修士論文　　2002 年度 (平成 14 年度)

P2P モデルを用いた
分散トラフィック測定機構に関する研究

慶應義塾大学 政策・メディア研究科
氏名： 豊野 剛
jr@sfc.keio.ac.jp

修士論文要旨 2002 年度 (平成 14 年度)

P2P モデルを用いた分散トラフィック測定機構に関する研究

論文要旨

本研究では P2P モデルを用いることで，大規模ネットワーク上でエンドノードのユーザによる分散したトラフィック測定機構を構築した．これにより，エンドノードのユーザ及びアプリケーションに必要なトラフィック測定情報を提供する環境が実現した．

近年のインターネットのバックボーンインフラストラクチャ，Last 1 Hop の広帯域化により，エンドノードのユーザのインターネット利用形態に変化が生じている．特にストレージ共有ツール，ネットワーク対戦ゲームなどの P2P アプリケーションの増加に伴い，大容量データが End-to-End で双方向に通信されるようになった．その反面，これらのアプリケーションがネットワークを効率的に利用するための，エンドノードのためのネットワーク情報の測定手法の整備が遅れている．

関連研究を検討することで，エンドノードによるネットワーク測定情報の利用における既存のトラフィック測定機構の不十分な点を明らかにした．既存の測定機構では，エンドノードからは主に片方向通信のトラフィック測定しか行うことが出来ない．また，測定したデータをユーザ間で相互に利用する機構や過去の測定データの参照ができない．

本研究では，P2P モデルを用いたエンドノード間の測定機構 Distributed mutual measurement mechanism(以下 Dm3) を実装し，評価を行った．本機構は P2P モデルでエンドノード間の双方向の測定環境を実現する．また，データを一元的に管理するデータベースサーバを構築することで過去の測定データの利用を可能にする．これにより，エンドノードのユーザおよびアプリケーションに End-to-End の通信に必要なトラフィック情報を提供できた．また，広域ネットワーク上で，測定情報をユーザ同士が相互に利用するための環境も提供した．

実際に本機構を運用して得られた測定データを検証し，エンドノードのユーザの利用モデルを満たす必要十分な測定データを得ていることを実証した．また，問題点と要件定義の面から本機構と既存の測定機構と比較し，定性的な評価も行った．その結果，本機構はエンドノードのユーザおよびアプリケーションに必要な測定を行える環境を提供したことを示した．

キーワード: 1: P2P モデル　2: トラフィック測定　3: エンドユーザ　4: 分散システム

慶應義塾大学 政策・メディア研究科
豊野 剛

Abstract of Graduation Thesis Academic Year 2002

Distributed Network Traffic Measurement with P2P model.

Abstract

This research implements a mechanism to measure traffic information within a massive network from the end user point of view applying the Peer-To-Peer (P2P) model. This mechanism creates an environment for the end node users and their applications to notice and utilize the traffic information needed.

Recent growth of the Internet backbone and widening of the Last 1 Hop connection results in the change of end user internet usage. The change mainly comes from the rise of P2P application, such as network storage, file sharing tool, and network gaming software. Users are now communicating end-to-end, exchanging massive amounts of information bi-directionally. On the other hand, an environment that is able to measure network information needed for these applications listed above is currently not available yet.

By studying the past related researches, past network measurement software is inadequate for measuring the current network from the end-node point of view. Past measuring mechanisms are mainly focused on measuring traffic uni-directionally. Also, measurement data of the past are not able to be referred.

This research proposes, implements and evaluates a Distributed Mutual Measurement Mechanism (Dm3) to measure network information between end nodes using the P2P model. This mechanism accomplishes a bi-directional traffic measurement between end-nodes and also obtains past information by referring the database server. As a result, an environment to provide End-to-End traffic information is constructed to be used by widely spread end-node users and their application.

By conducting group measurement of multiple nodes using this mechanism within actual internet traffic, measurement of bi-directional channel is possible which could not have been found by past measurement mechanisms. Finally, the problems of past measurement mechanisms and the "Dm3" mechanism are compared and evaluated qualitatively to conclude that "Dm3" meets the requirement discussed and proved that end-node user and their application is able to retain the information needed for End-to-End network.

Keywords: 1: P2P model 2: Traffic measurement 3: End user 4: Distributed system

Tsuyoshi Toyono
Graduate School of Media and Governance
Keio University

目 次

図 目 次

表 目 次

第1章　序論

本章では，エンドノード上で動作するアプリケーションによるインターネットの新たな利用形態について分析し，エンドノード間測定の必要性と本研究の目的を述べる．

1.1　背景

本研究ではインターネット (The Internet) 上において，末端に位置するエンドノード同士が協調し，ネットワークの状態やトラフィック測定の情報を共有し，利用できる環境の構築を目指す．

ネットワークの状態や動向を把握するために，さまざまなトラフィック測定 (traffic measurement) 手法が開発されてきた．特にネットワーク管理の観点でトラフィック測定は重要であり，現存するネットワークの測定手法の多くは，管理者が自ドメイン内の障害検知に用いるものや中継ノードのトラフィックモニタリングを行う目的で設計されている．

これに加え，近年ではエンドノードに接続するユーザに対してもネットワーク状態やトラフィック測定の情報を提供する必要性が高まってきている．その理由として，インターネットインフラストラクチャの整備により接続の形態が変化してきていること，広帯域，常時接続のユーザが増加し，エンドノードから直接大容量の情報を送信したり受信したりする新しい利用形態が増加してきたことが挙げられる．

1.1.1　ユーザのインターネット接続形態の変化

国内でも急速にインフラストラクチャ，接続サービスの多様化が進んでいる．しかし，インターネットは IP によるパケット交換網であり，その特性上，均一なサービスが保証されていない．そのため接続するポイントによって性能は変化し，同じポイントで接続しても常に同一の通信性能が得られるとは限らない．

例えば，近年普及した ADSL(非対称デジタル加入者線 Asymmetric Digital Subscriber Line)

の仕組みでは，実効帯域は収容する基幹局から接続ポイントまでの距離に影響され，同じサービスでありながら帯域幅が異なり，上りの帯域幅と下りの帯域幅も異なる．また，インターネットでは多くの場合上りの経路と下りの経路は異なり，時間によってもトラフィック流量が異なる (上りと下りのトラフィックについては 1.5 節にて論じる)．今後，Cable TV 各社，電力会社などの異種業者の参入，光ファイバによる専用接続，移動体接続サービス，ホットスポットの整備，ISP のマルチホーム化など，接続インフラストラクチャ及び接続サービスの多様化はさらに顕著になっていくと予測できる．

既存のトラフィック測定手法は，多くが管理者が自ネットワークを監視するためや研究者がネットワーク状況を測定するためのものがほとんどであった．また，エンドノードからのトラフィック測定では，主に上りの回線の通信性能しか把握することができず，下りの回線の通信性能，時間の推移によるトラフィックの変化などの情報を得る手段が少ない．

1.1.2 ユーザのサービス利用形態の変化

1.1.1 節で述べたような，広帯域，常時接続などの接続形態の変化に伴い，ユーザの利用形態にも変化が生じている．今までは WWW，FTP など主にネットワークのバックボーンの近辺に設置されたサーバから，ネットワークの末端に位置するエンドノードのユーザがクライアントとしてデータをダウンロードするようなクライアント・サーバモデルで接続されるサービスを享受することが一般的だった (図 1.1)．この利用形態では，細い上り方向のトラフィック (主にリクエストパケット)，太い下り方向のトラフィック (主にダウンロードデータ) が生じる．

しかし，最近ではエンドノード同士がストレージを共有したり，エンドノードから多数のユーザに向けてストリーミングを発信したりといった，いわゆる Peer-to-Peer(以降 P2P) モデルと呼ばれる形態での利用も増加してきている．

これらの利用ではユーザは End-to-End で直接双方向・大容量のデータを送受信するようになっている (図 1.2)．このような利用形態においては，ネットワークバックボーンと自ノードの間のトラフィック情報だけでなく，エンドノード間のネットワーク情報がアプリケーションのパフォーマンスに影響を与えるようになっている．既存のネットワーク測定手法では，ドメイン内部に限定された測定であったり片方向通信路のみの測定であったりするため，End-to-End の大容量・双方向の通信のトラフィック測定情報をユーザに提供することは難しい．

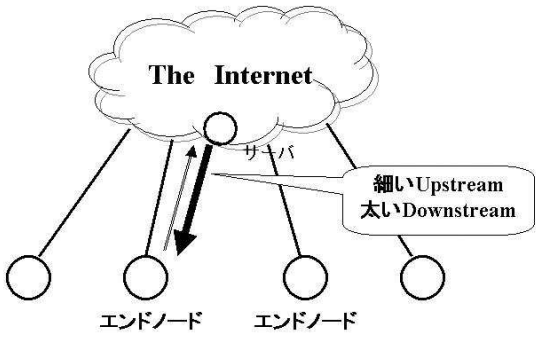


図 1.1: 今までの一般的なクライアント・サーバ型通信

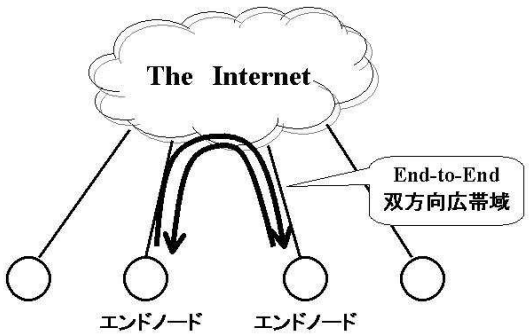


図 1.2: End-to-End の大容量データ通信

1.2 エンドノード測定の重要性

今後，1.1 節で述べたように一層ユーザのインターネット接続形態の多様化，サービス利用形態の変化が進むと考えられる．これらの変化により，ネットワークエンドノードに位置するユーザやユーザが利用するアプリケーションにも，自らの接続しているネットワークの状態や通信相手のエンドノードとの間のトラフィック情報を提供するような，ユーザ間の測定環境が重要となる．

1.3 エンドノードでの測定利用環境

以下に，本研究で想定するトラフィック測定の新しいニーズとして，エンドノードによる測定データの具体的な利用ケースを挙げる．

1.3.1 ユーザによる ISP 接続サービスの選択

光ファイバ，ADSL，無線，公衆電話回線，携帯電話に代表されるさまざまなインフラストラクチャに加え，常時接続，移動体接続などその上で多様な接続サービスが展開してきたことで，ユーザも自分のニーズに合わせたサービスを選ぶことができるようになった．

しかし，ISP などの接続サービスを選択する指標は価格，バックボーン帯域，ユーザサポート体制などの静的な情報及び掲示板や他のユーザの評判といったあいまいな情報が元となってしまっている．

エンドノード上でサーバを構築することで上りの通信量が増加したり，ストリーミングなどでダウンロードデータが多くなるなど，これからはエンドノードにおいても多様な利用形態が想定される．各地点に分散したエンドノードからの統一されたトラフィック測定のデータを実際に比較することで，インフラストラクチャ，接続サービス及び接続ポイントなど，具体的な指標を立てられる．

1.3.2 効率的なアプリケーションネットワーク形成

エンドノードからも，正しくトラフィック情報，トポロジ情報などが得られれば，ネットワーク上に余剰なトラフィックを流さずに IP ネットワークを考慮に入れたより効率的な P2P モデルのアプリケーションネットワークが構築できる．

P2P モデルでは，エンドノード上のアプリケーションが自律分散的に協調し合い，直接的に資源の交換，共有，制御などを行う．これらのサービスでは，アプリケーションがエンドノード (ユーザ) をグループ化し，アプリケーションレベルでのネットワークを形成し通信するケースが多い．これらの，アプリケーションが TCP/IP 上に覆い被さるように新しいトポロジを形成することを，本論文では「アプリケーションネットワーク Apllication Network」と呼ぶ．(アプリケーションネットワークは「オーバーレイネットワーク Overlay Network」とも呼ばれる．)

これらが用いるアプリケーションネットワークの特徴として，以下のようなものが挙げられる．

- アプリケーションがユーザをグループ化する，不均質な自律分散システムである
- ノードが任意に追加，削除，交換するオープンネットワークである
- ネットワークトポロジの変更が頻繁に起こる
- 非同期性を備えたものも多い

以下にアプリケーションネットワークを形成する P2P モデルのサービスの例とその特色を挙げる．

1. メッセンジャーツール
   指定したユーザリストに登録されているユーザとチャット，ファイルの交換などを行うシステム．Hybrid P2P モデルと呼ばれる，仲介サーバを用いていることが多い．サーバはノードリストとユーザステータス，IP アドレスリストなどだけを保持し，ユーザ同士のチャット，ファイル交換などは直接通信される．
   例) ICQ，MSN メッセンジャー，Yahoo メッセンジャー，AoL メッセンジャーなど．
2. P2P 型のストレージ共有システム
   アプリケーションがネットワークを構成し，ユーザ同士が直接ファイルを交換しあうシステム．ファイルの送受信はユーザ同士 1 対 1 だが，ファイルの検索クエリなどはアプリケーションネットワーク上を流れる．
   例) Napster，Gnutella，WinMX など．
3. ネットワーク対戦ゲーム
   2 人〜大人数でネットワークを形成し，インターネット上で対戦ゲームを行うもの．複数ノードをグループ化し，アプリケーションネットワーク上のうち 1 ノードがサーバとなってデータを送受信する．
   例) Microsoft Age of Empires，Blizzard Entertainment Diablo2 など．

同じ P2P モデルを用いたもので，ストリーミングアプリケーションの ShareCast，PeerCast，グループウェアの Groove，その他にも分散キャッシングシステムなどアプリケーションネットワークを形成するものは多い．

これらはアプリケーションネットワーク上のノードを仮想的にサーバに見立て，階層的な通信を行っている．しかし，アプリケーションネットワーク上で構築された仮想リンクが，IP ネットワークを必ずしも反映しているとは限らない．これらの仮想サーバノードが，IP ネットワー

クのトポロジ，経路制御の観点から見て非効率的なポイントに設置されると，ネットワーク上に冗長なトラフィックが流れることになってしまう (図 1.3)．アプリケーションネットワークと IP ネットワークのトポロジ，リンクの状況の反映の度合いを，本論文中では「ネットワークの親和性」と呼ぶ.

