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

2006－HPC－107 (21)
2006／8／1

UDPによるファイアウォール越えを用いたP2Pオーバーレイネットワーク

岡本高幸† 朴泰祐†
佐藤三久† 建部修見†

家庭やオフィスの遊休PCは潜在的に大きな計算能力を有しており，これらを接続して効率的に利用することができれば非常に大きな計算資源となる．しかし，NATやファイアウォールの中にあるこれらのPCを相互に接続するには，物理的なIPアドレスに依存しないノード識別子によるルーティング処理やUPnP，hole punchingなどのNAT越えの技術が必要である．これらをアプリケーションごとに実装していくことは煩雑であり，P2Pアプリケーションの開発における問題となっている．そこで本稿では，アプリケーションをネットワークの物理構成から独立させ，物理ネットワークに依存せず参加するすべてのノードを等しく接続可能とするオーバーレイネットワークを提案する．そして，その実現に必要なNAT越え技術の一つであるUDP hole punchingについてのテストシステムを作成し，市販の家庭用ルータを用いて性能評価を行った．UDP hole punchingと独自のライブラリを使うことによってTCPと比べて2割程度のスループットの低下でNATを越えて直接通信が実現できることを確認した．

P2P Overlay Network based on UDP Firewall Traversal

TAKAYUKI OKAMOTO,† TAISUKE BOKU,† MITSUHISA SATO†
and OSAMU TATEBE†

An enormous number of PCs at home or office potentially implies a great amount of computation power when they are out of the work, and there is an opportunity to utilize their power for a large scale computation. However, these machines usually exist behind the NAT or firewall and it requires various techniques to access and connect them, such as logical naming independent from the original IP addresses, efficient routing, or NAT traversing with UPnP or UDP hole punching. It is troublesome to apply these techniques adequately to each application, and this is a hazard in the development of P2P application. In this paper, we propose an overlay network to connect all attending nodes in logically flat layer independently from their physical network in order to encourage the easy development of various P2P applications. In our system, we implement a generic communication library based on UDP hole punching which is one of the most common NAT traversal techniques, and evaluated the communication performance on commodity personal broadband router widely used at home. We developed an original communication layer only with UDP protocol which is basically compatible with TCP. By the direct communication through NAT box without intermediate relay server, we confirmed that our method provides a communication performance with only about 20% of performance degradation compared with TCP communication.

1. はじめに

家庭や企業に散在するPCは非常に多く，遊休状態のPCをインターネット上で接続し分散コンピューティングに利用することができれば，非常に大きな計算資源となる．しかし，これらのPCは多くの場合，NAT（Network Address Translation）やファイアウォールにより保護されたネットワーク環境にあり，他のドメインからのインバウンド接続を受けることができない．ボランティア・コンピューティング等において数万台規模のPCを接続するためにはP2Pによるスケーラブルなネットワークが必要であり，すべてのピアがインバウンド接続できることが望ましい．そのため，ファイアウォールやNATを超えて直接接続を可能にするUPnP[1]やhole punching[2]などのNAT越えの技術が必要となる．そのなかでもUDP hole punchingは本来のNATの機能を利用して実現する手法で，多くのNAT box（家庭用ルータ）で利用できる．しかし，

† 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

UDP hole punching では UDP/IP による通信しか行うことができない．そのため，これを一般の P2P アプリケーションで利用するには UDP/IP 上で信頼性を保障する処理が必要となる．

