

2001 年度（平成 13 年度）秋学期

修士論文要旨

適応的放送型映像配信機構に関する研究

ADSL の普及によって広帯域インターネット利用者が増大し、数多くの人々が様々な情報をインターネットを用いて発信したり取得したりするようになった。今日、以前と比べて大量のテキスト情報、画像情報の流通が当たり前のように行われている。その結果、例えば World Wide Web(WWW) サービスでは、人気のあるコンテンツを提供している WWW サーバにクライアントのアクセスが集中するようになった。

アクセスが集中するコンテンツを一つのサーバで提供した場合、サーバとネットワークに非常に大きな負荷がかかることから、複数のキャッシュサーバをネットワーク上に分散させ、処理の集中を防ぐ負荷分散技術が利用されるようになった。この手法によって、サーバの処理能力を分散できるだけでなく、一箇所のサーバに対するトラフィックの集中も軽減できる。この手法の一つが Content Distribution Network(CDN) である。CDN は、オリジナルコンテンツのキャッシュをキャッシュサーバに配置する「リバースプロキシ」と、クライアントを適切なキャッシュサーバに誘導する「リクエストナビゲーション」の二つの技術を用いた負荷分散手法である。もともと CDN は、HTML ファイルからリンクされていた画像やテキストファイルなどを対象にしていたが、最近では動画像や音声を含んだ映像配信の分野に利用され始めている。

CDN を利用した負荷分散を行う際に最も重要なのは、クライアントを誘導すべきキャッシュサーバの選択方法である。現在、Akamai などの CDN のサービスを提供している企業の多くは、受信者の IP アドレスを参照し、Border Gateway Protocol バージョン 4(BGP4) によって交換される AS_Path 情報やキャッシュサーバの負荷情報を元に、クライアントが要求した URL に含まれるドメイン名 (FQDN) に対応する DNS の A レコードを書き換えることで、キャッシュサーバの選択を行なっている。しかし、特に映像配信で CDN を利用する場合、HTML などの静的なコンテンツとは異なり継続的にデータ転送が行われることから、通信経路の転送性能を考慮する必要がある。

そこで本研究では、映像・音声といった連続メディア配信を主眼におき、利用者が連続メディアにアクセスする際に行われる Hyper Text Transfer Protocol(HTTP) 転送を CDN 内の複数のキャッシュサーバとクライアント間で行った上で、それに要した転送時間を用いて最適なキャッシュサーバの選択を行う。これにより AS_Path やキャッシュサーバの負荷情報と言った抽象度の高い情報ではなく、ネットワークの状況に合ったキャッシュサーバを選択できるようになる。

また、これまで映像・音声配信において、受信者が直感的に選択していたアプリケーションとメディアタイプの組合せ (例 RealMedia 300Kbps 用等) を、データ転送の量と時間から得た帯域情報を指標に用いることで明確化する。これにより、受信者の直観というあいまいな指標ではなく、計測に裏付けられた具体的な指標を用いて、クライアントアプリケーションやメディアタイプが選択できるようになる。

キーワード 1. サーバ選択、2. キャッシュサーバ、3. インターネット、4. スループット、5. 映像配信

慶應義塾大学大学院政策・メディア研究科
廣石透

Academic Year 2001
Abstract of Master Thesis

Adaptive digital video broadcast system on the Internet, using throughput information between a client and a server

With the rapidly spreading use of ADSL technology, the number of high bandwidth Internet users has increased dramatically and many people now send out and receive information through the Internet. Large quantities of text and image information pass to and from in a manner never before seen. One result of such situation, is a concentration of client access to World Wide Web(WWW) services providing popular contents.

When offering popular contents through a single server, tremendous pressure is put on the server and network. To remedy this, it has become common practice to use load distribution technology by placing multiple cache servers on the Internet. By doing so, not only is the server's workload distributed but the concentration of network traffic in one spot is also lessened. Content Distribution Networks(CDN) is one such technique. CDN uses 2 techniques to distribute workload, these are "reverse proxy" which places the original contents in the cache servers and "request navigation" which guides the client to the optimum cache server. CDN was originally used for text and image links from HTML files, but recently it is being used more and more for video stream distribution which includes moving images and sound.

The most important aspect of CDN's load distribution is the manner of choosing the correct cache server to guide the client to. Currently with CDN server providers such as Akamai, cache server selection is done by rewriting the A records which corresponds to the domain name (FQDN) requested by the client, based on the receiver's IP address, AS_Path information exchanged by BGP4 and cache server load information. However, especially in situations where CDN is used for video stream distribution, because of the continuous data flow unlike that of static HTML, the data transfer capabilities of the network path must be kept in mind.

In this research, keeping streamed media distribution such as video and sound in mind, selection of the CDN cache server is done based on the results of time taken for HTTP transaction between the client and each of the CDN cache servers. In this way, cache server selection can be made based on actual network conditions instead of information with high levels of abstraction such as AS_Path and cache server workload.

