ネットワーク機器の高性能化と多機能化を両立するデバイスとして注目されている。報告者らは、パケットのカプセル化という汎用 CPU により比較的低速度で行っている処理を、パケット毎に処理が可能という観点から、ネットワークプロセッサを適用して高速化することを検討した。検討した方式はアクセスゲートウェイに適用した。アクセスゲートウェイは、加入者が Internet 等のバックボーンネットワークにアクセスするためのアクセスネットワークを構成するノードであり、ネットワークレイヤプロトコルとして IPv4 と IPv6 に対応するネットワーク構成を可能とすることで、次世代アクセスネットワークを構成するノードとする。

キーワード ネットワークプロセッサ、アクセスゲートウェイ、アクセスネットワーク、PPP、L2TP、IPv6

Packet Encapsulation Method with Network Processor

Minoru HIDAKA[†] Tatsuya KAMEYAMA[†] Mitsuru NAGASAKA[†] Jun NAKAJIMA[‡]
and Migaku OHTA[‡]

[†] Hitachi, Ltd., Central Research Lab 1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo, 185-8601 Japan

[‡] Hitachi Communication Technologies, Carrier Network Systems Division, 216 Totsuka, Totsuka-ku, Yokohama-shi, Kanagawa, 244-8567 Japan

E-mail: † {hidaka, kameyama, nagasaka}@crl.hitachi.co.jp

‡ {jun_nakajima, migaku_ohta}@hitachi-com.co.jp

Abstract Improvement in the speed of an access gateway and multi-functionalization are demanded with the high speed network for the Internet and an access network. Encapsulation is one of the basic functions for the access gateway and realized by relatively slow general purpose CPU so far. In this paper, the network processor is newly applied to the access gateway to enhance packet-processing speed and to realize advanced features. Functions to compose the access network for both of IPv4 and IPv6 protocols are also realized.

Keyword Network Processor, Access Gateway, Access Network, PPP, L2TP, IPv6

1. はじめに

近年のインターネットの普及により、ルータを中心に構成されたパケット交換網のトラフィックが増大している。その一端を担っているのが ADSL 等の高速アクセス接続サービスの普及であり、一加入者あたりのアクセス帯域が 1.5Mbit/s 以上となり、最近では 12Mbit/s 程度にブロードバンド化した加入者が増加し

ている。それに伴い、アクセス系ネットワーク機器の高速、大容量、低価格が加速されている。同時にアクセスネットワークでは、PPP などのユーザー認証、L2TP などのトンネルプロトコル、IPSec などの暗号化など、高速化と同時に多機能化が求められている。多機能化を市場の要求に対応してすばやく行うためにネットワークプロセッサ (NP) が注目されているが、

報告者は NP を適用したパケットスイッチ開発と NP 用プログラム開発を通じて、その利用方法の検討を行っている。特に NP を適用する対象装置は、ネットワーク内において、以下に述べる位置の装置である。

2. ネットワーク構成

インターネットへの接続は、図 1 のように、加入者からの接続をインターネットサービスプロバイダ (ISP) が取り纏めて IP 網に接続する BAS による接続形態と、加入者が一旦ホールセール事業者の提供する

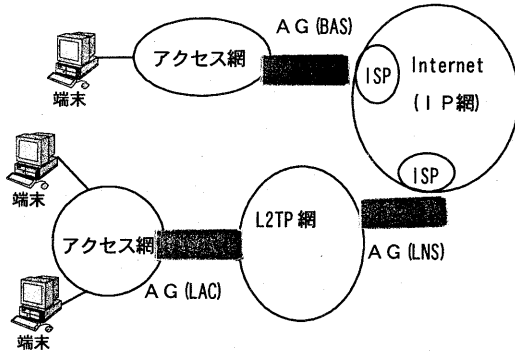


図 1 ネットワーク構成

网上的アクセス集線装置 (LAC) に接続し、L2 (データリンク層) トンネルを経由してネットワークサーバ (LNS) に接続し、ISP を介して IP 網に接続する LAC-LNS 接続形態がある。

