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

2007－HPC－111 (43)
2007／8／3

プライベートネットワーク内のノードをサーバとして外部に公開するための機構

山本剛之† 建部修見† 佐藤三久†

多くの未使用資源がプライベートネットワークの中に存在しているが，これらは外部ネットワークからの接続を受け入れることができないという制約を持つ．このためプライベートネットワーク内にサーバを設置しても外部へ公開することができなかった．そこでこれらの資源を利用するための機構を検討する．このような通信制約を取り除く機構としてはすでにいくつか研究があるが，本研究の特徴はネットワーク機器の設定を極力少なくしていることとサーバ側のコードはプライベートネットワークについて一切意識せずに書かれた既存のものを使用できるということである．実際にこの機構を Gfarm ファイルシステムに実装し，プライベートネットワーク内のファイルサーバを外部ネットワークから使用できることを確認した．

Mechanism for Using Nodes Inside Private Network as Public Servers

Takashi Yamamoto,† Osamu Tatebe† and Mitsuhisa Sato †

Many idle resources are present in private network, however, these can not accept connection from exterior network. Because of this restriction it was impossible for server inside a private network to be accessed from other network. Number of research have been already made to remove this limitation. Our approach is to accomplish this task by making minimal configuration to network devices, and without rewriting any part of the server code. We have implemented this system to Gfarm file system, and we were able to confirm that file server inside private network properly functioned when accessed from exterior network.

1. はじめに

一般家庭における PC の普及率はここ十数年で飛躍的に増加している．そしてこれらの PC の基本性能もまた急速に進化している．しかし一般ユーザが PC を使う主な目的はホームページ閲覧や文章作成といった，比較的 CPU に対する負荷が小さいものである．またストレージに関してもほぼ同様であり，現在市販されているデスクトップ PC は 100GB を超える容量の HDD を搭載するものがほとんどであるが，それほどの容量を要求するアプリケーションは限られている．

そこでこれらの未使用の資源を有効利用することを考える．ブロードバンド接続の普及により，これらの資源に対してはインターネットを経由してアクセスすることが可能である．すでに実用化されているものとして，CPU に関しては BOINC[2] といったボランティアコンピューティング，ストレージに関しては Gnutella[3] などの P2P ファイル共有アプリケーションが代表的である．

しかし一般家庭の PC を利用するには非対称ネットワークについて考える必要がある．NAT (Network Address Translation) がその代表である，例えばブロードバンドルータを導入しているような環境では PC はプライベートネットワークの中に存在するため外部ネットワークからの接続要求を受け取ることができない．プライベートネットワークの中から外部へ接続要求を出すことはできるが，このように通信を開始できるのは一方からのみとなってしまっている．サーバのようにクライアントからの接続を受け入れ，要求を処理するようなノードとして利用したいときにこのような制約はアプリケーションの設計に大きな影響を及ぼす．

本研究では家庭の PC に着目して，既存の任意のネットワークでサーバを無変更で実行するための汎用な機構を検討する．具体的には NAT などによるプライベートネットワークの PC をサーバとして外部に公開するためのものである．プライベートネットワークと外部ネットワーク間での通信制約を UPnP[5] (Universal Plug and Play) などのネットワーク的な設定なしにソフトウェアだけで解決する．クライアント側はライブラリの connect() 部分を修正してプライベートネットワークのサーバに接続させるが，サーバ側に

† 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

関しては既存のものを使い，コードに変更は一切加えない．

以後，第 2 章で提案手法の設計を，3 章では実装について述べる．第 4 章では実験及び評価について述べる．関連研究を第 5 章で述べ，最後にまとめを述べる．

2. 提案手法の設計

プライベートネットワーク内のノードをサーバとして外部に公開するための提案手法の枠組みについて述べる．

2.1 基本指針

本機構を適用するアプリケーションはクライアントサーバ型であることが前提である．元のアプリケーションのサーバのコードを変更せずプライベートネットワークの中で（かつ外部からの要求を受け取れるように）使うために，今回作成した PSA（Private Server Agent）と呼ばれるデーモンプロセスを導入しサーバと連係動作させる．サーバとはあるネットワークアプリケーションにおいてサービスを提供するノードであるが，サービスの内容については限定しない．プライベートネットワークの内外で接続を確立するには内から接続を開始するしかないがサーバにこのような機能はない．本来サーバは接続を受ける役割を持つため，自分から接続を行うということはしないためである．そこでプライベートネットワークの中から外に向けて接続を開始する動作が必要であり，これを PSA に担当させる．PSA は，外部へ公開したいサーバが存在する各プライベートネットワーク内に 1 プロセス以上用意する．

