高性能クラスタにおけるEthernetを利用した高性能マルチパス・マルチリンクネットワークシステムの開発

三浦信一[†]　岡本高幸[†]

安価なコモディティネットワークである Ethernet は，その高いコストパフォーマンスから，多くの PC クラスタで使用されている．しかしながら，大規模な HPC クラスタに用いるには，バンド幅・耐故障性および拡張性が低く問題があった．我々は，この問題を解決する RI2N/UDP および VFREC-Net をこれまで開発してきた．これらのシステムは複数リンクをノード入出力あるいは中間スイッチに多用することにより，バンド幅の増強と耐故障性を実現する．この二つの技術は直交しており，同時に使用することで既存の Ethernet が持つ問題点を解決することが可能である．しかし，それら実装上の問題により性能や運用に問題があることが分かった．そこでそれらの問題を解決するために 2 つのシステムの機能を 1 つに統合し，それらの主な機能を Linux のデバイスドライバとして実装した RI2N/Drv を開発している．実装が完了した高バンド幅化機能のみを実現する現在のバージョンで性能評価を行ったところ，過去の 2 つのシステムを組み合わせた場合と比較して高いバンド幅と低い遅延時間を実現できた．これにより新たに開発したデバイスドライバが，クラスタ向けのネットワークシステムとして有効に機能し得ることがわかった．

A High-performance Network System with Multi-Path / Multi-Link Ethernet for High-Performance PC Clusters

SHIN'ICHI MIURA [†] and TAKAYUKI OKAMOTO [†]

Ethernet is the most popular interconnection network to be used on various PC clusters. Especially, Gigabit Ethernet provides a very high cost/performance ratio as the most used commodity network. However, it is difficult to utilize it on a large scale HPC clusters because its absolute bandwidth and scalability is limited. To solve this problem, we have been developing both RI2N/UDP and VFREC-Net systems for wide bandwidth, high scalability and high dependability for large scale HPC clusters. In this research we implemented a special device driver on Linux system, named RI2N/Drv. Through the performance evaluation, we confirmed that the new implementation provides both higher bandwidth and low latency than the old systems.

1. はじめに

高性能 PC サーバを相互結合した PC クラスタは，HPC 分野の様々な局面で用いられている．これらの HPC 向け PC クラスタは，比較的安価でありながら高性能であるため幅広い分野で利用されている．しかし HPC 分野では，ノード間ネットワークに対しバンド幅・耐故障性そして拡張性に関して要求が厳しい．そのため主要部品の多くにコモディティ製品を活用しつつも，ネットワークだけは専用ネットワーク製品を選択することが多い．特に現在，クラスタの規模は大きくなり，ノード数の増加と高密度化が進んでいる．ノード数の増加とともに相対的に部品点数の割合が増大するため，ネットワークは耐故障性に優れていなければならない．これに加えてクラスタの規模に合わせて全体性能を向上させるためには，より柔軟なネットワークトポロジを許容し，高いバイセクションバンド幅を確保するネットワークシステムが必要になる．このような環境で，コモディティネットワークである既存の Ethernet を用いた場合，耐故障性が低くなり，またネットワークトポロジは原則 Tree 形状に制限されるためバンド幅ボトルネックが生じやすい．そして，これらの問題は並列計算の性能ボトルネックとなる可能性がある．そのためクラスタ規模が大きくなるに伴い，これらの問題を解決する専用ネットワークである Myrinet[1] や InfiniBand[2] といった SAN (System Area Network) が多くの HPC クラスタで採用されている．

† 筑波大学大学院 システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

しかし SAN は，コモディティネットワークである Ethernet と比較して高価である．今後クラスタがより大規模した場合，ネットワークの導入コストが問題になる．そのため既存の Ethernet を用いつつ，拡張性・耐故障性をもつネットワークシステムが必要とされている．

これまで我々はこれらの問題を解決するべく，ノード間の接続に対して高バンド幅と耐故障性を提供する RI2N[3]，および，スイッチ間の接続性に柔軟性を持たせ，クラスタの規模に拡張性を持たせる VFREC-Net[4] をそれぞれ開発してきた．しかし，より高性能なネットワークシステムを実現するためにこの 2 つのシステムを同時に利用したところ，いくつかの問題があることがわかった．本プロジェクトではこの問題を解決するべく，これら 2 つのシステムを新たに 1 つに統合し主要機能をネットワークデバイスドライバのみで実現する．これにより，より高性能な PC クラスタ用ネットワークをユーザに提供することを目指す．

次章において過去に実装・評価を行ったこの 2 種類のネットワークについて述べ，それらのネットワークシステムを組み合わせた場合に見受けられた問題を述べる．その後に新しく提案・実装する RI2N/Drv の概要を述べそれらの実装方法を示す．4 章および 5 章では新しく実装した RI2N/Drv の性能評価を行い，それで得られた結果の考察を行う．その後 6 章で今後の課題について述べ，7 章で関連研究を述べる．

2. 既存システム

我々は，ノード間の接続に対して高バンド幅と耐故障性を提供する RI2N[3]，および，スイッチ間の接続性に柔軟性を持たせ，クラスタの規模に拡張性を持たせる VFREC-Net[4] をそれぞれ提案・開発をしてきた．ここでまず，過去に開発した 2 つのネットワークシステムについて述べる．なお，それぞれのネットワークの詳細については文献[3),4)] を参照されたい．

