社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－ARC－167 (15)
2006－HPC－105 (15)
2006／2／28

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

岡本高幸[†] 三浦信一[††] 朴泰祐[††]
佐藤三久[††] 高橋大介[††]

PCクラスタはその対価格性能比の高さによって高性能計算分野で広く用いられている．PCクラスタのネットワークとしてはGigabit Ethernetが用いられることが一般的であるが，その一方で特に大規模化を考えた場合，その信頼性は必ずしも高くない．ハードウェアの完全な故障以外にもネットワークスイッチの一時的な不具合等が問題となる．これを解決する手段として，冗長リンクを利用することでEthernetを使って高バンド幅で耐故障性のあるネットワークを構築するRI2Nというシステムが提案された．本稿ではRI2Nの耐故障性をユーザレベルで実現することを目的とし，その実装システムとしてUDP/IPを利用したRI2N/UDPを設計・実装する．実装したシステムの耐故障性とネットワーク性能について評価し，故障時にも安定して通信を継続できることを確認した．

RI2N/UDP: Fault-tolerant network for PC-clusters based on multi-link Ethernet

TAKAYUKI OKAMOTO,[†] SHIN'ICHI MIURA,[††] TAISUKE BOKU,[††]
MITSUHISA SATO[††] and DAISUKE TAKAHASHI[††]

The PC-cluster has been used for high performance computing with its high performance/cost ratio. To save the cost, Gigabit Ethernet is used mainly for intercommunication network. However, the reliability of Ethernet is not quite high because of not only the hardware failure but also tentative error on network switches. To solve this problem, we have proposed an interconnection network system based on multi-link Ethernet named RI2N. In this paper, we develop a user level implementation by UDP/IP named RI2N/UDP. Through the evaluation on the performance and fault-tolerance, it is shown that our system can keep the computation on the network link failure to provide a high reliability on the system.

1. はじめに

クラスタの性能を大きく左右する要素として，ノード間をつなぐネットワークの性能が挙げられる．Myrinet[1]やInfiniband[2]などのSystem Area Network（以下，SAN）と呼ばれるネットワークは，高バンド幅と低レイテンシのための設計がなされており，並列数値計算を行うことを目的するHPCクラスタに適している．しかし，一方でこれらのネットワーク製品は一般への需要が少なく，非常に高価であるという欠点がある．そのため，多くのPCクラスタではネットワークとしてGigabit Ethernet（以下，GbE）を利用している．GbEはSANに比べてバンド幅やレイテンシなどの性能面では劣るものの，一般に広く普及しているため導入のコストが非常に小さい．

一方，クラスタ向けのネットワークとしては性能だけでなく信頼性が高いことも重要である．計算の最終段階でのネットワークエラーによって全ての計算が無駄になる可能性もあり，特に長時間の並列処理を要求するアプリケーションではネットワークの信頼性が問題となる．そのため，クラスタの通信における耐故障性を向上させる研究が行われている[3]．ネットワークはコンピュータの筐体から外に出ている部分であり，ケーブルが抜ける，断線するといった故障の発生しやすい部分である．また，ケーブルだけでなくそれを束ねるスイッチが故障する場合もある．Ethernetスイッチの故障はネットワークの集約点での故障であるためそこに接続されたノード全てに被害が及ぶ．このように，ネットワークの故障はノード内での故障に比べて比較的起こりやすく，クラスタのネットワークに耐故

† 筑波大学第三学群 情報学類
College of Information Sciences, Third Cluster of Colleges, University of Tsukuba
†† 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

障性を持たせることは，クラスタ全体の耐故障性能を向上させる上で重要である．特にEthernetスイッチでは，そのプロトコル上の特性から，通信の集中等によって著しく性能が低下し，ハードウェア的には故障していなくてもリセット等の措置が必要となる一時的な故障（soft failure）が多い．