このように，大規模な P2P コンピューティングのためのネットワークを構築することは容易ではない．そこで我々は，参加するすべてのノードが同じ手順で等しく通信することを可能にするオーバーレイネットワークを提案する．既存のネットワークの上に独自の名前空間によってルーティングを行うオーバーレイネットワークを構築し，物理的なネットワークの構成によらずアプリケーションを記述できる API を提供する．そして，通信レイヤには UPnP や UDP hole punching などを実装し，効率的な通信ができるようにそれらを自動的に選択して利用する．主にターゲットとするアプリケーションとしては，比較的 1 つのジョブの計算時間が長く，大きなデータを必要とするような科学技術計算を考えている．そのため，レイテンシ性能よりもバルク転送のスループットを高いレベルで維持し，グローバルネットワーク上で効率的に通信を行うことが求められる．本稿ではその初期段階として，UDP 上で信頼性のある通信を行うライブラリを開発した．そして，そのライブラリと UDP hole punching を用いて，互いに異なる NAT の下にあるノード同士が直接通信を行うことを可能にするテストシステムを構築しその性能評価を行った．

まず，2 章で現在のインターネット環境で P2P ネットワークを構築する場合の問題について述べ，3 章でこれを解決するためのオーバーレイネットワークを提案する．4 章と 5 章では，そのオーバーレイネットワークの実装に用いる NAT 越えの技術について述べる．そして，6 章でテスト環境における通信性能の評価について述べ，7 章でまとめる．

2. P2P コンピューティング

現在，ADSL や FTTH の普及によって，家庭やオフィスからインターネットに常時接続されたコンピュータが非常に多くなっている．これは，インターネット上で定常的に接続された計算資源が多いことを意味し，BOINC[3] などインターネット上で計算資源を確保する分散コンピューティングの大きな原動力となっている．しかし，サーバ・クライアントモデルによるネットワークではスケーラビリティの問題から実現可能なアプリケーションが限られている．また，サーバが single point of failure となり耐故障性についても問題がある．そのため，拡張性や耐故障性に優れる P2P コンピューティングが期待されている．しかし，インターネットに接続されているコンピュータの多くはファイアウォールや NAT の内側にあり，接続の自由度が制限された状態にある．P2P ネットワークは，ピア同士が直接通信を行うことによって実現されるネットワークであるため，インバウンド接続を制限された環境では十分な性能が発揮できない．

一方，NAT の下にあるコンピュータに対してインバウンドコネクションを作成する NAT 越えと呼ばれる技術がある．UPnP はネットワークを通してルータや家電製品の設定を行うための技術であり，これに対応した NAT box を用いることでプログラムから自動的に NAT の設定を変更できる．一方，hole punching は，NAT のアウトバウンド通信に付随する副作用を使って直接通信を行う方法である．hole punching には TCP を使うものと UDP を使うものの 2 種類があるが，TCP hole punching を行うには TCP ヘッダを操作したパケットを送信しなければならないためドライバレベルでの対応が必要である．それに対して UDP hole punching は，IP アドレスとポート番号の設定のみで実現できるためユーザレベルで実装することができる．また，UDP hole punching は TCP hole punching よりも多くの NAT box で利用可能であるという報告もある[2]．

実際にどのような NAT 越えの技術を選択するかを考えた場合，UDP hole punching は NAT 本来の機能を利用して実現されているため多くの NAT で利用できると考えられ，また，現在市販されている家庭用の NAT box ではそのほとんどで UPnP がサポートされている．そのため，この 2 つの方式をサポートすればリレーノードを必要とする状況はほとんどなくなると考えられる．

3. NAT 越えを用いたスケーラブルなオーバーレイネットワークの提案

前述の通り多くのコンピュータは NAT やファイアウォールによって保護された環境でインターネットに接続されている．そこで本稿では，参加するすべてのノードが LAN 上と同じように等しく接続可能なオーバーレイネットワークを提案する．オーバーレイネットワークとは，既存の物理ネットワークの上に構築する，独立した名前空間やルーティング機能を持ったネットワークのことである．この上では物理ネットワークに依存せず，どのノードとも同じように通信を記述できる．本稿で提案するオーバーレイネットワークは単にフラットなネットワーク環境を提供するだけでなく，ルーティングシステムの分散管理によって高いスケーラビリティを実現し，NAT 越えの技術を用いたノード同士の直接接続によって実ネットワーク上で効率の良い通信を行う．