本報告では、各網を次のように定義する。アクセス網は加入者宅に直接接続したアクセス回線を集約する網とする。Internet は、通信プロトコルとして Internet Protocol (IP) を採用した網とする。L2TP 網はホールセール事業者がアクセス網と IP 網とをトンネル接続する際に用いる転送網とする。本報告では、アクセスネットワークにおいて、加入者からの接続を処理するゲートウェイをアクセスゲートウェイ (AG) と呼ぶ。

アクセス網、L2TP 網では、どちらの場合も接続回線の帯域増加が要求されている。

3. ネットワークノード

3.1. パケット処理方式

本報告ではネットワークを構成するノードとしてのパケットスイッチについて検討する。

パケットスイッチの動作を簡単に説明すると以下の手順になる。

(1) 回線部から入力されるパケットをメモリに蓄積し、出力回線への出力決定を待つ。

(2) パケットヘッダ部の情報を元に、宛先を経路検索し、出力回線を決定する。

(3) スイッチにより、出力側の回線部にパケットを転送する。

(4) 出力に必要なパケットデータを書換えを行う。

(5) 書換えの終わったパケットデータをメモリから読出し、出力回線に出力する。

回線インタフェースの高速化、スイッチ性能の向上により、(1) (3) (5) の処理性能が向上したことで、パケットスイッチ全体の処理性能は、(2) の宛先経路検索処理の性能が支配的になっている。

ソフト処理では出力回線決定のための宛先経路検索処理が低速であったが、ハード処理では、検索処理を ASIC 等で構成した専用論理回路により高速処理することで、そのパケット処理を高速化している。

ネットワークプロセッサでは、パケット処理はソフト処理であるが、経路表の検索処理を専用のハードウェアで高速化する。また、パケット毎に異なるハードウェアを割当てて物理的に同時動作させることで、パケット処理を高速化する。

ソフト処理のメリットは、パケット形式、パケットの処理プロトコルの変更、特にトンネルインタフェース方式毎のパケットのカプセル化方式の変更に対して、柔軟に対応できる点である。

逆に、ハード処理では、専用論理回路にハード化された論理以外の処理を簡単に追加することができないという点がデメリットとされている。

また、カプセル化処理としては、(4) のデータ書換え処理時間が性能に大きな影響を及ぼす。

3.2. ハードウェア構成

パケットスイッチのハードウェア構成を図 2 に示す。一般に、パケットスイッチは、複数の回線部 (LIF) と宛先経路検索部 (RP) とスイッチと制御部 (CNT) から構成されている。

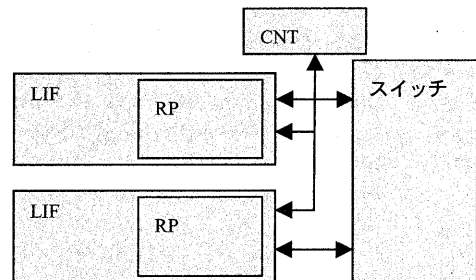


図 2 パケットスイッチのハードウェア構成

RPには、すでに述べたように、1. 専用論理回路(ASIC)、2. 汎用CPU、3. ネットワークプロセッサ(NP)が用いられる。

RPのハードウェア構成として、図3にNPを適用した場合であるNetwork Processor Card(NPC)を示す。NPCには、NPとNPC制御CPUと接続回路を搭載している。この構成例では、NPC制御CPUを構成要素としたが、装置の制御部(CNT)が、NPCの制御を兼ねる構成も考えられる。