そこで，ネットワークを冗長化することによって高性能化と信頼性の向上を同時に実現する手法としてRI2N (Redundant Interconnection with Inexpensive Network)[4]が提案された．これは，複数のEthernetリンクを1つの論理ネットワークとしてユーザプログラムに提供することにより，リンクが正常な期間は負荷分散によるバンド幅の向上を行い，故障が発生した場合にも他の冗長リンクを利用して安定した通信を提供し続けることのできるネットワークである．

RI2Nの実装システムとしては，高バンド幅化のためのプロトタイプが作成された[4]．HPCアプリケーションの並列処理では，ストリーム通信の高速化と信頼性向上が重要である．しかし，一般にEthernet上でのTCP/IP通信では，ソケットによる固定的なchannelを張るため，故障時にプログラムそのものがロックすることや，故障の検出が行えなくなることがある．作成されたプロトタイプはTCP/IPのマルチストリーム上にRI2Nストリームを構築する実装方法であったため，このような問題が発生し故障時の性能が低下していた．

そこで我々は，耐故障性に重点を置いたRI2Nのユーザレベル実装システムとしてRI2N/UDPを設計し，実装と評価を行う．RI2N/UDPはUDP/IPを用いてユーザプログラムにストリーム通信を提供する．UDP/IPはコネクションレス型であるためTCP/IPでの前述のような問題を回避できる．さらに，RI2Nで実現するマルチリンクによる耐故障性では，パケットの再送のようにTCP/IPでのプロトコルと重複する処理が必要となる．したがって，TCP/IPでの処理（信頼性確保やフロー制御）も含めた形でUDP/IP上のRI2N/UDPに集約して実装する方が効率的であると考えられる．

一般に，このようなネットワークシステムの実装としてはデバイスドライバあるいはOSカーネルレベルでの実装（システムレベル）と，ソケットなどのAPIを利用したネットワークプログラミングによるライブラリとしての実装（ユーザレベル）が考えられる．システムレベル実装では，仮想ネットワークデバイスとしてユーザに提供でき，またシステムイベントを低コストでキャッチできるため，透過性が高く効率的な実装が可能である．しかし，その反面でシステムへの依存性が高くなり開発コストも大きくなるという問題がある．これに対し，ユーザレベル実装は移植性の高さという大きなメリットがあり，POSIX等の標準にあわせることによりユーザに負担をかけることなく多くのシステムで利用することができる．RI2Nと同じくマルチリンクを利用した負荷分散，耐故障性技術としてLink Aggregation[5]があるが，その利用にはLink Aggregationに対応したスイッチが必要である．また，Link Aggregationプロトコルは一組のスイッチ間またはスイッチ・ノード間でのみ有効なものであり，複数スイッチに跨る実装は不可能である．これは，接続先のスイッチに故障が発生すると通信の復旧は不可能になることを意味する．それに対して，ユーザレベル実装であるRI2N/UDPはスイッチに依存することはなく，またUDP/IPが利用できるネットワークであればEthernetでなくとも利用することができる．

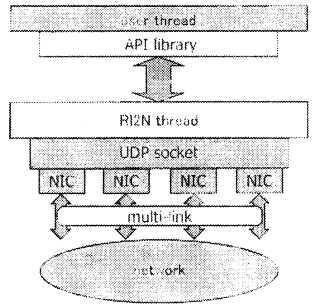

図1 RI2N/UDPの構成

本稿では，まずRI2N/UDPの設計について述べ，次に実装段階での種々の問題について述べる．そして，RI2N/UDPの耐故障性とネットワークとしての性能について評価と考察を行う．

2. 設　　計

2.1 RI2N/UDPの概要

RI2N/UDPはユーザプログラムに対して，TCP互換のAPIを通して信頼性のあるストリーム通信を提供するものとする．これをUDP/IPベースで実装するため，RI2N/UDPではTCP/IPと同様にコネクションの作成やロストパケットに対する再送を行う．これらはユーザプログラムと平行して処理を行う必要があるためユーザプログラムとは別に通信用スレッドを利用する．**図1**にRI2N/UDPの構成を示す．元々はユーザプログラムであった計算を行うスレッドをユーザスレッド，RI2Nの通信を行うスレッドをRI2Nスレッドと呼ぶ．RI2Nスレッドはノード内のユーザスレッドとの間で通信を行い，また，ネットワークを介して他のノードのRI2Nスレッドとも通信を行う．