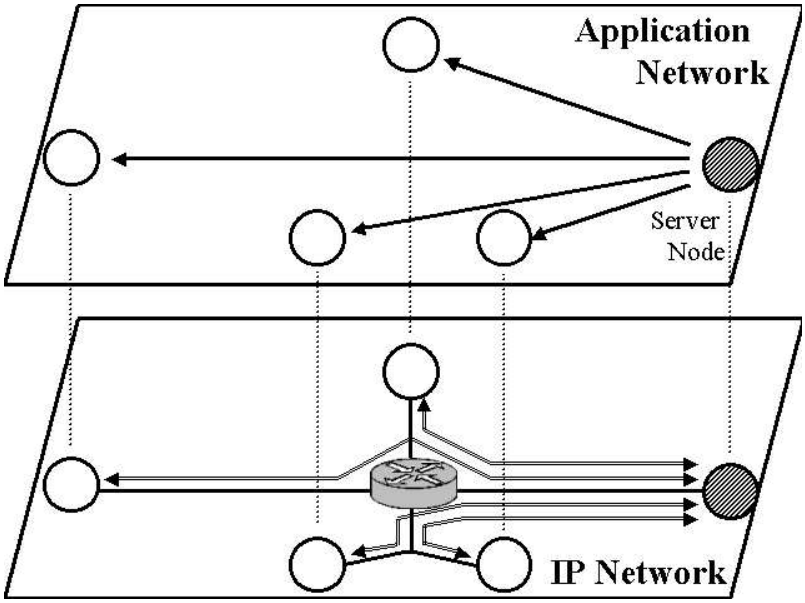


図 1.3: 冗長な構成のアプリケーションネットワークの例

図 1.4 に示すように，IP ネットワークの情報がエンドノードの P2P アプリケーションからも得られれば，IP ネットワークと親和性のある，より効率的な P2P モデルのアプリケーションネットワークが構築できる．これを実現するためには，エンドノードからも的確なトラフィック測定の情報，トポロジの状態などが得られる必要がある.

1.3.3 選択可能な End-to-End 通信の実現

エンドノードのユーザ間で大容量のデータが直接通信されるようになったが，現在ではユーザにトラフィック測定の情報をフィードバックする手段が乏しく，これらの通信が非効率的になされている例も多い.

図 1.5 は非対称なネットワークにおけるユーザの利用モデルをを示している．この図上で，ユーザ A がデータ X を取得したい場合を想定する．まず，ネットワーク上に同一のデータ X

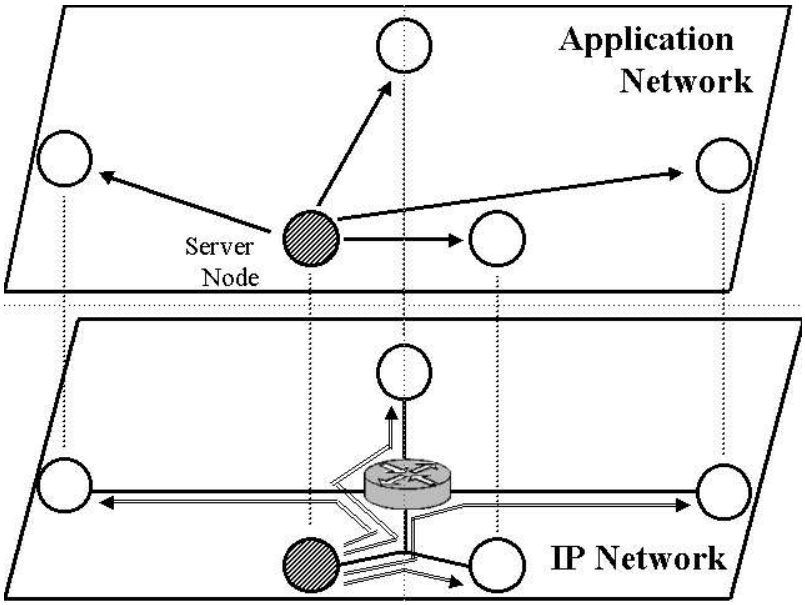


図 1.4: IP レイヤと親和性の高いアプリケーションネットワークの例

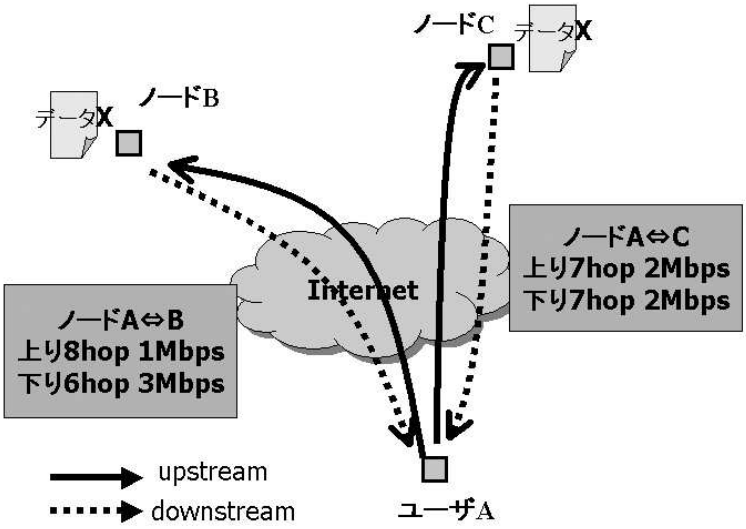


図 1.5: 非対称ネットワークのトラフィック測定のモデル

を持つ 2 つのノード B と C が検出されたとする．データのダウンロードを行う際，どちらか通信性能が良い方から取得したいので，ユーザ A からトラフィック測定を行うと，図のようにノード B まで 1Mbps，ノード C まで 2Mbps だった．この場合，ユーザ A からの測定だけではノード C の方が通信性能が良いと判断できる．

この図の例では，実際にはデータをダウンロードする下りの通信性能はノード B の方が上回っている．しかし，現状ではこのようなネットワークであっても，ユーザ A からの測定のみではこの下りのトラフィック情報を得ることができない．エンドノードのユーザに正しくトラフィック情報を提供することで，よりアプリケーションのニーズに即したサービスの利用が実現される．

1.4 目的

本研究では，エンドノードのユーザに，必要なトラフィック測定情報を提供するための基盤の構築と，そのトラフィック測定情報をユーザ間で互いに利用するための環境の実現を目的とする．

既存のネットワーク測定手法だけでは，得られるデータはエンドノードのユーザが欲している情報とは必ずしも合致しない．本研究ではエンドノード同士を協調させ，測定した情報を共有することにより，ユーザが利用したい情報を自由に得られるシステム Dm3(Distributed mutual measurement mechanism) を提案し，実装及び評価を行う．

1.5 用語の定義

本論文中での用語の定義を行う．

ノード ネットワークに接続されている機材を「ノード」と呼ぶ．一般的に，ノードとはインターネット上での TCP/IP のプロトコルを処理する能力を持ったネットワークの機器を指すが，本論文中ではこの他にアプリケーションネットワークを形成するための構成要素もノードと称する．

エンドノード 本論文中でエンドノードとは，インターネット接続サービスを用いてネットワーク上の各種サービスを利用している末端ノードと定義する．インターネット接続サービスとは，ISP からの PPP 接続，Cable TV 各社からの常時接続，各種移動体接続などの接続サービス全てを指す．サービスに用いられる各種無線，電話回線，ATM，xDSL，光ファイバなどの物理的な接続方式は問わない．本論文中で End-to-End の通信とは，このエンドノード同士の通信を指す．

トラフィック トラフィックとはネットワーク内を流れている情報そのもの，もしくはその情報の流通量を指す．ユーザがネットワークを利用する際には，2 つのデータのフローが発生する．一つは自ノードから相手ノードまでのフロー (行きのデータ Outgoing)，もう一つは相手ノードから自ノードまでのフロー (帰りのデータ Incoming) である．本論文中では，ノードの位置に関わらず自ノードを起点とし，統一して前者を上りのトラフィック (アップストリーム Upstream)，後者を下りのトラフィック (ダウンストリーム Downstream) と呼ぶ．

トポロジ ノード同士は，有線ないし無線の回線で接続されネットワークを構成する．このネットワークの全体構成，機器間の接続状況を一般的にネットワークの「トポロジ」と呼ぶ．本論文中では，この他にアプリケーションネットワークの形成ノード同士の接続状況もトポロジと称する．

1.6 本論文の構成

以下に本論文の構成を述べる．次の 2 章で，本論文における問題意識のまとめを行い，関連研究としてトラフィック測定手法，アプリケーションネットワークの効率化に関する研究を分析，考察する．3 章では，関連研究の問題点を踏まえ，本論文の目的を実現するための要件定義を行う．また，4 章で要件を元に新しいエンドノードユーザ間の測定，測定データ共有環境である Dm3 システムの提案及び設計を述べる．そして，5 章で実際に Dm3 システムの実装を行い，6 章においてシステムの運用および評価を行う．最後に，7 章で本論文の結論および今後の課題を述べる．

第2章　エンドノード測定における問題点と関連研究

本章では，まず本研究における既存の測定手法に関しての問題点を述べる．次に関連研究としてエンドノードが行うことの出来るトラフィック測定手法についてのまとめと，既存の広域ネットワーク測定プロジェクトの調査を行う．また，アプリケーションネットワークにおいてIP ネットワークと親和性を高める先行研究についても考察する．最後に問題点および関連研究の考察を踏まえ，本研究で設計する機構に必要な要件をまとめる．

2.1　エンドノード測定に関する問題点

既存のトラフィックの測定手法にはさまざまなものがあるが，1 章で述べたように，エンドノードのユーザ，及びエンドノードのアプリケーションにデータを提供できる枠組みが十分ではない．以下に既存の測定手法の，エンドノード測定に関する問題点を挙げる．

1. 片方向測定であること．
   エンドノードのユーザが行えるトラフィック測定の大部分は，自ノードを起点とした上りの片方向通信路の測定となっている．ユーザの利用形態を考えると，上りの通信のみならず，下りの通信のトラフィック測定も重要となってくるが，既存の測定手法ではこれは実現されていない．

2. 測定履歴を参照できないこと．
   ping や traceroute といったユーザが行うトラフィック測定では，過去の履歴を利用することができない．ネットワークを流れるトラフィックは時間によって変移するが，ユーザ測定情報だけではこれらのトラフィック推移を把握することは出来ない．

3. ユーザに測定データが提供されていないこと．
   広域なトラフィック測定プロジェクトのデータはユーザにとっても有用であるが，これら

の測定結果はわかりやすい形でユーザに提供されてはいない．また，データ自体非公開のものもある．

4. 測定コストが高いこと．
   ある特定のトラフィック測定のためにシステム (OS：Operating System) 自体に改変を加えたり，恒常的にサーバを運用したり，また中間ノード (ルータ) の運用権限やシステム管理権限が必要だったりすることは，その処理を行うシステムのオーバヘッド (Overhead) が大きいといえる．本論文では，これを測定を行うための「コストが高い」と定義する．既存の広域ネットワーク測定手法で，測定コストが高いことは，エンドノードのユーザが容易にトラフィック測定情報を得るための障害となる．

5. 汎用性に欠けていること．
   P2P モデルを用いた新しい End-to-End アプリケーションを設計する際に，そのネットワーク効率のチューニングのために独自にトラフィックの測定アルゴリズムを設計している例が多い．より汎用性のあるプロトコル設計を行うためには，エンドノードからの測定手法とそのデータの利用をアプリケーションの機構から切り分ける必要がある．また，P2P アプリケーションのネットワークは常にノードが入れ替わり，ネットワーク構成が変化するオープンネットワークであるが，このようなネットワーク構成にも対応できるような実時間性を保った測定が必要となる．

2.2 関連研究

以上のような問題意識を踏まえ，大きく分けて 3 つの分野の関連研究の考察を行う．まず一つ目に，エンドノードが行うことのできる測定手法について考察する．二つ目には，既に行われている複数のノードを用いた広域なトラフィック測定プロジェクトの取り組みについて考察する．三つ目に，既存のアプリケーションネットワーク上で行われているネットワーク形成の効率化に関する研究について考察する．

2.2.1 エンドノードからの測定手法

現在，エンドノード自身が行える測定手法の代表的なものは，次のようなものがある (表 2.1).

表 2.1: エンドノードが行える測定手法

名称	入力	測定項目	備考
ping	ICMP	rtt(round trip time), TTL(Time To Live), パケット損失率	
traceroute	ICMP, UDP	途中経路, rtt, パケット損失率	
bing	ICMP	RAW-IP 帯域幅 (bps)	2 点間測定
hping	ICMP, TCP, UDP, ICMP, RAW-IP プロトコル	ポートスキャン, Path MTU discovery, Remote OS フィンガープリントなど	RAW-IP パケット生成
pathchar, pchar	UDP, ICMP	帯域幅, スループット, パケット損失率, キュー長など	測定に時間がかかる
bprobe, cprobe	ICMP	帯域幅, congestion	現在は SGI IRIX に依存
ttcp, nttcp	load generator	スループット	
iperf	TCP, UDP	bandwidth, ディレイジッタ, パケット損失率など	対向でサーバの必要あり

ping は ICMP Echo リクエストメッセージを送信し，対象ノードからの ICMP Echo リプライメッセージを受信することによって測定を行う．ping によって rtt と パケット損失率を測定できる．ping の問題点は，得られる情報が対象ノードの情報のみであるという点で，途中継路上にあるノードの測定は行うことができない．

traceroute を用いることによって，自ノードから対象ノードまで，パケットの流れる経路を確認できる．traceroute は TTL(Time to Live) 値を 1 から順にひとつずつ伸ばしながら UDP パケットを送信し，途中経路にあるノードが返す ICMP 到達不可 (TTL 時間超過) メッセージを拾うことで経路とトラフィックの測定を行う．

ping，traceroute などは代表的なネットワーク測定手法であるが，片方向通信路 (上りの通信) のトラフィックデータしか得ることができない． また，これらの手法は主に障害検知や到達性の確認といった用途に用いられ，時間によるトラフィックの推移などを測定するためには MRTG (22) など，別の手法と組み合わせたりしなければならない．

bing は Bandwidth ping の略で，異なる 2 ノード間の帯域幅を推測するツールである．bing ではユーザの位置するノード A 上から，ノード N1 とノード N2 の帯域幅を推測することができる．図 2.1 のようなトポロジの場合，ノード N1 とノード N2 をペアで測定することにより，rtt 値からより正確な帯域幅を予測できる．bing では異なるパケットサイズの ICMP Echo リクエストメッセージを複数回観測し，それらの rtt 値から 2 ノード間の帯域幅を推測する．

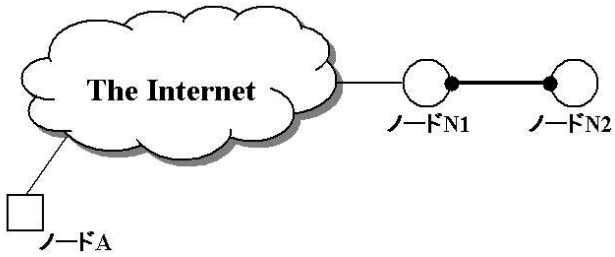


図 2.1: bing の測定トポロジ

hping は RAW-IP パケット生成ツールであり，さまざまな TCP，UDP 及び ICMP パケットを生成することが出来る．トラフィック測定ツールではないが，パケット損失率，帯域幅などを測定することができる．

pathchar (21) および pchar は，traceroute の機能と共にパケットペアスキームを用いて帯域計測が行えるツールである． pathchar はプローブ (probe) と呼ばれる一連の UDP パケットを作り，その IP ヘッダの TTL を 1 から順に増加させながら，ターゲットのノードまでの経路の rtt を計測する．traceroute とは異なり，以下の方法でより精密に rtt を計測する．

1 つのプローブは，複数の UDP パケットから構成される． デフォルトでは，IP データグラムのサイズにして 64 バイトから 1,500 バイトまで，44 バイトずつ変化させて複数のパケットを作りだす．各プローブで，どのサイズのパケットを送信するかはランダムに決められる．また，各プローブでは rtt の測定を行う．パケットサイズをランダムに変化させながら経路中のリンクを調べ，ゲートウェイ間の実効帯域，遅延 (us 精度)，及びパケットロス・レートを推定する．最終的には，目的のホストまでのホップ数，rtt，ボトルネックになっているリンクのバンド幅などがホップ毎に表示される (図 2.2).