2.1 RI2N

RI2N (Redundant Interconnection with Inexpensive Network) はノード間のネットワークに耐故障性と高バンド幅を同時に提供するシステムである．RI2N では，各ノードを複数の異なるネットワークに所属させ，これらのネットワークリンクを同時に利用する事で目的を達成する (**図 1**)．すべてのネットワークが正常な場合，これらの複数のネットワークを同時に利用することでユーザに高いネットワークスループットを提供する．また一部のネットワークが故障などの理由により利用できなくなった場合，そのネットワークを送受信の対象から除外し，残りの正常なネットワークだけを用いて送受信を継続する．

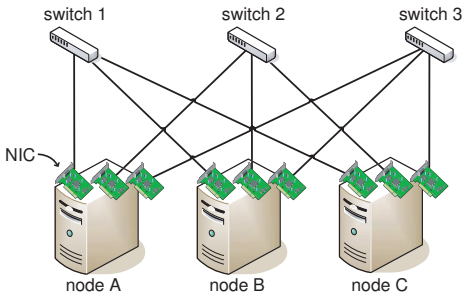


図 1 RI2N/UDP が想定するネットワークイメージ

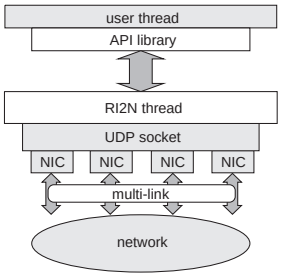


図 2 RI2N/UDP の実装イメージ

現在の RI2N の実装として RI2N/UDP[3] がある．RI2N/UDP の実装イメージを**図 2** に示す．RI2N/UDP では実装の簡便性と移植性を得るため，一般的な socket API を用いたユーザレベル実装とした．RI2N/UDP は下位プロトコルとして，一般的な TCP/IP ではなく UDP/IP を使用する．コネクションレス指向型の UDP/IP を利用することで，故障発生時においても安定動作することが可能になる．RI2N/UDP は，マルチリンク上で UDP/IP プロトコルによるパケット制御を行うことにより，TCP/IP 相当のストリーム通信機能を提供している．このため，複数リンクの監視機能，パケットの順序制御機能，ロストパケットの再送機能，フロー制御機能等がユーザレベルライブラリ上に実装されている．RI2N/UDP はアプリケーションの移植性を高めるため TCP/IP に類似した socket API を提供する．このことから，TCP/IP を用いる Open MPI[5] など現在 PC クラスタで利用されている多くの MPI 実装を容易に移植可能としている．

2.2 VFREC-Net

クラスタの性能をノード数に合わせて向上させるためには，スイッチ間のバンド幅をノード-スイッチ間よりも増強しなくてはならない．たとえばノード-スイッチ間に Gigabit Ethernet（以下 GbE）を用いる場合，スイッチ-スイッチ間の接続には GbE よりも高速なリンク，たとえば 10GbE 等を用いるべきである．しかし，

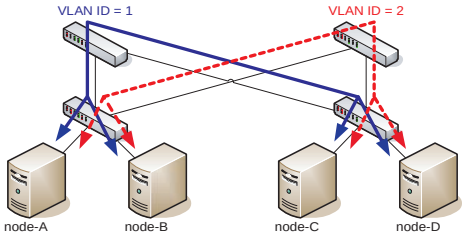


図 3 VLAN ルーティング法 イメージ図

それらの高速なリンクをサポートするスイッチは GbE のそれよりも高価であり，コストパフォーマンスのために GbE を利用している以上，上位リンクでの 10GbE の利用はそのメリットを乏しくする．バンド幅を高めるためにスイッチ間に複数のリンクを用意するという方法も考えられるが，安価な Layer-2(以後 L2) スイッチを用いる場合，それを実現することは困難となる．L2 スイッチを用いる場合ではネットワークにループができる構造はブロードキャストストームの原因となり利用できないためである．そのため複数のリンクをスイッチ間に用意できない．IEEE 802.3ad[6] を用いることでスイッチ間に複数のリンクを用意することが可能であるが，その規格制約上，やはりバンド幅は制限される．そのため，現在まで多くの PC クラスタでは単純 Tree 型のネットワークのみが用いられてきた．そこでこの問題を解決するために，tagged-VLAN[7] を用い，各ノードから送信するパケットの VLAN ID を制御することで，Ethernet を用いた場合でも柔軟なトポロジを可能にする VLAN ルーティング法が提案されている[8]．VLAN ルーティング法を用いた場合のイメージを**図 3** に示す． VLAN ルーティング法は，物理的にループのあるネットワーク構成を，VLAN 技術を用いることで論理的にループのない複数のネットワークに展開し，これらのネットワークをノード側から明示的に使い分けることで，既存の Ethernet でスイッチ間のバンド幅ボトルネックを解決する技術である．VLAN 機能をサポートする Ethernet スイッチは HPC クラスタで用いられるレベルの Ethernet スイッチでは標準的にサポートされており，特にコスト上の問題もない．我々は，この VLAN ルーティング法を改良し，システムレベルで実装したうえで，より PC クラスタ向けに利用しやすくした VFREC-Net (VLAN-based Flexible, Reliable and Expandable Commodity Network)[4] を開発した．