図2にネットワークの構成を示す．各ノードに複数のNIC（Network Interface Card）を用意してNICの冗長化を行い，それらを互いに独立したネットワークに接続する．スイッチに故障が発生すると全ての通信が止まってしまうためリンクごとにそれぞれ別のスイッチを使用しスイッチも冗長化する．また，スイッ

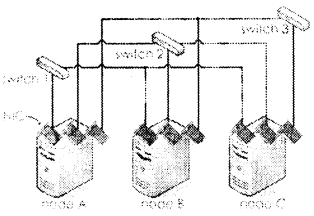

図 2 ネットワークの構成

チによって物理的に分割するだけでなく割り当てる IP アドレスもリンクごとに異なるネットワークアドレスにする．図 1 の通り RI2N/UDP では UDP ソケットで通信を行うため，NIC の MAC アドレスなど物理的な情報を使ってリンクを選択することができない．そこで，OS の IP ルーティングを利用する．IP ルーティングでは NIC に割り当てられたネットワークアドレスに従ってルーティングが行われるため，それぞれ違うネットワークアドレスにしておく．このようにしておけば，UDP ソケット上から使用するリンクを選択することができる．

2.2 ネットワークの故障と UDP/IP

RI2N/UDP は下位プロトコルとして TCP/IP ではなく UDP/IP を使用する．これはネットワーク故障時にも安定してシステムを稼働し続け，またより効率的な実装とするためである．耐故障性を実現する上でまず重要なことはネットワークに故障が発生してもユーザプログラムが安定して動作し続けられることである．UDP/IP はコネクションレス型であり，データの到着を保障していないため，ネットワークが故障しても単にパケットロスが増加するだけでシステムに影響はない．しかし，TCP/IP はコネクション型であり，パケットの再送や確認応答によって通信相手を意識した処理を行っているため，故障が発生するとシステムは何らかの形で影響を受ける．

故障が発生した後にはロストパケットの再送が必要になる．TCP/IP では，IP 層以下でロストしたパケットは OS カーネルによって再送が行われるため，通常のパケットロスに対して上位層が再送を意識する必要はない．しかし，故障が発生した場合には，その時故障したリンクのソケットバッファに保存されていたデータを別のリンクによって再送する処理が上位層で必要になる．そのため，通常のパケットロスと故障の両方に対する再送を同様に扱って UDP/IP 上に集約して実装する方が効率的であると考えられる．

2.3 RI2N プロトコル

RI2N/UDP ではノード間の通信に専用のプロトコルを用いる．これを RI2N プロトコルと呼ぶ．RI2N プロトコルは UDP/IP 上に実装した TCP に近いものであるが，RI2N/UDP ではマルチリンクを前提としており，また故障時の性能についても考慮する必要があるため，いくつかの点で TCP とは異なっている．

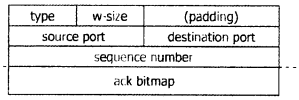

図 3 選択的確認応答パケットの構造

2.3.1 再送制御

RI2N/UDP ではマルチリンクを使って round-robin でリンクを選択してパケットを送信する．そのため，パケットの順序が数個分ずれて受信側に到着することがシングルリンクの場合と比べて非常に多い．また，故障時や soft failure 時にはパケットを 1 つおきや 2 つおきに落としてしまう．このような状態で，古くからの TCP と同様にある点から後のパケットを全て再送してしまうと，すでに受信できているパケットについても再送が行われ効率が悪い☆．これらの性質を考慮し，RI2N プロトコルではより大きな到着順序のずれを許容し，また到達確認応答には選択的確認応答を用いることによって再送するパケットの数を削減する．

