

卒業論文 2002年度 (平成14年度)

オーバレイネットワークの最適化手法の提案

慶應義塾大学 環境情報学部

氏名： _____

指導教員

慶應義塾大学 環境情報学部

村井 純

徳田 英幸

楠本 博之

中村 修

南 政樹

平成15年2月22日

オーバレイネットワークの最適化手法の提案

本研究では、中継転送を用いるオーバレイネットワークにおいて、ネットワークやノードの状態に基づいて構成を動的に変更し最適化する機構を提案した。

従来の手法では、オーバレイネットワークの構成に IP ネットワークのトポロジやノードの状態を考慮しないため、通信が非効率的である。この問題を解決するためには、ネットワークやノードの状況に応じて適切な性能が得られるように動的に構成を変更する必要がある。

オーバレイネットワークを実現する手法は現在までに数多く提案されているが、既存の手法では匿名性を実現できていなかったり、動的な構成変更を行わないため、十分な性能を得られない。

本研究では、ノード広告メッセージと優先度を利用してオーバレイネットワークの構成変更を行うシステムを実装し、評価を行った。本システムではノードの情報をノード広告メッセージとして他のノードに広告する。ノード広告メッセージを受信したノードは IP ネットワークのホップ数を測定し、ノード情報とホップ数から算出した優先度に基づいてノード間リンクを動的に再構成する。

本論文で行った実装を用いて動作実験を行い、正しくこのシステムが機能することを確認した。さらに、既存のオーバレイネットワークと比較し、本システムが匿名性を実現しながらネットワークやノードの状態に基づいてオーバレイネットワークの構成を動的に最適化するのに有効であることを示した。

本論文で提案する手法を利用することにより、オーバレイネットワークでの中継転送を利用した匿名通信路を実現するうえで最適化されたオーバレイネットワークを構成できる。

キーワード

1. オーバレイネットワーク, 2. 匿名通信路, 3. リンク最適化,

Approach to optimization of overlay network

This paper describes a method of dynamic overlay network construction based on the state of a network or a node. In current techniques, performance of communication is not efficiently adjusted to its network conditions. This is because there are no considerations about the state of a node or an IP network topology. To solve this problem, we have to construct the topology of an overlay network with the state of a network or a node. There are some techniques for realizing an overlay network. However, they are not enough from the view point of communication anonymity or throughput. We designed and implemented a system for optimizing overlay network based on node advertising messages and node priority. This system advertises a node advertising message to other nodes. A node advertising message contains a state of the sending node. At first the node which received the node advertising message measures the hop count to the source on IP network. At the same time the node calculates the link priority based on the state of the source node and the measured hop count. And then, the link s dynamically optimized.

We constructed an experimental testbed, and evaluated our implementation on this testbed. As a result, it was shown that our implementation performed as we expected. On the other hand, we compared our system with existing overlay networks. We also showed that our system had advantages in adjusting a network composition based on the state of a network or a connected node with anonymity.

By using the technology described in this paper, the efficiently optimized overlay network can be realized, which has anonymous communication paths based on the relay transfer on the overlay network.

Keywords :

1 . overlay network , 2 . Anonymous communication path , 3 . Link optimization ,

Keio University , Faculty of Environmental Information

Masahiro NAKAYAMA

目次

第1章	はじめに	7
1.1	背景	7
1.1.1	サービスの通信形態の変化	7
1.1.2	匿名の必要性	7
1.1.3	オーバレイネットワークによる匿名通信路	8
1.2	オーバレイネットワークの問題点	8
1.3	目的	8
1.4	用語の定義	9
1.5	本論文の構成	9
第2章	既存技術の問題点	10
2.1	既存の研究・アプリケーション	10
2.1.1	Gnutella	10
2.1.2	Winny	10
2.1.3	Peer-to-Peer ストリーミング	13
2.2	まとめ	13
第3章	本研究のアプローチ	14
3.1	オーバレイネットワークの効率化	14
3.2	考慮すべきパラメータ	16
3.2.1	ノードの状態	16
3.2.2	ネットワークの状態	16
3.2.3	本研究で利用するパラメータ	17
第4章	設計	18
4.1	システムの概要	18
4.1.1	システムの構成	18
4.1.2	リンクの種類	18
4.2	ノード間メッセージ	19
4.2.1	メッセージのフォーマット	19
4.2.2	メッセージ受信時の動作	20
4.3	ノード間リンク	20
4.3.1	ノード間リンクの確立	20

4.3.2	ノード間メッセージの交換手順	23
4.3.3	ノード間リンクの切断	23
4.4	ノード情報の測定	23
4.5	リンク優先度の算出	24
第5章	実装	25
5.1	実装環境	25
5.2	ノードの構成	25
5.2.1	リンクマネージャ	25
5.2.2	リンクバッファ	25
5.2.3	メッセージプロシージャ	27
5.2.4	ノードバッファ	27
5.2.5	測定モジュール	27
5.3	メッセージ受信時の動作	28
5.4	API	29
5.4.1	初期化	29
5.4.2	接続	29
5.4.3	メッセージハンドラの登録	29
5.4.4	メッセージの送信	29
第6章	評価	30
6.1	動作実験	30
6.1.1	実験の概要	30
6.1.2	実験環境	30
6.1.3	実験手順	30
6.1.4	実験結果	31
6.2	定性評価	31
第7章	おわりに	33
7.1	本論文のまとめ	33
7.2	今後の課題	33
7.2.1	測定するパラメータ	33
7.2.2	メッセージの経路制御	33
7.2.3	オーバレイネットワークの最適化の指標	34

目次

2.1	Ultrappeers による階層化されたオーバーレイネットワーク	11
2.2	Winny のオーバーレイネットワーク	12
3.1	IP ネットワークのトポロジ	14
3.2	効率の悪いオーバーレイネットワークの例	15
3.3	効率の良いオーバーレイネットワークの例	15
4.1	システムの概要	18
4.2	ノード広告メッセージ	19
4.3	動作概要	21
4.4	ノード広告メッセージの動作 (優先度が高い場合)	21
4.5	ノード広告メッセージの動作 (優先度が低い場合)	22
4.6	ノードの状態遷移図	22
4.7	ハンドシェイク	23
5.1	ノードの構成	26
5.2	メッセージ受信時の動作	28
6.1	動作実験	31

表 目 次

1.1	サーバとオーバレイネットワークの比較	8
2.1	既存のアプリケーションの比較	13
3.1	パラメータ一覧	17
6.1	本システムと既存の提案の比較	32

第1章 はじめに

本章では、インターネット上でのコミュニケーションにおける匿名性の必要性を明らかにし、匿名性を実現した通信に必要となるオーバーレイネットワークの動的構成による最適化の必要性と本研究の目的を述べる。

1.1 背景

1.1.1 サービスの通信形態の変化

多くの人が参加するコミュニケーション手段としてインターネットが確たる地位を占めるようになった。特にBBS(電子掲示板)やIM(インスタント・メッセージ)のような人と人とを結ぶサービスが広く利用されている。

こういったサービスはクライアント・サーバ型で提供され、情報を一括で管理するサーバが中央に存在するのが一般的である。しかし利用者の個人情報が一ヶ所に集中することはそのサーバの管理主体のセキュリティレベルに利用者全ての情報が依存することになり、悪意を持つ者にとって格好の攻撃対象となる。実際に、悪意を持つ者による攻撃や運用上の不備等によって個人情報が漏れるという事件が多発している。

さらに近年の技術革新によってADSLや光ファイバを安価に利用できるようになったのを受け利用者へのサービスが広帯域化しまた利用者も増えていることからサーバの転送量が大幅に増大し、そのためサービスの維持に大きなコストがかかるようになった。その一方で利用者の帯域幅が増えたことからオーバーレイネットワーク上で様々なサービスを行うことが現実的となった。