PSA の接続先となるのは，PSA とクライアントの双方からの接続を受けれるような環境にあるプロキシサーバである．プロキシサーバはプライベートネットワークの内外の接続を仲介する役割を持つ本機構独自のサーバである．本稿では便宜上プロキシサーバはグローバル IP アドレスをもつものとして議論を進める．プロキシサーバは PSA との接続を常に維持しておく．クライアントはプロキシサーバと PSA を経由して目的のサーバと通信することができる．ただしクライアントは要求を出す前にまず最初にプロキシサーバに接続を行うよう動作に変更を加える必要がある．

プロキシサーバと PSA を追加したシステムの構成を図 1 に示す．本研究のターゲットは一般家庭の PC であるため，プライベートネットワークの構成にはブロードバンドルータを用いる．

以下，本機構で用いるアドレス空間と実アドレス空間でのアドレス変換，クライアントに加えるべき機能，そして対応しているプロトコルについて詳細を述べる．

2.2 アドレス変換

NAT 内のノードはグローバル IP アドレスを持たないため，外部ネットワークから識別できるような仮想的なアドレスを割り振る必要がある．現段階では暫定的にクラス E の IP アドレスを使用している．

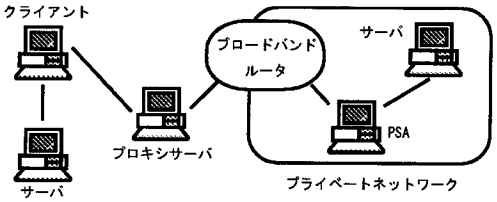

図 1 プロキシサーバと PSA の追加

このような仮想アドレスを割り振る前に，まず PSA は各プライベートネットワーク内のサーバのプライベート IP アドレスに対して ID をつける．ただしこの ID をそのまま外部へと公開する仮想アドレスとして使用することはしない．本来プライベート IP アドレスは各プライベートネットワーク内で閉じているものなので，この ID で外部へ公開すると他のプライベートネットワークのものと衝突する可能性がある．

本機構で使用する仮想アドレスはプロキシサーバに管理させる．プロキシサーバは PSA が各サーバに対して与えた ID と，PSA が存在しているプライベートネットワークを構成しているブロードバンドルータの WAN 側の IP アドレスの組に対して一意に定まるような仮想アドレスを決定する．ブロードバンドルータの WAN 側に設定されているのはグローバル IP アドレスであるためこの組み合わせであれば衝突の危険性はない．

2.3 クライアントへの機能追加

通常，クライアントはサーバに直接接続を行うがプライベートネットワーク内のサーバに対してはできない．そこで以下の処理を行う機能をクライアントに加える．

まずクライアントは目的のサーバの IP アドレスを取得すると，それがクラス E のものかそうでないかを調べる．クラス E 以外であれば直接接続が可能なアドレス空間であるため本機構を用いず通常の動作を行う．クラス E のものであれば本機構により割り振られた仮想アドレスを有するサーバとみなしてプロキシサーバへの接続を行い，この仮想アドレスを送信する．そしてプロキシサーバ及び PSA の応答を元にサーバまでの通信経路を確立する．

2.4 TCP と UDP

本機構を適応させるアプリケーションは，基本的に通信は TCP を用いて行っているということを前提としている．ただしアプリケーションによってはブロードキャストやマルチキャストを用いる必要があり，これには UDP の使用が推奨されている．[6] そこで UDP パケットに対してもリレーが可能であるようプロキシサーバと PSA を設計する．

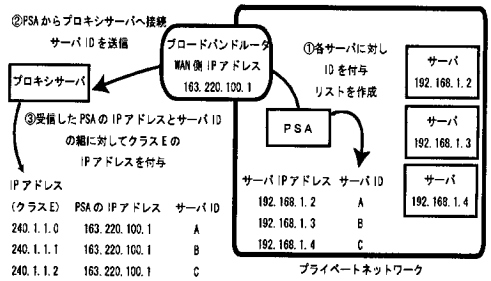

図 2 フェーズ 1 の様子

3. 実　　装

第 2 章での枠組みを元に，提案手法の実装に関する詳細を 2 つのフェーズに分けて述べる．フェーズ 1 はクライアントが要求を出す前に行う準備をするための段階である．フェーズ 2 ではクライアントがどのように要求を出し，サーバまでの通信経路を確立するかを述べる．