NPは、コアCPUと複数のマイクロエンジン(ME)から構成される。コアCPUは、NPに埋め込まれた汎用CPUである。MEは、パケット処理に特化したCPUである。

コアCPUは、NPCの制御CPUあるいは装置の制御部と連携して、パケットスイッチが行う制御パケットの処理を担当している。

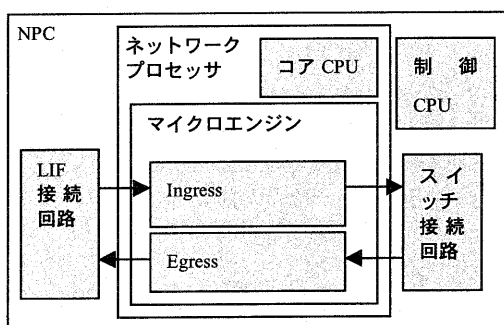


図3 NPCのハードウェア構成

3.3. マイクロエンジンとコアCPUの処理分担

NPのマイクロエンジン(ME)は、パケット毎にパケットヘッダ解析とパケット転送を繰り返す処理を効率的に実行することに適している。一方、パケット間に関連した情報を継続的に保持することが必要なPPPのセッション状態管理などの処理は、メモリ量や命令領域の大きさ等の計算機資源の制約により、MEでの処理の実行に適していない。

後者ようなパケット間情報の処理は、NP内に埋め込まれた汎用CPUや外部汎用CPUにより実行する。

つまり、MEでは、データパケットを処理し、経路表に示された出方路に転送する。経路表設定処理やプロトコルに従い制御情報を管理する処理を汎用CPUで行う。また、プロトコルにより分類した制御パケットを汎用CPU上の処理プロセスと送受信するため、ME-コアCPU間で転送する。

MEでは、以下の処理を分担する。

- (1) データパケット転送処理
 - (2) 制御パケットのコアCPUへの転送処理
- コアCPUでは以下の処理を分担する。
- (1) 制御パケットの処理
 - (2) データパケットの方路決定
 - (3) 回線インタフェース制御情報設定

3.4. NPのAGへの適用

AGの適用形態は、次の3つの場合を想定している。

(1) BAS (Broadband Access Server)

BASは、加入者からのPPP接続を終端処理し、主としてPPPフレームを転送する網となっているアクセス網とIP網との接続を行う。

(2) LAC (L2TP Access Concentrator)

アクセス網とIP網の間をL2TPを用いてトンネルにより接続する場合に、アクセス網からの接続を集線する。L2TP網のレイヤ2トンネル終端点となる装置である。

(3) LNS (L2TP Network Server)

アクセス網とIP網の間をL2TPを用いてトンネルにより接続する場合に、IP網との接続を行う。L2TP網のレイヤ2トンネル終端点となる装置である。

AGでは、BAS、LAC、またはLNSの各適用形態によって異なるプロトコルとパケットフォーマットを処理する必要がある。汎用CPUを用いたソフトウェア処理による実装では、実行性能が十分とは考えられず、そこで、実行性能に期待できるNPを適用することにより、処理の柔軟性と高速化を両立する。

AGでは、Ethernet、ATMのLIFを収容可能である。また、レイヤ2トンネルインタフェースも備える。そのため、データリンク層のフレームフォーマットが多種類となり、その処理をNP部が行うことで対応した。

特にATMの場合は、PPPのフレームフォーマットにPPPoA、PPPoEoA、IPoAの3種類のフォーマットがあり、さらにVC多重化方式とLLC多重化方式があるが、両方式ともNPのソフト処理によりフォーマット毎にフレーム処理を行う。

4. ネットワークプロセッサの利点

4.1. カプセル化処理の柔軟性

AGでは、その適用形態により、パケット交換を行うレイヤが異なる。従って、宛先検索プロトコルの変更に対処することが求められる。

LACとして適用されるノードでは、パケットはデータリンクフレームとして交換される。

それに対し、BASあるいはLNSとして適用される

ノードでは、パケットはネットワークデータグラムとして交換される。AG では、ネットワーク層プロトコルとして IPv4 と IPv6 に対応した。

また AG は、LAC あるいは LNS に適用するために、PPP および L2TP に従ったトンネルインタフェースを備える。使用形態に応じ、つぎのカプセル化処理を行う必要がある。