1.1.2 匿名の必要性

1.1節で述べたようにコミュニケーションを提供するサービスが普及しているが、中でも利用者の匿名性を保証した匿名掲示板や、個人間商取引等で信頼できる宅配業者が仲介を行うエスクローサービスなど匿名性を提供するサービスの利用者は増加の一途をたどっている。こういったサービスの利用目的として、悪意を持つ人に個人情報が渡らないようにする自衛と、掲示板や個人間商取引での行為と実世界の個人とを結びつけないという2点があげられる。しかし匿名性をサーバに依存することは前節で述べたような問題がある。このため、通信の匿名性に対する要求は非常に高い。

表 1.1: サーバとオーバレイネットワークの比較

	単一サーバ	オーバレイネットワーク
個人情報	全てサーバ上に置かれる	各ノードに分散
ボトルネック	サーバの帯域幅 全トラフィックが集中	各ノードの帯域幅 トラフィックは分散

1.1.3 オーバレイネットワークによる匿名通信路

サーバを利用せずに匿名性を保ったまま通信を行う手法の一つに、オーバレイネットワーク上で中継者をはさむ方法がある。この方法では通信を行っているエンドノード同士が互いに直接接続しないため、通信者がお互いに匿名のまま通信を行うことができる。この匿名性を持った通信を実現するオーバレイネットワークを匿名通信路と呼ぶ。

ここで匿名とは通信者の識別子となる IP アドレスが知られない事を指す。IP アドレスは ISP のトポロジ等に依存し地域等を知る手がかりとなるほか、ネットワークによって管理者情報として本名や住所、電話番号といった確実に個人を識別できる情報につながる識別子になり得ることから、IP アドレスを匿名化することは非常に重要である。

表 1.1 に単一のサーバに依存する場合とオーバレイネットワーク上での違いを個人情報の場所と通信のボトルネックの 2 点から示す。単一サーバによるサービスでは全利用者の個人情報がサーバ上に保存されるが、オーバレイネットワークでは個人情報をサーバに集中させる必要が無い。ボトルネックという観点では単一サーバによるサービスは全てのトラフィックがサーバに集中するため規模性が低いがオーバレイネットワークではトラフィックが分散し集中しないため規模性が高い。

1.2 オーバレイネットワークの問題点

1.1.3 節で述べたようにデータの転送を第三者を経由して行うことで匿名性を確保できるが、オーバレイネットワークには以下に挙げる問題がある。

IP[1] ネットワークのトポロジを考慮せずにオーバレイネットワークのノード間リンクが構成されているため、通信が非効率になりやすい。

また各ノードがオーバレイネットワーク上に流れるデータを処理する際、処理性能の低いノードを経由するデータが多いとオーバレイネットワーク全体のパフォーマンスが低下する。

1.3 目的

本研究では匿名通信路を実現するため、ネットワークやノードの状態に基づいて構成を動的に変更しオーバレイネットワークを最適化するシステムを設計、実装する。こ

れにより、複数の中継者を介在した通信においてパフォーマンスの低下を低減できる。

1.4 用語の定義

本論文で利用する用語を、以下のように定義を行う。

オーバーレイネットワーク 本論文では IP ネットワーク上に存在する複数のノード同士がリンクを張り合う事によって論理的に構築されたネットワークをオーバーレイネットワークと呼ぶ。

このオーバーレイネットワーク上に検索要求等のメッセージを転送することで、オーバーレイネットワーク全体として何らかの機能を提供する。オーバーレイネットワークを利用したアプリケーションとして gnutella[2] に代表される分散ファイルストレージやシェアキャスト [3]、PeerCast[4] といったストリーミングなどがある。またオーバーレイネットワークを利用して匿名のまま情報発信を行うものとして Freenet[5]、Winny[6] が存在する。

ノード オーバーレイネットワークの構成要素として何らかの機能を提供するアプリケーションプロセスを指す。

リンク ノード同士で接続される TCP コネクションをリンクと呼ぶ。

オーバーレイネットワークの構成 オーバーレイネットワークに参加するノード間のリンクの構成をオーバーレイネットワークの構成と呼ぶ。

IP ネットワークのトポロジ ノード間の IP ネットワーク上における接続状態を IP ネットワークのトポロジと呼ぶ。

匿名通信路 オーバーレイネットワーク上の中継者を経由して通信を行うことで通信を行っているノードの IP アドレスを匿名としたまま通信を実現するオーバーレイネットワークを匿名通信路と呼ぶ。

1.5 本論文の構成

本論文では、まず第 2 章において既存のオーバーレイネットワークにおける問題点をあげ、第 3 章で問題点を解決するために必要な要件について論じる。第 4 章では本研究で提案するシステムの設計について述べる。第 5 章では、実装を行ったシステムの詳細を述べる。第 6 章では実装をもとに評価を行い、要件を達成できたかを検討する。第 7 章でまとめと今後の課題について述べる。

第2章 既存技術の問題点

本章では、既存のアプリケーションや過去に提案された枠組みから本研究に関連の深いものについて述べ、現在のオーバーレイネットワークにおける問題点を明らかにする。

2.1 既存の研究・アプリケーション

2.1.1 Gnutella

Gnutella[2] は、Justin Frankel と Tom Pepper によって開発されたファイル共有アプリケーションである。このシステムではオーバーレイネットワーク上に検索要求メッセージを転送することで特定のサーバに依存することなくオーバーレイネットワーク上のファイルを発見できる。この検索要求は目的のファイルを見つけるまで転送されるが、オーバーレイネットワーク上に目的のファイルが存在しない場合に検索要求で溢れないよう TTL(Time To Live:生存時間)として指定されたノード数を経由すると削除される。ファイルが見つかった場合には、検索者とファイルの保持者が直接接続し HTTP[7] によって転送される。

最初に公開された Gnutella 0.4 プロトコルは不必要なメッセージの拡散などのプロトコル上の欠陥のために規模性が低かったが、その後プロトコルの改善や規模性を確保するための工夫を取り入れている。そのうちオーバーレイネットワークの構成に階層化を取り入れたものとして Ultrapeers[8] がある。Ultrapeer は Ultrapeer ノードと Leaf ノードから構成され、複数の Leaf ノードが Ultrapeer ノードに対し接続することで Gnutella ネットワークへの参加ノード数を削減し規模性を改善している。図 2.1 にこの Ultrapeers の全体概要を示す。