なお性能向上のためのチューニングなどは現段階では行っていない．

3.1 フェーズ 1：プロキシサーバと PSA の導入

PSA はあらかじめ同プライベートネットワーク内のサーバのプライベート IP アドレスを取得しておく．これらの IP アドレスに対してそれぞれ一意に ID を振り分け，対応表を作成する．そしてプロキシサーバに接続し，これらの ID を伝える．

プロキシサーバは PSA からサーバ ID を受け取ると，これを送ってきた PSA の IP アドレス（ブロードバンドルータの WAN 側に設定されている IP アドレス）と ID を組としてクラス E の IP アドレスを付与し対応表を作成する．ここまでの様子を図 2 に示す．この図では，プライベートネットワークの中にプライベート IP アドレス（192.168.1.2, ...）を持つサーバを 3 台，ブロードバンドルータのグローバル IP アドレスとして 163.220.100.1 が設定されている構成で説明している．

3.2 フェーズ 2：クライアントの要求処理

クライアントはサーバの IP アドレスが実 IP アドレスか仮想 IP アドレスであるかを判定する．仮想 IP アドレスであれば，プロキシサーバに接続しこれを送信する．プロキシサーバはその仮想 IP アドレスを自分のリストの中から検索し，もし発見できたなら次にサーバと同じプライベートネットワークにいる PSA にクライアントの IP アドレスとサーバの ID の二つを送信する．ここの通信にはフェーズ 1 で確立した接続を使用する．

そして PSA は受信したサーバの ID からサーバのプライベート IP アドレスを割り出しそれに接続を行う．PSA とサーバは同一ネットワーク内に存在するのでこの接続は問題なく行える．

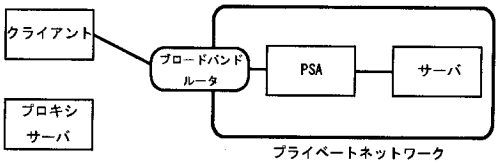

図 3 PSA からクライアントへ到達可能なときの通信経路

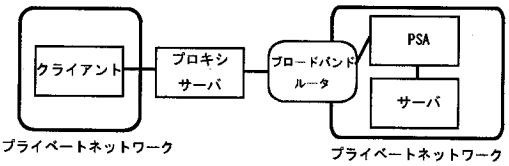

図 4 PSA からクライアントへ到達不可なときの通信経路

次に，プロキシサーバから受信したクライアントの IP アドレスに対して PSA は接続を試みる．PSA からクライアントへ到達可能であるかどうかで以降の通信過程が変わってくる．現実装では 2 秒以内に接続が確立されなかった場合に到達不可と判定している．到達不可であるとは，クライアントもまた別のプライベートネットワークの中に存在しているために PSA からの接続要求を受けれないといったことが原因として考えられる．

以下，到達可能であるか否かの 2 つのケースに場合分けをして説明していく．

3.2.1 PSA からクライアントへ到達可能

到達可能であればこの時点で PSA はクライアントとの接続が確立されているはずである．これ以降 PSA はクライアントとサーバの間でリレーサーバとして機能する．クライアントは以降プロキシサーバと通信を行わないためこの接続は切断する．

サーバまでの通信経路を図 3 に示す．クライアントは PSA を accept() してできた接続を利用して通信をする．

3.2.2 PSA からクライアントへ到達不可

到達不可の場合は PSA とプロキシサーバがリレーサーバとして機能する．PSA はプロキシサーバとサーバ間の通信をリレーし，プロキシサーバは PSA とクライアント間の通信をリレーする．

これにより，クライアントはプロキシサーバと PSA を経由してサーバと通信が行える．この場合の通信経路を図 4 に示す．使用する接続はクライアントが最初にプロキシサーバに connect() を行って生成したものである．

表 1　マシン構成

部品	部品名
CPU	Dual Opteron 2.6GHz x 2
メモリー	4GB
NIC	Gigabit Ethernet
OS	linux 2.6.18

表 2　ネットワーク機器

ネットワーク機種	機種名
スイッチ	PowerConnect2724 (1000Base-T)
ブロードバンドルータ	BEFSR41C-JP V3 (100Base-T)