RI2N プロトコルでは到着順序がどの程度ずれているかの確認は受信側で行う．そして，ある閾値を超えてずれるとそのパケットはロストしたものと判断して再送の要求を行う．この到着確認応答に使用するパケットの構造を**図 3** に示す．TCP とは異なり RI2N プロトコルのパケットには，確認応答番号専用に使うフィールドを用意しない．sequence number フィールドはデータパケットではそのパケットのシーケンス番号を格納するが，確認応答時には次に送信することを要求するパケットのシーケンス番号（以下，ACK 番号）を格納するために使う．また，その次の ack bitmap フィールドはパケットを選択的確認応答に指定するためのフィールドである．ACK 番号以降のパケットで各パケットが既に到着したか否かを各々1 ビットを使って表現したビットマップ情報を格納する．

2.3.2 故障の検出

RI2N/UDP では耐故障性を実現することを最も重視している．しかし，故障していない状態でも Ethernet を使っている以上，パケットロスは発生する場合がありそれを補完するために前節のような再送処理を行っている．この再送処理があるため，ネットワークに故障が発生しても正常に動作するリンクが 1 つ以上ある限り，RI2N/UDP 上では通信を続けることができる．しかし，この段階では特定のリンクが故障しているか否かを判断するわけではないため，round-robin による送信リンク選択の結果，次の送信機会でそのリンクにもパケットを送出してしまう．故障したリンクへ送出したパケットはすべて消えてしまい，通信効率が著しく低下する．これを回避するには，故障を検出しそのリンクでの送信を停止しなければならない．

RI2N プロトコルでは故障の検出をパケットの受信

☆ Linux kernel 2.4 以降では確認応答が実装されている．

側で行う．まず，受信側ではどのリンクからどれだけのパケットが到着したかを常に記録しておく．送出リンクはround-robinで選択されているため，すべてのリンクが正常である場合にはどのリンクに到着するパケットの数もほぼ等しくなる．この値が相対的に見て非常に少ないリンクがあればそれは故障もしくはそれに近いレベルのパケットロスを起こしているリンクであると判断できる．受信側でこのような評価を定期的に行い，故障が検出された場合にはまずそのリンクの使用を停止する．そして，検出した故障情報を通信相手に知らせる．故障情報パケットには通常のヘッダに加えて故障しているリンクのアドレス情報を格納する領域がある．このパケットを受け取った側はその情報を元に，リンクの使用と停止を判断する．そして，その確認のための応答パケットを送信する．この確認応答が届くまでは故障情報パケットも一定間隔ごとに再送する．これは，故障情報パケットがロストして相手に故障情報が伝わらなくなることを防ぐためである．

故障の検出と同時に故障が回復したことの検出も重要である．RI2Nプロトコルではリンクが故障から回復し再び使用できる状態になったことも自動的に検出する．これも前述の故障検知とほぼ同じ手法で行う．故障検知のために各リンクに届いたパケットの数をカウントしていることは前述の通りであるが，故障したリンクについてもこの処理を続けていく．また，1つでも故障したリンクがある状態では一定時間ごとにすべてのリンクに対して適当なパケットを送出するようにしておく．通常は故障リンクに対してパケットを送出しないため受信側でいくら待っていても受信パケット数は0のままであるが，このようにしておけば故障から回復したリンクからは何らかのパケットが受信できることになる．そこで，故障検出の場合と同様に定期的に各リンクからの受信パケット数を確認して，すでに故障と判断されているにもかかわらず受信パケットのあるリンクは故障から回復したリンクであると判断する．この回復情報も故障情報と同様に扱い，故障情報パケットを使って相手ノードに通知する．

3. 実　　装

RI2N/UDPをLinux上のユーザレベルライブラリとして実装した．スレッドの実装にはpthreadを用いた．pthreadによるスレッドプログラミングでは，スレッド間の同期を取るためにmutexや条件変数などの同期機構が用いられる場合が多い．しかし，RI2Nスレッドではユーザスレッドからの要求とネットワークの状況を同時に監視していなければならない．ネットワークの監視はselect()システムコールによって行うためpthreadの同期機構による同期と相性が悪い．そこで，RI2N/UDPではsocketpairを利用することでスレッド間の同期を行っている．socketpairはファイルディスクリプタとして扱うことができるためselect()システムコールによる監視ができる．しかし，スレッド間の全てのデータ転送をsocketpairを経由して行うとスループットの面で不利である．そのため，同期やメタ情報の通信はsocketpairで行い，実際のデータの受け渡しは通常のメモリコピーによって行っている．