ホップ数及びノードの生存時間の測定により接続先ノードを選択する試み [9] もあり、より安定したオーバーレイネットワークを維持できる。

Gnutella は匿名性確保に主眼を置いておらず、中継転送を行うといった仕組みを用意していない。そのため相手の IP アドレスを特定できる。

2.1.2 Winny

Winny[6] は中継転送とキャッシュにより公開者や取得者の双方が匿名のままファイルの配布を行うことができるファイル共有アプリケーションである。

Winny ではファイルの保持者がオーバーレイネットワーク上の近隣のノードに保持情

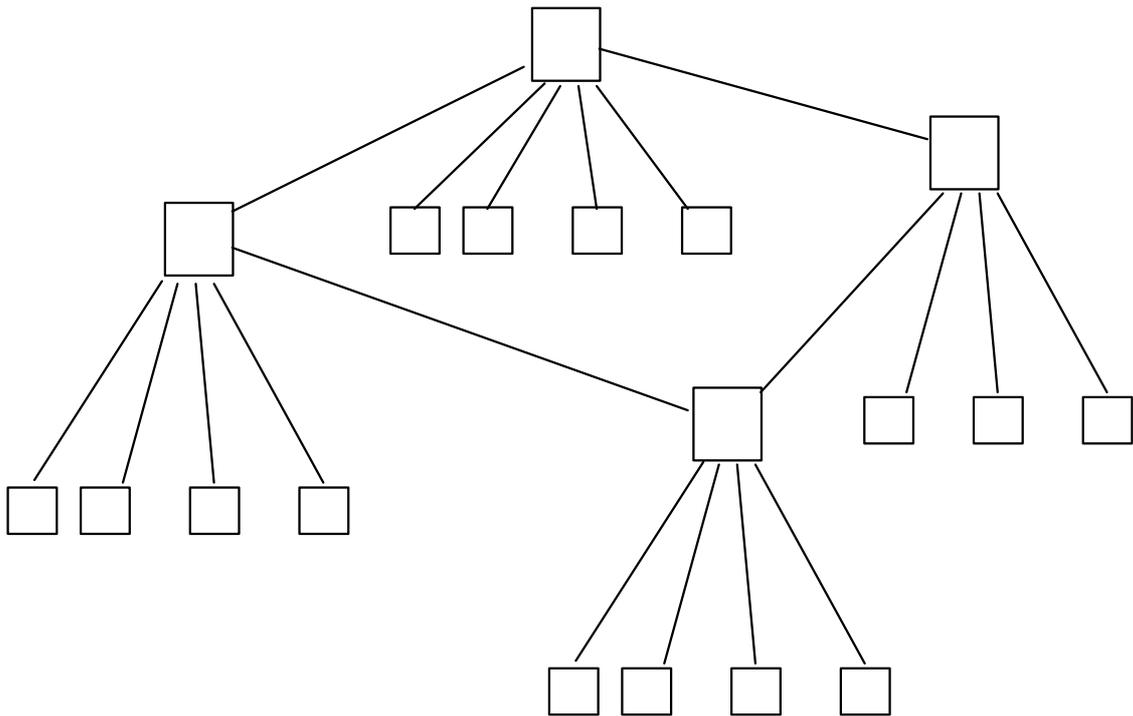
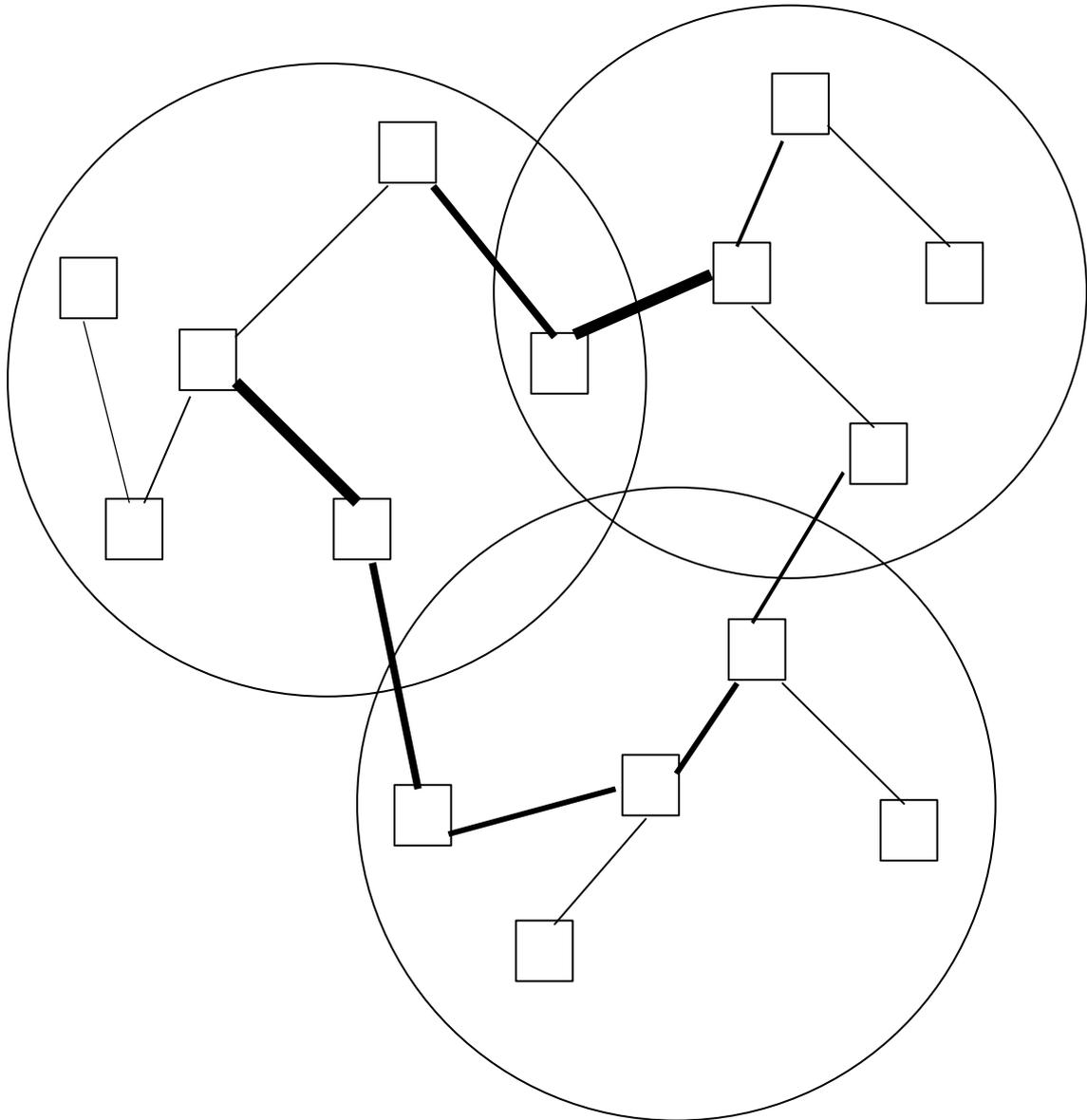


図 2.1: Ultrapeers による階層化されたオーバーレイネットワーク

報を広告する。保持情報を受け取ったノードはそれぞれ自分の取得条件に適合するかを調べ、適合すればそのファイルを取得する。オーバーレイネットワーク上で遠くにあるファイルを取得するために Gnutella のように検索要求を発行することもできる。さらに保持情報をオーバーレイネットワーク上の各ノードが転送する際に保持者 IP アドレスを書き換える。これによりファイルの保持者に匿名性を提供する。ファイルの転送を行う場合には書換を行ったノードを逆にたどり最終的な保持者まで接続する。

このオーバーレイネットワークは帯域幅と優先度の 2 つのパラメータから動的に構成される。まず高速な回線を持つノードがより多くの保持情報や検索要求を処理できるように帯域幅によって段階的にぶらさがるようにノード間のリンクを張る。さらに似た傾向のファイルを持つノードをオーバーレイネットワーク上で近くに配置するため任意に指定したクラスタ化キーワードをリンク接続時に交換し、その類似度をノードの優先度とする。さらにそのノードからのファイル転送が成功した場合に優先度を増加させる。決められた接続数上限を越えて接続された場合には、優先度の低いノードから順にリンクを切断する。図 2.2 にこのオーバーレイネットワークの全体概要を示す。

アクセス回線だけがボトルネックであると仮定しているため、通信者から中継者に至る経路上に狭い帯域幅や遅延の大きい回線があると直接転送する場合に比べ極めてパフォーマンスが落ちる。



Winny では帯域幅による階層化と要求あるいは保持している情報の傾向によるクラス
 タ化を併用している。この図では線の太さで帯域幅をあらわし、同じ傾向のファイル
 を要求ないし保持しているノードの集まりを円であらわしている。高速な帯域幅をも
 つノードが中心となり、同じ傾向を持つ低速なノードがそれらにぶらさがるように
 になっている。

図 2.2: Winny のオーバーレイネットワーク

2.1.3 Peer-to-Peer ストリーミング

オーバーレイネットワークを用いて擬似マルチキャストを実現する Peer-to-Peer ストリーミング配信システムがある。これらはストリーミングを受信しているノードがさらに別の受信ノードに中継送信を行うことで動的に配信ツリーを形成し、配信主の帯域を圧迫することなく大量のノードに同一のストリーミングを配信できる。

この Peer-to-Peer ストリーミングには、オーバーレイネットワークへの参加ノードの一覧を保持する仲介サーバを利用するものと、仲介サーバ無しで配信ツリーを構築するものに分けられる。前者に含まれるものとしてシェアキャスト [3]、後者に含まれるものとして PeerCast[4] がある。PeerCast では Gnutella を検索ネットワークとして利用しその上でストリーミングを配信しているノードを検索する。配信主ノードとそのストリーミングを受信し中継送信しているノードを区別しないため、配信主ノードの匿名性が保たれている。