Furthermore, the pairing of application and media type (such as RealMedia, 300Kbps stream) for video and sound distribution which was done on the user's intuition, can now be made based on concrete bandwidth information expressed by the amount of data transferred versus the time it took. In this manner, client application and media type pairings can be made based on actual measurements.

Key Words 1.Server selection, 2.Cache server, 3.Internet, 4.Throughput, 5.Broadcasting

Keio University Graduate School of Media and Governance
Toru Hiroishi

慶應義塾大学 大学院 政策・メディア研究科
2001年度（平成13年度）秋学期 修士論文

適応的放送型映像配信機構に関する研究

学籍番号 80032146

廣石透

主査

村井純

平成14年2月25日

目次

第 1 章	序論	1
1.1	背景	1
1.2	研究概要	2
1.3	本研究の構成	2
第 2 章	映像配信機構の問題点	3
2.1	映像配信をする際の問題点	3
2.2	インターネットにおけるトラフィック負荷分散	4
2.2.1	トラフィック負荷分散の重要性	4
2.2.2	様々なレイヤーでのトラフィック負荷分散	4
2.2.3	Content Distribution Network	5
2.2.4	CDN におけるリクエストナビゲーション	7
2.2.5	リクエストナビゲーションにおける問題点の整理	8
2.3	再生アプリケーション、メディアタイプの選択	12
2.3.1	インターネットの特徴	12
2.3.2	動的コンテンツを配送する際の問題点	12
第 3 章	本研究のアプローチ	13
3.1	映像配信の位置付け	13
3.1.1	コンテンツ配信の仕組み	13
3.1.2	HTML の構造	13
3.1.3	CDN の利用	14
3.2	データ転送時間を利用したリクエストナビゲーション機構	15
3.2.1	目的	15
3.2.2	システム概要	15
3.2.3	転送性能の測定	16
3.2.4	利用可能帯域の時間変化	17
3.3	データ転送時間を利用した再生アプリケーション、メディアタイプの選択	18
3.3.1	目的	18
3.3.2	手法	18
3.3.3	様々なアプリケーション	19
第 4 章	本研究の設計	21
4.1	システム全体像	21
4.1.1	データ転送時間を指標として用いたサーバ選択の先行研究との比較	23
4.2	転送性能の優位を判別	24

4.2.1	キャッシュサーバの画像へのリンク	24
4.2.2	データ転送時間の計測方法	25
4.2.3	ダウンロード時間データベース	26
4.2.4	優位の判別方法	26
4.3	リクエストナビゲーション機構	27
4.4	メタファイルの生成	27
4.5	映像品質	28
4.5.1	データ転送時間を利用したアプリケーション選択機構の設計	28
4.5.2	前提条件	28
4.5.3	システム概要	28
4.5.4	アプリケーションのリストを取得	30
4.5.5	メディアタイプの選択	31
4.5.6	メタファイルの生成	32
第 5 章	評価	33
5.1	定性評価	33
5.1.1	データ転送時間を利用したリクエストナビゲーション	33
5.2	各機能の動作証明	34
5.2.1	実験環境	34
5.2.2	実験概要	34
5.2.3	ハードウェア環境	35
5.2.4	ソフトウェア環境	35
5.2.5	アプリケーションを選択	35
5.2.6	データ転送時間の計測	36
5.2.7	クライアントに応じたキャッシュサーバ選択	37
5.2.8	帯域情報を元に適切なメディアタイプの選択	38
5.3	サーバが用意すべきメディアタイプの提案	39
第 6 章	結論	42
6.1	本研究の成果	42
6.2	今後の課題	42
6.2.1	統計的手法を用いたトラフィック変動の予測	43
6.2.2	帯域情報をサブネットレベルで集約	43
6.2.3	他の指標と協調	43
	謝辞	44
	参考文献	45
付録 A	DSL 普及率	48
付録 B	インターネットバックボーンにおけるトラフィックの傾向	49