RI2N/UDPではselect()システムコールによってネットワークとユーザスレッドの監視を行っている．リアルタイム性を追求するのであれば1回のselect()システムコールでは各ファイルディスクリプタとも1回しか入出力を行わず，毎回select()システムコールによる状況の確認をしてから処理を行うべきである．しかし，この方法ではselect()システムコールを頻繁に発行せねばならず，また一般的なx86系CPUでは1回当たり10μs近いオーバヘッドがある．そのため，RI2N/UDPでは非ブロッキング通信とselect()システムコールを組み合わせて利用している．非ブロッキング通信であればファイルディスクリプタの状況確認と実際の入出力を同時に行うことができるためオーバヘッドはほとんどない．そこで，複数のファイルディスクリプタを監視する処理はselect()システムコールを利用し，ファイルディスクリプタが決まってからの連続した入出力を制御するためには非ブロッキング通信を利用している．これによって，相対的にselect()システムコールのオーバヘッドが減少しスループットが向上する．

表1　測定環境

項目	スペック
CPU	Intel Xeon 2.8GHz 2-way x 2nodes
memory	DDR 1024MB
kernel	linux 2.6.12-1.1381
NIC	Intel PRO1000MT dual port 1000base-T (PCI-X 64bit/100MHz)
switch	Dell Power Connect 2724 (24 ports GbE switch)

4. 評　　価

RI2N/UDPの耐故障性能を評価するため，故障時のスループットの変化の様子を実験によって確認する．またネットワークとしての基本的な性能であるレイテンシとバンド幅についても測定する．測定環境は**表1**に示すとおり，2-way Xeonマシン2台を2つのGbEリンクによって接続した環境である．実際のシステムではスイッチの故障に対する耐故障性のためにリンクごとに別のスイッチを使用するべきであるが，今回はシステムの動作確認が目的であるため1台のL2スイッチをVLANによって仮想的な2つのスイッチに分けて使用する．

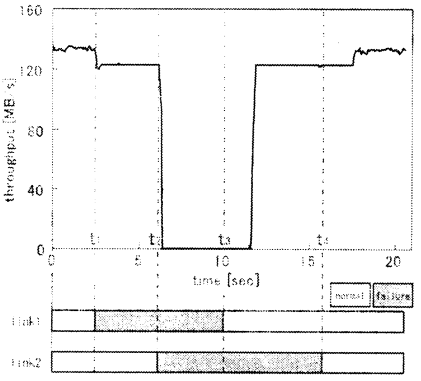

図 4　故障時のスループットの変化

4.1　故障時の変化

ネットワーク故障時にも安定して通信が続けられているか，また故障と回復が正しく検出できているかを実験によって確認する．今回は，これをスループットの変化によって確かめた．まず，一方向のバースト転送を長時間続け 100ms ごとにその時点のスループットを出力するプログラムを作成した．そして，そのプログラムを実行中にケーブルを抜き差しすることで意図的に故障を発生させ，故障時及び回復時のスループットの変化を観測した．

図 4 に測定結果を示す．横軸は時間，縦軸はその時点でのスループットを表している．グラフ下のバーは各リンクの状態（normal=正常か，failure=リンク切断）を表している．MTU は 6000byte で RI2N パケットのサイズは 5000byte とした．また，故障とみなすか否かの取得パケット数の比は 100:2 とし，いずれかのリンクでパケットを 100 個受信したときに受信パケット数が 2 より少ないリンクを故障とみなした．人の手による作業であるため時間はあまり正確ではないが，グラフ中の最初の時刻 t_1 で 1 つ目のケーブルを抜き，時刻 t_2 で 2 つ目のケーブルを抜いた．そして，時刻 t_3 で最初に抜いたケーブルを挿し，時刻 t_4 でもう一つのケーブルを挿した．