2.2 まとめ

表 2.1 にこれまでに述べたアプリケーションの比較を示す。この表から、これまでの中継転送を行うアプリケーションでは十分にノードやネットワークの状態を利用していないことが分かる。

本章では既存のアプリケーションが採用している手法について述べ、それらの手法では本研究が目的とする匿名通信路を実現するオーバーレイネットワークにおける問題点を解決できていないことを明らかにした。

表 2.1: 既存のアプリケーションの比較

	転送	ノードの状態	ネットワークの状態
Gnutella Protocol 0.6	直接 (HTTP)	階層化 (Ultrappeers)	-
測定による 接続先ノード選択	直接 (HTTP)	階層化 (Ultrappeers) 稼働時間	ホップ数
Winny	中継転送	帯域幅 クラスタ化キーワード	-
P2P ストリーミング	中継転送	-	-

転送 情報の転送に中継ノードを経由するかどうかを示す。

ノードの状態 ノードの状態や設定に基づいてオーバーレイネットワークの構成を行っているかどうかを示す。

ネットワークの状態 ネットワークの測定に基づいてオーバーレイネットワークの構成を行っているかどうかを示す。

第3章 本研究のアプローチ

本章では、2章で述べた問題点を基に、効率的なオーバレイネットワークを動的に構成するシステムについて考察する。

3.1 オーバレイネットワークの効率化

2章で述べたように匿名通信路のパフォーマンスを向上させるには、ネットワークやノードの状況に応じてオーバレイネットワークを効率的に構成しなければならない。IPネットワーク上に分散してノードが存在する状態(図3.1)で、オーバレイネットワークの構成で非効率な部分を動的に最適化する。(図3.2、図3.3)

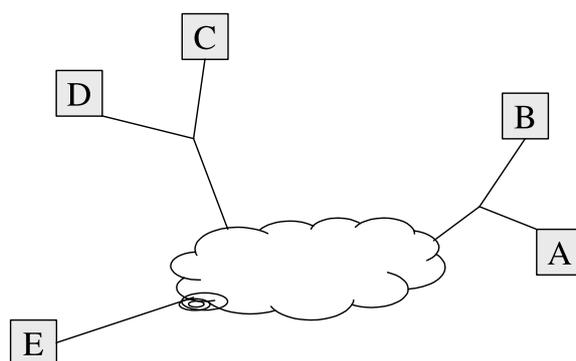


図 3.1: IP ネットワークのトポロジ

本研究では、ノード広告メッセージとノードの優先度を用いてこれを実現する。ノード広告メッセージには送信元のノードの情報が含まれ、オーバレイネットワーク上を転送される。ノード広告メッセージを受信したノードは発信元ノードに対しネットワーク状態の測定を行う。測定結果とノード広告メッセージに含まれるノード状態から別に定める式を用いてノードごとに優先度を求め、既に接続しているノードより優先度が高い場合にはその相手に接続を行う。

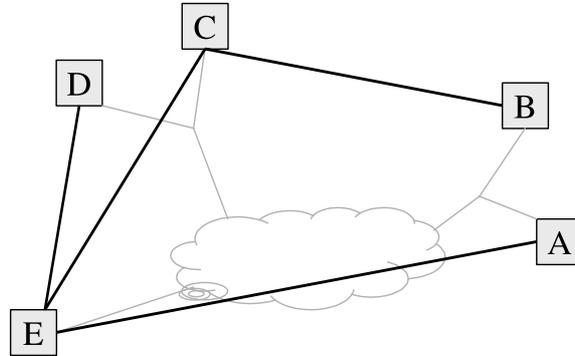


図 3.2: 効率の悪いオーバレイネットワークの例
 ノード A、B および ノード C、D は IP ネットワーク上で近いにもかかわらず離れた
 ノードに接続を行っている。

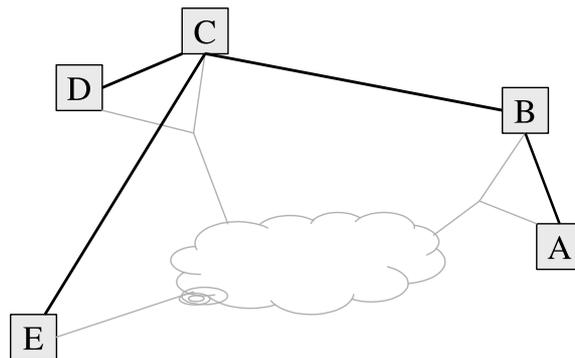


図 3.3: 効率の良いオーバレイネットワークの例
 IP ネットワーク上で近いノードに接続できている。

3.2 考慮すべきパラメータ

動的に構成を変えるにはネットワークやノードの情報を知らなければならない。本節ではオーバーレイネットワークの動的構成を行う際に考慮すべき測定要素について述べる。

3.2.1 ノードの状態

帯域幅 オーバレイネットワーク上のボトルネックの一つとして、ノードがインターネットに接続しているアクセス回線の帯域幅が考えられる。帯域幅の情報を利用し、帯域幅が太いノードに優先して接続することで転送速度が向上する。このパラメータは利用者による設定とすることで測定が不要という利点がある。

処理性能 検索要求の転送処理や通信路の暗号化などを行う場合にはノードの計算処理性能や記憶容量等の処理性能をパラメータとして考慮するべきである。処理性能の高いノードを優先することで、メッセージ毎の転送処理性能や中継転送速度が向上する。処理性能は他に動いているタスクなどに左右されるために、動的に変化することが考えられる。

稼働時間 オーバレイネットワークを安定させるには検索要求の転送やファイル転送の中継を行うノードが連続して稼働することが重要である。ここでは既に長く稼働しているノードはこれからも長く稼働し続けると仮定する。稼働時間の長いノードを優先することでオーバーレイネットワークが安定する。

3.2.2 ネットワークの状態

ネットワークの状態はノード広告メッセージを受信したノードがその発信元に対して測定を行うことで求める。

一般的にオーバーレイネットワークの利用者はネットワークの管理者等ではないユーザを対象としている。そのため経路制御プロトコルや経由するルータの SNMP[10] MIB からの情報を取得することは困難である。よって、エンドノード間で測定できる項目を利用する。

ホップ数 IP パケットの送信時の TTL をあらかじめ決め、到着時の TTL と比較することで IP ネットワーク上で経由したルータの数を知る。ホップ数を測定することで、IP ネットワーク上での距離を推定することができ、ノード間リンクの RTT を改善できる。

ホップ数の計測では PPPoE[11] 等によるトンネリングが行われている場合にそれを知ることができないという制限がある。また、マルチホームなどにより往復の経路が異なる場合には方向ごとに別に扱う必要がある。

表 3.1: パラメータ一覧

パラメータ	測定トラフィック	変化	効果
帯域幅	x	変わらない	転送速度
処理性能	x	変わる	メッセージ毎の転送処理性能 中継転送速度
稼働時間	x	増加する	オーバーレイネットワークの安定
ホップ数	少ない	基本的に変わらない	遅延、転送速度
RTT	普通	変わる	遅延
スループット	多い	変わる	転送速度

RTT ノード間でパケットが往復する時間を RTT(round trip time) と呼ぶ。RTT を測定することで、ノード間の遅延がわかる。アクセス回線の帯域幅が細い場合などには瞬間的な他の通信に左右されやすいため一般的には数度繰り返し測定した値の平均を用いる。

実効帯域幅 各ノード間で実際に利用できる帯域幅を実効帯域幅と呼ぶ。スループットの測定にはパケットペアスキームを利用した pathchar[12] などの手法が存在するが、測定に必要な通信量が多いため規模性を欠く。

3.2.3 本研究で利用するパラメータ

表 3.1 に以上のパラメータをまとめた。本研究では、計測のしやすさとそれに対する効果から、ノードの状態として帯域幅と処理性能、ネットワークの状態としてホップ数を採用する。