3.1 通信モデル

アプリケーションにはインターフェイスとして TCP ライクなソケット API を提供する．これにより，UDP/IP をそのまま用いる場合のように信頼性や

パケットの流量制御に傾注する必要がなく，従来のTCP/IPを利用した幅広いアプリケーションを容易に移植可能となる．そのため，基本的に1対1の通信モデルのみ提供する．P2Pネットワーク向けの集合通信（ブロードキャストやノードリストの取得など）はアプリケーション側で実装するものとする．

3.2 名前空間

本システムはオーバーレイネットワーク上で“IPアドレス+ポート番号”の代わりとなる独自の識別子（以下，IDと呼ぶ）を提供する．IDはオーバーレイネットワーク上のエンドポイントに対して付けられる名前で，これを指定することによって一意に通信相手を特定することができる．オーバーレイネットワーク上に新たなエンドポイントを作成するときにはそれに一意なIDを与える．IDはアプリケーション側から値を指定することもできるが重複は許可しない．アプリケーションから指定しない場合にはランダムかつユニークなIDがシステムよって割り振られる（IPネットワークのDHCPに相当）．ノードの故障などによって，実ネットワーク上の別の場所で再度同じIDを利用したい場合には，これまで登録されていたエンドポイントがすでに到達不可能な状態になっていることを確認してから再利用を許可する．

3.3 ルーティングとノードの管理構成

図1は提案するオーバーレイネットワークが，実際の通信レイヤで構成するノードの管理構造の例である．

まず，インバウンド接続が可能で長期的に利用できるノードをスーパーノードに設定する．各一般ノードはそれぞれひとつのスーパーノードの管理下に置かれ，管理するスーパーノードにアウトバウンド接続して他のノードからの接続要求を待ち受ける．これは，一般ノードがインバウンド接続を受けられないノードである可能性があるためである．この構成では，接続開始時にスーパーノードを経由した通信を行うことになるが，実際のデータ通信を行うコネクションはNAT越えの技術を用いた直接接続になるため，スーパーノードのスループットが通信のボトルネックとなることはない．また，スーパーノードと一般ノードが実ネットワーク上で非常に遠い位置にあったとしても，接続開始時にしかその影響は受けない．

次に，どの一般ノードがどのスーパーノードで管理されているのかを分散環境上で効率よく検索するため，スーパーノード間でDHT（Distributed Hash Table）[4]を構成する．すべてのスーパーノードはインバウンド接続を受けることが可能であるため，既存のDHTシステムを利用することができる．このDHT上で，一般ノードのIDをkeyにして，そのノードを管理するスーパーノードの実アドレス情報を検索する．

多数の一般ノードが一部のスーパーノードに偏ると，そのスーパーノードがボトルネックとなる．そのため，一般ノードは分散して管理されることが望ましい．本システムでは，先にも述べたとおりスーパーノードと一般ノードが物理ネットワーク上で遠くとも問題はない．そこで，単純にIDのハッシュ値から管理するスーパーノードを決定する．ハッシュ空間がスーパーノードによって等間隔に分割されていれば，各スーパーノードが管理する一般ノードの数は確率的に均等になる．

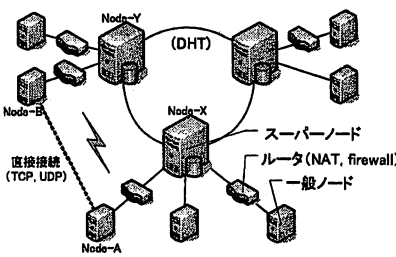

図1 オーバーレイネットワークの管理構造

3.4 一般ノード間の接続方式

一般ノード間で実際に通信を行う場合には次のような4つの接続方式を適宜選択して用いる．

(1) 単純なTCPによる接続（NATによるポートフォワーディング設定も含む）
(2) UPnPを用いたTCPによる接続
(3) UDP hole punchingを用いたUDPによる接続
(4) リレーノードが中継するTCPによる接続