4.2　バンド幅

バンド幅の測定では，送信側と受信側を決め一方向のデータ転送に要する時間を測定してバンド幅を求めた．**図 5** に測定結果を示す．横軸は RI2N のパケットサイズ，縦軸はバンド幅である．各 NIC の MTU は 6000byte に設定し，2 リンクを使用した場合と 1 リンクのみの場合のそれぞれについて測定した．この 1 リンクの状態は故障時の場合とは違って，最初から 1 つのリンクしかないものとして通信を行った場合である．測定の結果，2 リンクでのバンド幅の最大値は 136MB/s であった．また，同じ環境の 1 リンク上で TCP による通信を行った場合のバンド幅は 123MB/s であった．

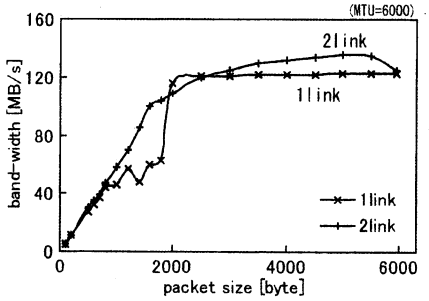

図 5　バンド幅の測定結果

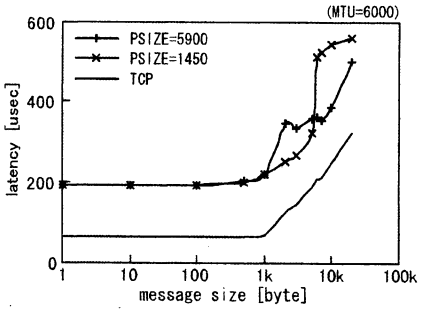

図 6　レイテンシの測定結果

4.3　レイテンシ

レイテンシはノード間でメッセージの ping-pong を 1000 回行ってそれに要する時間から算出した．**図 6** に測定結果を示す．横軸は ping-pong するメッセージの大きさ，縦軸はレイテンシである．MTU は 6000byte で RI2N のパケットサイズを 5900byte にした場合と 1450byte にした場合，および TCP を用いた場合の測定を行った．RI2N/UDP では最小で 190μs，TCP では 62μs のレイテンシがあった．

5.　考　　察

故障および回復時のスループットの変化について見てみると，RI2N/UDP では 1 つでも正常なリンクがあれば通信を継続できていることがわかる．図 4 を見ると，時刻 t_1 でスループットが階段状に低下している．この時点で故障（ケーブルを抜くこと）を起こすため，これ以後は使用可能なリンクが 1 つになりスループットが低下している．しかし，故障検出後にも安定して 120MB/s 程度を継続している．このスループットは図 5 より 1 リンクでの最大のバンド幅と同等であり，故障検出によって設計通りに故障リンクの使用を停止していることがわかる．

また，故障検知にかかる時間も十分に短いものであるといえる．予備実験によれば故障リンクの使用を停止せずに round-robin による送信選択から外さなかっ

た場合のスループットは 10MB/s 程度という非常に低い値であったが，図 5 を見る限りにおいてそのような測定点はない．これは，測定間隔の 100ms よりも十分に短い時間内に故障が検出され 1 リンクでの動作に切り替えられたためであると考えられる．

回復の検出について見てみると，故障の検出ほど短時間では行われていない．図 4 の時刻 t_3 で link1 が正常な状態に戻されているが，その後約 2 秒間のスループットは 0 のままである．また，link2 を正常に戻した時刻 t_4 でも同様に，ケーブルを挿してからそれがスループットに反映されるまでに数秒程度の時間を要している．これは，ハートビートの間隔を 3 秒ごとにしているためである．この間隔を短くすることは可能であるが，その場合はネットワークに流れるパケットの量が多くなり性能に影響が出る可能性がある．また多くの場合，故障からの回復には人の手による物理的な作業が必要になるため，人間の作業にかかる時間を基準とすればこの程度の間隔で十分であると考えられる．