第4章 設計

本章では本研究で実現するシステム的设计について述べる。

4.1 システムの概要

4.1.1 システムの構成

本研究で提案するシステムは、IP ネットワーク上に複数存在するノードとそれらを結ぶノード間リンクから成り立つ。このノード同士がノード間リンクを通じてメッセージをやりとりしたりネットワークの状態を測定することで、オーバレイネットワークの構成の最適化を行う。本論文が対象とする範囲はオーバレイネットワークの動的構成に関する部分となる。図 4.1 にこの全体概要を示す。

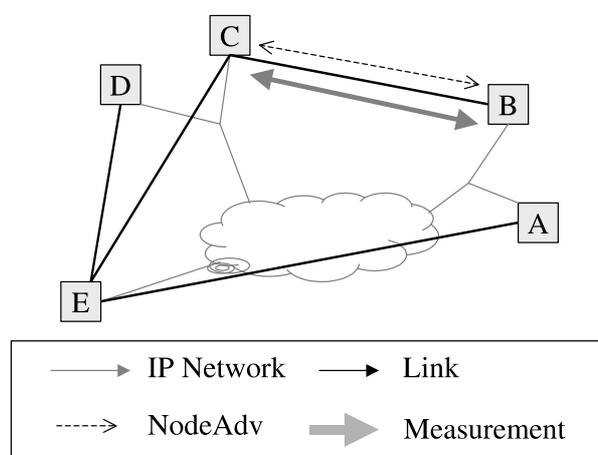


図 4.1: システムの概要

4.1.2 リンクの種類

ノード間を接続するリンクとしてTCP を用いて通信を行う。ポート番号はノードごとに任意に指定できるものとする。

リンクは以下の二種類に分けられる。

- 上流リンク自分から接続を行ったリンク。
- 下流リンク他ノードから自分に対し接続されたリンク

上流リンクと下流リンクにはそれぞれ接続上限が設けられ接続数がある上限値を越えた場合には、上流リンクの場合には後述の優先度の最も低いノードとのリンクを切断し、下流リンクの場合には最後に接続したノードを切断する。

本システムでは各ノードがこの上流リンクを能動的に張り替える。

4.2 ノード間メッセージ

4.2.1 メッセージのフォーマット

本システムではリンク上でメッセージを交換することで動作する。メッセージにはメッセージの種別を示すメッセージタイプとメッセージの長さを示すメッセージ長が含まれる。メッセージ長はバイト数で表現される。

本システムではノード広告メッセージを定義する。

ノード広告メッセージ 本システムではノード広告メッセージのフォーマットを定義する。

図 4.2 にノード広告メッセージのフォーマットを示す。

0	7 8	15
Type: 1	Length	
Protocol	(reserved)	
Address		
Portnum		
SubType: 1	Bandwidth	
SubType: 2	Performance	

図 4.2: ノード広告メッセージ

メッセージタイプは1である。ノード広告メッセージは利用するプロトコルとアドレス、ポート番号と、ノードの情報が含まれる。プロトコルに依存してアドレスのフォーマットが変わるが、IPv4 では 32 ビット、IPv6[13] では 128 ビットのアドレスが入る。ノード情報としてはサブタイプが1の帯域幅とサブタイプが2の処理性能の二組のノード

ド情報が含まれる。帯域幅はノードが接続しているアクセス回線の帯域幅で、毎秒バイト数で表記される。処理性能はノード上で実際の計算能力を計測した値を用いる。

4.2.2 メッセージ受信時の動作

本システムは、ノード広告メッセージによって以下のように動作する。

1. 新規参加ノードが既にオーバーレイネットワークを構築しているノードの一つに接続する。
2. その接続を用いて、新規参加ノードは直接接続したノードに自ノードの情報を含めたノード広告メッセージをオーバーレイネットワークに送出する。
3. ノード広告メッセージは、さらにその隣接ノードに転送される。
4. ノード広告メッセージを受信したノードは、送出元ノードまでのネットワークの状態を測定し、メッセージに含まれるノード情報と総合し優先度を算出する。この優先度を算出する方法については4.5節で述べる。
5. 既に接続しているノードよりも優先度が高ければそのノードに接続する。接続可能なリンク数の上限を越えた場合は接続中のノードでもっとも優先度が低いノードとのリンクを切断する。
6. 優先度が低ければそのノードには接続せず、ノード広告メッセージはさらに別のノードへ転送される。

図 4.3 としてこの動作を示した。

さらに図 4.4、図 4.5 としてノード広告メッセージの流れに基づくオーバーレイネットワークの動的構成に関する動作を示す。図 4.4 では測定結果から算出した優先度が高かったため発信元ノードに接続し、図 4.5 では優先度が低かったためノード広告メッセージを他のノードに転送している。

また図 4.6 にノードの状態遷移図を示した。

4.3 ノード間リンク

4.3.1 ノード間リンクの確立

接続先が本システムであるかを相互に確認しリンクを確立するため、TCP の接続が確立した直後にハンドシェイクと呼ばれる動作を行う。図 4.7 にハンドシェイクの動作を示す。

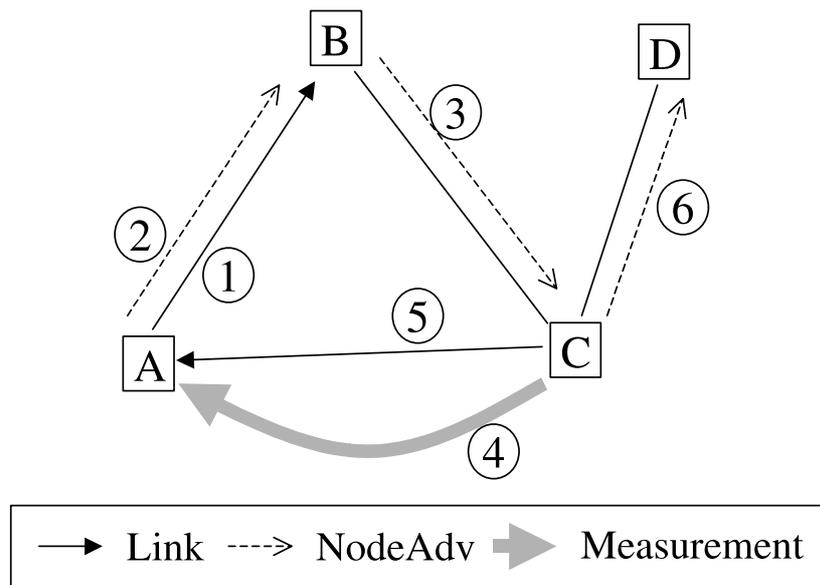


図 4.3: 動作概要

1. TCP コネクションの確立

接続先ノードに TCP を用いて接続する。以後この接続を TCP コネクションと呼ぶ。

2. プロトコル識別子の送信

接続先ノードは TCP コネクションが成立した直後にプロトコルを識別する文字列を接続元ノードに送信する。接続元ノードはそのプロトコル識別子を確認し以後の手順を続ける。プロトコル識別子を受信できない場合はその時点でハンドシェイクを中断し、TCP コネクションを切断する。

本システムではプロトコル識別子として”Prits/1.0” とそれに続く改行文字を用

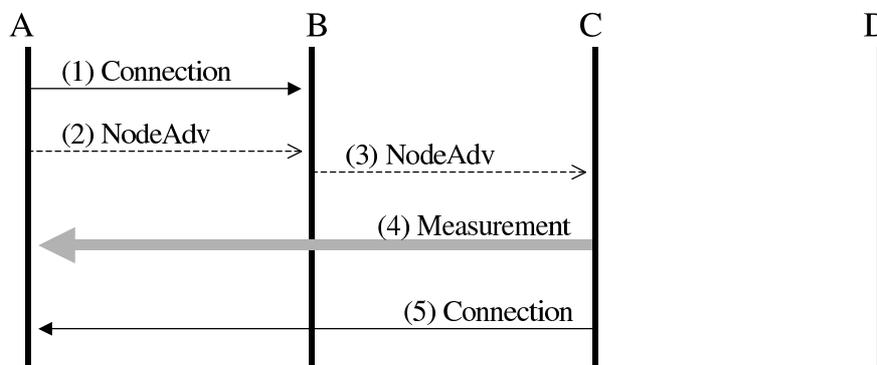


図 4.4: ノード広告メッセージの動作 (優先度が高い場合)