しかし，pathchar の測定は大変時間がかかり，ネットワークの現在の状況を的確に捉えることは難しい．またプローブのパケットサイズが大きく，ネットワークに負担をかけてしまう．

bprobe 及び cprobe は SGI ハードウェア上の IRIX OS アーキテクチャに依存している．どちらのツールも ICMP Echo リクエストメッセージとパケットペアスキームを利用し，bprobe は与えられたパスに沿った最大帯域幅，cprobe は同じく現在のコンゼッション Congestion を測定できる．

```
riesling ~ 79% 14:24: pathchar ka9q.ampr.org
pathchar to ka9q.ampr.org (129.46.90.35)
 mtu limitted to 1500 bytes at local host
 doing 32 probes at each of 64 to 1500 by 44
 0 192.172.226.24 (192.172.226.24)
 |   9.3 Mb/s,   269 us (1.83 ms)
 1 pinot (192.172.226.1)
 |    85 Mb/s,   245 us (2.46 ms),  1% dropped
 2 sdscdmz-fddi.cerf.net (198.17.46.153)
 |    45 Mb/s,   -13 us (2.70 ms)
 3 qualcomm-sdsc-ds3.cerf.net (134.24.47.200)
 |   8.8 Mb/s,   1 us (4.07 ms)
 4 krypton-e2.qualcomm.com (192.35.156.2)
 |   5.2 Mb/s,   1.02 ms (8.42 ms)
 5 ascend-max.qualcomm.com (129.46.54.31)
 |   53.2 Kb/s,   4.20 ms (243 ms)
 6 karnp50.qualcomm.com (129.46.90.33)
 |    12 Mb/s,   -172 us (243 ms),  +q 8.96 ms (13.0 KB) *3,  6% dropped
 7 unix.ka9q.ampr.org (129.46.90.35)
7 hops, rtt 11.1 ms (243 ms), bottleneck 53.2 Kb/s, pipe 4627 bytes
riesling ~ 80% 15:30:
```

図 2.2: pathchar の実行例

ttcp や nttcp は，古くからあるロード・ジェネレータ，処理能力のベンチマークツールである．トラフィック測定ツールとして用いることにより，パケット損失率やスループットなどを得ることができる．ttcp や nttcp を元に sting，nettest，SProbe など大変多くの測定ツールが開発されている．最新のものは TCP UDP の各種パラメータの他にも，データ・パターン生成，オフセット・コントロール，マルチキャストなどをサポートしている．

iperf (24) は NLANR によって開発された ttcp (23) に類似する帯域幅測定ツールである．POSIX 互換プラットフォーム上で動作し，TCP の MSS/MTU，パケット損失率，ジッタ，TCP および UDP の帯域幅，スループットを測定できる (図 2.3)．しかし，iperf を用いるためには対向ノード側で 専用のサーバを上げる必要がある．

```
18:51[~]% iperf -c mistral.sfc.keio.ac.jp
------------------------------------------------------------
Client connecting to mistral.sfc.keio.ac.jp, TCP port 5001
TCP window size: 32.5 KByte (default)
------------------------------------------------------------
[  5] local 133.27.36.174 port 1665 connected with 202.249.25.24 port 5001
[ ID] Interval       Transfer     Bandwidth
[  5]  0.0-10.0 sec   8.6 MBytes   7.2 Mbits/sec
18:51[~]%
```

図 2.3: iperf の実行例 (TCP スループット)

これらのエンドノードが行うことのできる測定手法には，大きく分けて 2 つの問題点が挙げられる．まずひとつは，先に述べた片方向通信路の測定に限定されていることである．エンドノード測定は，対向ノードを指定して直接 End-to-End のトラフィック測定を行うことができるというメリットがあるが，現状では iperf のように限られたノード間でしか双方向測定が実現されていない．1 章で述べたように，現状では P2P アプリケーションにより，エンドノード間の大容量・双方向通信が広まってきている．これらのアプリケーションがネットワークの状況を把握するために，片方向通信路の性能測定だけでは不十分である．もうひとつは，測定した情報はそのノード内だけに限定され，また過去の履歴を保持しておくための機構に乏しいことである．現状でこれらの情報を得るためには，測定のためのサーバを恒常的に設置するなどの手段が必要となる．

2.2.2 大規模測定プロジェクト

企業や iDC(Internet Data Center) の Web サーバ，FTP サーバの性能などを，インターネット上の分散した地点から行い，その評価結果をリポートするといった，広域ネットワークのトラフィック測定情報の提供は既にコマーシャルサービスになっている ((25)，(26)，(27))．

また，AS 間経路制御やトポロジ構築など，研究分野としても Caida CoralReef (16)，UCSB Mantra (17)，WIDE Project MAWI (28) など複数のプロジェクトが広域なトラフィック測定を行っている．これらでは，ルータに独自の測定アプリケーションを注入したり，OSPF や BGP のタッピングを行ったり，また SNMP を用いてネットワーク情報を収集したりといった手法を用いているものが多い．本節では，ユーザからの測定により関連のある，末端ノードを用いた広域ネットワーク測定プロジェクトについて以下で検討する．

skitter(caida)

skitter (1) は CAIDA プロジェクト (29) によって運営されている広域ネットワーク測定システムである．skitter では，システムを構成するサーバから，traceroute に似た多数の測定パケットを複数の Destination IP アドレスに向けて送出し，広域なトポロジデータを収集する．skitter は Site-to-Site のトラフィック測定として，RouteViews (2) など複数の他プロジェクトと協調し，AS 間の Peering に関する調査などを行っている (図 2.4)．

skitter サーバを運用するにはカーネルに改変が必要である．2002 年 6 月現在，18 つのサーバが運用されている．測定は片方向 (上り) のみでサーバ間での協調測定などは行っていない．収集されたデータはメンバー以外には非公開となっている．

Rocketfuel

Rocketfuel (3) は，ISP(Internet Service Provider) 内部のトポロジを，複数の public traceroute サーバを用いることによって作成するプロジェクトである．Rocketfuel では，traceroute の中から 1) 対象 ISP を途中で横断する，2) 測定元 (Source IP Address) が測定対象 ISP の中にある，3) 対象 ISP の中に Destination IP Address を設定する，の 3 つのパターンを抽出し，対象の ISP のトポロジ測定を行う．

現在 10 の ISP のトポロジを外部からの測定のみで生成できたとしているが，解析とトポロ

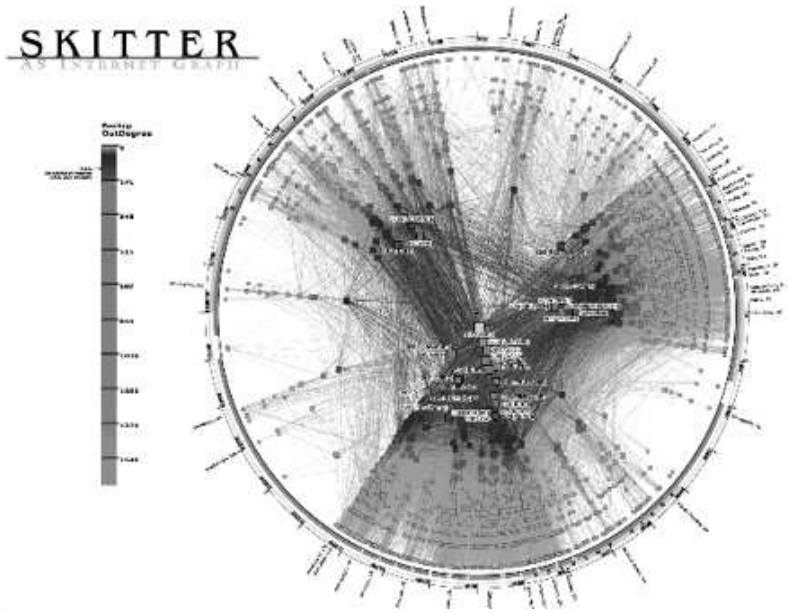


図 2.4: skitter による AS マップ

ジ作成は個々の ISP に特化した手動分析で行っている．得られる情報はバックボーン，バックボーンルータ，エッヂルータ，それらのリンクなどとなっている．

Net100 プロジェクト

Net100 プロジェクト (30) は Web100 プロジェクト (31) からの派生プロジェクトで，Netlogger (32)，NTAF (33) などがフレームワークに取り込まれている．

Netlogger は End-to-End で対抗ホストのメモリ状況，プロセス，ファイルの入出力情報などの OS のパラメータを監視できるもので，アプリケーションボトルネックとネットワークボトルネックの双方が把握できる．また，TCP MIB などを用いた Web100 カーネルを利用することで，TCP buffer など OS パラメータの変更も実現する．

NTAF は，これら Web100 カーネルによって得られた情報をデータベース化して共有している．Net100 プロジェクトでは特定 2 ノード間で相互に OS パラメータを監視しあい，お互いの最適値にコンフィグレーションすることにより，その 2 ノード間で最大のネットワークパフォーマンスを引き出すことを目的としている．

IEPM/PingER

IEPM/PingER プロジェクト (18) は，スタンフォード大 SLAC によるプロジェクトで，地理的に分散した 35 のサーバ (2001 年現在) からの継続的な ping データを集積し，分析している．1995 年からのデータを保有し，Abilene の weather map，Andover Internet Traffic Report (34)，Network Weather Service (14) など，近年の Internet Weather Report 系プロジェクトの先行事例ともなっている．しかし個々のサーバが個別に ping データを集計，蓄積しており，サーバ同士での協調は行っていない．

これらの広域なネットワーク測定プロジェクトによって得られるデータは，ネットワークの時間による推移，ドメイン間のトポロジ把握など，エンドノードや，アプリケーションネットワーク形成の際に有用な情報が多い．しかしながらこれらの測定データの多くは研究利用を目的とし，アプリケーションが利用できるまでに至っていない．また，これらのプロジェクトは測定のために大変高いコストを費やしており，エンドノードのための測定に応用できるフレームワークとは言いがたい．

2.2.3 アプリケーションネットワークの最適化

途中ノードに改変を加えずにエンドノード同士の双方向通信を促進するアプリケーションネットワークは，今後ますます発展していくと予想できる．特に，オーバーレイネットワークマルチキャストを用いたアプリケーションに関してはさまざまな研究がなされている．

これらの研究では，基本的に IP ネットワークに手を加えず，アプリケーションレイヤ上のプロトコルのみで動的なネットワーク形成，スケーラビリティ，データ検索，共有などの効率化を図ろうとしているものが多い．しかしながら，アプリケーションネットワークの構成と，IP ネットワークとの親和性が低ければ，これらの研究を効果的に生かすことはできない．

NICE (4)，Multicast CAN (5)，SCRIBE (6) などに代表されるアプリケーションネットワーク上の階層化マルチキャストに関する研究は，スケーラビリティに加え，データ共有率の最適化，検索アルゴリズムの高速化などについて述べている．最近ではこれに加え，参考文献 (7)，(8)，(9)，(10) などにおいて，効率的なアプリケーションネットワークを構築するために，トラフィック測定機構を導入したり，エンドノード同士にトポロジ情報を共有させたりといった試みが行われている．これらは，それぞれ独自にアプリケーションネットワークのプロトコルとして組み込まれている．そのため，これらの測定手法は応用，拡張が行えず柔軟性に欠ける．

Marcel らは，アプリケーションネットワーク構築に関してのトポロジ最適化の研究 (11) を行っている．この研究で提案されている MITHOS アルゴリズムでは，まずブートストラップノードに接続し，自ノードの情報を提供すると，マルチキャストなどいくつかの手法によって物理トポロジ的により近いノードが応答を返すようになっている．この手法を用いれば，例えば 10,000 ノードでネットワークを構築した場合でも最悪のケースを想定して 12Hop 以内に収まるとし，ネットワーク直径は小さく抑えられるとしている．

2.3 関連研究の考察

2.2.1 小節では，既存のエンドノードからの測定手法について述べた．これらの手法は，主に片方向の測定であり，例えばユーザがデータをダウンロードする際などに影響する下りの通信のトラフィックの測定が行えない．

2.2.2 小節で述べたように，広域なトラフィック測定プロジェクトは片方向測定であったり，測定コスト (測定コストに関しては後述) が高い．更に，これらのプロジェクトはユーザにトラフィックデータが還元されていない．

2.2.3 小節では，主にアプリケーションネットワークにおけるトラフィック測定について述べた．これらの研究ではアプリケーションネットワーク内部のデータ転送効率や，スケーラビリティを考慮したトポロジ再構成などの試みが行われている．しかしながら，これらの研究ではそれぞれのアプリケーションが形成するネットワーク上で独自のプロトコルを用い測定を行うことで最適化を実現しようとしており，汎用性，柔軟性に欠けている．エンドノードからのトラフィック測定のフレームワークを切り分けることができれば，よりシンプルなプロトコル設計が行える．

また，アプリケーションネットワークには利用モデルによってさまざまなニーズがあり，いつも物理的トポロジが優先されるとは限らない．例えばストリーミングアプリケーションならば遅延やジッタを考慮したネットワーク構築，データの共有システムの際にはスループットを優先したネットワーク構築，などといったようにアプリケーションの要求に応じて優先するべきパラメータを変える必要がある．

表 2.2 に，エンドノード測定に関する問題意識と，既存の先行事例の比較をまとめる．

本章では既存のトラフィック測定機構で，これらのユーザに必要なトラフィック情報を提供できるかの検討を行った．この表が示すように，各関連研究は各々の分野においてある程度の要

表 2.2: 関連研究と要求事項のまとめ

	1	2	3	4	5
エンドノード測定手法 (一般)	×	×	○	○	○
広域測定プロジェクト	×	○	×	×	○
アプリケーションネットワーク上の測定機構	○	×	○	○	×