VFREC-Net は仮想 Ethernet デバイスのドライバとしてシステムレベル実装されている．そのため TCP/IP や UDP/IP といったネットワークプロトコルを一切変更することなく使用することが可能になっている．この VFREC-Net を用いることで，大型計算機や高価な SAN などでしか実現不可能であったネットワークトポロジを安価な Ethernet を用いて構成することが可能になった．一例として**図 4** に示すような Fat Tree 構成が初めて Ethernet でも構成可能になり (これを SAN などのネットワークとし区別するために VBFT (VLAN Based Fat Tree)[8] と呼ぶ)，ノード数に応じて高いバイセクションバンド幅を得ることが可能になっている．

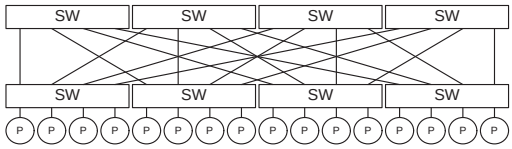


図 4 VLAN Based Fat Tree ネットワーク

2.3 問 題 点

このように，現在まで開発してきた RI2N/UDP と VFREC-Net は共にクラスタ向けネットワークシステムである．それらは共に高いスループットを得ることを主目的としているが，RI2N/UDP ではノード-ノード間，VFREC-Net ではスイッチ-スイッチ間とターゲットが異なっている．そこでこれら 2 つのシステムを組み合わせることで，より高性能なネットワークをクラスタ向けに提供できるものと考えた．現状では，この二つのシステムは実装の階層が明確に分離されている．RI2N/UDP は，前述のように通常の UDP/IP を用いたユーザレベル実装であり，一方の VFREC-Net はドライバを用いたシステムレベル実装となっている．そのため，この 2 つのシステムを同時に利用することが可能であった．VFREC-Net でスイッチ間のバンド幅を高め，そしてそれらのネットワークを複数用意して，そのネットワークを RI2N/UDP で同時に利用することで，高バンド幅・耐故障性のあるネットワークを実現できると考えた．そこで，われわれはこれら 2 つのシステムを組み合わせた，ネットワークシステムを過去に評価した[9]．その結果，いくつかの問題があることがわかった．特に以下の問題が RI2N/UDP に存在した．

- ユーザにプログラムの改変・再コンパイルを要求し，プログラムのポータビリティの面で課題がある．
- 通常の socket プログラムを用いたユーザレベル実装であることに加えて，Thread の同期や多くの送受信データのメモリコピーを必要とするため遅延時間が大きく，また処理能力が相対的に低いプロセッサではネットワーク性能の向上が期待でき

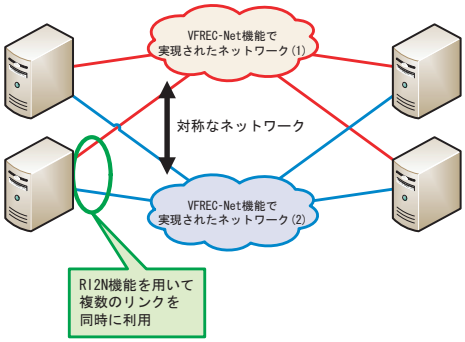


図 5　システムの最終的な目標

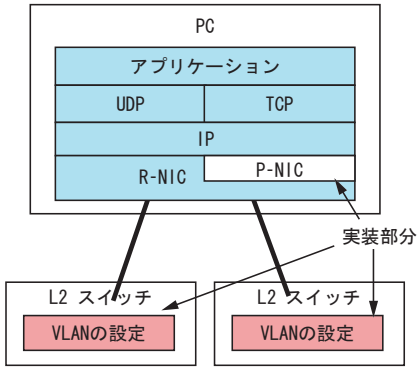


図 6　実装のレイヤ

ない．

一方で VFREC-Net は，RI2N/UDP と異なり Ethernet デバイスドライバとして実装したため，RI2N/UDP のような問題は発生していない．このまま，2 つのシステムを統合し運用した場合，RI2N/UDP の問題が大きく残ることになる．また同じネットワークシステムでありながら，これら二つのシステムは実装及び設定方法などが異なるため，システムが複雑化しシステム管理や運用の面でユーザへの負担が大きくことも考えられる．

3. 提案システム

前述の問題を解決するために，われわれは VFREC-Net と RI2N/UDP の機能を統合した新たなネットワークシステムを開発する．最終的に実現するシステムイメージを**図 5** に示す． 以後この新たに開発するシステムを RI2N/Drv と呼ぶこととする．RI2N/Drv は，今までの RI2N と VFREC-Net の機能を包括するものである．まず，VFREC-Net 機能に相当する機能で高いスケラビリティをもつネットワークトポロジを構成する．たとえば前述のような VBFT(図 4) のようなネットワークトポロジが考えられる．このようなネットワークを複数用意し，各ノードがいずれのネットワークにも所属するよう接続する．そして RI2N に相当する機能によって，これらのネットワークを同時に利用することで，ノード間に高バンド幅化と耐故障機能を提供する．

3.1 実 装 方 針

RI2N/UDP の問題は，その実装がデータの送受信のために特別の API を用意したユーザレベルの実装であり，そのため，アプリケーションの可搬性と性能が低かったことである．その点を踏まえて RI2N/Drv では，ユーザからのインタフェースとしては，OS が提供する socket API をそのまま利用可能にすることを目標にする．一方，先行研究として実装した VFREC-Net はデバイスドライバとして実装したため，実装がスマートになり，上位の通信プロトコルとして TCP/IP をそのまま利用できることで性能評価においてもオーバヘッドは最小限に抑えられていることが確認できた．そこで新たに開発する RI2N/Drv は，既存の VFREC-Net をベースとし，RI2N の基本機能を VFREC-Net ドライバ中に実装する．また，今回は特別なネットワークプロトコルは用いない．この理由として，このシステムで要求するプロトコル機能が標準の TCP/IP のみで実現可能であると考えられるためである．今回開発するシステムを Linux の Ethernet デバイスドライバとして実装し，ユーザは標準の TCP/IP を通して RI2N/Drv を使用する．