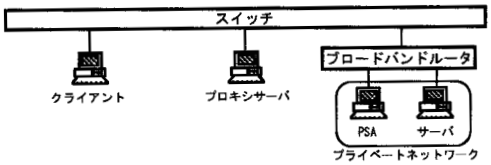

図 5　ネットワーク構成 1

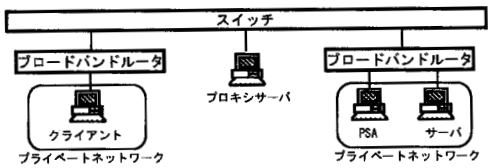

図 6　ネットワーク構成 2

4. 性能評価

4.1 評価環境

クライアント，プロキシサーバ，PSA として使用した各マシンの構成を表 1 に示す．使用したネットワーク機器を表 2 に示す．本研究の目的である一般家庭の PC の資源利用ということから，より環境を近づけるためにプライベートネットワークを構成するゲートウェイには一般家庭で使用されているようなブロードバンドルータ BEFSR41C-JP V3 を使用した．

PSA からクライアントへ到達可能な場合のネットワークの構成を図 5，到達不可の場合のネットワークの構成を図 6 に示す．

4.2 評価項目

4.2.1 スループット

スループットを測ることで本機構が元のアプリケーションの性能に与える影響について調べる．計測には Iperf[4] を用いた．Iperf はメモリ to メモリのデータ転送をネットワークを介して実行できるため，ハードディスクがボトルネックになることはない．よって Iperf のクライアント及びサーバが十分なスペックを持っていればネットワークそのものの性能を測ることができる．また Iperf もクライアントサーバ型のアプリケーションであるため，本機構を Iperf に実装し実行することで本機構のスループットを計測できる．実装に関しては Iperf クライアントのコードを修正しプロキシサーバと PSA の導入を行う．

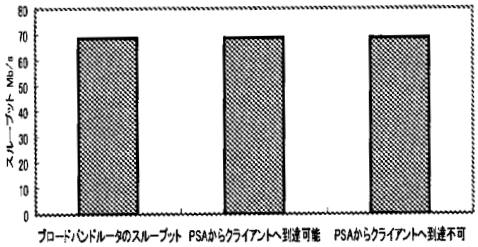

図 7　スループット

ブロードバンドルータの性能を測るため，ブロードバンドルータが構成するプライベートネットワークにクライアント，その外にサーバを設置しスループットを計測する．クライアントからサーバへ 300 秒間に送れるデータ量からスループットを計算すると 68.5Mb/s であった（10 回計測したものの平均値）．この値をこのブロードバンドルータのスループットとする．

到達可能である場合の構成でクライアントとサーバ間で同様の実験を行った結果，68.5Mb/s のスループットを得た．到達不可の場合でも 68.5Mb/s のスループットが得られているため，本機構の導入によるスループットの低下はこのブロードバンドルータを使っている環境ではほとんどないといえる．3 つの結果をまとめたものを図 7 に示す．

また，到達可能である場合プロキシサーバは通信経路に含まれないためその分の通信帯域とプロキシサーバの処理能力を節約することができる．

4.2.2 接続確立に要する時間

接続確立時間は通信開始における遅延時間の一部である．アプリケーションによっては通信遅延が大きいとシステム全体のパフォーマンスを低下させてしまう．遅延時間を測ることで適応すべきアプリケーションの対象が見えてくる．クライアントが接続確立にかかる時間は gettimeofday() 関数をコード内に記述し計測した．本機構にはクライアント，プロキシサーバ，PSA，サーバの 4 種類のノードが存在するためいずれかの持つクロックを基準にしなければ測定できない．今回はクライアントのクロックを基準に用いた．

PSA からクライアントへ到達可能の場合と到達不可の場合で接続を確立するまでにかかる時間をそれぞれ表 3，表 4 に示す．ただし第 3 章で示したフェーズ 1 の処理はすでに完了しているものとしている．(1) プロキシサーバへの接続と (2) 仮想アドレスの送信は両ケースにおいて同じ処理であり，経過時間に大きな違

表 3　クライアントが接続確立までに要する時間
（クライアントから PSA が到達可能である場合）

過程	時間 [s]
(1) クライアントからプロキシサーバへ接続	0.001084
(2) クライアントが仮想アドレスを送信	0.000042
(3) PSA からの接続受入まで	0.001693
(4) 合計	0.002819