1 双方向測定

2 測定履歴の共有と参照

3 ユーザへの測定データ提供

4 測定コストの削減

5 汎用性，応用性

求は満たされているといえる．しかしながら，1.3 節で述べたような利用形態に即した，エンドノードのユーザに包括的にトラフィック測定環境を提供し，ユーザ間で測定データを共有するフレームワークはまだ実現されていないことが分かった．本考察を踏まえ，次章ではエンドノードに必要なトラフィック測定の要件を定義し，測定機構の提案を行う．

第3章　エンドノード間トラフィック測定機構の提案

本章では，第 2 章でまとめた，エンドノード測定に対する既存の測定手法の問題点を踏まえた上で，エンドノード間の協調測定，及びデータ共有のために必要な要件を定義し，要求を満たすシステムの提案を行う．

3.1　要件の定義

2.3 節にて，既存のトラフィック測定手法及びプロジェクトではこれからのエンドノードのユーザの利用形態に合わせたユーザにトラフィック測定情報を提供するための枠組みが不十分であることを述べた．本節ではこれを踏まえた上で，エンドノードのユーザからトラフィック測定を実現するための要件定義を行う．

1. ドメイン間にまたがった広域なトラフィック測定
   P2P アプリケーションの発達と Last 1 Hop の広帯域化，常時接続の普及でユーザとユーザが直接データを送受信することになった．すなわち，ドメイン内部のトラフィック測定情報をユーザに還元するだけでは不十分であり，実際の利用形態により近似した，広域なトラフィック測定環境を提供する必要がある．

   例えばユーザ A が行ったトポロジ測定結果，同様に B，C が行ったトポロジ測定結果があるとすると，これらを一つのデータベースにマージすることにより，ネットワーク全体のトポロジの様子が把握できる (図 3.1)．

2. 上りと下りを切り分けたトラフィック測定
   次に，トラフィックの非対称性が挙げられる．ユーザ同士は相互にデータを送受信するが，インターネットの性質上，上りと下りのネットワーク性能は同一ではない．さらに ADSL，無線，PHS からの移動体通信などの接続形態の多様化により，スループットの非対称性

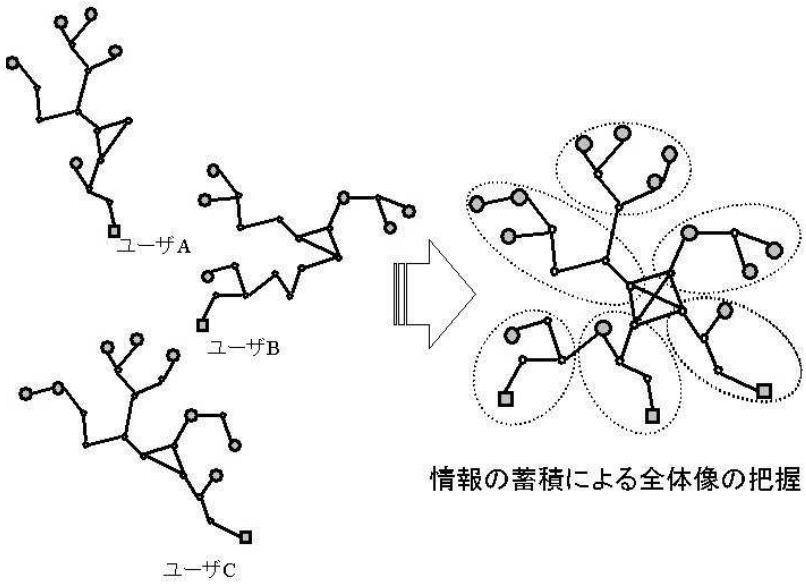


図 3.1: データを一元管理すると効果的な例 (トポロジ)

も一層増している．1 章で述べたように，エンドノードからも，双方向のトラフィック測定環境が求められる．

3. 測定情報の共有と再利用
   さらに，アプリケーションネットワークにより，複数ユーザがグループを形成するケースが増加している．これらの複数ユーザが互いにネットワーク性能を共有しあうことにより，より的確なアプリケーションネットワークの形成が行える．ユーザ間で測定情報がシェアできるトラフィック測定環境が求められる．また，これを測定情報を測定時間と共に共有することで，過去の情報も再帰的に利用可能となり，トラフィックの時間変移もユーザに提供できる．

4. 測定コストの軽減
   2.1 節にて，既存の手法の測定コストについて述べた．本研究の要件として，エンドノードのユーザにトラフィック測定環境を提供する際に，既存のシステムに改変を加えないようにする．また，P2P アプリケーションネットワークの多くはノードの入れ替わりが頻繁に起こるオープンネットワークである．これらのアプリケーションに対応したトラフィック測定を行うためには，常時接続のサーバを設置したり，中間ノードにおけるモニタリングなどの測定コストも削減する必要がある．これらの測定のためのコストを軽減することで，拡張性，汎用性及びユーザの利便性を高める．

本研究では，ネットワークのトラフィック情報の共有を論点とする．実際に効率的にネットワークを利用するためには，本来トラフィック状態の他にエンドノード自体のハードウェアスペック，利用するソフトウェアスペック，プロトコルの設定などが影響を及ぼすが，これらは別のフレームワークでユーザに提供されるべきと考えられる．以降，本研究ではエンドノード同士が共有するべき情報としてトラフィック測定情報のみを扱う．

3.2 測定要素

本研究では，エンドノードのユーザに必要なトラフィックの測定を包括的に提供するためのフレームワークを対象とする．トラフィックに影響を与える要素として，ルータの処理をSNMP(Simple Network Management Protocol) MIB で監視したり，ルーティング制御プロトコルをモニタリングしたりといった，ネットワーク中間ノードによる測定項目も考えられる．しかし，これらはユーザからみると，データ伝送の遅延やパケット損失率などの別のパラメータに置き換えられる．また，ユーザがこれらの測定を行っても，ルータに対して何らかの改善を要求することは通常考えられない．

これを踏まえ，本研究では測定要素として，エンドノードから End-to-End で対向ノードとの間で測定可能な項目を対象とする．現状では以下のようなものが考えられる．また，ネットワーク測定手法は日々進化しており，ここに挙げたものの他の要素にも測定要素の増加に柔軟に対応できるよう，システム構築の際には拡張性を持たせる必要がある．

rtt(round trip time) ネットワーク上のノード間で，パケットが往復する時間を rtt(往復時間 (round trip time)) と呼ぶ．rtt を測定することによって，ネットワークの大まかな遅延を知ることができる．rtt はパケットの往復時間なので，相手ノードでの処理時間も含まれる．

経路 データがネットワーク上のどのノードを介して到達したかを示す道のりを経路と呼ぶ．パケットの TTL(Time To Live) を調整することによって ICMP により途中に経由したノード情報を得ることができる．この経路からトポロジを推定することができる．

パケット損失率 発信されたメッセージに，相手ノードから到着確認メッセージが届く仕様に

なっているプロトコルがある．このプロトコルで送信パケット数と到着確認パケット数の比を取ると，どの程度のメッセージがネットワーク上で損なわれているかを知る目安となる．これをパケット損失率 (packet loss rate) と呼ぶ．パケット損失率によって，ネットワークのトラフィック流量が推測できる．

実効帯域幅 データがノードからノードへ通信される時，途中でいくつかのネットワークを経由することになるが、その際ネットワークの帯域の太さが一定であることは稀である．多くのネットワークでは途中経路の帯域幅は異なり，また同じ帯域を持っていても時間によるトラフィックの流量などによって有効な帯域幅は異なる．この実際に利用できる帯域を実効帯域幅とする．

帯域がそれまで経てきた経路のものよりも太いまたは同じ帯域幅である場合，パケットが中継ルータから送り出される間隔はそれまで伝送されてきたパケットの送信間隔と同じに保たれる．しかしより細い帯域をもつネットワークに中継される場合，ルータは送り出すことのできる速度より早くパケットを受け取ることになりパケットはルータ中に滞留する．このため，パケットが送り出される間隔はそれ以前よりも長くなる．一旦細い帯域を経由したパケットは次に太い帯域のネットワークに中継されることになっても，細い帯域でのパケット間隔より短い間隔で送り出されることはない．この原理を利用した帯域計測の方法が、パケットペアスキーム (packet pair scheme) である．パケットペアスキームを用いた測定ツール pathchar (21) では，2 つの連続する (時間差ゼロ) パケットを宛先のノードに対して送信し，それぞれのパケットに対する ACK の間隔を調べることにより宛先ノードまでのネットワークにおける実効帯域幅を知ることができる．

スループット ネットワーク上で実際にどの程度のデータが転送できるかを示すのに「スループット (Throughput)」を用いる．このスループットの値によって，ユーザの用いるアプリケーションの出力のおよその性能を知ることができる．ネットワークにおいて，一般的にスループットは実際に転送されたデータ量を転送時間で割ることにより求められる．TCP において，受信側のノードが何バイトの処理を一度にできるかの値を「ウィンドウサイズ Window size」と呼ぶが，特に，TCP のスループットはこのウィンドウサイズに比例する．TCP スループットの計算にはパケット損失率，タイムアウト時間などさまざまな要素が絡むが，TCP のスループット T はおよそ以下の式 (式 3.1) で近似することができる．

$$T = \frac{WindowSize}{RTT} + \alpha \tag{3.1}$$

この値は実効帯域幅に依存する．文献 (15) にて，スループットは最終的には帯域幅に依存する安定した一定値を示すことが述べられている．

遅延とジッタ IP によるデータパケットの送信は非同期であるため，ネットワークの混雑やルーティング等の状況によっては途中のルータのメモリ上に滞留することがある．また、送信元から宛先まで情報を伝播するためにも，ある程度の時間が必要である．このようにして各パケットは送り出した時刻からある程度遅れて宛先に到着する．これを遅延と呼ぶ．また遅延は一定ではなく、ネットワーク状況によって到着時間に揺らぎが生じる．これをジッタと呼ぶ．遅延とジッタによってネットワークの混雑状況及び経路を推定することができる．

3.3 エンドノード測定支援モデルの提案

前章までで，ユーザに有用なトラフィック測定要素の定義と，既存の関連研究の測定手法がユーザに還元するに当たって不十分な点をまとめた．本研究では，3.1 節の要件を満たしたエンドノードからのトラフィック測定及びそのデータの共有機構を提案する．

まず，トラフィック測定を行い，そのデータをユーザ間で共有するためのネットワーク基盤として，クライアント・サーバモデルに代表されるセントラル (中央管理) 型と，P2P 型の検討を行った．本研究でターゲットとしているユーザが必要としている測定は固定点のサーバを基点としたトラフィック情報ではなく，各々のエンドノード間を直接測定したデータであることが望ましい．さらに，これらのエンドノードは常にトラフィック測定を行っているわけではなく，自らが測定情報を必要なときに，この測定環境を利用できる必要がある．以上の理由より，本研究では協調およびデータの共有モデルとして，P2P 型のシステム構築を行う．

測定する項目は，3.2 節に述べたようなエンドノード測定手法を用いる．これらの測定を双方向で行うことで，今後のアプリケーションネットワークの多様化に必要なトラフィックデータを収集する．

P2P 型のシステム基盤を用いることにより，測定を行うエンドノードの頻繁な入れ替わりや，測定を行うエンドノードの増加などに柔軟に対応できる．しかしながら，P2P モデルを用いる

とデータの蓄積が行えず，要件に挙げた測定データをユーザ同士で共有する環境が実現できない．そこで本研究では測定したデータは統一したフォーマットでデータベースに集約し，一元的に管理する．このデータベースを参照することにより，時間によるトラフィックの推移，過去の測定データの参照，ユーザ同士の共有を実現する．

提案するエンドノードのためのトラフィック測定システムの概要を以下にまとめる．

- 本システムでは，P2P モデルを用いエンドノードのユーザからの分散したネットワーク測定を行う．P2P モデルでユーザ同士が協調して測定し合うことで，上りの通信と下りの通信の双方向測定が行えるようになり，かつ既存のシステムよりも低コストで広域なネットワーク測定が実現される．

- また，個々の測定情報を統一したフォーマットで，時間軸に沿ってデータベースに集積する．この測定情報をユーザ同士が自由に共有できるようにする．これにより，自ノードを含まない測定情報の参照や，時間推移に沿ったネットワークトラフィックの変化などがユーザからも利用，分析可能になる．

次の 4 章以降で，これらの要件を踏まえたエンドノード間トラフィック測定システム Distributed mutual measurement mechanism(Dm3) について詳しく設計を述べる．

第4章　Dm3の設計

3 章で，既存のネットワーク測定手法で不足しているエンドノードのためのトラフィック測定の要件を定義した．本章では，エンドノード間の協調測定，及びデータ共有のためのシステム Distributed mutual measurement mechanism(以下 Dm3) の具体的な設計について記述する．

4.1　システム概要

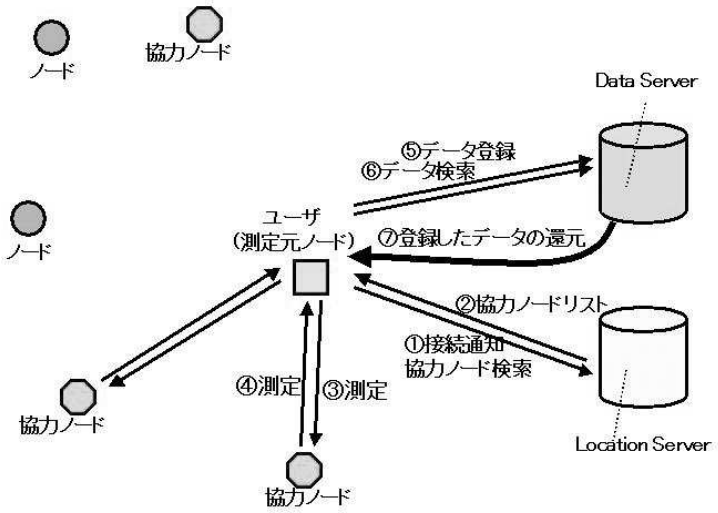
本研究で提案する End-to-End の分散測定システム Dm3 は，以下の要素で構成される (表 4.1)．

表 4.1: Dm3 の構成要素

	機能と役割
測定ノード (ユーザ)	測定を行う主体ノード
測定ノード (対向)	主ノードの要求に基づき協調して測定を行う対向ノード
ロケーションサーバ	ネットワーク上にアクティブな測定ノードの一覧を保持し，測定を行うユーザに一覧を渡す
データベースサーバ	測定データを管理，保持するデータベース

図 4.1 に Dm3 システムの全体概要および測定までの通信の流れを示す．

Dm3 は ICQ (36) や Napster (37) に代表される，いわゆる Hybrid P2P モデルと呼ばれる分散した環境を用いる．測定ノードは P2P アプリケーションとして動作することで，広域に分散したエンドノード間の双方向のトラフィック測定を，いつでも，どこからでも行えるようになる．また，これらのノードを結びつけるために，アクティブなノードのリストを管理するロケーションサーバを設ける．トラフィックの測定自体は複数の測定ノード間にて End-to-End の通信で直接行う．システム的には測定を行うノード間で差異はないが，便宜上，以降ではトラフィック測定を行いたいユーザ (主体) を測定元ノード，対向で測定に協調するノードを協力ノードと呼ぶことにする．また，これらの測定情報を時系列で管理し参照するためにデータベースを設け，過去の測定履歴を一元的に管理する．ユーザはこのサーバに問い合わせることで，測定に



1. 測定ノードが接続通知，および P2P 測定協力ノードの検索要求クエリをロケーションサーバに送信.
2. ロケーションサーバは測定ノードをノードリストに登録し，測定ノードに対し検索結果を返信.
3. 測定ノードは協力ノードに対して測定を行う．また協力ノードに測定要求クエリを送信.
4. 協力ノードは測定要求に基づき測定を行い，測定データを測定元ノードに返信.
5. 測定ノードは測定した全てのデータをデータベースサーバに登録.
6. 検索の際にデータ検索クエリを送信.
7. データベースサーバはデータ検索結果を返信.