付録 C 予備実験	51
C.1 実験概要	51
C.2 実験結果	51
C.3 考察	55

目次

2.1	局所的なサーバ選択	5
2.2	ユニキャスト通信におけるトラフィックの重複	6
2.3	キャッシュサーバを利用した負荷分散モデル	7
2.4	AS の相互接続	11
3.1	他のサーバにある画像へのリンク	14
3.2	システム概要図	16
3.3	メディアタイプ選択	19
3.4	アプリケーションが使用する帯域の範囲	20
4.1	システム概要図	22
4.2	画像ファイルの埋め込み(例)	24
4.3	キャッシュサーバから画像をダウンロード	25
4.4	帯域情報の計算式	26
4.5	メタファイルによる誘導	27
4.6	メタファイルの記述例 1(Real Media)	28
4.7	メタファイルの記述例 2(Windows Media Technology)	28
4.8	システム概要図	29
4.9	Javascript によるプラグイン検出	31
5.1	実験環境	34
5.2	ダウンロードした HTML の表示画面	36
5.3	アクセスログ (spring)	37
5.4	アクセスログ (hiro)	37
5.5	各キャッシュサーバからの帯域情報 (turtle)	37
5.6	Real Server へのアクセス (hiro)	38
5.7	www.sfc.wide.ad.jp にアクセスした全てのクライアントから計測された帯域情報の累積度数分布と度数分布	39
5.8	www.sfc.wide.ad.jp にアクセスした中の約 50%のクライアントから計測された帯域情報の累積度数分布と度数分布	40
A.1	DLS 加入者の推移	48

表目次

5.1	リクエストナビゲーションの機能評価	33
5.2	ハードウェア環境	35
5.3	ソフトウェア環境	35
5.4	64Kbps で区切った時のクライアントの度数	41
B.1	Top 25 TCP application categories seen during February, 2000	49
B.2	Top 25 UDP application categories seen during February, 2000	50
C.1	クライアント・サーバ間を 28.8Kbps に設定した際に計測された帯域	51
C.2	クライアント・サーバ間を 64Kbps に設定した際に計測された帯域	52
C.3	クライアント・サーバ間を 128Kbps に設定した際に計測された帯域	52
C.4	クライアント・サーバ間を 300Kbps に設定した際に計測された帯域	53
C.5	クライアント・サーバ間を 500Kbps に設定した際に計測された帯域	53
C.6	クライアント・サーバ間を 1Mbps に設定した際に計測された帯域	54
C.7	クライアント・サーバ間を 10Mbps に設定した際に計測された帯域	54
C.8	クライアント・サーバ間を 100Mbps に設定した際に計測された帯域	55

第1章 序論

1.1 背景

1998年と2000年に発表された調査結果 [1, 2] によれば、インターネット上に流れている最も多いトラフィックは HTTP[3] であり、以下 Real Audio、DNS(Domain Name System)[4] などが並んでいる。このことから、インターネットが主にコンテンツ配信の手段として利用されていることが分かる。近年、この傾向がますます顕著になると共に、利用者の増加や回線帯域の向上によるトラフィックパターンの変化と合わせて、いくつかの問題が起きている。その最も一般的な例が、サーバへのアクセスの集中によるサービスの一時的な利用不能やトラフィックの輻輳である。最近では、テキストデータや画像データのような静的なデータだけではなく、映像・音声のような動的コンテンツが数多く用いられるようになっている。これらの動的なデータは、静的なデータに比べて情報量が非常に多いため、他のトラフィックへの影響する可能性が高い。このことから、インターネット上で安定したコンテンツ配信を実現するためには、サーバへのアクセス集中やトラフィック集中を避ける手法が必要とされてきた。

この問題に対し、Content Delivery System(CDS) あるいは Content Distribution Network(CDN) と呼ばれる手法が用いられている。この手法は、サーバの負荷分散を行うためにネットワーク上に複数のキャッシュサーバを配置し、オリジナルサーバのコンテンツをキャッシュすると共にクライアントをキャッシュサーバに誘導する、というものである。この手法を効果的に活用するために重要なのは、利用者に最適なキャッシュサーバを選択し誘導する手法である。現在、クライアントが複数のキャッシュサーバから最適な一つを選択する手法の多くは、Round Trip Time(RTT) などの応答時間や BGP4 や OSPF などが交換する経路情報を元にしている。またキャッシュサーバへの誘導は、要求された URL に含まれるドメイン名 (FQDN) に対応する Domain Name System(DNS) の A レコード (IP アドレス) を、キャッシュサーバに書き換えることによって行われている。既に Akamai[5] などの企業がこの技術を利用し、世界各地に配置されたキャッシュサーバと高性能な DNS サーバを組合せた、安定した WWW による情報配信を提供している。

最近では、キャッシュサーバの性能向上と利用者環境の整備によって、動的コンテンツ配信においても CDN を用いるケースが増加している。静的なデータの場合と同様、オリジナルサーバから転送されるデータをキャッシュすると共に、クライアントをキャッシュサーバに誘導する。しかし、テキストデータや画像データなどの静的なデータの断続的なデータ転送とは異なり、動的コンテンツは継続的なデータ転送を行うため、キャッシュサーバを選択する際にはその特性を十分に考慮する必要があると考えられる。また、全ての利用者が同様のチューナを用いる地上波放送や衛星デジタル放送と異なり、インターネット上での映像・音声配信では、コンピュータの性能や回線の種類が一樣ではないため、利用者毎に動的コンテンツを再生する環境が異なると言える。このため、一般的にインターネットを利用した放送型の動的コンテンツ配信では、送信者は利用者の環境に合わせ再生アプリケーション、品質などが異なる数種類の転送データを用意する。ところが、利用者はそれらの複数のメディアを何の指標もなく、ただ直感というあいまいな判断基準で選択している。したがって、利用者の直感ではなく裏付けを持った明示的な指標が必要であると考えられる。