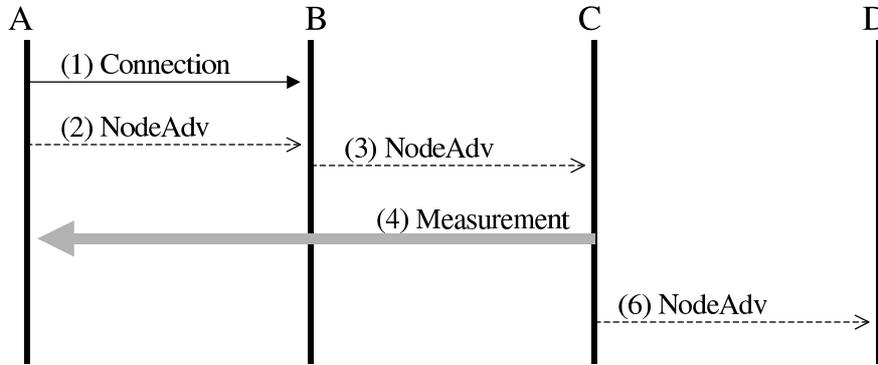


図 4.5: ノード広告メッセージの動作 (優先度が低い場合)

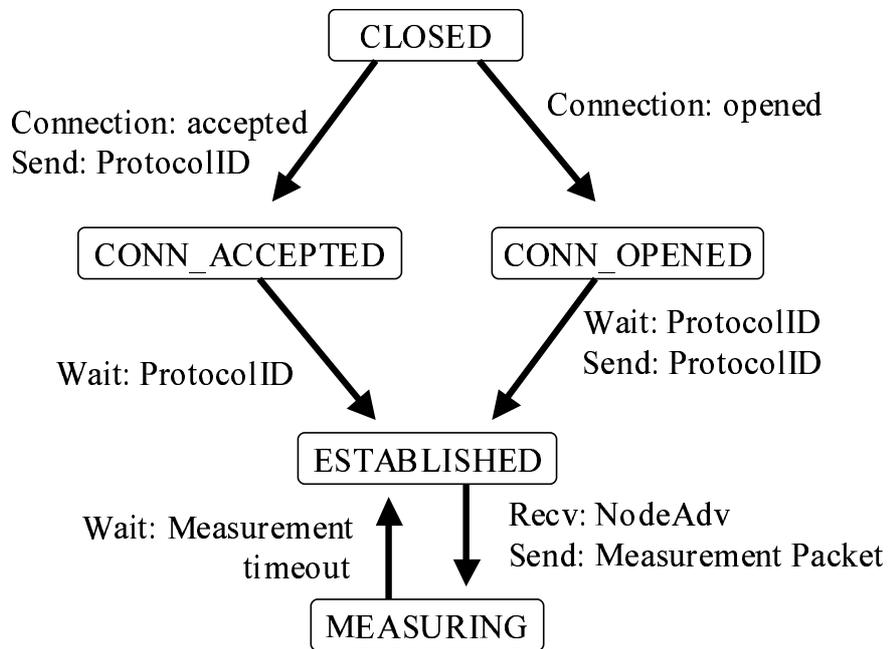


図 4.6: ノードの状態遷移図

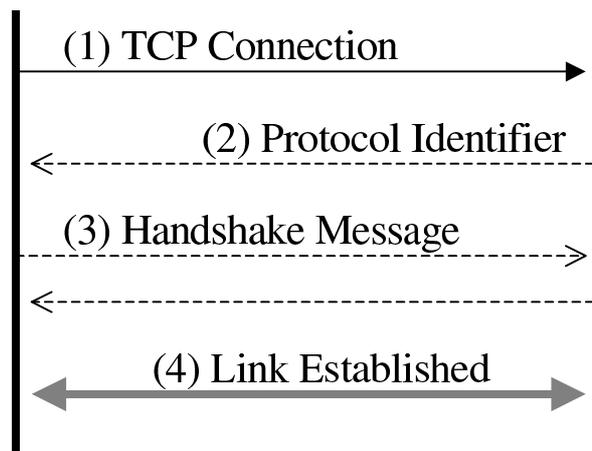


図 4.7: ハンドシェーク

いる。

3. ハンドシェークメッセージの交換

プロトコルを識別完了後、接続元ノードはそのノード情報を含むノード広告メッセージを接続先ノードに送信する。

この時点でリンクが成立したものとみなす。

4.3.2 ノード間メッセージの交換手順

メッセージはリンク上を連続して送信され、各メッセージに含まれるメッセージ長を単位として処理される。

4.3.3 ノード間リンクの切断

リンクの切断は、TCP の接続を切ることで行う。

4.4 ノード情報の測定

ホップ数の測定は以下の手順で行う。

1. IP パケットの TTL を 1 に設定し、その TTL を内容とした測定パケットを UDP[14] で測定先ノードに送信する。これを TTL の測定上限まで一つずつ TTL を増加し繰り返す。また、測定開始の時刻を記録する。
2. 測定パケットを受信した測定先ノードは、受信した内容をそのまま測定元ノードに送り返す。

3. 測定元ノードは戻って来た測定パケットの内容のうち最小のものを相手までのホップ数として記憶する。この測定は測定開始から5秒後にタイムアウトし、その時点でのホップ数をノードバッファに保存する。

4.5 リンク優先度の算出

ノード広告に含まれるノード情報と測定で得たネットワーク情報からノードごとの優先度を求める。以下の方針で式を決める。

- ホップ数が低いノードを最優先する。
- ホップ数が同じであれば帯域幅が高いノードを優先する。
- 処理性能がある程度低いノードはホップ数や帯域幅に限らず優先度を下げる。

以上をもとに、優先度は式 4.1 によって算出される。

$$\text{優先度} = A \times \text{ホップ数} + B \times \log(\text{帯域幅}) + C \times \text{処理性能} \quad (4.1)$$

第5章 実装

本章では、本研究で提案するシステムの設計に基づいて実装を行う。

本システムは、さらに上位のアプリケーションを実装するためのライブラリとして実装される。このライブラリを Prits(PROgramming Interface for Tokumei System) と呼称する。

5.1 実装環境

本システムの実装は、FreeBSD[15] 4.7-STABLE(2002/12/16) 上で行った。プログラミング言語には C 言語を用い、コンパイラとして gcc[16] を利用した。本実装は、FreeBSD、NetBSD[17] 上で動作する。

5.2 ノードの構成

ノードの構成を、図 5.1 で示す。ノードは、リンクマネージャ、リンクバッファ、メッセージプロシージャ、ノードバッファ、測定モジュールの 5 個のモジュールによって構成される。

5.2.1 リンクマネージャ

メッセージプロシージャあるいはアプリケーションからの指示に従い他のノードにリンクを張りメッセージを転送する。接続中のリンクを管理し、他ノードから受け取ったメッセージをメッセージプロシージャに渡す。