図 4.1: システム概要

協力するノードが存在しなくとも過去のトラフィック測定情報を参照し，現状の推測を行うすることができる．

以下に測定するエンドノード (測定元ノード) のユーザが実際に本システムを用いて測定を行うまでの流れを，UML シーケンスダイアグラムに基づいて示す (図 4.2)．

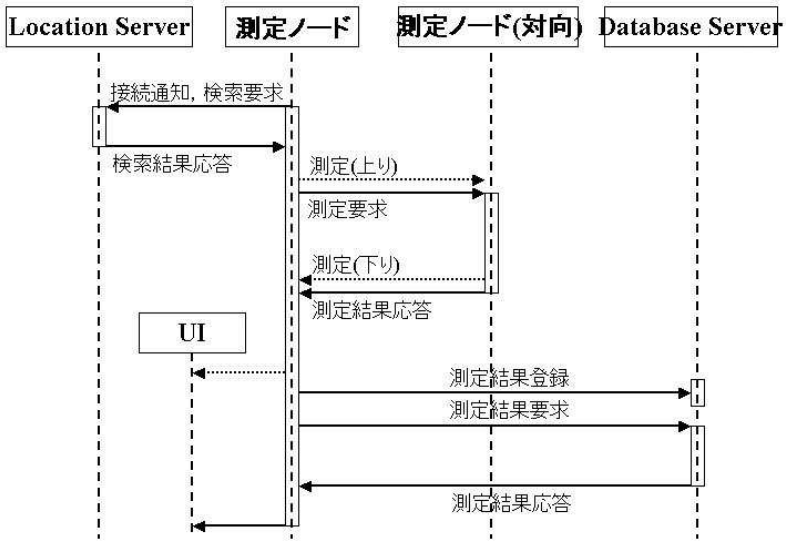


図 4.2: Dm3 シーケンスダイアグラム

ロケーションサーバはアクティブな測定ノードのリストのみを管理する．測定を行うユーザは，まずこのリストを参照することで測定対向の協力ノードを探すことができる．協力ノードに測定リクエストを送り，トラフィック測定を代行してもらうことにより，下りの通信の測定データを得ることが可能となる．各測定ノードは P2P アプリケーションとして動作し，リクエストが来たときだけ協力して測定を行う．

また，各エンドノードが行った測定のデータは測定時間と共にデータベースサーバに送信され，他のエンドノードが行った測定データとともに一元的に管理される．ネットワーク的に分散した地点の測定データを，統一したフォーマットで蓄積することで，2 章で挙げたユーザ間の測定データの共有を実現する．

4.2 測定ノード

測定ノードの役割は，ユーザがトラフィック測定を実際に行う主体になることと，他のユーザがトラフィック測定を行う際にはそれを補う協力ノードとして働くことである．

測定ノードは，ネットワークに接続されるとまず規定のロケーションサーバに接続通知メッセージを送信する．この接続通知は生存確認 (KEEPALIVE) メッセージとしてオンライン上に存在する間は，一定時間間隔で送信され続ける．実際にユーザが測定を行う際には，まずロケーションサーバで協力ノードの検索を行う． この検索には検索要求メッセージ (図 4.4) が用いられる．アプリケーションネットワークなどが既に構築され，協力ノードの位置が明らかな場合はこの手順は省かれる．実際の測定を行うには，図 4.6 に示すの測定要求メッセージを用いる．測定要求メッセージでは測定元 IP アドレス，測定先 IP アドレス，測定項目とそのオプションが指定される．測定項目及びそのオプションパラメータの指定方法は実装に依存する．

実際に測定が行われた後，自ノードからの測定データ，協力ノードからの測定データ共にユーザに表示された後，全ての測定結果はデータベースサーバに登録される．なお，プライバシなどの観点から，トラフィック測定のデータをデータベースに登録しないよう指定できる仕様も実装することが望ましい．

4.3 ロケーションサーバ

ロケーションサーバは現在オンライン上に存在する (アクティブな) 測定ノードの一覧リスト (以下リスト) を管理する．リストの Update 方法は以下の通りである．まず測定ノードがオンラインになり接続通知を受けると，そのノードの IP アドレスをリストに加える．アクティブなノードは定期的に KEEPALIVE メッセージを送信してくるので，メッセージを受け取るとそのノードをリストの一番上に更新する．一定時間以上 KEEPALIVE メッセージがない測定ノードはオフラインもしくは接続不可になったとみなし，リストから廃棄する． これによりロケーションサーバは現在オンライン上に存在する，アクティブな測定ノードだけのリストを保持する．

ノードの検索を受けるとこのアクティブな測定ノードリストから，条件を満たす協力ノードの IP アドレスリストを応答する．ロケーションサーバと測定ノードの応答は図 4.4 及び図 4.5 のデータフォーマットを介して行われる．また，データベースサーバのアドレスも保持し，測定ノードからリクエストがあった場合データベースサーバの位置も応答する．

4.4 データベースサーバ

データサーバは各測定ノードが収集したデータを統一されたフォーマットで一元的に管理し，蓄積する．また，測定データの時間推移や自分の所属しない他ドメインネットワークとのトラフィック測定データの比較を行うためのデータベースとして，検索クエリに基づき測定データを切り出す．このデータベース検索のためのインタフェースは実装に依存する．

4.5 Dm3 プロトコル

4.5.1 通信の流れとメッセージタイプ

前節までに解説した Dm3 システムを実現するためには，4 種のノード間をで通信するための以下のプロトコルが必要となる．

測定クライアント→ロケーションサーバ 接続通知及び生存確認

測定クライアント→ロケーションサーバ アクティブな測定協力ノードの検索要求メッセージ

ロケーションサーバ→測定ノード ノード検索結果の返信

測定ノード→協力ノード 測定代行の要求メッセージ

協力ノード→測定ノード 測定結果の返信

測定ノード→データベースサーバ 測定結果のデータベースへの登録

本システムでは特に断りがなければトラフィック測定アルゴリズム以外のメッセージは TCP において通信される．なにも指定しない場合のデフォルトの PORT はサーバ，測定ノード間の通信ともに 5963 と定める．

4.5.2 パケットフォーマット

Dm3 で用いるアプリケーションパケットフォーマットを以下に示す．Dm3 フォーマットでは，基礎となる 4 つのデータフィールドに複数のオプションフィールドが付く．基礎となる 4 つのフィールドは以下のとおりである．

VER 測定に用いる IP のヴァージョン (v4，v6) および Dm3 プロトコルのバージョンを指定する．

MSG_TYPE 上記の通信プロトコルのうち，どのメッセージであるかを示す．詳細は表 4.2 に後述する．

ID パケットのシーケンス番号を記述する．シーケンス番号は測定元ノード側でインクリメントしながら一意に割り振る．これにより複数の測定要求メッセージが同時に発せられたときにも測定データの一意性を保つ．

TIME タイムスタンプを格納する．本設計では 8 オクテットの timeval 型を想定している．

各メッセージタイプをまとめたものを以下の表 4.2 に示す．

表 4.2: メッセージタイプ

メッセージの種類	TYPE
ノード検索クエリ	2
ノード検索結果	3
KEEPALIVE	4
測定要求	7
測定結果	8
データベース登録	9

Dm3 の通信では，この基礎データフィールドと，各々のメッセージに必要な複数個のオプションフィールドを合わせてパケットを生成する．

KEEPALIVE メッセージ

KEEPALIVE(生存確認) メッセージは測定ノードがアクティブあることを一定時間間隔でロケーションサーバに通知するメッセージである．KEEPALIVE メッセージはノード検索に関わりなく，一定時間毎に測定ノードが送信し，ロケーションサーバはこれを返送する．デフォルトでは 20 分毎に送出される．ロケーションサーバ上ではアクティブな測定ノードリストが保持され，KEEPALIVE メッセージが一定時間以上送信されてこないノードは非アクティブになったと見なし，リストから削除する．デフォルトではノードの生存時間は 1 時間と定める．

2 オクテットの MSG_TYPE はどの種類のメッセージかをを示す．KEEPALIVE メッセージのメッセージタイプの値は 4 である．KEEPALIVE メッセージ中の 16 バイト目から 24 バイト目までは 0 が埋め込まれ，原則的に参照されない．

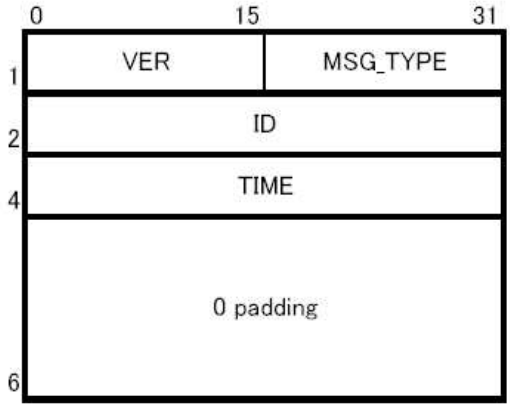


図 4.3: KEEPALIVE メッセージ

Version IP4，6 及び protocol ヴァージョン.

Message type keepalive メッセージの type は 4.

ID 測定ノードによって割り振られるメッセージシーケンス番号.

Time Time Stamp.

Padding 0 で埋められる.

ノード検索クエリメッセージ

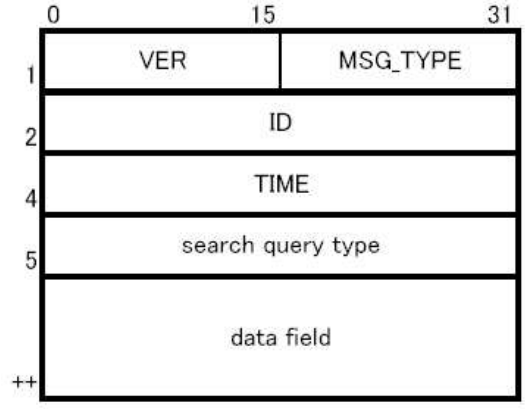


図 4.4: ノード検索クエリメッセージ

測定ノードが，対向の測定に協力してくれる P2P ノード (以下協力ノード) を検索する際，及びデータベースサーバのアドレスを検索する際には，ロケーションサーバにこのノード検索ク

エリメッセージを送信する．ノード検索クエリメッセージのメッセージ番号は 2 である．

ノード検索の際の検索の種類を node search type フィールドに記述する．データベースサーバの IP アドレスを検索したい場合はここに 2 を記述する．協力ノードを検索する際にはいくつかの検索タイプがあり，何も指定しない場合は最新のノードリストを要求する 3 が埋め込まれる．検索の種類として AS 番号，アドレスプレフィックス，ドメイン名を指定できる．これらを指定した場合はロケーションサーバ内で longest match によって検索され，最も一致するノードが返信されることになる．

Version IP4，6 及び protocol ヴァージョン．

Message type ノード検索クエリメッセージの type は 2.

ID 測定ノードによって割り振られるメッセージシーケンス番号．

Time Time Stamp.

Node search type Node Search Query
(Default(newset) 3, AS Number 4, Prefix 5, Domain 6),
Database server 2.

Data field 検索キー．デフォルト (最新ノード要求) の場合は参照されない．

ノード検索結果メッセージ

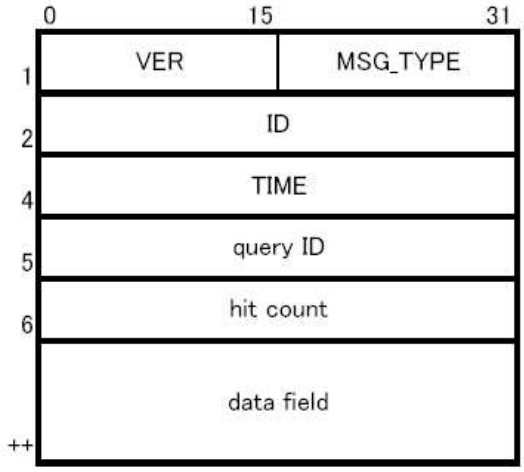


図 4.5: ノード検索結果メッセージ

ロケーションサーバから測定ノードへの応答に用いられるノード検索結果メッセージには，検

索条件に合致したアドレスの個数，および検索に合致したアドレスのリストがオプションフィールドに記述される．測定ノードから要求された検索キーに関わらず，ノードリストは IP アドレスで応答される．これはマルチホーム環境や，複数のネットワークインターフェースを持つノードの重複などを防ぐためである．これにより，ユーザは協力ノードを IP アドレスで指定することになる．

Version IP4，6 及び protocol ヴァージョン．

Message type ノード検索結果メッセージの type は 3.

ID サーバによって割り振られるメッセージシーケンス番号．

Time Time Stamp.

Query ID 検索を要求したリクエストパケットの ID.

Hit IP address count 検索にマッチしたアドレスの個数．

Data field 検索にマッチした IP アドレス．

測定要求メッセージ

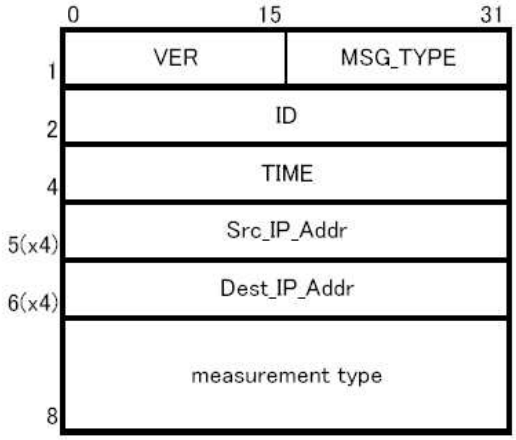


図 4.6: 測定要求メッセージ

測定ノードから協力ノードへ測定代行を要求するために，この測定要求メッセージを送信する．測定要求メッセージのメッセージ番号は，6 である．このメッセージにより，測定元，測定先の IP アドレス，及び測定する項目が指示される．測定する項目は 3.2 節に述べたような手法及びパラメータを想定する．測定項目は，パケットサイズ，パケット数，測定回数などの各々の測定項目に固有のオプションパラメータと共に Measurement type フィールドで指定される．Measurement type フィールドの定義は実装に依存する．

Version IP4，6 及び protocol ヴァージョン.