(1)～(3)の方式については，広域ネットワークで測定したスループットの経験からどれが適しているかの優先度付けを行う．(4)については他の方式よりも明らかに多くの資源を無駄に消費してしまうため，優先度は最も低い．

4. UDP hole punching

UDP hole punchingはSkype[5]などのP2Pアプリケーションで利用されている一般的なNAT越えの技術である．UDP通信を許可しているNATでは，ローカルノードからアウトバウンドのUDPパケットが送出されたときにその送信元，送信先IPアドレスとポート番号を一定時間記憶しておく．そして，NAT上の送出元ポートに外から送られてきたパケットを，最初のパケットの送信ノードへ転送するという処理が行われる．そのためアウトバウンドパケットから始まる通信はNATを通しても利用可能なのである．しかし，このままでは通信を行うノードが両方ともNATの内側にあった場合には通信ができない．これを例を用いて示す．**図2**はNode-AとNode-Bがそれぞれ別のNAT boxに接続されており，Node-Cだけがグローバルネットワークに直接接続している様子を表している．Node-Aのポート1111（以下Node-A:1111と表す）からNode-C:3333へパケットを送った場合，Node-C:3333からのパケットをNode-A:1111に転送

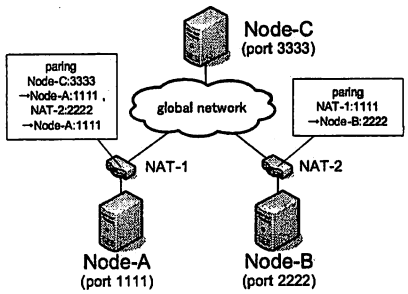

図 2 UDP hole punching の仕組み

するように NAT-1 が設定される．そのため，Node-C からの応答パケットが NAT-1 に届くと Node-A に転送される．一方，Node-A:1111 から NAT-2:2222 へパケットを送った場合，同じように NAT-1 上では転送の設定が行われるが，NAT-2 のポート 2222 は，まだどのローカルノードとも関連付けられていないため NAT-2 でパケットは破棄される．

UDP hole punching では通信を行う両者が互いにパケットを送信することでこの問題を解決する．先の例で NAT-2 でパケットが破棄されてしまうのは NAT-2 のポート 2222 にパケット転送用の設定がされていないためである．そこで，相手に届けるためではなくパケット転送の設定をするために Node-B から NAT-1 へ一度アウトバウンドのパケットを送信するのである．上記の Node-A から NAT-2 への送信の前に，Node-B:2222 から NAT-1:1111 へパケットを送ったとするとそのパケット自体は破棄されてしまう．しかし，NAT-2 上には "NAT-1:1111 から NAT-2:2222 へのパケットは Node-B:2222 へ転送する" という設定がされるため，その後に NAT-1 から送られてくるパケットが先の結果とは違って Node-B へ転送される．さらに，もう一度 Node-B:2222 から NAT-1:1111 へパケットを送ると今度は NAT-1 にも転送設定がされているためそのパケットは Node-A へ転送され，結果として Node-A と Node-B の間で直接通信が実現するのである．

UDP hole punching を実現するためには，送られてくるであろうパケットの送信元と送信先を正確に知り，"反対"のパケットを送り返す必要がある．そのためには，パケットを送信したときに NAT 上で使用される送信元ポート番号を知ることと，別の経路を使ってその情報を交換することが必要である．図 2 の例では NAT 上のポート番号はローカルノードのポート番号と等しくなるものとしたが，実際には NAT box ごとに対応付けは異なっており，hole punching を行う前にスーパーノードを介して対応付けの仕方を調査し結果を交換する．