ここでいうカプセル化とは、被カプセル化パケットの宛先検索の結果、宛先がトンネルインタフェースの場合にカプセル化ヘッダを付加する処理を行うことである。

デカプセル化とは、トンネルインタフェースに到着したカプセル化パケットからカプセル化ヘッダを削除する処理を行うことである。

(1)BAS 使用時の処理

a. アクセス網→IP 網

加入者端末から送信された PPP フレームからデカプセル化して取出した IP データグラムを IP 網へルーティングする。

b. IP 網→アクセス網

IP データグラムをルーティングし、宛先加入者の PPP トンネルインタフェースに送信する。この時 PPP フレームにカプセル化する。

(2)LAC 使用時の処理

a. アクセス網→L2TP 網

加入者端末から送信された PPP フレームを PPP 接続情報から宛先決定した L2TP トンネルへ送信する。PPP フレームを L2TP フォーマットにカプセル化し L2TP 網に転送する。

b. L2TP 網→アクセス網

L2TP トンネルからデカプセル化して取出した PPP フレームを宛先加入者の PPP トンネルに送信する。

(3)LNS 使用時の処理

a. L2TP 網→IP 網

L2TP トンネルからデカプセル化して取出した PPP フレームから、さらに IP データグラムをデカプセル化して取出し IP 網へルーティングする。

b. IP 網→L2TP 網

IP データグラムのルーティングを行い、転送経路が L2TP トンネル内の PPP トンネルの場合、IP データグラムを PPP フレームにカプセル化し、さらに L2TP パケットにカプセル化して L2TP トンネル転送を行う。

ネットワークプロセッサは、プログラミングが可能という観点から、機能追加が容易とされているが、

パケット処理性能のカタログ値は、プログラムステップ数が IP パケットのルーティングを行う場合の性能を元に表示されていることが多い。高機能化の為に複雑な処理をプログラミングする場合は、ステップ数が増大し、パケットあたりの処理時間が増加する。またプログラマビリティやプログラム可能な命令数なども考慮する必要がある。

4.2. カプセル化処理の高速化

L2TP トンネルインタフェースに対応したカプセル化処理の高速化を行うために、NP ハードウェアアシストを考慮した処理方式を検討した。

4.2.1 カプセル化

L2TP トンネルインタフェースでは、L2TP ヘッダの生成と、UDP へのカプセル化を行う。

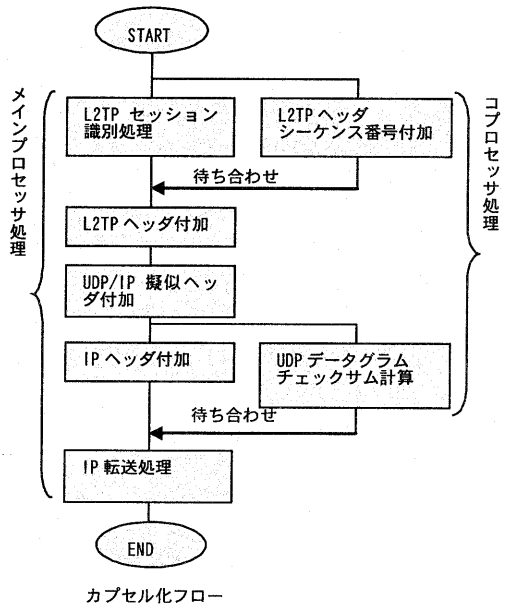


図 4 L2TP カプセル化処理流れ図

L2TP ヘッダの生成には、L2TP トンネルの識別情報のテーブルとシーケンス番号管理が必要である。L2TP シーケンス番号の管理には、カウンタを管理するコプロセッサを利用した。

さらに、UDP ヘッダと擬似 IP ヘッダを生成し、UDP チェックサムの計算を行う。チェックサム計算は、チェックサムコプロセッサを利用する。この処理は被カプセル化パケットが長いほどチェックサム計算に必要