Message type 測定要求メッセージの type は 6.

ID 測定ノードによって割り振られるメッセージシーケンス番号.

Time Time Stamp.

Source IP Address 測定元ノードの IP アドレス.

Destination IP Address 測定先ノードの IP アドレス.

Measurement type 測定する項目.

測定結果メッセージ

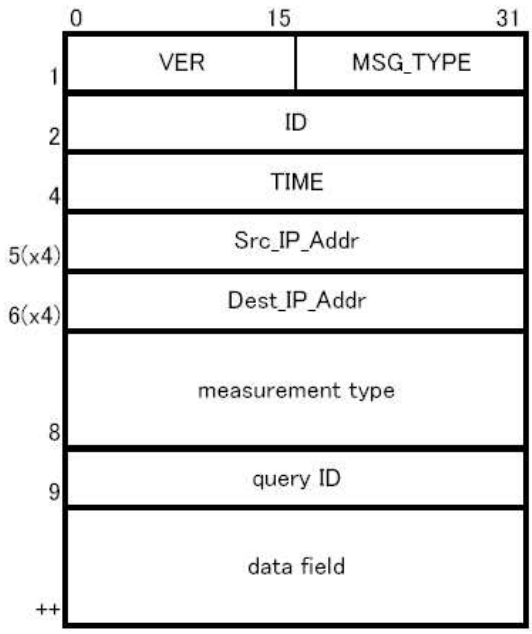


図 4.7: 測定結果メッセージ

協力ノードによって行われた測定結果は測定元ノードに返信される．測定結果メッセージのメッセージタイプは 7 である．測定結果メッセージには，測定元ノードの IP アドレス，測定先ノードの IP アドレス，測定要求メッセージの ID 番号など，測定要求メッセージの必要なフィールドはコピーされる．実際の測定結果はデータフィールドに入力され，協力ノードから測定元ノードへデータが返信される．

Version IP4，6 及び protocol ヴァージョン.

Message type 測定結果メッセージの type は 7.

ID 協力ノードによって割り振られるメッセージシーケンス番号.

Time Time Stamp.

Source IP Address 測定元ノードの IP アドレス.

Destination IP Address 測定先ノードの IP アドレス.

Measurement type 測定する項目.

Query ID 測定を要求したリクエストパケットの ID.

Data field トラフィック測定データ.

データベース登録メッセージ

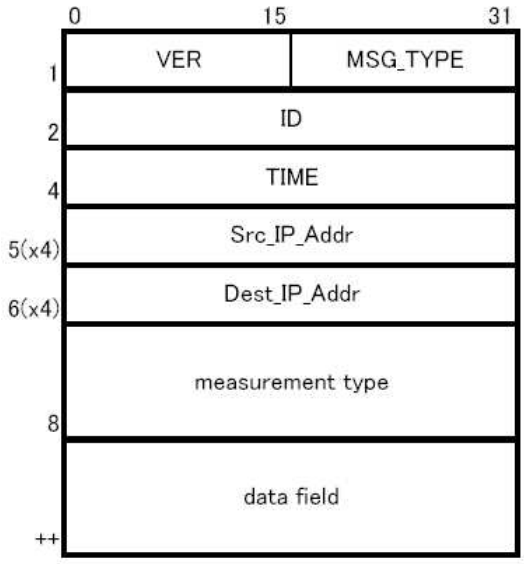


図 4.8: データベース登録メッセージ

測定された全てのデータは，このデータベース登録メッセージによってデータベースサーバに登録される．データベース登録メッセージのメッセージタイプは，9 である．測定ノードの測定データは，自ノードを起点とした測定もこのパケットフォーマットに格納され，データベースサーバに送信される．

Version IP4，6 及び protocol ヴァージョン.

Message type 測定結果メッセージの type は 9.

ID 測定ノードによって割り振られるメッセージシーケンス番号.

Time Time Stamp.

Source IP Address 測定元ノードの IP アドレス.

Destination IP Address 測定先ノードの IP アドレス.

Measurement type 測定した項目.

Data field トラフィック測定データ.

第5章　Dm3の実装

本章では，前 4 章での要件定義及び設計に基づき，実際にプロトタイプとなる実装を行う．

5.1　実装環境

本システムは，以下の環境で実装を行った (表 5.1)．

表 5.1: 実装環境

	クライアントノード	サーバ群
OS	FreeBSD4.7-REL	FreeBSD4.6.2-REL
Compiler	gcc version 2.95.4 20020320 [FreeBSD]	gcc version 2.95.3 20010315 (release)
Database		PostgreSQL 7.2b2

測定ノード，ロケーションサーバ及びデータベースサーバ共に FreeBSD 上の C 言語で実装した．開発に用いたマシンは PentiumIII 600MHz(Mem 512M NIC fxp0) および Athlon 1.4GHz(Mem 768M NIC rl0) である．

5.2　測定ノード

図 5.1 に測定ノードの実装概念図を示す．

測定ノードシステムは大きく分けてロケーションサーバと通信をする Node list manager 部，測定を制御する Measurement controller 部，最終的に測定したデータをユーザにアウトプットし，またデータサーバに送信する Data Sender 部に分かれる．

測定ノードで取り扱う測定項目は，各測定機能を独立したモジュールとして切り分けることにより，新しい測定手法にも柔軟に対応するようにした．なお，本実装ではあらかじめ表 5.2 の

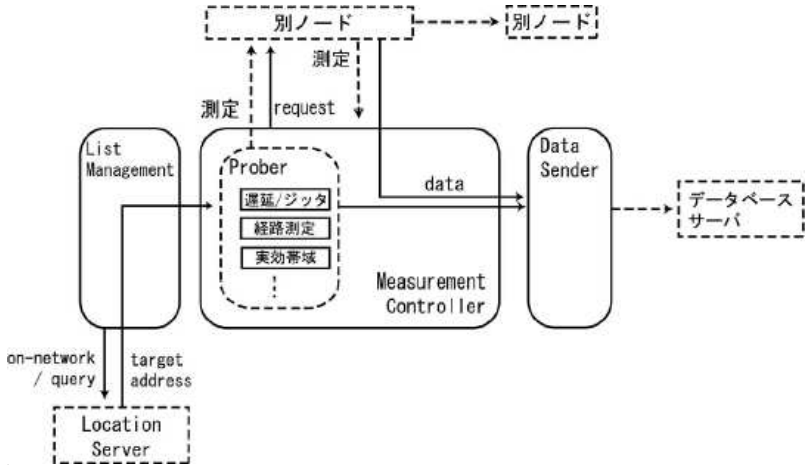


図 5.1: 測定ノードシステム概念図

ような測定モジュールを実装した．各々の測定を行う際には，モジュールの指定および必要なオプションを引数として指定する．測定項目を独立することにより，新しい測定手法に対応する際も，本機構にモジュールを加えるだけで容易に機能を追加できる．時間の同期に関しては，各測定ノード間は SNTP(Simple Network Time Protocol) で同期が取られ，誤差はミリ秒以内に押えられていると想定している．

表 5.2: 測定モジュール

モジュール	パラメータ	入力
遅延測定	上りの遅延，ジッタ，パケット喪失率	測定パケット数，パケットサイズ
途中経路	途中経路，RTT，パケット喪失率	測定パケット数，パケットサイズ
パケットペア	実効帯域	プロトコル，測定パケット数，パケットサイズ
TCP スループット	TCP スループット	測定パケット数，パケットサイズ
UDP スループット	UDP スループット	測定パケット数，パケットサイズ

実際のデータフォーマットは，測定モジュール毎に構造体を構築し，測定結果フォーマットのデータフィールド及びデータベースサーバではこのフォーマットで格納される (図 5.2 ：rtt 測定モジュール構造体)．また，測定したデータはユーザに表示される (図 5.3) と同時にデータベースサーバに送信される．

```
struct rttm{
       ...(snip)
  int16_t type;
  int16_t hop;
  int16_t recvpkt;
  int16_t sendpkt;
  struct timeval rtt_min;
  struct timeval rtt_sum;
  struct timeval rtt_max;
       ...(snip)
};
```

図 5.2: 測定データ構造体 (rtt 測定モジュール)

```
jr@mistral[~]%dm3 -s -t addr.config
----------------------------------------
[module]:TCP Throughput
[time]: Jan 5 15:26:35 2003 JST
[ID]:14785
[window size]: 56.0 KByte
----------------------------------------
203.178.143.82 -> 219.184.14.5 :5900 Kbits/sec
203.178.143.82 <- 219.184.14.5 :4220 Kbits/sec
203.178.143.82 -> 198.66.251.26 :327 Kbits/sec
203.178.143.82 <- 198.66.251.26 :867 Kbits/sec
203.178.143.82 -> 202.249.25.24 :69.6 Kbits/sec
203.178.143.82 <- 202.249.25.24 :725 Kbits/sec
219.184.14.5 -> 198.66.251.26 :117 Kbits/sec
219.184.14.5 <- 198.66.251.26 :773 Kbits/sec
198.66.251.26 -> 202.249.25.24 :57 Kbits/sec
198.66.251.26 <- 202.249.25.24 :43.6 Kbits/sec
jr@mistral[~]%
```

図 5.3: Dm3 によるスクリーン出力 (例：TCP スループット測定モジュール)

5.3 ロケーションサーバ

ロケーションサーバは図 4.5 のフォーマットにより TCP で各測定ノードと通信し，図 5.4 のようなノードリストのデータを保持する．

接続通知があると RADB と DNS により AS number と Hostname を追加し，アドレスをノードリストに加える．登録されているノード数が一定値を超過しても現実装では生存確認応答は行わず，時間毎にタイムスタンプの古いノードを順に破棄していく．これによりノードリストは常に最新を保つ．

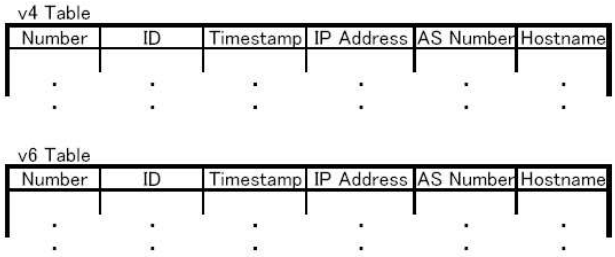


図 5.4: ロケーションサーバ・データベース

協力ノードを検索する際には，ノード検索メッセージが，AS 番号，IP address，Domain 名のいずれかの条件と共に送信されてくくるので，longest match を行い，より条件に近いノードの IP アドレスをリストにして返信する．なお，アプリケーションネットワーク上で用いている場合など，協力ノードが既知である際には接続通知のみを行い，ロケーションサーバでノード検索を行う必要はない．

5.4 データベースサーバ

データサーバは各測定ノードと TCP の任意の PORT(実装ではデフォルトで 5963) で通信する．データベースは FreeBSD 上の PostgreSQL で実装し，データの抽出，加工が容易となっている．登録された測定データはリレーショナルデータベースで階層的に保持する．実際には，メインデータベースが検索のキー ID を生成し，測定データは項目毎の各サブデータベースに ID と共に格納している (図 5.5).

また，これらは hash アルゴリズムを用いて生成したキー ID を検索に用いることによって検索の高速化を図る．また新しく追加される測定モジュールがあっても，データベース側ではサ

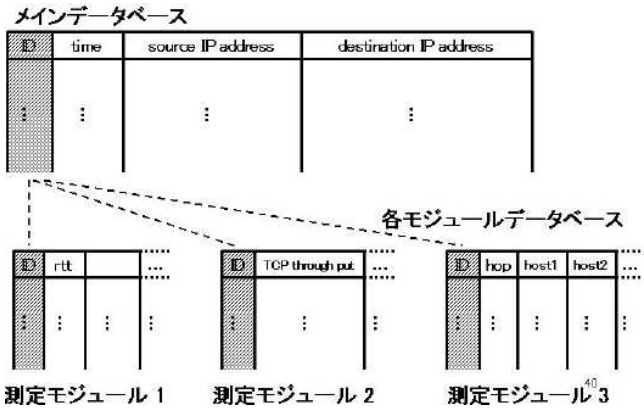


図 5.5: データサーバ・データベース

ブデータベースを追加するだけでよく，拡張性も確保される．なお，本実装では，データの解析のために，検索プロトコルとは別に PHP を用いたフロントエンドを用意した (図 5.6).

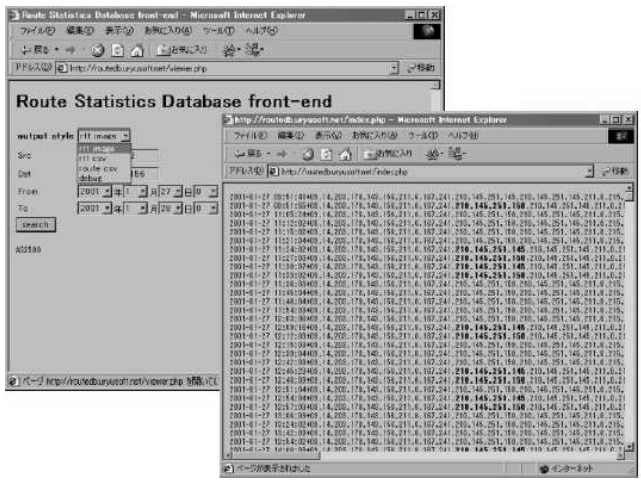


図 5.6: データベースサーバのフロントエンド

第6章　本機構の評価

本章では，実装した Dm3 をインターネット上で実際に運用し，得られた測定データを 1.3 節で提案したユーザの利用モデルに沿って分析し，評価を行う．また設計に挙げた問題点および要件の定義を再考し，関連研究との定性的な比較を行う．

6.1　エンドノードによる利用モデルからの評価

本機構 Dm3 を実際にインターネット上で運用し，測定データを収集した．表 6.1 に評価環境を示す．なお，この表では複数ノードをグループ化し同時に複数測定を行った場合でも 1 回の測定と計算している．

表 6.1: 評価環境

測定ノード数	12 ノード (WIDE，NTT Verio，Yahoo BB，OCN，Asahi-net 他)
述べ測定回数	2436 測定
述べ測定期間	約 366 時間
測定モジュール	rtt，経路，Throughput

本機構 Dm3 を 12 ノードで運用し，測定データを収集した．具体的にはそれぞれの測定項目ごと，30 分に 1 回の UNIX cron による自動測定と，マニュアルによる測定を併用して行った．自動測定の場合，3 ノード間の測定を行った．これは測定元ノードと，残りの 11 ノードの中からロケーションサーバを用いて無作為に抽出された 2 つのアクティブノードの計 3 ノードで測定を行った．本機構は P2P モデルを用いた分散機構である．そのため 12 ノードで Dm3 が運用されていても，測定を行いたいときに全てのノードがアクティブであるとは限らない．自動測定では無作為に 3 ノードを抽出することで，実際の利用形態により近い測定環境を想定している．