5. UDP 上での reliable な通信

UDP hole punching は UDP/IP 通信のための NAT 越えの技術である．しかし，一般のアプリケーションでは信頼性のある通信が求められており，本稿で提案するオーバーレイネットワークでもアプリケーションプログラムに対して TCP/IP と同じような信頼性のあるストリーム通信を提供する．そのため，UDP hole punching を使うには UDP 上でパケットの到達確認や再送を行って信頼性を確保する処理が必要になる．また，スループットを重視する場合，フロー制御を行って効率的に通信が行えるようにしなければならない．

UDP 上で信頼性のある通信を行うプロトコルとして Reliable UDP[6] がある．しかし，到達確認と再送処理についてしか規定されておらず，また，利用可能な実装システムも公開されていない．そこで本研究では，我々が別の研究プロジェクト[7] で開発している RI2N/UDP[8] というライブラリを改良して独自に通信ライブラリを開発した（以下，これを RUDP ライブラリと呼ぶ）．RI2N（Redundant Interconnection with Inexpensive Network）は Ethernet のマルチリンクを利用して耐故障性と高バンド幅化を実現するクラスタ向けネットワークで，RI2N/UDP はこれを UDP/IP 上のユーザレベルライブラリとして実装したものである．RI2N/UDP ではアプリケーションプログラムに対して TCP と同等の API を提供するためにライブラリ内でパケットの到達確認や再送処理，輻輳制御などを行っている．RUDP ライブラリではこの大部分をそのまま利用し，UDP ポート番号の設定等の部分に多少の変更を加えることで UDP/IP 上での信頼性のある通信を実現する．本稿ではこの RUDP ライブラリと UDP hole punching を組み合わせることによって，UDP/IP による信頼性のある直接通信を実現する．

6. 通信性能の評価

実装前の予備評価として通信性能の評価を行う．家庭用の NAT box を用いて TCP/IP の通信性能と UDP hole punching + RUDP の通信性能を測定し，直接接続の有効性について評価する．

6.1 評価環境

評価は**図 3** のような LAN 上のテスト環境で行う．図中の "Test network" とある部分を仮想的なグローバルネットワークであるとする．ただし，今回の評価では Test network は 1 台の Fast Ethernet スイッチで構成し，本来のグローバルネットワークのような大きな遅延は再現しない．server はグローバル IP アドレスを持っているが，Node-A，Node-B はそれぞれ異なる NAT box を通してグローバルネットワークに接続しており，それ自身はグローバル IP アドレスを持っ

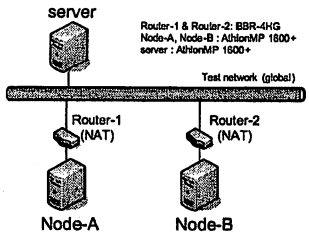

図 3 評価環境

ていないという状況を想定している．そのため A-B 間で通信を行うには NAT の設定や hole punching，リレーなどが必要となる．NIC はすべて Intel PRO/100 を用いる．2 台の NAT box には家庭用に市販されている BUFFALO BroadStation BBR-4HG を用いる．

6.2 評価方法

評価は Node-A と Node-B 間の通信性能の測定によって行う．A-B 間の通信の方法としては以下の 4 つの方法を用いる．

(1) NAT box を手動で設定して，TCP で通信する方法（TCP DMZ）
(2) NAT box を手動で設定して，UDP 上で RUDP ライブラリを使って通信する方法（RUDP DMZ）
(3) 両方から server に TCP アウトバウンド接続をして，server に通信をリレーさせる方法（TCP relay）
(4) server でアドレス情報を交換して UDP hole punching によって NAT 越えを行い，RUDP ライブラリを使って通信する方法（UDP hole punching + RUDP）

