

第 VII 部

ネットワークプロセッサを用いた アプリケーション開発

第7部

ネットワークプロセッサを用いたアプリケーション開発

第1章 はじめに

ネットワークプロセッサ(以下NPと略記)とは、ネットワークトラフィックを処理するための専用プロセッサである。

NPの特徴の1つとして、ACICなどとは異なり、ソフトウェアによるプログラムが可能であるという特徴がある。NPの基本的構成は、パケットの転送やコピー、廃棄など単純だが短時間で処理が要求されるパケットの処理を行う Processing Engine(以下PEと略記)と、例外処理やルーティングテーブルの変更など複雑だが高速処理を要求しないパケット処理を行うコアプロセッサの2つにフォワーディングプレーンとコントロールプレーンを分けている。NPではPE、コアプロセッサそれぞれに対し、ソフトウェアでプログラム可能な制御用コードをロードしてパケット処理を行う。

近年NPが注目されている理由は、NPの持つ処理能力の高さとソフトウェアによりプログラム可能という特徴のためである。NPには、ハードウェア化における開発期間の短縮、中継ノードにおける柔軟なサービスの提供、エンドホストにおけるTCP/IPやアプリケーション層での処理性能の向上など、さまざまな面で期待を集めている。しかし、NPの処理性能や可能性はまだ未知数な部分が多く、NPによって実現可能なこと、NPを用いることで解決できる問題、NPの特性を生かしたアプリケーションが何であるかといったことなどは明らかになっていない。また、NPを用いたアプリケーション開発において、PE上で動かすプログラムには、その性質上、非常に低レベルなプログラミングが要求される。開発環境や開発に用いる言語、用いるNPの特性を熟知する必要があるが、若い分野であるため参考となるドキュメントも乏しく、熟知するまでには多くの労力と時間を費やす。そのため、参考となるドキュメントの多いC言語などを用いた汎用プロセッサ上

でのソフトウェアプログラミングよりも開発のノウハウを覚えるまでに時間がかかり、NPを研究の実装手段として用いる際の大きな障壁となっている。

そこでWIDE Project内において、NPに関する技術調査・研究を目的として2003年3月にネットワークプロセッサワーキンググループ(以下NPWGと略記)が設立された。NPWGの活動は、さまざまな観点から評価・検討し、NPの持つ可能性または技術的限界を議論していくこと、NPを用いたアプリケーション開発における経験や開発ノウハウをドキュメント化して公開することで共有し、NP上でのプログラミングの障壁を下げること、さまざまなアプリケーションをNP上で実現していくことでNPの可能性を追求することを目的としている。

2004年度の活動成果は、情報共有用Webページの設立と、奈良先端科学技術大学院大学の犬宅、飯村らによって開発が行われてきたNPを用いたDVTSスプリッタが挙げられる。以下、2章でDVTSスプリッタに関する活動報告を行う。そして3章でまとめと今後のNPWGの活動の方向性を述べる。

第2章 ネットワークプロセッサを用いたDVTSスプリッタの実装

2.1 DVTSスプリッタの開発と実装

インターネット上で高品質な映像データをリアルタイムに伝送する技術として、DVTS(Digital Video Transport System)[66]の研究・開発が進められている。DVTSを用いることで、市販のDVカメラやビデオデッキとPCをFireWire(IEEE1394インタフェース)ケーブルで接続し、インターネットを経由した映像配信を行うことができる。DVTSの利点としては、安価に高品質な映像の配信システムを構築できること、インターネットを利用することでFireWireケーブル長の制限(4.5m以内)を超えての映像配信が可能になること、などが挙げられる。しかしながらDVTSでは、非圧縮の映像の送受信を行う場合は約30Mbpsのネットワーク帯域を消費し、

かつ送信側・受信側の双方に相応の処理能力を要求することになる。とくに、同時に多数のホストに映像データを配信する場合、各ホストにユニキャストで配信するため送信側ホストの負荷増大は著しいものになる。また、マルチキャスト配信時においても、末端スイッチがマルチキャストに非対応の場合、フラディング（全ポートへの送信）が発生し、ネットワーク帯域の圧迫は著しいものになる。

そこで奈良先端科学技術大学院大学（NAIST）の大宅裕史・飯村卓司らのグループは本年度、ネットワークプロセッサ（以下 NP と略す）を用いた DVTS スプリッタの開発に取り組んだ。図 2.1 は DVTS スプリッタを用いた配信システムの構成を示した図である。DVTS スプリッタは主として受信ホストの付近上流に設置されることを想定している。DVTS 送信ホストでは 1 つの受信ホストに対して DVTS ストリームの送信を行う。DVTS 送信ホストと受信ホストの間に位置する DVTS スプリッタは、通常は L3 forwarder として動作する。コピーした DVTS ストリームの受信ホスト（コピー側受信ホスト）が登録されている場合、DVTS Splitter ではオリジナルの DVTS ストリームの受信ホストに対してパケット転送を行うとともに、DVTS ストリーム用の UDP パケットをコピーし、あて先 IP アドレスを変更してコピー側受信ホストに対して送信する。

本システムの開発には、奈良先端科学技術大学院大学と横河電機が共同で製作した NAPPI ボードを使用した。NAPPI には NP として Intel IXP1200 が搭載されており、ネットワークインタフェースは 10BASE-T/100BASE-TX で接続可能な Ethernet