測定に用いたモジュールは rtt 測定モジュール，途中経路測定モジュール，TCP スループット

測定モジュールの 3 つである．測定を行ったノードは，WIDE，NTT Verio，Yahoo BB，OCN，XEPHION，GlobalVision，Asahi-net，Dolphin International など，国内外のネットワーク組織に分散して位置している．これらの測定結果は全てデータベースサーバに登録される．

6.1.1 アプリケーションネットワーク上のサーバノード選出

1.3 節では，測定データの利用モデルとして，アプリケーションネットワークのトポロジの最適化を挙げた．本小節ではこの利用モデルについて議論する．

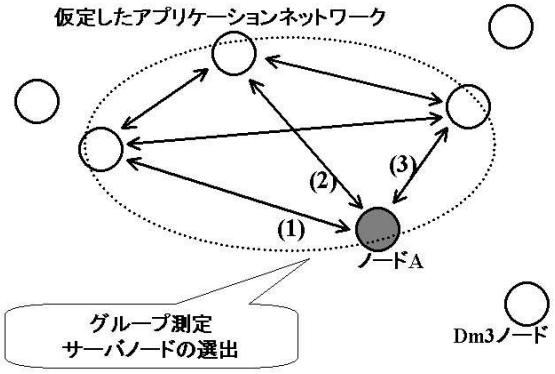


図 6.1: 4 ノード間サーバ選出測定の概要

アプリケーションネットワークのトポロジ最適化を実現するために，本機構を用いて，実際に複数ノードから構成されたグループの中からサーバに適したノードの選出を行った．具体的には 12 ノードの中から任意の 4 ノードを抽出し，この 4 ノード間で 1 時間毎にに全トポロジ (フルメッシュ) に対して双方向測定を行った (図 6.1)．本評価では，これを 200 回繰り返した．使用したモジュールは TCP スループット測定モジュールと rtt 測定モジュールである．

表 6.2 はその結果を示したものである．

表中の測定ノードの位置とは，4 ノードのネットワーク上の位置を示す．サーバとなるノードの選出アルゴリズムは以下の通りである．

1. 測定結果から，rtt，TCP スループットどちらについても，ノード A からの測定値 (1)(2)(3) を全て合計する．

表 6.2: サーバノード選出測定の結果

測定ノードの位置	rtt[1]	TCP スループット [2]	両条件同時 [3]
WIDE	60	179	56
Yahoo BB	121	0	0
AI3net	15	7	1
NTT Verio	0	0	0

1 rtt 合計が最小値でサーバに選出された回数

2 TCP スループット合計が最大値でサーバに選出された回数

3 同時刻の測定で [1]，[2] の場合両方ともにサーバであった回数

2. 他の 3 ノードについても同様の計算を行う．

3. rtt に関しては，この合計値が少ないノードほど，他の 3 ノードに対して近くに位置すると考えられる．よってこの 4 ノードで 遅延を考慮してネットワークを構築する際には rtt の合計値が最小のノードがサーバを担う．

4. TCP スループットに関しては，この合計値が大きいほど，他の 3 ノードに対する帯域が太いと考えられる．よってこの 4 ノードで帯域を考慮してネットワークを構築する際には TCP スループットの合計値が最大のノードがサーバを担う．

本測定では，同じ 4 ノードのグループで定期的に測定を繰り返しているにもかかわらず，時間帯によってサーバとなるべきノードが移行し，不確定である様子が分かる．

まず，例としてストレージ共有を行い，ファイルを交換するアプリケーションネットワークを想定する．このようなアプリケーションネットワークの構築には，ネットワーク帯域幅が求められる．測定結果から TCP スループットを優先すると，この場合 WIDE に位置するノードがサーバを担うことが望ましい．

次に，例としてストリーミングコンテンツを共有するアプリケーションネットワークを想定する．このようなアプリケーションネットワークの構築には，実時間性が求められる．測定結果から rtt を優先すると，Yahoo BB に位置するノードがサーバを担うことが望ましい．しかし表 6.2 から WIDE に位置するノードの方がサーバとして的確なケースも伺える．本データは 1 時間毎に行った測定の集計なので，時間帯によりサーバに適するノードが変動していることが分かる．

6.1.2 本機構を用いた選択可能な End-to-End 通信の実現

1.3 節にて，ユーザによる測定データの利用環境として選択可能な End-to-End 通信を挙げた．本小節では実際のネットワーク測定において，本機構がこの利用モデルを満たせているかを評価する．まず本機構を用いて 3 ノード間の双方向測定を行った (表 6.3)．まず，1 つの測定元ノードに対し，複数の測定ノードの中から任意の 2 ノードを抽出した．この 3 ノード間で TCP スループット及び rtt の測定を行い，その比較を行った．

表 6.3: 3 ノード間グループ測定の概要

測定元ノード	測定先ノード	測定回数	測定項目
12 ノード	${}_{12}C_2$(ランダム)	各 200 回	rtt，Throughput

測定ノードの抽出は ${}_{11}C_2$ 通りの中から無作為に行った．図 6.2 は本環境における 3 ノード間の測定のモデルを示したものである．なお，図中のノード B，ノード C は各々無作為抽出された 2 つの測定ノードを意味し，特定の固有ノードを指しているわけではない．

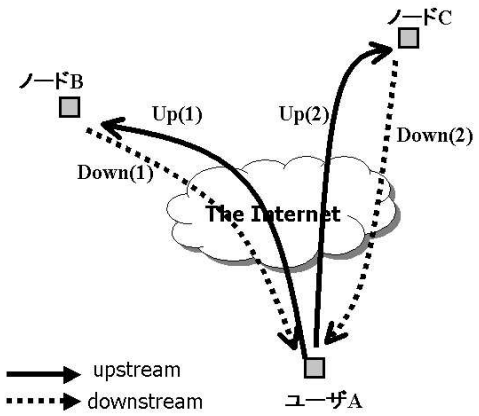


図 6.2: 3 ノード間測定のモデル図

例として，測定項目が TCP のスループットだった場合の検討を行う．1.3 節で述べたように，既存の測定手法では，測定ノードのユーザ A からの Up(1) のトラフィック及び Up(2) のトラフィックの測定を行っている．通常のネットワークでは，式 6.1 で示されるようなトラフィックが推測される．この場合，上りのトラフィック測定でノード C よりノード B の方が良いスループットを出したならば，通常，下りのスループットもノード B の方が性能が良いと考えられる．

$$Throughput : Up(1) > Up(2) \cap Down(1) > Down(2) \tag{6.1}$$

本研究ではこれに加え，ユーザ A からも Down(1) 及び Down(2) のトラフィックを測定することが出来る．このような測定を行った場合，実際には式 6.2 で示されるように，上りのトラフィック測定結果と下りのトラフィック測定結果で帯域の太いノードが異なることも考えられる．仮にユーザ A がデータのダウンロードなど下りのトラフィックの利用を目的としているならば，上りのスループットではノード B の方が性能が良いにも関わらず，ノード B ではなくノード C と通信した方が効率が良いことになる．

$$Throughput : Up(1) > Up(2) \cap Down(1) < Down(2) \tag{6.2}$$

このように，実際に双方向通信路の測定を行った結果，上りの場合と下りの場合で良い性能を示すノードが異なる現象を便宜上「優位なノードの逆転」と呼ぶことにする．

TCP スループットの測定

本機構を用いた 3 ノード間の TCP スループットの測定結果を表 6.4 に示す．

表 6.4: 3 ノード間の双方向スループット測定結果

測定元ノード A	式 6.1 のケース obverse	式 6.2 のケース reverse	優位なノードの逆転の割合 (%)
Node1	152	43	22.05
Node2	3	195	98.48
Node3	194	5	2.51
Node4	206	0	0
Node5	87	111	56.06
Node6	109	87	44.38
Node7	199	0	0
Node8	161	38	19.09
Node9	82	112	57.73
Node10	165	34	17.08
総計	1358	625	31.51

TCP スループットはデータ転送量であるので，数値が大きければ大きいほど通信性能が良いと定義する．表 6.4 では，2 つのノードとの上りと下りの計 4 つのフローを比較し，上りと下りともにノード B または C が優位だった場合は obverse，上りと下りで優位なノードが異なる

場合には reverse として計算した．この表から実際のネットワーク上では，上りのスループットで優位なノード (式 6.2 の場合ノード B) と下りのスループットで優位なノード (式 6.2 場合ノード C) が逆転しているケースが，全体の約 30 %も発生していることが観測された．

先にも論じた通り，測定した項目が TCP のスループットなので，上記の図 6.2 のモデルの例で，ユーザ A がデータのダウンロードを目的としているならば，ノード B ではなくノード C と通信した方が効率が良いことが分かる．

rtt の測定

表 6.5: 3 ノード間の双方向 rtt 測定結果

測定元ノード A	式 6.1 のケース obverse	式 6.2 のケース reverse	優位なノードの逆転の割合 (%)
Node1	181	22	10.84
Node2	96	99	50.77
Node3	195	0	0
Node4	122	81	39.90
Node5	178	22	11.00
Node6	110	90	45.00
Node7	117	82	41.21
Node8	183	16	8.04
Node9	191	8	4.02
Node10	192	2	1.03
Node11	180	17	8.63
Node12	193	1	0.52
総計	1938	440	18.50

同様に本機構を用いた，3 ノード間の rtt 測定の結果を表 6.5 に示す．rtt は遅延時間なので，比較した際に rtt 値が少ない方を優位な通信とした．この表から，実際のインターネット上では rtt 値においても上りと下りで優位なノードが異なるケースが約 18 %発生していることが観測された．これは，マルチホーム AS や複数のネットワークインターフェースを持つ観測ノードが含まれていたためと考えられる．

実時間性が求められるストリーミングやネットワーク対戦ゲームなどの P2P アプリケーションを利用する際には，このような rtt 値を元にしたノードの選択が重要となる．

6.1.3 利用モデルからの本機構の評価

本節では Dm3 によって得られた測定データを，1.3 節に挙げたエンドノードによる測定データの利用モデルに当てはめて評価を行った．既存の測定手法では，End-to-End 間で双方向通信のトラフィックは測定できなかった．大容量の End-to-End 通信を行う際や，アプリケーションネットワークを構築する際には，本機構が行う P2P モデルの双方向トラフィック測定のデータが重要なパラメータとなる．本機構で実際に双方向通信のトラフィック測定を行うことにより，実際のエンドノードの利用モデルを解決するための測定データをユーザに提示できていることを示した．以上から，エンドノードの測定のための本機構の有効性が示せた．

6.2 本機構のデータベースの評価

本機構で構築したデータベースサーバでは，個々の測定ノードの測定データを一元的に管理することにより，広域ネットワークの測定情報を分析することができる．

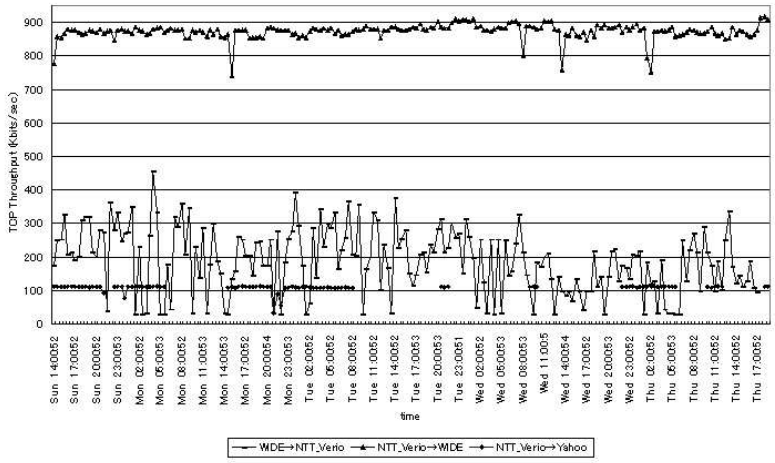


図 6.3: Dm3 による時系列のデータ参照 (例：3 点間 TCP スループット)

図 6.3 のグラフは，本機構を用いて異なる 3 ノード間で測定した TCP スループットを同一の時系列推移上で示した例である．この例では，測定時には関連性のなかった 2 つのノードの測定情報を，全く同一の軸で比較できていることが分かる．

以上から，本機構ではユーザの測定データをデータベースサーバで管理することにより，過去の履歴の参照を実現し，またネットワーク上の異なるノードで測定したデータを同一のフォーマットで一元的に扱うことが可能なことを示している．

6.3 本機構と各測定手法との比較

エンドノード測定に必要な要件定義を元に，本機構と既存の測定手法を比較した結果を表 6.6 に示す．

表 6.6: 本機構と他の測定手法との比較

	1	2	3	4	5
エンドノード測定手法 (一般)	×	×	○	○	○
広域測定プロジェクト	×	○	×	×	○
アプリケーションネットワーク上の測定機構	○	×	○	○	×
本機構 (Dm3)	○	○	○	○	○

1 双方向測定

2 測定履歴の参照

3 ユーザへの測定データ提供

4 測定コスト

5 汎用性，応用性

項目 1 エンドノードを起点とした既存の測定手法では，片方向通信路の測定情報しか得ることが出来なかった．本機構では P2P モデルを用いてエンドノード同士を協調させることにより，End-to-End の双方向測定を実現した．

項目 2 および項目 3 本機構では全ての測定データを統一されたフォーマットに格納し，データベースサーバで一元的に管理することによって，測定履歴を検索する枠組みを整えた．このデータベースを参照することにより，他のノードが行った測定情報を透過的に利用できる環境も提供された．

項目 4 既存のトラフィック測定，特に広域ネットワークの測定プロジェクトにおいては常に測定のためのサーバを挙げておく必要があった．しかし，本機構ではエンドノードがオープンな P2P ネットワークを構築することにより，コストをかけることなくいつでも必要なときに，必要な測定を行う環境が実現できた．

項目 5 ユーザの用いるアプリケーションによって，必要な測定項目は変化する．本システム Dm3 では，測定機構は各項目ごとに独立し，モジュール化した．これによって，ユーザ

は必要な項目の測定項目だけを行うことができる．また，モジュール化によって新しい測定項目への拡張が容易である．

6.4 本章のまとめ

本章では，実装した Dm3 を実際にインターネット上で運用し，収集した測定データを元に評価を行った．

Dm3 によって得られた測定データを，1.3 節に挙げたエンドノードのユーザによる利用モデルに沿って分析し，評価を行った．まず，利用モデルとしてアプリケーションネットワークのトポロジ最適化に関して検証を行った．具体的には，複数ノードのグループ間を本機構で双方向測定し，その中からサーバノードの選出を検討した．ネットワークを利用するアプリケーションの要求事項によって優先すべき項目を変えた場合でも，要求に合ったサーバノードが選出できていることを示した．次に，利用モデルとして選択可能な End-to-End の通信に関して検証を行った．実際に 3 ノード間で双方向測定を行い，ユーザが下りのトラフィック測定情報を得ることにより，ニーズに適した通信ノードを選択できていることを示した．これらから，本機構で実際に双方向通信のトラフィック測定を行うことにより，エンドノードのユーザによる利用モデルを実現するための測定データをユーザに提示できていることが示せた．