それぞれの方法で，レイテンシと最大スループット，接続開始にかかる時間を測定する．

レイテンシは，Node-A と Node-B の間で 1000 回の ping-pong 通信を行い，それに要した時間から求める．最大スループットは A-B 間で 100MB 程度のデータを片方向転送し，それに要する時間から求める．接続開始にかかる時間は，通信相手が接続待ちの状態で connect() を呼んだときにかかる時間を 10 回測定し，その最小値を測定結果とする．ただし，UDP hole punching + RUDP については実装上の問題から，それぞれ独立に実行したときの時間しか測定できない．そのため，UDP hole punching 自体にかかる時間のみ測定する．実際にはその結果に RUDP 単体での接続開始時間を加えたものが UDP hole punching + RUDP の接続開始にかかる時間になる．

6.3 結果と考察

レイテンシの測定結果を図 4 に示す．最もレイテンシが短かったのはメッセージサイズが 100byte の時の UDP hole punching + RUDP の結果で，335μs であった．しかし，その大きさに対して，他の方式と

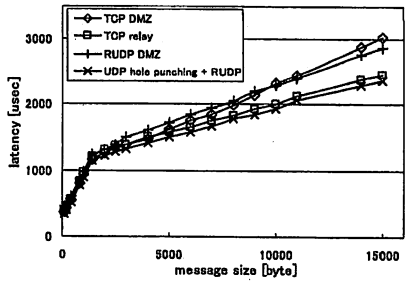

図 4 2 ノード間のレイテンシ測定結果

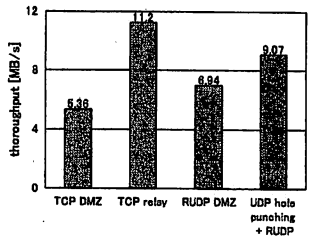

図 5 2 ノード間のスループット

の差は小さい．一般にスイッチのみで接続した Fast Ethernet の TCP，UDP のレイテンシは 100μs に満たないが，図 4 の結果では MTU（1500byte）以下のメッセージでも 1ms 以上の時間がかかっている．このことから，この実験におけるレイテンシはそのほとんどがルータを通したことによるもので，各 NAT box から server の間のレイテンシ，つまり，TCP relay によって余分に増えるレイテンシは相対的に小さいことがわかる．実際に広域ネットワーク上で評価した場合には各ルータと server との間に大きな遅延があるので今回とは違った結果が得られるものと考えられる．今回の測定の目的は細かなレイテンシの大小を比較することではなく，TCP 以外の各方式を用いることによって致命的な問題が発生しないことを確認することである．その点について見てみると，今回の測定では各方式によってレイテンシに大きな違いは見られず，TCP の代わりに RUDP を用いたとしてもレイテンシに問題はないことがわかる．また，リレー通信と比べれば，広域ネットワーク上ではレイテンシが小さくなることも期待できる．

スループットの測定結果を図 5 に示す．測定結果から TCP relay が 11.2MB/s で最もスループットが高いことがわかる．一方，もっともスループットが低いのも TCP/IP を用いた TCP DMZ であった．また，RUDP ライブラリを用いた 2 つの方式でも手動でポートフォワーディングの設定をした RUDP DMZ の方がスループットが低い．これについては，あわせて行った Iperf 2.0.2 による server と各ノード間のバンド幅測定でも同様の結果が出ている．server をサーバ，各ノードをクライアントとした場合には，TCP，UDP

表 1 接続開始に要する時間

接続方式	接続開始時間 [ms]
TCP DMZ	0.914
TCP relay	7.46
RUDP DMZ	1.65
UDP hole punching	8.89

とも 90Mbps 以上であったが，サーバとクライアントを逆にした場合，つまり，ルータにポートフォワーディングの設定をして server から各ノードに接続した場合には TCP で 45Mbps 程度，UDP で 75Mbps 程度に低下した．このことから，今回使用した NAT box，BBR-4HG では，手動でポートフォワーディングを設定した場合は，NAT の機能によって自動的に設定される場合よりもスループットが低くなるという性質があると考えられる．この性質のため今回の測定結果では，手動で NAT のポートフォワーディングを設定した場合には TCP を利用するよりも RUDP ライブラリを利用した方が高いスループットが得られている．この性質が家庭用の NAT box 全般にあるとは言えないが，TCP よりも UDP の方が，NAT box の負荷が小さくなると考えれば，RUDP ライブラリの方が高いスループットが得られる場合もあり得る．また、UDP hole punching + RUDP は、TCP relay と比較してもスループットの低下は 2 割程度であり，リレーノードを必要とする TCP relay よりも大規模な P2P コンピューティングに適しているといえる．