1.2 研究概要

本研究はインターネット上の映像配信機構における以下の問題点の解決を行った。

- 動的コンテンツのコンテンツ配信機構におけるキャッシュサーバ選択指標の提案

キャッシュサーバ選択機構には様々なアプローチがある。DNS ラウンドロビンを用いて複数のアクセスを振り分ける方法、キャッシュサーバとクライアントとの間の Autonomous System(AS)[6] 数を用いる手法、キャッシュサーバの負荷を測定し、低負荷のものを選択する手法などである。ネットワーク上の位置を利用する手法は、単独では十分ではないことが示されている [7]。また、経路情報をもとにサーバを選択する手法は、既存の CDN での end-to-end の応答遅延を測定した評価から、最適なサーバの選択は困難であることも示されている [8]。これらの手法は、クライアント・キャッシュサーバ間の実際の帯域情報について何も考慮していない。本研究では、クライアント・キャッシュサーバ間の帯域情報をキャッシュサーバ選択の指標として用いる手法を提案する。

- 動的コンテンツのコンテンツ配信をする際にアプリケーション、利用帯域別のメディアタイプの選択をユーザの主観に頼っている

一般にインターネット上で映像を観る場合、必ずと言っていいほど複数のアプリケーションとそれに対応した複数のメディアタイプのリンクがある。例えば、Windows Media Player[9] 28Kbps 用、Real Player[10] 200Kbps 用と書かれているリンクである。通常、そのリンクをクリックすることでメタファイルを参照し、サーバにある動的コンテンツを再生する仕組みになっている。その際、動的コンテンツを受信する利用者の主観によって、それらのリンクを選択している。この主観は直観的であいまいであるため、サーバ・クライアント間の実際の帯域情報に即さない場合が多い。本研究では、HTTP によるデータ転送時間を指標とすることで、無駄なトラフィックが流れることを避ける。また、利用者に対してアプリケーション、メディアタイプ選択についての定量的な指標を与えることができる。

1.3 本研究の構成

本研究ではインターネットにおける放送型映像配信機構に対して以下の二つの提案をする

- データ転送時間を利用したキャッシュサーバ選択機構
- データ転送時間を利用したアプリケーション、メディアタイプ選択機構

第2章では、インターネットにおける映像配信をする際の問題点について述べる。特に現在のインターネットにおけるトラフィック負荷分散技術とアプリケーション、メディアタイプ選択について述べる。トラフィック負荷分散については、アプリケーションレベルでの負荷分散機構である Contents Distribution Network を概説し、CDN におけるキャッシュサーバ選択機構（リクエストナビゲーション）の問題点について述べる。

第3章では、本研究で提案するシステムの概要について述べる。

第4章では、本研究で提案する、実帯域を用いたキャッシュサーバ選択について述べる。また、キャッシュサーバ選択で測定した帯域情報を利用することで利用者のアプリケーション、メディアタイプの選択に指標を与えるシステムについて述べる。

第5章では、第4章で述べたシステムを評価する。

第6章では、本研究の成果を総括し、これからの展望について述べる。

第2章 映像配信機構の問題点

2.1 映像配信をする際の問題点

第1章にも述べたように、映像配信のような動的コンテンツ配信はテキスト情報や画像情報を配送するよりも非常に広い帯域を消費する。また、今年もDSL加入者の増加が著しく [11]、ラストワンマイルの回線速度が広帯域化することによって、これからますます動的コンテンツがインターネットバックボーンに流入することは確実である。コンテンツを提供する側にとって、映像をより多くの利用者に再生してもらうためには、バックボーンにおけるトラフィック流量の増加を食い止めなければならない。途中回線の輻輳によって、利用者の再生品質を劣化させてしまわないためである。また、クライアントからのアクセスの集中によってサービスが継続できなくなるようなことがあってはならない。これらの問題を回避するためには、映像配信によるインターネット上に流れるトラフィックの負荷分散を行うことが重要である。

また、コンテンツ利用者にとっては良好な品質で映像を再生したいという要求がある。映像配信では、データ転送量と映像の品質の関係は比例しているため、クライアント・サーバ間の転送性能によって映像の品質が変化する。そのため、コンテンツ提供者はサーバ・クライアント間の転送性能を最大にし、その転送性能に適したアプリケーション、メディアタイプの動的コンテンツを提供する必要がある。また、インターネットにおける映像配信では、地上波のテレビ放送などと違い、利用者のメディア再生環境が異なる。そのため、利用者の環境に適した動的コンテンツを提供する必要がある。