また，本機構で構築したデータベースサーバにより，ユーザの測定データの過去の履歴の参照を実現していることを示した．要求事項であった，測定データのユーザ間共有はの異なるノードで測定したデータを同一のフォーマットで一元的に扱うことで実現された．

また，問題点に挙げた，エンドノード測定に関する既存の測定手法の不足な点と比較しながら本機構の定性的な評価を行い，本機構 Dm3 が要件を満たしエンドノードのユーザおよびアプリケーションに必要十分なデータを提供できる環境を実現していることを示した．

以上から，本機構が目的に挙げた要求をを満たし，エンドノードに必要な測定のために有効であることを実証した．

第7章　結論

本章では本研究の結論として，まとめおよび今後の課題について述べる．

7.1　まとめ

本研究では，エンドノードのユーザおよびアプリケーションに End-to-End のネットワークトラフィック情報を提供し，かつ広域な測定情報をユーザ同士が利用するための機構を構築し，エンドノードのユーザの現状のネットワーク利用形態に即したトラフィック測定環境を実現した．

インターネットの広帯域化，サービスの多様化を受け，ユーザ及びアプリケーションのネットワーク利用形態は，今までのクライアント・サーバモデルから，End-to-End の双方向大容量データ通信に移行してきている．これに伴い，エンドノードへもネットワークトラフィックの情報を提供する必要性が生じている．特に P2P アプリケーションによるアプリケーションネットワーク上の通信及びその形成において，これらの End-to-End のトラフィック測定は重要性を増している．しかしながら，既存のトラフィック測定手法ではこのニーズを十分に満たすことが出来なかった．エンドノードのユーザへトラフィック情報を提供する際に問題となる点は，以下の 5 点である．

1. 片方向測定であること．
2. 測定履歴を参照できないこと．
3. 測定コストが高いこと．
4. ユーザに測定データが提供されていないこと．
5. 汎用性に欠けていること．

このような本研究における問題点から，関連研究を分析，検討する．まず，エンドノードが行うことが出来る既存のトラフィック測定手法は，主に自ノードを起点とした片方向の測定である．エンドノードが双方向・大容量の通信を行っている現状では，この測定だけではトラフィックの状態を把握するのに不十分であるといえる．次に，広域ネットワークの大規模測定プロジェクトについて検討した．これらのプロジェクトでは広域なネットワークの帯域，経路制御，トポロジなどを把握することが可能だが，測定のためのコストが高いこと，エンドノードに測定データが還元されないことなどが不十分である．さらに，既存のアプリケーションネットワーク上でのトラフィック測定について検討した．これらの測定手法はそれぞれの P2P アプリケーションの独自拡張であり，汎用性・応用性に欠ける．また，主なパラメータとして物理的トポロジを測定対象としているが，アプリケーションの利用形態を考慮すると，他のトラフィック測定情報もユーザに提供されなければならない．

これらの関連研究の問題点および不足な点を踏まえ，エンドノードのユーザに必要なトラフィック測定機構への要求事項をまとめた．この要件から，本研究では新しい P2P モデルのエンドノードユーザ間の測定，測定データ共有環境である Distributed mutual measurement mechanism(Dm3) システムを設計した．Dm3 では Hybrid P2P モデルを採用し，エンドノードのユーザ同士が協調して測定し合うことで，上りの通信と下りの通信の双方向測定が行えるようになった．また，特定のノードに依存しないオープンネットワークアーキテクチャにより，ユーザの利用モデルに即した柔軟な測定環境が提供された．また，各々の測定ノードが End-to-End で直接測定を行うことで，既存のシステムよりも低い測定コストで，広域なネットワークの測定が実現された．さらに，個々の測定情報を統一したフォーマットで，時間軸に沿ってデータベースに集積し，測定情報をユーザ同士が自由に共有，利用可能なデータベースサーバを構築した．これらにより，自ノードを含まない測定情報の参照や，時間推移に沿ったネットワークトラフィックの変化などがユーザからも利用，分析可能となった．

本論文では，実装した Dm3 をもとに評価を行った．実際に，インターネット上で複数ノードをグループ化し本機構で測定を行うことで，従来のエンドノード測定手法では測定できていなかった，複数ノード間の双方向測定データを得られた．このデータをもとにエンドノードのユーザによる測定データの利用モデルに則して，本機構の有効性を検討した．本機構で実際に双方向通信のトラフィック測定を行うことにより，エンドノードのユーザによる利用モデルを実現するための測定データをユーザに提示できていることを実証した．また，本機構で構築したデータベースサーバにより，ユーザの測定データの過去の履歴の参照を実現していることを示した．さらに，問題点と要件定義に基づき，本機構がエンドノードのためのトラフィック測定の要求事項を満たしていることを定性的に示した．

以上，本論文では，インターネット上でのエンドノードのアプリケーションによる End-to-End 通信の台頭による，エンドノード間トラフィック測定情報の共有の必要について述べ，関連研究の問題点と要件について議論した．このニーズを踏まえ，エンドノードのユーザおよびアプリケーションに End-to-End のネットワークトラフィック情報を提供し，かつ広域な測定情報をユーザ同士が利用するための機構を構築した．本論文で提供するシステム Distributed mutual measurement mechanism(Dm3) は，オープンネットワークとして測定を行うエンドノードを相互に利用することで，低コストで広域測定を行う．これにより，本研究ではエンドノードのユーザおよびアプリケーションに，利用形態に即したトラフィック測定情報を提供する環境を実現した．

7.2 今後の課題

今後の研究課題として，以下の 3 点を挙げる．

- スケーラビリティの確保
- セキュリティ
- データベースの活用
- 既存のモニタリング手法との親和性の向上

以下に，それぞれの課題について詳しく述べる．

7.2.1 データベースのスケーラビリティの確保

多数のエンドユーザの測定データを集める本機構では，スケーラビリティの再考が重要課題と言える．測定ノード自身は P2P アプリケーションとして動作するので大規模ネットワークにおいてもスケーラビリティは確保される．しかしながら本機構で提案したデータベースサーバでは，このような規模性については配慮されていない．5 章で述べたように，本研究では hash 化したリレーショナルデータベースによって負荷の軽減を測ったが，今後データの大規模化に対応し，分散データベースなどの負荷分散とスケーラビリティの確保を考慮するべきである．DNS(Domain Name System) サーバのようにデータベースを階層化しロケーションサーバのノード検索応答メッセージにより分散する方法や，利用するアプリケーションネットワーク毎にデータベースを設置するなど，データベースサーバの負荷分散に関しては検討中である．

7.2.2 セキュリティ

本論文中では，セキュリティに関する議論を行わなかった．しかしながら，本来各測定ノード及びデータベースサーバの通信は認証されるべきである．セキュリティ機構を組み込まなければ，本機構は測定ノードのなりすましや不正な測定リクエストによる DDoS の踏み台になる危険性がある．また，データベースサーバに登録する際にも，タイムスタンプや測定データの正当性を確認する機構が必要となる．解決方法として，PGP による公開鍵認証や，ロケーションサーバによるノード情報の正当性の管理など，幾つかの方法が考えられる．

7.2.3 データベースの活用

6.2 節において，本研究で構築したデータベースについての評価を行った．本研究で構築したデータベースはエンドノードによって行われた測定を統一したフォーマットで一元的に管理することができるが，そのデータの利用に関しては別の議論として，本論文中では検討を行わなかった．3.1 節の要件で示したように，これらの一元的に管理したデータを統合して分析することにより広域なネットワークの時間推移や，トポロジの状態変化が把握できる．しかし，そのためにはデータの視覚化や，複数のネットワークインターフェースを持つルータやマルチホームノードの処理など，解決しなければならない問題がある．このデータベースの利用形態に関しては今後の課題とする．

7.2.4 既存のモニタリング手法との親和性の向上

本研究では，エンドノード間の測定データの共有を保証するために，独自のデータベースサーバを構築し，トラフィック情報の交換を実現している．また，ネットワークの測定情報のみを対象とし，ハードウェアスペックやプロトコルチューニングなどのパフォーマンス向上手法との互換性は考慮しなかった．本機構を実ネットワーク上でより効果的に利用するためには，これらの既存の測定手法及びモニタリング手法とデータの協調を行うことが望ましい．関連技術として，NLANR による End-to-End Performance Framework (38) を挙げる．このプロジェクトでは，各種測定項目，モニタリング項目を XML などの一般汎用記述言語で交換しようとしているが，まだ研究途上であり，仕様は確定していない．今後，これらの汎用化手法を用いることで，よりユーザのニーズに即したトラフィック測定環境が実現できる．

謝辞

本稿を締めくくるにあたり、本研究を行なう機会を与えて下さり，多くのご指導並びにご助言をいただいた慶應義塾大学環境情報学部 村井純教授，徳田英幸教授，楠本博之助教授，中村修助教授，南政樹専任講師に心より感謝申し上げる．また親切かつ丁寧にご指導いただいた慶應義塾大学環境情報学部 斎藤信男教授にも重ねて心より感謝申し上げる．

そしてご多忙であるにも関わらず，日頃より懇切なご指導を賜わった 慶應義塾大学政策・メディア研究科の土本康生氏，同 石田剛朗氏，同 小川晃通氏，独立行政法人 通信総合研究所の杉浦一徳氏に厚く御礼申し上げる．

また，本研究の全般に渡り貴重なご意見をいただいた慶應義塾大学 徳田・村井研究室のメンバ全員，ならびに様々な形で協力してくれた KG STREAM のメンバ一同，学生コンサルタントのメンバ一同に感謝する．中でも多大な協力をいただいた鳥谷部康晴氏，本波友行氏，臼井健氏に特に深く感謝の意を表する．さらに忙しい中，貴重なご意見を下さった国際大学 Center for Global Communications の石橋啓一郎氏，株式会社 NTT Communications の林亮氏，慶應義塾大学 理工学研究科の 江木啓訓氏に深く感謝する．

公私の両面にわたり，様々に協力しながら 6 年に渡り共に研究を進めてきた同輩一同にもここで感謝の意を示す．長きに渡る執筆活動を支えてくれた，数多くのカップラーメンと蜜柑にも感謝の念を禁じえない．

最後に，苦楽を共にし心の支えとなってくれた友人たち，本研究にご協力頂いたすべての皆様に，厚く御礼申し上げる．

以上を持って本論文の謝辞とする．

参考文献

参考書籍

(1) K. Claffy, T. E. Monk, and D. McRobb, "Internet tomography. In Nature", January 1999

(2) Broido, A, claffy, k, "Analysis of RouteViews BGP data: policy atoms", Network Resource Data Management Workshop, 2001

(3) N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP Topologies with Rocketfuel", In Proceedings of ACM/SIGCOMM '02, August 2002

(4) S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", In Proc. ACM Sigcomm, Aug. 2002

(5) S. Ratnasamy,M. Handley, R. Karp, and S. Shenker, "Applicationlevel multicast using content-addressable networks", In Proc. 3rd International Workshop on Networked Group Communication, Nov. 2001

(6) M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure", IEEE Journal on Selected Areas in communications(JSAC), 2002

(7) Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal, "Building low-diameter p2p networks" 42th IEEE Symp. on Foundations of Computer Science, 2001

(8) H. Balakrishnan, M. Kaashoek, D. Karger, R. Morris, I. Stoica "Chord 'A Scalable Peer-to-peer Lookup Service for Internet Applications'", ACM SIGCOMM, San Diego, Califomia, U.S.A., August 2001

(9) Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", In Proceedings of Multimedia Computing and Networking(MMCN), 2002

(10) S. Banerjee and B. Bhattacharjee, "Analysis of the NICE Application Layer Multicast Protocol", UMIACSTR 2002-60 and CS-TR 4380, Department of Computer Science, University of Maryland, June 2002

(11) Marcel Waldvogel, Roberto Rinaldi, "Efficient Topology-Aware Overlay Network", ACM Computer Communication Review, 2002

(12) S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologicallyaware overlay construction and server selection", In Proceedings of IEEE INFOCOM'02, New York, NY, June 2002

(13) J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design", ACM Transactions on Computer Systems, pages 277-288, 1984

(14) B. Gaidioz, R. Wolski, and B. Tourancheau, "Synchronizing Network Probes to avoid Measurement Intrusiveness with the Network Weather Service", Proceedings of 9th IEEE High-performance Distributed Computing Conference, pp. 147-154, August, 2000

(15) J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation", ACMSIGCOMM, September 1998

(16) David Moore, Ken Keys, Ryan Koga, Edouard Lagache, k Claffy, "CoralReef software suite as a tool for system and network administrators", Usenix LISA, 2001

(17) P. Rajvaidya, K. Almeroth and k. claffy, "A Scalable Architecture for Monitoring and Visualizating Multicast Statistics", UCSB CS Technical Report, June 2000

(18) Warren Matthews, Les Cottrell, "The PingER Project: Active Internet Performance Monitoring for the HENP Community", IEEE Communications Magazine on Network Traffic Measurements and Experiments, 2000

(19) Vern Paxson, "Towards a Framework for Defining Internet Performance Metrics", INET'96, 1996

(20) Akira Kato, "Traffic Repository", INET'99, June 1999

参考資料/URL

(21) LBNL's Network Research Group,
http://www-nrg.ee.lbl.gov/

(22) MRTG,
http://www.ceres.dti.ne.jp/ riocat/webtools/mrtg/

(23) ttcp source code,
ftp://ftp.arl.mil/pub/ttcp/

(24) Distributed Application Support Team, "Iperf",
http://dast.nlanr.net/Projects/Iperf

(25) Go'mez Performance Network,
http://www.gomeznetworks.com/

(26) KeyLabs,
http://www.keylabs.com/

(27) KEYNOTE,
http://www.keynote.com/

(28) WIDE Project,
http://www.wide.ad.jp/

(29) CAIDA,
http://www.caida.org/

(30) Net100 Project,
http://www.net100.org/

(31) Web100 Project,
http://www.web100.org/

(32) NetLogger,
http://www-didc.lbl.gov/NetLogger/

(33) NTAF,
http://dpsslx05.lbl.gov/WK/NTAF/

(34) Andover Internet Traffic Report,
http://www.internettrafficreport.com/

(35) NSPIXP2,
http://nspixp.sfc.wide.ad.jp/

(36) ICQ,
http://web.icq.com/

(37) Napster,
http://www.napster.com/

(38) NLANR End-to-End Framework,
http://dast.nlanr.net/Projects/Framework/