表 4　クライアントが接続確立までに要する時間
（クライアントから PSA が到達不可である場合）

過程	時間 [s]
(1) クライアントからプロキシサーバへ接続	0.001279
(2) クライアントが仮想アドレスを送信	0.000031
(3) プロキシサーバからの応答まで	2.000391
(4) 合計	2.001701

いは見られない．(3)PSA 又はプロキシサーバからの応答までの処理は (4) 接続確立にかかる全ての時間から (1) と (2) の時間を引いたものである．(3) の中で行われていることとしては，プロキシサーバによる仮想アドレスの検索，それにマッチしているサーバ ID とクライアントの IP アドレスの PSA への送信，PSA からクライアントへの接続試行，PSA からサーバへの接続などにかかる時間が含まれる．PSA からクライアントへ到達可能である場合と不可である場合では，後者はタイムアウトで設定されている 2 秒間を全て待ってしまうので (3) において前者とは大きな差が出ている．ただしそのタイムアウト時間を除くと到達不可であるときのほうが短い時間で接続を確立している．これは，クライアントは PSA からの接続を受け付ける必要がなく，すでに確立しているプロキシサーバとの接続を使用すればよいからである．

PSA からクライアントへ到達可能であるときに要する接続時間において支配的となっているのは，ブロードバンドルータを越えて接続をするのにかかる時間である．BEFSR41C-JP V3 だとこれには約 1ms かかることが実験によりわかった．到達可能であるときはこの動作を 2 回行う，クライアントからプロキシサーバへ接続するときと PSA からクライアントへ接続するときである．到達不可の場合ではブロードバンドルータを越えての接続はクライアントとプロキシサーバ間の一回だけである．なお，同ネットワーク（BEFSR41C-JP V3 の構成するプライベートネットワーク）内での接続に要する時間は予備実験より約 0.1ms であった，これは PSA がサーバに接続するときにかかる時間である．これらのことよりブロードバンドルータの内外で接続をするのにネットワーク機器の処理による遅延時間が接続そのものにかかる時間と比較してとても大きいことがわかる．

PSA からクライアントへ到達不可であるときの接続確立時間はタイムアウトに設定した時間程度かかってしまうため，接続を頻繁に繰り返しかつ一回の接続において流すデータ量が小さいアプリケーションを本機構を用いて使用するとこの接続確立時間が大きなオーバヘッドとなる．到達可能であるか否かの判定をもっと短い時間で行えるような方法を検討したい．しかし大きなデータを扱うようなアプリケーションであれば接続確立は最初の一回だけであるので通信性能に大きな影響は与えないと考えられる．到達可能であればクライアントの接続に要する時間は数 ms である，この値を大きいと見るか小さいと見るかは到達不可であるときと同様，適用するアプリケーションの通信性質に依存する．

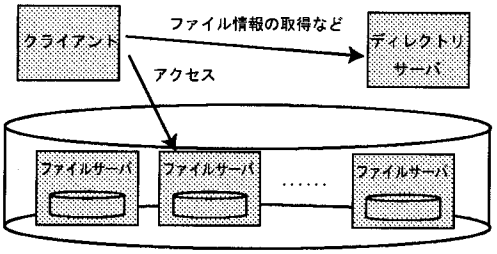

図 8　Gfarm ファイルシステム

4.2.3　Gfarm への適応

提案手法が汎用的に利用できることを示すために Gfarm ファイルシステム[1] にも適応した．

Gfarm はグリッド環境向けに設計開発された広域分散ファイルシステムである．Gfarm の全体像を図 8 に示す．Gfarm にはクライアント，ファイルサーバ，ディレクトリサーバの 3 種類のノードが存在する．主な機能として，ネットワークで接続されている複数のファイルサーバの持つローカルストレージを統合し，仮想的な巨大なストレージを構築することができる．ファイルサーバは自分のストレージをこの仮想ストレージに対して提供するノードである．クライアントはこの仮想ストレージに対して読み書きを行えるが，実際にどのファイルサーバのストレージに対してアクセスするかは自分自身では決定しない．クライアントはこの統合された仮想ストレージにアクセスする際，まずディレクトリサーバに問い合わせを行う．ディレクトリサーバはどのファイルがどのファイルサーバに保存されているかといった情報を持っているため，クライアントはディレクトリサーバに指示されたファイルサーバに対してアクセスすればよい．