接続の開始に要する時間の測定結果を**表 1** に示す．接続の開始にかかる時間は TCP DMZ が 0.944ms で最も短かった．これに対して，最も時間がかかった UDP hole punching はその 10 倍程度の 8.89ms であった．最終的なオーバーレイネットワークのフレームワークでは，これらの時間にノードの探索やスーパーノードによる仲介処理の時間を加えたものが実際の接続開始に要する時間となる．これらはグローバルネットワーク上で複数回のパケットの交換を必要とする処理であり，今回の測定結果に比べて十分に大きなものになる．しかし，本研究でターゲットとしているアプリケーションは，1 つのジョブの計算時間が数分以上あり，また比較的大きなデータを必要とするものであるため，接続開始にかかる時間はある程度長くとも許容される．これらのことから，今回測定した各接続方式による接続開始時間の差は提案システムにおける要求に対して問題のないレベルであるといえる．

7. ま と め

本稿では NAT やファイアウォールを意識せずに通信を行うことを可能にするオーバーレイネットワークを提案した．また，そのオーバーレイネットワークの通信方式の一部となる UDP hole punching を用いた通信方式を実装し LAN 上でその性能を測定した．その結果，スループットを重視するアプリケーションにおいて既存の TCP 通信と比べても十分な性能があることを確認した．

しかし，実装に用いる通信方式の一つである UPnP で NAT の設定を行う方式について性能測定ができていない．また，本稿での評価は LAN 上の評価環境で行ったものであり，グローバルネットワーク上でのレイテンシが再現できていない．そのため，UPnP を用いた方式も含めて，グローバルネットワーク上での性能評価を行う必要があると考えられる．UDP/IP 上で信頼性のある通信を実現するために用いた RUDP ライブラリも元々クラスタ上での利用を目的としたライブラリであるため，グローバルネットワーク上で使用した場合には著しく性能が低下する可能性がある．その場合にはフロー制御等の処理に改良を加える必要があると考えられる．

謝辞 本研究の一部は文部科学省 科学研究費補助（基盤研究 (A)17200002 及び基盤研究 (C)17500031）による．

参 考 文 献

1) UPnP Forum: UPnP. http://www.upnp.org.
2) B. Ford, P. Srisuresh and D. Kegel: Peer-to-Peer Communication Across Network Address Translators, *Proceedings of the 2005 USENIX Annual Technical Conference*.
3) D. Anderson: BOINC: A System for Public-Resource Computing and Storage, *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pp. 4–10 (2004).
4) I. Stoica, R. Morris, D. Karger, M. Kaashoek and H. Balakrishnan: Chord: A scalable peer-to-peer lookup service for internet applications, *Proceedings of the 2001 SIGCOMM conference*, Vol. 31, No. 4, pp. 149–160 (2001).
5) Skype Technologies SA: skype. http://www.skype.com.
6) T. Bova and T. Krivoruchka: Reliable UDP Protocol, *"draft-ietf-sigtran-reliable-udp-00.txt"* (1999).
7) H. Nakashima, H. Nakamura, M. Sato, T. Boku, S. Matsuoka, D. Takahashi and Y. Hotta: MegaProto: 1 TFlops/10kW Rack Is Feasible Even with Only Commodity Technology, *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pp. 28–28 (2005).
8) 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, 情報処理学会研究報告, 06-HPC-105, pp. 85–90 (2006).