3.2 実 装 方 法

実装するシステムの階層モデルについて**図 6** に示す．前述のように，本システムは主にネットワークデバイスドライバとして実装する．本ドライバは OSI 参照モデルにおけるネットワーク層とデータリンク層の間に位置する．ネットワーク層で生成された IP パケットは，データリンク層の仕様に基づいて Ethernet フレームへと変換され，通常はそのまま，OS の処理によって決定されたネットワークデバイスより送信される．本デバイスドライバを用いる場合，仮想的なネットワークデバイス (以後 P-NIC と表現) を生成し，RI2N/Drv 用の Ethernet フレームが P-NIC へ渡されるように IP アドレスを設定する．P-NIC で取得した Ethernet フレームは，仮想デバイスドライバ中で必要な処理を加えたのちに，関連付けられた本物のネットワークデバイス (以後 R-NIC と表現) より送信する．

3.2.1 高バンド幅化の実現

本システムでの高バンド幅化を行うターゲットは 2 つある．1 つはスイッチ間の高バンド幅化，もう 1 つはノード間の高バンド幅化である．それぞれのアプロー

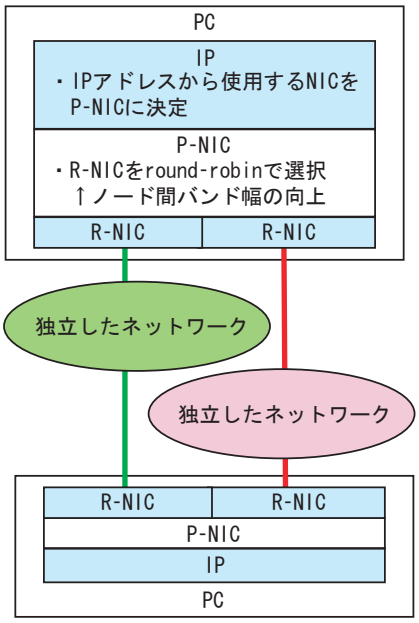


図 7　ノード間の高バンド幅化の実現

チについて述べる．

ノード間の高バンド幅化

基本方針として，送信側でシステムに関連付けられた n 個の R-NIC より，ラウンドロビンで Ethernet フレームを送出する (**図 7**)．受信側の n 個の R-NIC で，ラウンドロビンで送られてきた Ethernet フレームを受信し，それらの Ethernet フレームをあたかも RI2N/Drv システムが用意した P-NIC から受信したかのように処理を加えた上で上位のネットワーク層 (e.g. IP) へと引渡す．この実装の場合，一度に送信するアプリケーションにおけるデータサイズが Ethernet での MTU サイズ以下の場合，送信を行う R-NIC を複数用意することによるは効果は望めない．n 個以上の Ethernet フレームを送信するケースの場合，バンド幅 (又は通信遅延時間) への効果は単純に n 倍なることが期待できる．しかし実際には，Ethernet フレームを単純にラウンドロビンによって各 R-NIC に割り付けるだけでは，バースト転送時における性能向上と後述する耐故障機能に対応できない．そこで本実装では，各 R-NIC の Ethernet フレームの送信キューの状況を確認し，キューが詰まっている状況ではその段階で処理に余裕がある他の R-NIC から送信する仕組みを備える．

このシステムでは，Ethernet フレームの単位で複数の R-NIC から分散してフレームを送信するため，Ethernet フレームの到着順序が保障できない．しかし，本来 Ethernet を用いたシステムではネットワーク層 (e.g. IP) または，トランスポート層 (e.g. TCP) は Ethernet

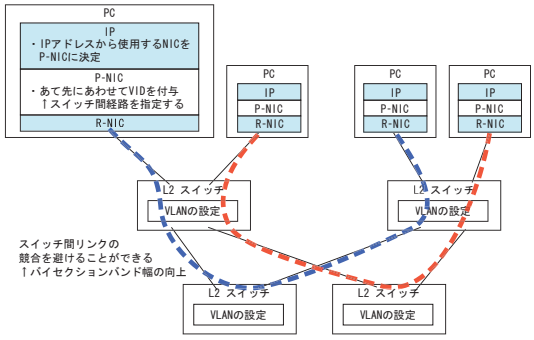


図 8　スイッチ間の高バンド幅化の実現

フレームの到着順序が正しくないことを想定した設計となっている．特に TCP ではデータの信頼性を確保するために，これらの到着順序を補正する仕組みを持っている．そこで，それらの処理をトランスポート層の TCP で処理されることを期待し，RI2N/Drv では特別な処理を行わない．

上位スイッチ間での高バンド幅化

上位スイッチ間での高バンド幅化のための実装は，VFREC-Net と同様である (**図 8**)．本ドライバは VFREC-Net を基にして開発されているため，既存システムでの説明に記述した内容 (2.2 節) となる．また具体的な実装については文献[4] を参考にされたい．