要求を整理すると、以下のようにまとめることができる。

- コンテンツ提供者の視点
 - サービスを安定的に提供したい。
 - より多くの人にコンテンツを再生してもらいたい。
- コンテンツ利用者の視点
 - 持っている再生環境に適した映像を受信したい。
 - なるべく良好な品質で映像を受信したい。

コンテンツ提供者の視点、利用者の視点の双方の要求を満たすためには、大規模な動的コンテンツ配信のトラフィック負荷分散とコンテンツ利用者・提供者間の転送性能に適したアプリケーション、メディアタイプの選択が必要である。

本章では、インターネットにおけるトラフィック負荷分散について概説し、その問題点を探る。また、インターネット利用者のデータ受信環境が異なることから起きるアプリケーション、メディアタイプ選択に関する問題について述べる。

2.2 インターネットにおけるトラフィック負荷分散

2.2.1 トラフィック負荷分散の重要性

電話網に代表される回線交換方式では、通信の両端を結ぶ回線の中継システムが管理することで、信頼性のある通信を保証している。しかし、回線に流れるデータ量は様々であり、単位時間毎に観測した場合、回線帯域を有効に利用できていないことが多い。また、同時に利用できる通信の端点の組の数は中継システムの能力によって限定されるため、中継システムの配置は、資源が競合しないように考慮する必要がある。

一方、インターネットに代表されるパケット交換方式では、回線をパケット単位で共有するため、回線交換網と比べ多数の利用者が同時にネットワークを利用できる。つまり、多くの人々が同時に利用するような広域ネットワークに適していると言える。しかし、パケット交換方式による中継システムはパケットの確実な配送を保証しない。このため、中継システムの処理負荷が大きかったり、中継システムに接続されたネットワーク資源が過度の競合状態にある状況では、パケットが破棄されることがある。このように、パケット交換方式は、回線の占有が行われぬ反面、通信を行う終端ノードからネットワークに流入するデータ量の総量が、ネットワークの許容量を超えることがあるという欠点がある。例えば、オリンピックのようなアクセス頻度の高いコンテンツを持つサイトには世界中からのアクセスが集中し、中継回線には過度の輻輳が起きる。また、2001年9月に起きたアメリカ同時多発テロ事件では、刻一刻と知らされる最新情報を得るために、全世界からのアクセスが短期間に集中し、サービスを継続できなくなった例もある。

今日のインターネット環境において、このような状況は特別なことではない。したがって、アクセスの集中を回避し安定的にデータ転送を行うために、インターネットを介して転送されるトラフィックの負荷分散機構が重要になる。

また、インターネットでは1対多の通信が可能のため、終端サーバにクライアントからのアクセスが集中した場合、終端サーバの処理能力を超えることがある。サービスを安定的に提供するためには、終端サーバへのアクセスの負荷分散が重要である。

2.2.2 様々なレイヤーでのトラフィック負荷分散

トラフィック負荷分散は以下のように、OSI 参照モデル [12] における各階層で負荷分散の手法が提案されている。

- ネットワーク層
 - マルチキャスト
中継ルータによって配送データのコピーが行われ、ユニキャストと比べ、中継ルータに流入するデータ量を削減できる。放送型通信を行う場合に特に有効である。しかし、全てのネットワークがマルチキャストに対応しているわけではないという欠点がある。
- トランスポート層
 - TCP の輻輳制御
中継回線に輻輳が起きた際に終端ノード間で輻輳を回避するメカニズム。
 - L4 スイッチ
複数のサーバを一つのアドレスを持ったサーバとして仮想的に運用する。終端ノードからのアクセスの集中を分散することができる。

- アプリケーション層

- Content Distribution Network(CDN)

ネットワーク地図上に複数のキャッシュサーバを置き、クライアントはネットワーク地図的に距離が近いキャッシュサーバへ接続する。分散して配置したキャッシュサーバがオリジナルサーバが持つコンテンツを配信する。

- その他

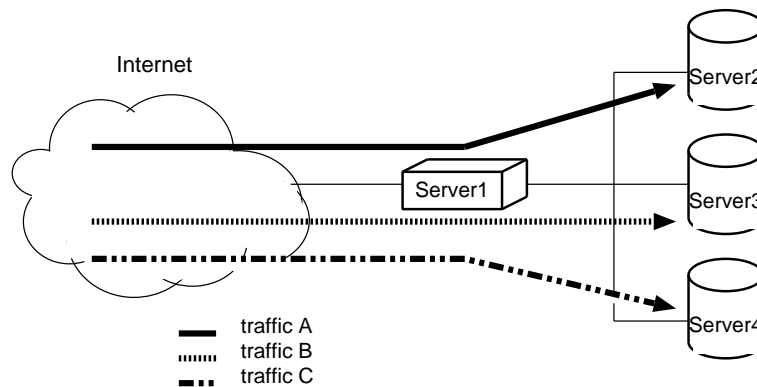


図 2.1: 局所的なサーバ選択

図 2.1 にあるように、クライアントにとっては Server2,3,4 は仮想的に Server1 であるように見え、Server1 はクライアントからのアクセスを Server2,3,4 へ振り分ける。つまり、NAT 技術を用いて局所的に集まったサーバのサーバ選択を行う。CDN にこの技術を用いたキャッシュサーバ群を複数配置することでより効率的な負荷分散を行うことが出来る。ACEdirector,BIG-IP,LSNAT,Equalizer 等が代表的なツールである。