バンド幅の測定結果について見てみる．図 5 ではほとんどの部分で 1 リンクよりも 2 リンクの方が高いバンド幅を示している．また，2 リンクを使用した場合には最大バンド幅で 1 リンクの TCP に勝っている．しかし，その向上の度合いは 10 %ほどにとどまっており物理的に帯域が 2 倍になっていることを考えるとマルチリンクによるバンド幅の向上は少ない．ここには示していないが，バンド幅と同時に測定したパケット損失率は RI2N パケットサイズが 5000byte 以上の範囲では 0.1%にも満たない．パケットロスがないということから，受信側の受信タイミングがパケットの到着間隔に追いついていないのではなく，送信側のパケット送出速度の遅さがバンド幅のボトルネックになっているものと考えられる．

レイテンシの測定結果はシステム構成から予想される以上に悪い結果になった．UDP のレイテンシと TCP のレイテンシはほぼ同程度であるため，RI2N/UDP が UDP 上に実装されている以上はレイテンシ性能が TCP よりも良くならないのは当然である．しかし，スレッド間通信のレイテンシなどを含めたとしても 200μs という値は考え難い．ネットワークのレイテンシ 60μs に，スレッド間通信によるレイテンシ 20μs の送信側と受信側の 2 回分を加えて，100 から 120μs 程度がシステム構成から考えられる妥当なレイテンシであると思われる．このような結果となった原因はプロトコルの設計にあると考えられる．現在のプロトコルでは，メッセージサイズが小さい場合には相対的に多くの確認応答パケットが送信されることになる．これは，バンド幅を重要視して連続した大きなメッセージを扱うことを前提にプロトコルの設計を行っているためである．そこで，メッセージサイズが小さい場合はレイテンシが優先されるように，確認応答パケットを意図的に遅らせて通信のない期間にまとめて送信するといった変更を加えれば，レイテンシは改善されるものと考えられる．

6. まとめ

本稿では RI2N のユーザレベル実装システムとして RI2N/UDP を設計し実装と評価を行った．耐故障性能については，故障時にも安定して通信を継続できていることが確認でき，また，故障によるスループットの低下も最低限に抑えられていることが確認できた．ネットワークの性能としては，GbE リンク 2 本の上で 136MB/s のバンド幅を達成し，マルチリンクによるバンド幅の向上が確認できた．しかし，これは理論性能の 55 %程度の値であり更なる改良が求められる．レイテンシは 200μs とシステム構成から予想以上に悪い結果となってしまったが，設計の改良によって改善の余地はあるものと思われる．

今後は耐故障性のための機能拡張として，故障情報をユーザに通知する機能や上位のクラスタ管理システムとの連携についても検討する．また，バンド幅の向上とレイテンシの低減のためにメモリコピーの回数を減らすように設計および実装を見直す．

謝辞 本研究を行うにあたり，貴重な助言をいただいた CREST 「メガスケールクラスタ研究チーム」のメンバーに深く感謝します．本研究の一部は科学技術振興機構「戦略的創造研究推進事業 (CREST) -情報社会を支える新しい高性能情報処理技術-『超低電力化技術によるディペンダブルメガスケールコンピューティング』」及び文部科学省 科学研究費補助 (基盤研究 (C)17500031 及び基盤研究 (A)17200002) による．

参考文献

1) N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic and W.-K. Su: Myrinet: A Gigabit-per-Second Local Area Network., *IEEE Micro*, Vol. 15, No. 1, pp. 29–36 (1995).
2) InfiniBand Trade Association: InfiniBand. http://www.infinibandta.org/.
3) R. T. Aulwes, D. J. Daniel, N. N. Desai, R. L. Graham, L. D. Risinger, M. W. Sukalski and M. A. Taylor: Network Fault Tolerance in LA-MPI., *PVM/MPI*, pp. 344–351 (2003).
4) S. Miura, T. Boku, M. Sato and D. Takahashi: RI2N - Interconnection Network System for Clusters with Wide-Bandwidth and Fault-Tolerancy Based on Multiple Links., *ISHPC*, pp. 342–351 (2003).
5) IEEE: IEEE 802.3ad "Link Aggregation" (2000).