3.2.2　耐故障機能の実現

RI2N/Drv では複数の R-NIC をラウンドロビンで選択して通信を行っており，またロストパケットは TCP によって再送される．そのため単純に考えた場合，すべてのネットワークが故障しない限り，完全に通信が停止することは無い．故障したネットワークを送信ネットワークとして選択したことで消失した Ethernet フレームは，TCP の再送機能によって，いずれ他の正常なネットワークを用いて再送される．しかし実際には，故障ネットワークを通信に継続して使用し続けた場合，フレーム消失がいつまでも続き，TCP の Window コントロール機能により，送信スピードは自動的に低下する．送信スピードが低下した場合においてもフレームのロスの原因は混雑ではなくネットワークの故障であるため，そのロス率は変化せず，最終的に送信は停止に近い状態に落ち着く．このままでは，耐故障機能が実現できているとはいえない．そこでネットワークの故障状態を常に監視し，故障が発生した場合には，そのネットワークに所属する R-NIC を利用するデバイスリストから除外する必要がある．それにはなんらかの手段を用いて故障を検知し，送受信するネットワークとして不適切な R-NIC を排除する必要がある．これらの故障検出の方法として**図 9** に示した，以下の 2

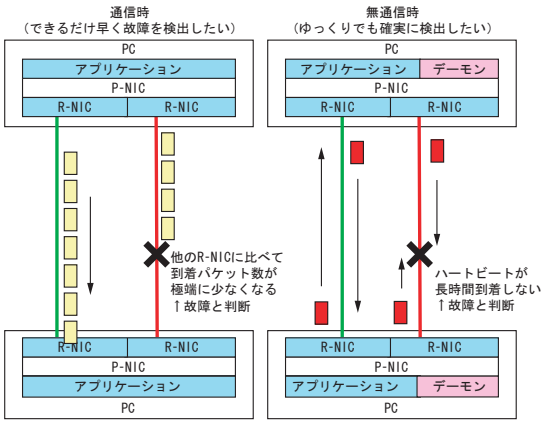


図 9　故障検出機能

表 1　評価環境

CPU	Intel Xeon 3 0GHz 1-way (Hyper Thread ON)
Memory	1 0 Gbytes DDR2/400MHz (Dual Channel)
NIC	Intel PRO/1000MT Dual Ports (MTU=8000)
	PCI-X 64bit/133MHz
OS	Linux Kernel 2 6 19
Compiler	GCC ver 4 1

表 2　スループットの測定結果

Normal Ethernet	a – b	121 Mbyte/sec
	a – e	121 Mbyte/sec
RI2N/UDP	a – b	203 Mbyte/sec
	a – e	203 Mbyte/sec
RI2N/Drv	a – b	246 Mbyte/sec
	a – e	246 Mbyte/sec

つの機構を用いる．

到達パケットの偏り検出機構　複数の R-NIC から送信するタイミングは，基本的にはラウンドロビンで決定される．そこで，受信側でどの R-NIC にどれだけのフレームが到着したのかを，送信元ごとに記録する．すべてのリンクが正常である場合は，どの R-NIC に到着する Ethernet フレーム数もほぼ等しくなる．この値を比較し他の R-NIC に比べて Ethernet フレームが非常に少ないものがあれば，その R-NIC の所属するネットワークが故障，もしくはそれに近いレベルの Ethernet フレームロスを起こしているネットワークであると考えられる．受信側でこのような評価を定期的 (e.g. 一定数のパケットが到着するごと) に行うことで，到着フレーム数が相対的に少ない R-NIC のネットワークに故障が発生したと判断できる．

ハートビート機構　到達パケットの偏り検出機構では，ある程度の送受信が行われているノードペア間でのネットワーク故障のみが検出可能である．しかし実際には，ショートメッセージのみしか通信しないノードペアも存在する．このようなノードペア間では元々高いバンド幅を必要としていないため，低いスループットでも通信が継続できていればよいが，故障したネットワークを故障と判断せず使い続けることは望ましくない．そこで，ハートビートパケットを用いて，各ノード間のネットワークをある一定間隔で監視する．もしハートビートパケットを頻繁にロスするネットワークがある場合，そのネットワークは故障しているとシステムが判断可能である．また，故障時においても，定期的にハートビートパケットを送信することで，故障の復帰についても検出可能になる．

これらの機能は，デバイスドライバのみで実装することが難しい．特にハートビート機構はその特性上タイマ等が必要であり，デバイスドライバとは別に独立したデーモンプロセスで制御することが望ましい．これらの実装については考案中であり，現在のところ RI2N/Drv の機能として組み込まれていない．

3.3　現在の実装状況

現在，ノード間およびスイッチ間の高バンド幅機能について実装を完了している．ネットワークの耐故障化については現段階で実装を完了していない．以後の性能評価は，おもに高性能化についてのものである．