ところで，Gfarm はグリッド環境のように全てのノードがグローバル IP アドレスを持つような環境を想定して設計されている．つまりプライベートネットワーク内のノードをファイルサーバとして利用することはできない．そこで本機構を適用することを試みた．

Gfarm の持つコマンドを用いてクライアントからファイルサーバへ 512MB のデータ転送を行い，その

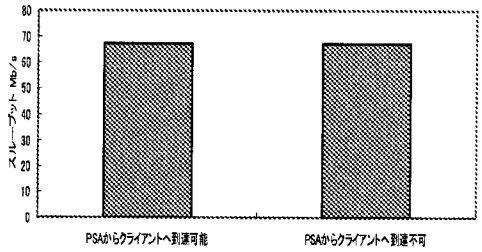

図 9 Gfarm システム内でのデータ転送時のスループット

スループットを測定した．その結果を図 9 に示す．

PSA からクライアントへ到達可能と到達不可のいずれの場合においてもスループットは 67.9Mb/s で，Iperf での測定結果とほぼ同じとなっている．正しくデータの転送が行えたことからも本機構がプライベートネットワーク内のサーバに対して有効であることが示せた．

5. 関連研究

NAT 越えの技術として STUN[7] がある．これはクライアントもサーバも別々のプライベートネットワーク内であるときに通信をするための有効な方法だが UDP のみの通信しか行うことができず TCP を使うことができないという制限がある．Skype[8] などで使用されている．

6. おわりに

本稿ではプライベートネットワークの中の一般家庭の PC に焦点を当て，これをサーバとして外部に公開するための汎用的な機構を提案した．サーバ側のコードは既存のまま，変更を一切必要とせずに使用できる．これを実現するためにプロキシサーバと PSA を導入する．また，クライアントのネットワークの状態から二通りの通信方式を提案した．この機構を用いて Gfarm のプライベートネットワーク内のファイルサーバを使用できたことを確認した．

また，本機構を Iperf に実装しスループットを測定した．その結果は本機構を用いずに Iperf で測定した値とほぼ同じ値であったため，本評価環境においては本機構を用いることによるスループットの低下はないといえる．またプライベートネットワーク内のサーバとの接続を確立するために必要な時間を測定した．この時間にはクライアントのネットワークの状態を調べるという過程が含まれる．これには PSA からの接続要求をクライアントへ出し一定時間内に成功するかをみる．接続を試みる時間は設定可能であるが，正確な結果を得るためには数秒程度必要である．接続可能であれば大きな遅延とはならないが，接続不可である場合はタイムアウト時間とほぼ同じ時間がかかってしまう．これはネットワーク機器や接続処理そのものの遅延時間のオーダを大きく上回っている．よってクライアントの状態を調べるためのより高速な方法を検討したい．

今回は TCP と UDP のリレーまでを実装したが，ブロードキャストとマルチキャストに対する実装はまだできていないためこれを今後の課題としたい．また，プロキシサーバ及び PSA を 1 台ずつの実装をしたが，そのいずれかに障害が発生した場合プライベートネットワーク内のサーバ全てに対してアクセスができなくなってしまう．家庭の PC が常にオンライン状態であることは少ないため，より実用的なものとしていくために冗長化を行うことでネットワーク障害や PC の不測のネットワーク脱退による被害を抑えられるようにしていきたい．

謝辞 本研究の一部は，文部科学省科学研究費補助金特定領域研究課題番号 19024009 および基盤研究 (A) 課題番号 17200002 による．

参考文献

1) 建部 修見，森田 洋平，松岡 聡，関口 智嗣，曽田 哲之，「ペタバイトスケールデータインテンシブコンピューティングのための Grid Datafarm アーキテクチャ」，情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム，情報処理学会，Vol.43，No.SIG 6 (HPS 5)，pp.184-195，(2002).
2) D. Anderson: BOINC: A System for Public-Resourcing Computing and Storage, Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on, pp. 4-10, (2004).
3) Gnutella. http://gnutella.com.
4) Iperf. http://dast.nlanr.net/Projects/Iperf.
5) UPnP Forum: UPnP. http://www.upnp.org.
6) W. Richard Stevens. UNIX NETWORK PROGRAMMING Networking APIs : Sockets and XTI Volume1, Second Edition. Pearson Education Japan, (1999).
7) J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). RFC3489. (2003).
8) Skype Technologies SA. http://www.skype.com