OSI 参照モデルにおける個々の層において、トラフィック、アクセスの負荷分散を行っている。OSI 参照モデルで言うトランスポート層で行うよりも、キャッシュサーバを利用したアプリケーション層で行われるのが最近の主流である。なぜなら、インターネットトラフィックの増加傾向は 1 年で 2 倍になる [13] が、ムーアの法則からホストの性能は 18ヶ月で 2 倍となり、ルータやサーバの性能を向上させるよりも、トラフィック自体を分散させた方がコストが低いからである。よって、本研究ではアプリケーション層によるトラフィック負荷分散に焦点をあて、特に映像配信を行う場合の Content Distribution Network におけるキャッシュサーバ選択機構への提案を行う。

2.2.3 Content Distribution Network

CDN は、インターネット上に大量のキャッシュサーバを世界中に配置することで、コンテンツ配信を行う際のインターネットの途中経路に起こる輻輳を軽減させるシステムである。本論文では、特に同一ポリシーでキャッシュサーバを管理しているネットワークのことを CDN と呼ぶ。

現在のコンテンツ配信において、主に利用されているのは HTTP などのユニキャスト通信である。ユニキャスト通信の場合、同じコンテンツに対し複数クライアントからリクエストがあると、図 2.2 のように、必ず

途中経路のどこかでトラフィックの重複が起きる。図 2.2 にユニキャスト通信におけるトラフィック重複を示す。

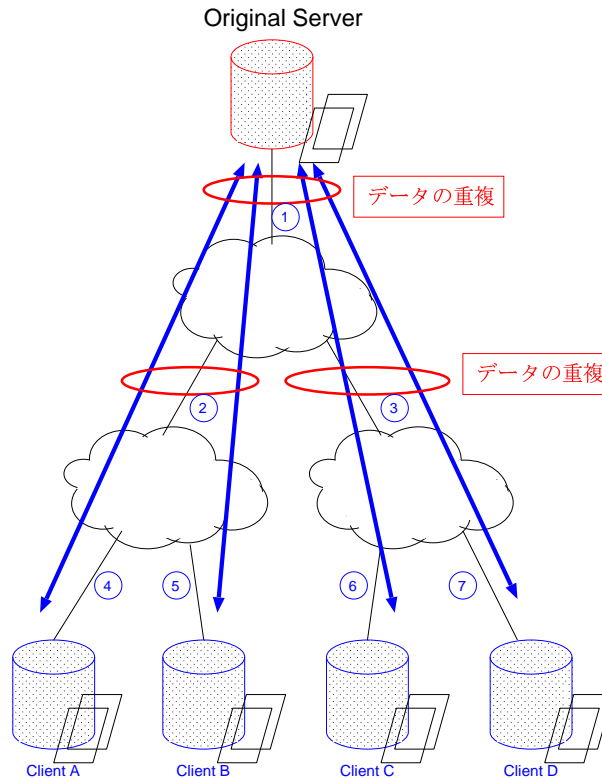


図 2.2: ユニキャスト通信におけるトラフィックの重複

図 2.2 のクライアント A,B,C,D がオリジナルサーバに同時に接続することで、オリジナルサーバの接続している回線 1 に同じトラフィックが 4 回流れてしまう。また中継回線である回線 2,3 においても、同様のデータの重複が起きる。

このようなトラフィックの重複を最小限に留めるために、ネットワーク上に大量のキャッシュサーバを設置する。クライアントはキャッシュサーバに接続することによって、オリジナルコンテンツを持っているサーバ(オリジナルサーバ)と同様のコンテンツにアクセスできる。図 2.3 にあるように、CDN を利用することによって、オリジナルサーバはキャッシュサーバだけにコンテンツを配信すればよく、中継回線のトラフィックの重複を軽減する。

また、トラフィックの重複だけでなく、オリジナルサーバへのクライアントのコネクション数も減少させる。クライアントにとっても、ネットワーク地図的にオリジナルサーバより距離が近いキャッシュサーバを選ぶことで、応答時間を短くすることができる。つまり、コンテンツをダウンロードする時間が短くなる。図 2.3 にキャッシュサーバを利用した負荷分散モデルを示す。