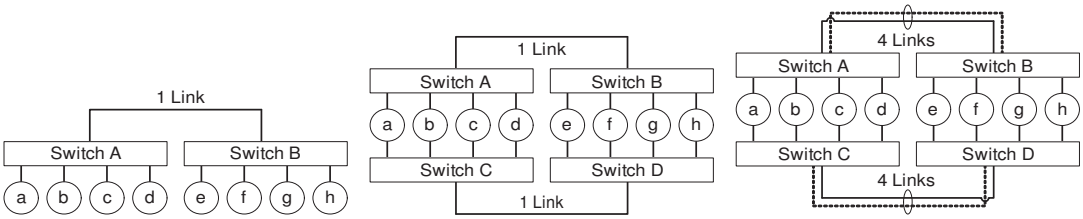
4.　性 能 評 価

新たに実装した RI2N/Drv の性能評価を行う．評価環境として**表 1** に示す 8 ノードからなるクラスタを構築した．この環境を用いて基本性能としてスループット，遅延時間を RI2N/Drv を用いた場合について評価する．なお性能比較として RI2N/Drv を用いない通常の Ethernet 上で TCP/IP を用いた場合と過去の実装である RI2N/UDP についても同時に評価を行う．それぞれのネットワーク構成を**図 10** に示す．

4.1　スループット

はじめにスループットの評価として，あるノードペア間のスループットを求める．測定の対象として，スループットの評価と同様に，同一のスイッチで接続されているノード間 (a – b) での遅延時間と，異なるスイッチに接続され Ethernet フレームが 2 台スイッチを経由する場合 (a – e) の遅延時間について計測する．測定結果を**表 2** に示す．

スループットの結果ではノード間に経由するスイッチの段数で値に差はない．Normal Ethernet では，1 つのネットワークのみにしか接続されていないため，その



(a) Normal Ethernetで用いるネットワーク　(b) RI2N/UDPで用いるネットワーク　(c) RI2N/Drvで用いるネットワーク

図 10　ネットワーク接続図

最大スループットは Gigabit Ethernet の最大スループット (125Mbyte/sec) と等しい．評価結果から実測上でほぼ理論ピーク性能が出ていることがわかる．RI2N/UDP および RI2N/Drv では，複数あるマルチストリームを同時に利用することで，理論ピーク性能は 250Mbyte/sec となる．これに対し，従来実装の RI2N/UDP の評価結果は 203Mbyte/sec であった．RI2N/UDP は UDP/IP を利用したユーザレベル実装となっており，そのため様々なオーバヘッドが常にかかるものと考えられる．過去の性能評価では異なる環境において最大スループットが 245Mbyte/sec の場合もあったが，RI2N/UDP はその実装上の問題で最大性能を出すために細かいシステムパラメータをチューニングする必要がある．今回は時間の都合上それらのパラメータについてチューニングを行わなかった．一方の RI2N/Drv の実装はドライバレベルとなっており，RI2N/UDP と比較してより高い性能を期待することができる．結果は 246Mbyte/sec であり，理論最大性能である 250Mbyte/sec に近く，ヘッダ等でのオーバヘッドを考えた場合，ほぼ理想的な性能であるといえる．RI2N/UDP の場合と異なり，特に特別なパラメータチューニングを行わずとも，OS の TCP/IP 実装を用いてこれだけの性能を得ることができる．この評価結果より，今回実装した RI2N/Drv は過去の実装である RI2N/UDP よりも高い性能がより簡単に得られることがわかった．

4.2 遅 延 時 間

次に遅延時間について評価する．スループットの評価と同様に，同一のスイッチで接続する場合と，2 台スイッチを経由する場合について評価を行う．それぞれの評価結果を**図 11**, **図 12** に示す．

まずメッセージサイズが 20 Kbyte 以下の部分について注目する．RI2N/Drv ではメッセージサイズが MTU (8000 byte) の部分までは標準の Ethernet とほぼ値が重なる．この場合，メッセージサイズが MTU を超えないため，1 個の Ethernet フレームサイズにすべてのメッセージが収められる．そのため，2 系統の R-NIC が同時に利用できないため，遅延時間を短くすることができない．一方 RI2N/UDP では同様な条件にもかかわらず標準の Ethernet を利用する場合より大きな時間を必要としている．これはユーザレベルで実装することによるオーバヘッドであり，この差は 30–50 μsec 程度ある．メッセージサイズが MTU サイズを超えたのち RI2N/Drv は 2 系統 R-NIC を利用することで，通信遅延時間を通常の Ethernet と比較して低くすることができる．RI2N/UDP でも MTU サイズ以降は 2 系統のネットワークを有効に活用できるが，初期のオーバヘッドのために通常の Ethernet よりも性能を十分に引き出すためには，より大きなメッセージサイズを必要としている．