な処理が増えるためにパケット処理時間が増加する。

カプセル化処理に必要なプログラムステップは、シミュレータによる見積もりによれば、IP 転送に対して、1.95 倍に増加した。

4.2.2 デカプセル化

汎用 CPU で処理される L2TP トンネルでは、デカプセル化は、IP ヘッダと UDP ヘッダのアドレスポートペアからソケットを特定し、一致したソケットからその UDP ペイロード部分を読み出した後、L2TP ヘッダの処理を行う。つまり、IP ヘッダ解析、UDP ヘッダ解析、L2TP ヘッダ解析の順であり、プロトコルスタックに従った順序で処理される。その結果として PPP フレームを取り出す。

ヘッダ解析をプロトコルスタックレイヤごとに順に処理する方式により NP のプログラミングを行った場合、NP のプログラム量が増加し、高速処理性能を損なうと考える。

NP では、テーブル検索コプロセッサを利用し、IP ヘッダ、UDP ヘッダ、L2TP ヘッダを同時にテーブルにより一致検索する方式を実装した。

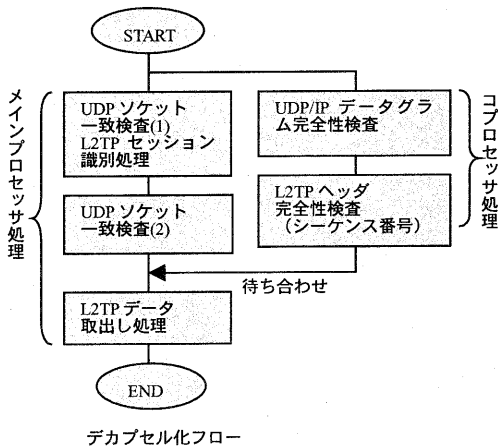


図 5 L2TP デカプセル化処理流れ図

NP におけるフレーム処理段階で、IP ヘッダ、UDP ヘッダ、L2TP ヘッダから、宛先 IP アドレス、宛先 UDP ポート番号、L2TP ヘッダ内のセッション番号とトンネル番号を取り出し、検索データベース内のエントリと一致検索する。一致したエントリがある場合は、そのパケットは L2TP トンネルから取り出すパケットの候補であるので、残りの特定項目、すなわち送信元 IP アドレス、送信元 UDP ポート番号の一致を確認する。

また、これに平行して、チェックサムコプロセッサによりパケットデータのチェックサム検査を行う。

デカプセル化処理に必要なプログラムステップは、シミュレータによる見積もりによれば、IP 転送に対して、1.34 倍に増加した。

5. パケット間に相関がある場合の問題

ネットワークプロセッサでは、パケット毎に処理を行うことを前提に、ハードウェア補助による高速化処理を行っている。しかし、トランスポート層より上位の層では、パケット間に相関がある処理がある。例えば L2TP で同じセッション ID とセッション毎に増加するシーケンス番号をヘッダ内に付加する処理、また同様にシーケンス番号検査する処理などがそれにあたる。ここで同じセッション ID を持つパケット群をパケットフローと呼ぶこととする。パケットフローを区別して管理するためには、パケットフロー毎にそのフローの継続期間中シーケンス番号を保持する為に、メモリなどの計算機資源が必要である。すでに述べたようにこのような処理は、ME による処理に適さないと考えている。しかし、例外として次の場合は、ME により処理することとした。これは、コア CPU などによるソフト処理がパケット処理に介在した場合、極度に処理速度が減じ、ME による処理速度高速化の妨げとなるためである。

- (1) IP 識別子の付加
- (2) L2TP データのシーケンス番号付加
- (3) L2TP データのシーケンス番号検査

これらは、パケットフロー毎に一連の番号を管理するという共通の方式により処理することが可能である。パケットフロー数を ME の利用可能な計算機資源のみで実装可能な範囲内とすることで対策した。