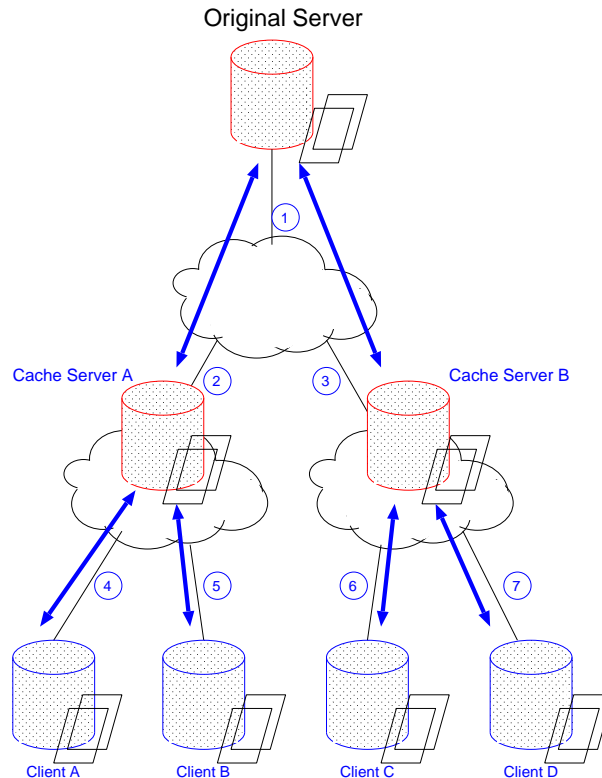


図 2.3: キャッシュサーバを利用した負荷分散モデル

キャッシュサーバを設置することによって回線 1、また回線 2,3 に起きるトラフィックの重複を避ける。このように CDN を利用することで、トラフィック負荷分散を行い、人気のあるコンテンツへのトラフィック集中を回避する。CDN を利用して効果的にトラフィック負荷分散を行うために、クライアントは適切なキャッシュサーバを選択する仕組みが必要である。図 2.3 で言えば、クライアント A,B はキャッシュサーバ A を選択し、クライアント C,D はキャッシュサーバ B を選択するという仕組みである。

2.2.4 CDN におけるリクエストナビゲーション

異なるポリシーを持ったキャッシュ機構の協調も含めて、CDN は以下のような機能で分類ができる。[14, 15]

- キャッシュサーバにおけるコンテンツ同期

図 2.3 におけるキャッシュサーバ A・キャッシュサーバ B 間のコンテンツの同期やオリジナルサーバとのデータ同期についての技術。

- 認証、課金システム [16, 17]

課金を行って配信したいコンテンツがオリジナルサーバにある場合、それをキャッシュサーバを利用して配信した際にも同様の認証を行う仕組み。

- 最適なキャッシュサーバ選択 (リクエストナビゲーション)[18]

大量のキャッシュサーバがネットワーク上にある場合に、クライアントにとって最適なサーバを選択する仕組み。

CDNにおけるこれらの機能の中でも本研究では最適なキャッシュサーバ選択(リクエストナビゲーション)について扱う。

2.2.5 リクエストナビゲーションにおける問題点の整理

参考文献 [19] によるサーバ選択機構への要求に照らし、既存のリクエストナビゲーションについて機能と問題点を考察する。

リクエストナビゲーションへの要求事項

- 規模適応性

サーバ選択機構は規模適応性のあるサーバを構築するために生まれた技術である。そのため、サーバ選択において本来の目的である規模適応性が損なわれるべきではない。

- 利用者透明性

規模適応性のあるサーバ構築の目的の一つは利用者の利便性を増すことである。よって、サーバ選択機構をとりいれることで、利用者の利便性が損なわれるべきではない。そのためには利用者にとってサーバ選択機構が透過的にあるべきである。

- サービス非依存性

構築するサーバの対象とするサービスは様々である。したがって、サーバ選択機構はサービスに対して非依存であるべきである。

- 実装非依存性

サービスは複数の実装があることが一般的である。WWWサーバには apache や Microsoft Internet Information Server などの様々な実装がある。サーバ選択機構の利用にあたって、実装の選択範囲を狭めるべきではない。

- サービス提供者非依存性

サーバ選択を行うのはサービスの提供者のみとは限らない。例えば、ある組織のネットワーク管理者が、その組織のネットワーク構成に応じてサーバ選択を行うこともある。よって、サービス提供者とは独立してサーバ選択機構が利用できた方がよい。

- 選択ポリシーの柔軟性

サーバ選択の判断基準はサービスの内容など様々な要因に依存する。選択ポリシーは画一的なものではなく、サービス提供者あるいは利用者によって異なる。よってサーバ選択機構は様々な選択ポリシーを取り扱うことが可能であり、それらを柔軟に切り替えられることが望ましい。