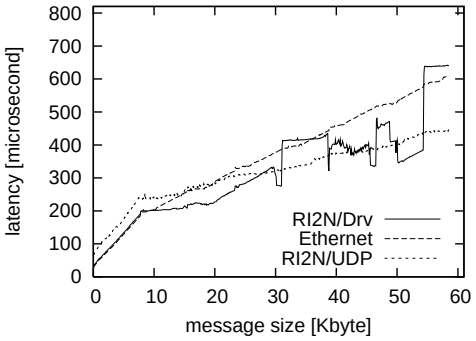


図 11　通信遅延時間 (a – b) 結果

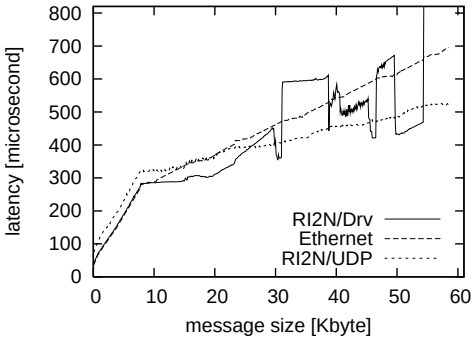


図 12　通信遅延時間 (a – e) 結果

次にメッセージサイズが 20 Kbyte 以上の部分に注

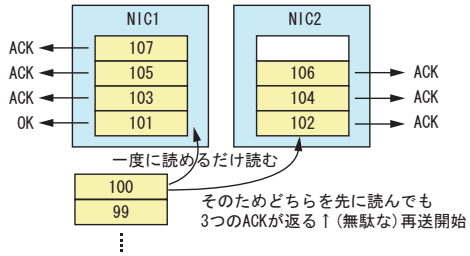


図 13 TCP 再送制御アルゴリズムと RI2N/Drv の問題

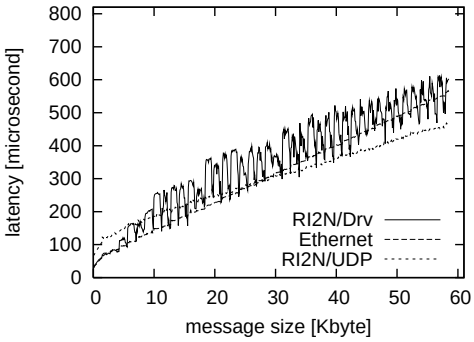


図 14 通信遅延時間 (a – b) MTU=1500 の場合

目する．ここで RI2N/Drv の評価結果が大きく変動しその値は安定しない．他のネットワークが安定していることから，これは RI2N/Drv の実装の問題であると考えられる．次章でこの問題点について考察を行う．

5. TCP 再送制御アルゴリズムに関する考察

前述の遅延時間の評価で，ある一定以上のメッセージサイズにおいて，RI2N/Drv の性能が悪化する場合があった．我々はこの原因として，RI2N/Drv の送信アルゴリズムと TCP の再送制御アルゴリズムのミスマッチが起こっていることを推測した．**図 13** を用いて，ここで発生している現象を説明する．実際に送受信する R-NIC として NIC1 と NIC2 がある．現在までに受信している TCP の順序番号を 100 とすると，次に受信するべきパケットは 101 となる．NIC1 にパケットが到着することで，NIC が OS に対して割り込みをかけて Ethernet フレームの受信を要求する．このときわずかなタイムラグのあと，103,105,107 のパケットも NIC1 に到着する．現在の Linux 上の標準的な TCP 実装では，ネットワークの性能を出すため，少ない割り込み回数で多くのパケットを取得しようと試みる．そこで現在届いている残り 3 つの Ethernet フレームを一度に受信する．そのため，103,105,107 と到着順序に矛盾がある TCP パケットを受信することになる．実際には NIC2 において 102,104,106 の Ethernet フレームが到着しているためそちらを受信すれば到着順序の矛盾はない．しかし，OS はそれらのことを認識できないため，パケットがどこかでロスした可能性があることを伝えるため ACK を送信側に送信する．このとき，103,105,107 の 3 つのパケットの到着において，それぞれで「まだ 102 が届いていない」という順序の矛盾を検出するため，合計 3 つの同じ ACK 番号（次にそこから送信することを望むシーケンス番号）の ACK パケットを送ることになる．現在の TCP には 3 度同じ内容の ACK を受け取ると再送を開始するというアルゴリズムがあるため，実際には何もパケットが失われていないにもかかわらず再送を開始することになる．このため，余計なパケットの送信を必要とし通信性能が低下する場合がある．この問題が発生する条件は Ethernet における MTU サイズに大きく依存する (MTU サイズが小さい場合，一度の割り込みの間に同時に多くのパケットを受信できてしまうため)．これらの仮説をもとに，実際に Ethernet の MTU を小さくし遅延時間の評価を行ったところ常に遅延時間の分散が発生する現象が確認でき (**図 14**)，この仮説が正しいと推測できる．

この問題を解決する一つの方法は TCP の再送アルゴリズムを改良し，再送処理を開始する順序不一致の Ack パケットに関してより寛容にすることである．具体的には現状の 3 回以上というパラメータを 6 回程度まで引き上げることで，ある程度の効果があると推測される．また，低メッセージサイズ時における並行リンクの使用効率の低下につながるが，送信 R-NIC を選択する際，現在の単純ラウンドロビンを改良することで，この問題を解決可能ではないかと期待している．具体的には Ethernet フレームを 1 つ送るごとに R-NIC を切り替えるのではなく，ある一定の回数以上同一の R-NIC から連続して送信を行う．これを行うことで，ある程度連続した順序番号をもつ Ethernet フレームを同時に取得できるようになると考えている．

6. 今後の課題

現状では，RI2N/Drv は基本的な機能の実装にとどまり，また性能評価についても十分ではない．今後よりよいシステムを開発する上で以下の課題を挙げる．

TCP 再送制御を考慮した実装

バースト転送時のスループットの評価では，おおむね我々の意図した性能を得ることができたが，遅延時間の評価では，あるメッセージサイズにおいて結果が悪くなる場合があった．この問題は TCP 再送制御と RI2N/Drv の実装上の問題であり，これを解決するた

めの方法について 5 章で提案した．今後これらの方法を適用しどのような実装が適切であるかを評価する．

耐故障機能の実装

今回のシステムでは高バンド幅化についての実装は完了したが，耐故障機能の実装については，その方針を述べるにとどまった．今後はこれらの実装を進めるとともに，耐故障機能の評価を行う．