6. 装置機能仕様変更への対応

NP を搭載したことにより、同じハードウェア構成でプログラムを変更することで、異なる機能仕様を持った装置を構成することが可能である。AG の場合は、プログラムの機能変更により、IPv4-IPv6 変換機能を持つ装置を構成することができる。

7. 評価実験

NP におけるパケットのカプセル化処理性能を評価するために、AG システムを使用してパケット転送実験を行った。AG のパケット処理能力として、(1) IPv4 ノードとしてのスループット性能、(2) IPv6 ノードとしてのスループット性能、(3) BAS としてのスルー

プット性能 (4) LAC としてのスループット性能、(5) LNS としてのスループット性能を表 1 に示す。スループットは、NP 一個当たりの値である。イーサネットフレーム長を 64byte, 512byte, 1518byte とした。

AG の NP 一個あたりの 64B イーサネットフレームの目標スループットは、IP 転送で 1Gbit/s、L2TP 転送 0.5Gbit/s である。1 ギガイーサネット回線に対応するパケット処理性能を達成している。

表 1 AG のパケット処理性能

項番	測定項目	方向	スループット (Mbit/s/NP)					
			64byte/frame		512byte/frame		1518byte/frame	
			計算値	実測値	計算値	実測値	計算値	実測値
1	IPv4転送	IP→IP	1351	1311	*3072	2948	*3064	2787
2	IPv6転送	IP→IP	1354	1309	3072	2897	3064	2745
3	BAS性能	PPP→IP	1052	1127	3072	2847	3064	2764
		IP→PPP	1245	1351	3072	2940	3064	2740
5	LAC性能	PPP→L2TP	972	605	3070	1771	3063	1702
		L2TP→PPP	808	797	3070	2742	2993	2779
7	LNS性能	L2TP→IP	635	742	3070	2677	2985	2726
		IP→L2TP	911	643	3070	1807	3063	1705

8. さいごに

ネットワークプロセッサを適用することで、高速にカプセル化可能となる方式を検討した。

L2TP プロトコルに従ったカプセル化は、

(1) ネットワークプロセッサに実装された複数の ME によるパラレル処理とコプロセッサの利用による高速化により、1 ギガイーサネットをサポートできる。

(2) UDP プロトコル処理におけるチェックサム計算処理に課題があるが、汎用 CPU 処理よりもパケットのカプセル化処理性能が高い。

また、ネットワーク層プロトコルとして、IPv4, IPv6 に対応した。

さらに、ネットワークプロセッサを適用したハードウェア構成とすることにより、プログラムを変更することで、IPv4-IPv6 変換機能を追加することが可能である。

アクセス網は、今後、IPv4 と IPv6 のネットワークが混在した状況になるものと考えている。現在、両プロトコル間のトランスレートを行うことができるゲートウェイの導入が検討されているが、AG により、IPv6 環境への移行環境を構築し、移行方式を検討していく。

文 献

[1] IBM, IBM PowerNP NP4GS3 Network Processor Datasheet, 2002.

[2] 日高, 亀山他, アクセスゲートウェイへのネットワークプロセッサの適用, 信学技報 Vol.102 No.53 NS2002-36, May. 2002.
 [3] 総務省, 情報通信白書平成 13 年度版, Jul. 2001.
 [4] T. A. Maufer, IP ルーティング徹底解説, 生田りえ子(訳), pp.229-256, 日経 BP 社, Oct.2000.
 [5] G. R. Write, and W. R. Stevens, TCP/IP Illustrated, Volume2, pp.755-794, Addison-Wesley, 1995.
 [6] W. R. Stevens, UNIX Network Programming Networking Volume1 APIs: Sockets and XTI, pp.261-272, Prentice-Hall, 1998.
 [7] W. R. Cheswick, S. M. Bellovin, Firewalls and Internet Security Repelling the Wily Hacker, pp226-231, Addison-Wesley, 1994.