5.2.2 リンクバッファ

接続中のリンクに関する情報を保持する。ソケットディスクリプタ、リンクの状態、接続先アドレス、リンクの方向を含む。

C 言語の構造体による定義を以下に示す。

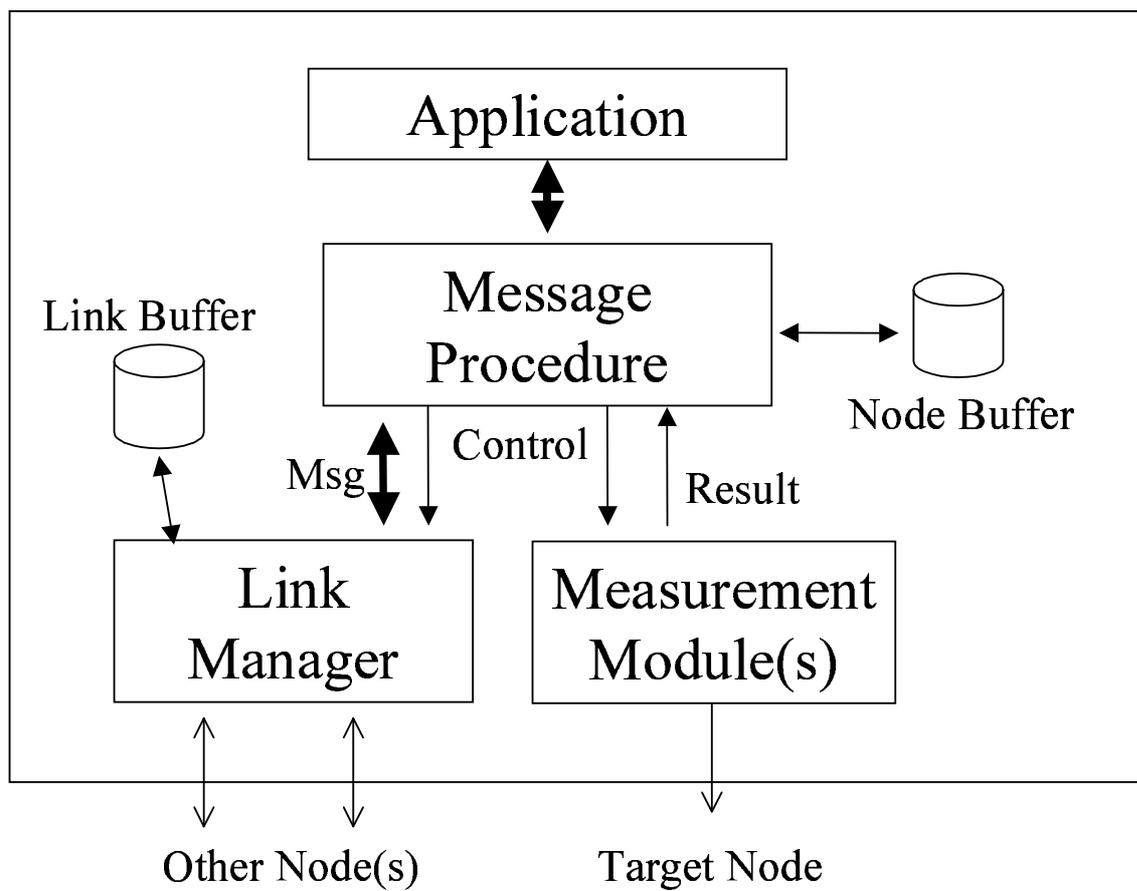


図 5.1: ノードの構成

```

struct link_buffer
{
int socket;
int state;
struct sockaddr_in addr;
int port;
int direction;
};

```

5.2.3 メッセージプロシージャ

他ノードから受信したメッセージをリンクマネージャから受け取り、メッセージの種類に応じてアプリケーションが登録したメッセージハンドラを呼び出す。本システム上ではノード広告メッセージを認識し測定モジュールに測定を依頼したり測定結果をノードバッファに保存する等の処理を行う。

5.2.4 ノードバッファ

ノード広告メッセージに含まれるノード情報と測定モジュールによる測定結果を保存する。

C言語の構造体による定義を以下に示す。

```

struct node_buffer
{
struct sockaddr_in addr;
int16_t priority;
uint16_t      port;
uint8_t      hops;
uint8_t      bandwidth;
uint8_t      load;
time_t      measurement_start;
};

```

5.2.5 測定モジュール

メッセージプロシージャからの要求によりノード間のホップ数の測定を行う。

5.4 API

本システムは上位のアプリケーションに対し、以下のインターフェースを提供する。

5.4.1 初期化

initialize 本システム全体を初期化する。これにより、リンクマネージャを起動し指定されたポート番号で他ノードからの接続を受けられるようになる。

待ち受けを行うポート番号を指定する。

5.4.2 接続

connect 他のノードに接続を行う。

接続先のノードのアドレス、ポート番号を指定する。

disconnect 指定したノードとのリンクを切断する。ただしそのノードと接続してなければ何もしない。

切断するノードのプロトコルとアドレス、ポート番号を指定する。

5.4.3 メッセージハンドラの登録

addhandler 指定した種別のメッセージが届いたときに呼ばれる関数を登録する。

取得したいメッセージ種別とその際に呼ぶ関数を指定する。

5.4.4 メッセージの送信

sendmessage 指定したメッセージを他ノードに転送する。

送信したいメッセージの種別とその内容を指定する。

第6章 評価

6.1 動作実験

6.1.1 実験の概要

4章で述べた設計が正しく機能することを確認するため、5章の実装のプロトタイプを用いて実環境上で実験を行った。

6.1.2 実験環境

実験に参加するノードとして、以下の4ノードを用意した。なおホスト名はドメイン名を省略して表記する。

sh 通信相手のノードを想定し、他のノードと別のIPネットワークに属するノードを選択した。

omoikane IPネットワーク上で帯域幅が狭くかつ離れた場所にあるノードを想定し、128kbpsの専用線で接続されている。

ec 新しく参加するホストにIPネットワーク上で近いホストを想定している。慶應義塾大学村井研究室のネットワークに接続されている。

jam 新しくオーバーレイネットワークに参加するホストを想定している。ecと同様に慶應義塾大学村井研究室のネットワークに接続されている。

6.1.3 実験手順

実験は以下の手順で行われた。

1. あらかじめ sh と ec 間および ec と omoikane 間を接続する。
2. jam から omoikane に対して手動で接続する。
3. 最終的なオーバーレイネットワークの構成を確認する。

6.1.4 実験結果

jam から omoikane に接続後、ec から jam に対しリンクが確立された。これにより接続直後の状態比ベオーバーレイネットワークの構成が最適となったことが確認された。図 6.1 にこの動作を示した。