クラスタシステム全体での性能評価

今回の性能評価では，ネットワークの基本性能を評価するにとどまった．RI2N/Drv では，スイッチ間のバンド幅についても高いバンド幅を実現可能であるが，それらの評価を行っていない．また，ネットワークの性能向上に伴い，クラスタ全体でどのような性能が得られるのかを評価できていない．今後は，クラスタ向けの各種ベンチマークを実行し，改善したネットワーク性能がどのようにアプリケーションに作用する評価する．

7. 関 連 研 究

Ethernet のバンド幅や耐故障性を高める技術として，IEEE 802.3ad[6] がある．この技術は，すでに多くの OS やスイッチで実装されているため，ユーザは既存環境を維持したまま使用することが可能になっている．しかし IEEE 802.3ad は HPC クラスタ向けに設計されておらず，接続形態やネットワークトポロジに制約が多い．特にスイッチ間並列接続については同一スイッチ間での直接接続のみ許されている点が最大の問題点である．並列リンクもラウンドロビンによる送信といった単純な方法でしか実装されていない．このようなことから，HPC クラスタでは IEEE 802.3ad が適切に機能できないと思われる．

HPC クラスタ向けに開発されたネットワークシステムとして PM/Ethernet[10] がある．PM/Ethernet は Ethernet 上でマルチリンクを使用した上で低遅延・高バンド幅を実現している．しかし，PM/Ethernet は直接デバイスドライバを変更するため，使用できる NIC 等に制限が生じるなど，システムへの依存が大きい．また，耐故障機能についても現在実装されていない．これに加えて，使用できる API も TCP/IP などと違い独自プロトコル PM を用いるため既存の socket API を用いたプログラムが流用できないといった問題もある．

HPC クラスタ向けの MPI ライブラリとして Open-MPI[5] がある．OpenMPI は動的に複数のネットワークを検出しこれらを同時に利用することで，高バンド幅化と耐故障機能を実現している．しかし，この機能を利用するためには，MPI を用いたアプリケーションを用意する必要があり，一般的な TCP/IP を用いたアプリケーションには適用できない．また，スイッチ間の高バンド幅化には対応できないという問題点もある．

8. お わ り に

本研究では，我々が過去に提案実装してきた RI2N/UDP と VFREC-Net を，新たに一つのシステムとして統合し高バンド幅・耐故障性を持ちつつ，かつ高い拡張性を有する PC クラスタ向けネットワークを提案・実装した．高バンド幅化の機能を中心に実装した現状の RI2N/Drv ではその評価結果から，そのオーバヘッドは極めて少なく，比較的大きなメッセージ転送時には高いバンド幅性能を示すことがわかった．また 2 つのネットワークを同時に使用した場合，バースト転送時において最大で 246 Mbyte/sec の結果を得ることができた．現在の実装では上位のプロトコルとして想定している TCP/IP のアルゴリズムと本システムのパケット再送処理における不整合の問題から，特定のメッセージサイズにおいて遅延時間が大きくなるという問題があるが，TCP/IP プロトコルの一部分の改良，もしくは RI2N/Drv システムの送信アルゴリズムの改良で回避できると期待できる．今後耐故障機能を実現することで，安価な Ethernet をベースにした，より高性能なネットワークを PC クラスタで用いることが可能になると考えられる．

謝辞　本研究の一部は，魅力ある大学院教育イニシアティブ「実践 IT 力を備えた高度情報学人材育成プログラム」による．

参 考 文 献

1) Myricom, inc.: Myrinet. http://www myri.com/.
2) InfiniBand Trade Association: InfiniBand, http://www.infinibandta.org/.
3) 岡本高幸, 三浦信一, 朴泰祐, 佐藤三久, 高橋大介: Ethernet マルチリンクによる PC クラスタ向け耐故障ネットワーク RI2N/UDP, ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS2007), 情報処理学会, pp. 41–48 (2007).
4) 三浦信一, 岡本高幸, 朴泰祐, 佐藤三久, 高橋大介: VFREC-Net：ドライバ制御による VLAN を用いたマルチパスネットワーク, 情報処理学会論文誌コンピューティングシステム, Vol. 47, pp. 35–45 (2006).
5) Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R.H., Daniel, D. J., Graham, R. L. and Woodall, T. S.: Open MPI: Goals, Concept, and Design of a Next Generation

MPI Implementation, *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, pp. 97–104 (2004).
6) IEEE: P802.3ad Link Aggregation Task Force. http://grouper.ieee.org/groups/802/3/ad/.
7) IEEE: 802.1Q - Virtual LANs. http://www.ieee802.org/1/pages/802.1Q.html.
8) 工藤知宏, 松田元彦, 手塚宏史, 児玉祐悦, 建部修見, 関口智嗣: VLAN を用いた複数パスを持つクラスタ向き L2 Ethernet ネットワーク, 情報処理学会論文誌 コンピューティングシステム, Vol. 45, No. SIG06(ACS6), pp. 35–44 (2004).
9) 三浦信一, 岡本高幸, 朴泰祐, 佐藤三久, 高橋大介: tagged-VLAN とマルチリンクに基づく PC クラスタ向け高性能・耐故障ネットワークの実装と評価, ハイパフォーマンスコンピューティング研究会 研究報告, 情報処理学会, pp. 25–30 (2006).
10) Sumimoto, S. and Kumon, K.: PM/Ehernet-kRMA: A High Performance Remote Memory Access Facility Using Multiple Gigabit Ethernet Cards, *3rd IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2003)*, IEEE Computer Society, pp. 326–333 (2003).