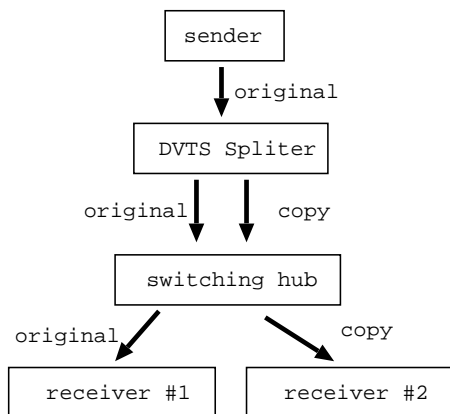


図 2.1. DVTS スプリッタを用いた配信システムの構成

ポートが 4 ポート使用可能である。開発環境は IXP1200 Developer Workbench を用い、DVTS スプリッタのマイクロコードの記述にあたっては、Layer 3 でのルーティングおよびパケット転送処理を行うサンプルプログラム「L3forwarder」を参考にした。開発の成果として、入力 DV ストリームを 2 つの特定ホストに配信するマイクロコードを記述し、実機での動作を確認することができた。L3forwarder に加えたマイクロコードの具体的な処理手順は以下の通りである。

1. 受信データが DV ストリームであるかの判定（異なる場合は以下の手順を省略）
2. 該当パケットの複製
3. 各複製パケットの宛先 IP アドレスの付替え

手順 1 は、現状ではパケットが UDP を利用していることと、特定のポート宛のものであることを判定基準としている。この部分は随時より精密な判定条件と置換可能であると考える。手順 2 は、L3forwarder でパケットの生成処理を行っている部分を単純に複数回行うことで実現している。手順 3 は、あらかじめ設定された配信先情報に基づき、手順 2 で生成したパケットの IP アドレスの付替えを行っている。このように DVTS スプリッタは、マイクロコードの実装としてもシンプルなものとなっている。

2.2 報告活動

DVTS スプリッタ開発の成果に関する報告活動として、2004 年 3 月に行われた WIDE 春合宿と DVTS コンソーシアム第 2 回メンバ総会、2004 年 7 月に行われた DVTS コンソーシアムインターオペラビリティワークショップに参加した。

2004 年 3 月の WIDE 春合宿では、実機を用いた DVTS スプリッタのデモ展示を行った。デモ展示での配信システムの構成は図 2.1 に示す構成で行った。このデモでは、送信ホストが送出する映像を DVTS スプリッタに接続された 2 台の受信ホストが受信する構成で行った。DV receiver #1 は送信側が指定している受信ホストで、DV receiver #2 は DVTS スプリッタでコピーした UDP パケットの受信ホストである。デモ展示では DV receiver #1、DV receiver #2 で受信した映像を並べて表示した。受信映像は receiver #1、receiver #2 と同じ場所でブロックノイズが発生していたが、それ以外はとくに問題なく DVTS 用の UDP パケットのコピーと転送が行え、映像データ

の複製配信が成功していることを示した。合宿内の BoF では、当時の成果をまとめたものを発表した。加えて同月の DVTS コンソーシアム第2回メンバ総会にも出席し、研究活動の経過を発表した。

また、2004年7月には、DVTS コンソーシアムによる「第1回インターオペラビリティワークショップ」に参加した。このワークショップで、開発した DVTS スプリッタと他機器との相互接続性について検証を行う予定であった。しかし、実機の動作が不安定であったためデバッグ作業を中心に行い、ワークショップ内では DVTS スプリッタの概略を紹介するに留まった。ワークショップでのデバッグ作業と後日の調査の結果、動作が不安定であった原因は OS ごとの DVTS sender のデバイスドライバの実装の差異により送出パケットの取り扱いが異なるためであることが判明した。具体的には、Windows と Mac OS 上での DVTS sender の実装ではパケットをバースト的に送出するため、NP 上でパケット処理が追いつかずパケットロスが発生し DVTS スプリッタの動作が不安定になるためである。これに対し、FreeBSD 上での実装を送信ホストとして用いると送出が非バーストであるため DVTS スプリッタのパケット処理速度で十分対応でき、パケットコピーに影響がないようであった。バースト転送に対する処理に関しては解決策を模索中である。

2.3 今後の DVTS スプリッタ開発の予定

現在本プロジェクトに関しては、最大配信先ホスト数の拡張・配信先などの設定作業の容易化・実機での安定動作・性能の評価に焦点を当てて、開発作業を継続中である。

第3章 まとめと今後の方針

本年度の活動報告は、主に2003年から奈良先端科学技術大学院大学のプロジェクト実習の一環として行ってきた DVTS スプリッタの開発についてまとめた。NP を用いた開発はプログラミング言語を覚えるだけでなく、プロセッサアーキテクチャ、メモリアーキテクチャ、またプロトコルの仕様についてなどの基礎知識を広い範囲にわたって求められる。そ

のため、開発を始める前に NP のアーキテクチャや基礎的なプログラミングを理解するまでに、2003年度は約半年ほど時間がかかってしまった。また、今回用いたプログラミング言語のドキュメントが開発を始めた段階ではほとんどなかったため、開発はなかなか思うように進まなかった。しかし、その教育効果は非常に高く、開発を通してソフトウェアアーキテクチャとハードウェアアーキテクチャ双方について広範な知識を得ることができたと開発者らから回答を得ている。

また、2003年度での開発で培ったノウハウや NP の開発に必要な知識をドキュメントとして残しておいたことにより、2004年度のプロジェクト実習では2003年度に半年かかった基礎知識の習得が1ヶ月半で終わることができた。2003年度、2004年度の開発において蓄積した NP の開発におけるノウハウは Network Processor ホームページ [226] において WIDE Project に対して公開を行っている。

NP WG では引き続き NP の研究動向の調査や、NP を用いたアプリケーション開発を行っていく。また、NP にとどまらず、Web サーバ用の TCP/IP スタックのハードウェア化についても奈良先端科学技術大学院大学の河合を中心に模索していく。