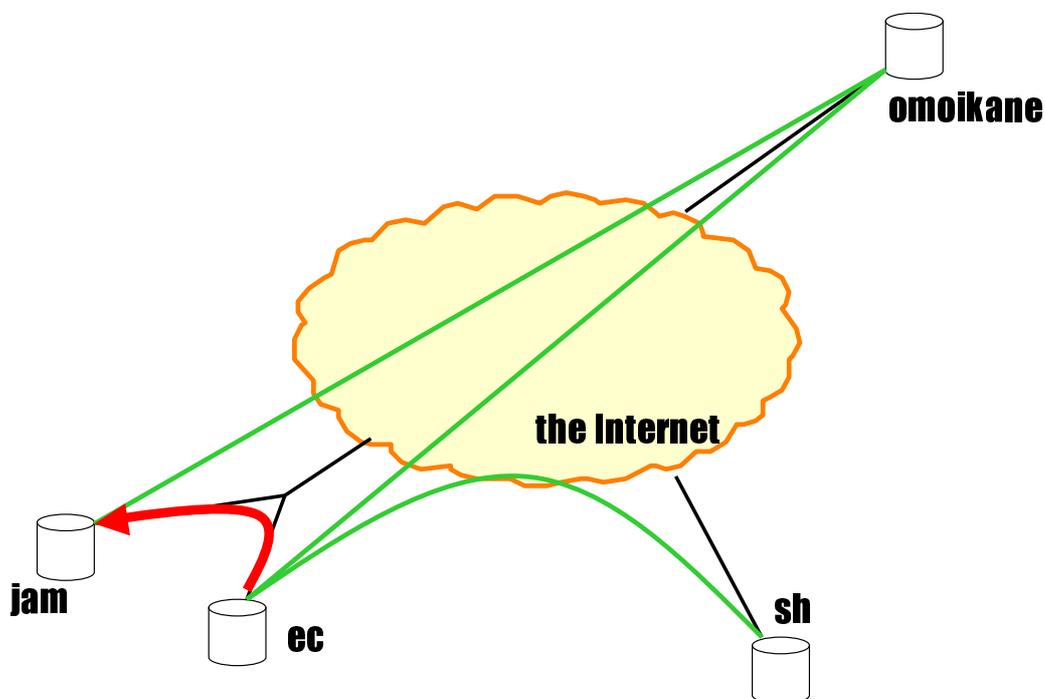


図 6.1: 動作実験

6.2 定性評価

本節では、これまで述べて来た本システムについて 2 章で取りあげたほかの提案との定性評価について述べる。

比較は、以下の点について行った。

転送方法 匿名通信ができるかどうか。

ノードの状態 ノードの状態をオーバーレイネットワークの構成に反映できているか。

ネットワークの状態 ネットワークの状態をオーバーレイネットワークの構成に反映できているか。

表 6.1 に比較結果を示す。

匿名通信の可否という点においては、中継転送によって匿名性を提供できる Winny、P2P ストリーミングおよび本システムを ○ とし、それ以外を × とした。

ノードの状態という点では、Ultrappeer による二段階の階層化 (○) を行う Gnutella と帯域幅による階層化 (○) を行う Winny に比べ本システムでは帯域幅に加えノードの処理性能という新しい指標を加えた 2 点を利用している (○)。これにより、暗号化処理が必要な匿名通信路においてパフォーマンスが向上する。

ネットワークの情報という点では、ホップ数を用い IP ネットワークのトポロジを利用してオーバーレイネットワークの最適化を行っている。Gnutella におけるネットワーク測定による接続先ノード選択の提案においても同様にホップ数を利用しているが、本システムでは新しくノード広告メッセージを導入することで最適化に必要な通信量を抑えている。

以上の比較から、本論文で提案する手法では本システムは他のシステムに比べてオーバーレイネットワークのパフォーマンスを向上することができ、有効であるといえる。

表 6.1: 本システムと既存の提案の比較

	匿名通信	ノードの状態	ネットワークの状態
Gnutella Protocol 0.6	×		×
ネットワーク測定による 接続先ノード選択	×		
Winny			×
P2P ストリーミング		×	×
本システム			

第7章 おわりに

7.1 本論文のまとめ

本論文では、ノードの持つ帯域幅や処理性能をノード交換メッセージによって交換し測定で求めたノード間のホップ数を加えた3つのパラメータに基づいて構成を動的に変えることでオーバレイネットワークの最適化を行うシステムを提案した。

現在のオーバレイネットワークを用いたサーバに依存しない通信モデルの必要性とその上で匿名通信路を実現する手法について述べ、それを実現する上での問題点について論じた。

既存のオーバレイネットワークを利用したアプリケーションや技術を取り上げ、匿名通信路を実現するためのオーバレイネットワークを設計するうえでの問題点について考察した。

本論文では現在のオーバレイネットワークの問題点を解決するため、オーバレイネットワーク上で交換したノードの情報と測定によって求めたネットワークの情報の両方を用いて動的にオーバレイネットワークの構成を変化させる手法を提案した。さらにこの手法を用いたシステムを設計・実装し、本論文の提案が他の手法と比較して優れていることを示した。

7.2 今後の課題

本節では、今後の課題と展望について述べる。

7.2.1 測定するパラメータ

本論文ではパラメータとしてホップ数を測定しているが、より効果的な測定手法を摸索する。また、パラメータや優先度の算出式を調整することで匿名通信路以外の性質を持つオーバレイネットワークの最適化に応用することもできる。

7.2.2 メッセージの経路制御

本論文では対象外としたが、ノード広告メッセージをオーバレイネットワーク上で転送する際の経路制御方法を工夫することで、最適化のパフォーマンスを向上できる。

7.2.3 オーバレイネットワークの最適化の指標

現在はオーバレイネットワークをどれくらい最適化できたかをあらわす指標が存在しない。本研究の効果を評価するためにもトポロジの類似度や転送のスループット等のオーバレイネットワークの効率に対する指標を定めることが考えられる。

謝辞

本研究を進めるにあたり、御指導を頂きました、慶應義塾大学環境情報学部教授の村井純博士、徳田英幸博士、同学部助教授の楠本博之博士、中村修博士、同大学環境情報学部専任講師の南政樹氏、重近範行氏に感謝致します。

絶えずご指導とご助言を頂きました慶應義塾大学大学院理工学研究科後期博士課程の江木啓訓氏、慶應義塾大学大学院政策・メディア研究科後期博士課程の西田視磨氏には執筆開始が大変遅くなってしまったにもかかわらず見捨てずにご指導頂くことができ、感謝の念が絶えません。慶應義塾大学大学院政策・メディア研究科の西村祐貴氏、須子善彦氏をはじめとする neco kg の皆様にはさまざまな形でのご支援を頂きました。慶應義塾大学環境情報学部工藤紀篤氏をはじめとする後輩諸氏には、公私両面にわたり惜しみない協力を頂きました。ここに深く感謝いたします。またこれまで4年間の私の本研究室での活動にあたり書ききれないほどのご指導を頂くとともに、本研究に際し忙しい中深い暗号学に関する知識と見聞から、多大なる貢献を頂いた国際大学 Center for Global Communications の石橋啓一郎氏に心から感謝申し上げます。

IRC チャンネルなどを通じて精神的な面でご支援を頂きました砂田和也氏、中村勇樹氏をはじめとする友人諸氏に感謝します。快適な執筆環境を提供して頂いたスターバックス秋葉原 CLICK&BRICK 店ならびにカフェ・ド・モア東京国際フォーラム店に感謝いたします。さらに本システムの名前の由来となりました声優グループ Prits の皆様ならびにラジオトークで疲れた心に笑いを頂いた福圓美里氏、その歌声で執筆を後押しして頂いたメロキュアの皆様に感謝いたします。

最後に、研究活動を美味しい夕食と快適な布団によって支えて頂いた両親に対する感謝をもって謝辞を締めさせていただきます。

参考文献

- [1] J. Postel. RFC791 Internet Protocol, September 1981.
- [2] Clip2. The gnutella protocol specification v0.4.
<http://www.clip2.com/gnutellaprotocol04.pdf>.
- [3] Inc. Bitmedia Inc. & ANCL. シェアキャスト
<http://www.scast.tv/>.
- [4] peercast.org. peercast.org
<http://www.peercast.org/>.
- [5] Freenetproject.org
<http://www.freenetproject.org>.
- [6] 47@Download.2ch. Winny web site
<http://www.geocities.co.jp/siliconvalley/2949/>.
- [7] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. RFC2616 Hypertext Transfer Protocol – HTTP/1.1, June 1999.
- [8] Anurag Singla, Christopher Rohrs, and Lime Wire LLC. Ultrapeers: Another step towards gnutella scalability working draft
<http://rfc-gnutella.sourceforge.net/proposals/ultrapeer/ultrapeers.htm>, 2001.
- [9] Yoshihiro Gotou, Shingo Ata, and Masayuki Murata. Methods on logical network construction in peer-to-peer services based on tra.c measurements.
- [10] J. Case and M. Fedor and M. Schoffstall and J. Davin. RFC1157 A Simple Network Management Protocol (SNMP), May 1990.
- [11] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, R. Wheeler. RFC2516 A Method for Transmitting PPP Over Ethernet (PPPoE), February 1999.
- [12] Lbnl's network research group
<http://www-nrg.ee.lbl.gov/>.

- [13] S. Deering and R. Hinden. RFC2360 Internet Protocol, Version6 (IPv6) Specification, December 1998.
- [14] J. Postel. RFC768 User Datagram Protocol. *<http://www.ietf.org/rfc/rfc0768.txt>*, pp. 1–3, August 1980.
- [15] The FreeBSD Project. The freebsd project
<http://www.freebsd.org/>.
- [16] The GCC team. Gcc home page - gnu project - free software foundation (fsf)
<http://gcc.gnu.org/>.
- [17] Inc The NetBSD Foundation. The netbsd project
<http://www.netbsd.org/>.