振り分け方法

リクエストナビゲーションには大きく分けて以下の二つの手法がある。

1. サーバ側が動的にクライアントを誘導する手法
2. クライアントがキャッシュサーバを選択する手法

クライアント側が自ら測定してキャッシュサーバを選択する手法は、サーバ側がもっている AS_PATH 情報やキャッシュサーバの負荷状況等の様々な情報との協調が難しい。また、クライアントはネットワーク上に大量にあるキャッシュサーバの情報を保持している必要があり、利用者透明性に欠ける。サーバ提供者も全ての利用者に対してサーバの増減を知らせなければならず、手間がかかる。そのため、本研究では前者のサーバ側が動的にクライアントを誘導する手法について述べる。

- DNS を利用

オリジナルサーバへアクセスすることが一度もなく、クライアント・キャッシュサーバ間の通信だけでクライアントからのアクセスを処理することが出来る（規模適応性）。また、利用者は DNS を参照するという普段行われている作業しか行わないため、利用者透明性にも優れている。サービス非依存性にも優れている。

しかし、DNS データベースが伝播するまでの時間がかかり、動的なネットワーク状況に対応するには向いていない。また、頻繁にデータを更新したとしても、DNS のアップデートによってネットワークに負荷がかかってしまう。DNS を利用した負荷分散機構には以下の種類がある。

- DNS ラウンドロビン

あるホストに対して複数のアドレスを登録することで、アクセスしてきたクライアントを順番にそのアドレスへ振り分ける。全てのサーバが平均的に選択される。これは選択指標に欠点がある。単純な順番の選択しかできないからである。また、一般的に DNS の登録はサービス提供者が行うのでサービス提供者に依存する。

- 重み付けラウンドロビン

DNS ラウンドロビンに選択肢を導入した仕組みである。サーバ毎に処理能力に応じた数の IP アドレスを割り当て、擬似的にサーバ処理能力を反映できるようにしている。しかし、基本的にはラウンドロビン選択のため、選択ポリシーの柔軟性がない。

- SWEB[20]

DNS ラウンドロビンの改良である。まず、第一にラウンドロビンでサーバ選択を行う。その後、選択されたサーバが他のサーバの状態を判断して、最終的なサーバを選択しリクエストを返す。しかし、HTTP でしか利用できないのでサービス非依存性に欠ける。

- Cluster DNS[21]

DNS サーバにサーバ選択機構を組み込んだものである。しかし、選択ポリシーの柔軟性に欠け、サービス提供者と独立にサーバ選択を行う場合を考慮していない。

- Smart Clients[22]

専用のクライアントを用いて、クライアント側でサーバ選択を行う。Java applet を用いてサーバ情報をクライアントに伝える。この方式はサーバ・クライアント側の双方の実装を変更する必要があり、実装非依存性にかける。

– 単一 IP アドレスアーキテクチャ[23]

この手法は長野オリンピックで使用された WWW サーバの負荷分散を行うために利用された。複数化したサーバ群に同一 IP アドレスを割り当て、ネットワークの経路制御によるサーバ選択を行う。インターネット上のすべてのパケットは経路制御機構の影響を受けるので適用可能範囲は広い。選択ポリシーと経路制御のためのポリシーを分離することができないため、選択ポリシーの柔軟性に欠ける。

– DistributedDirector[24]

商用サービスでは広く使われている。本方式を用いるには周辺ルータにも同じシステムを組み込む必要がある。

– 3DNS

商用サービスでは使われている。キャッシュサーバの送出している Packet Rate, RTT, Completion Rate, Topology を任意に組み合わせ、その算出値に基づいてロードバランスする。

– Akamai[5]

商用のコンテンツ配信システムである。選択方法は DNS を用いている。選択ポリシーに関しては Akamai から提供されたものを使う必要があり、柔軟性に欠ける。

– TENBIN[19]

選択方法に DNS を用い、選択手法を柔軟に切り替えることができる。また、DNS データベースに登録されている IP アドレスとは異なる IP アドレスを返すことができる。そうすることで、サービス提供者非依存性を解決している。

当該システムのプロトタイプ実装では、TENBIN 側からキャッシュサーバまでのラウンドトリップタイムを定期的に計測している。そして、クライアントが TENBIN へリクエストを出すことで、キャッシュサーバのリストを取得し、ラウンドトリップタイムを利用して選択を行う。しかし、この実装では、TENBIN とクライアントがネットワーク上の近い距離にないと効果がない。

● メタファイルの書き換え

– メタファイルにキャッシュサーバのアドレスを記述

オリジナルサーバに一度接続し、その後はキャッシュサーバとの通信を行う。これは必ずオリジナルサーバを参照することから動的な転送性能の変化にも対応したサーバ振り分けに適している。しかし、映像配信というサービスのみに限られたクライアント振り分け方法であるため、サービス非依存性には優れていない。

選択の指標

クライアントにとって適切なサーバを選択するための指標は以下のものが考えられる。

● ラウンドロビン (DNS を利用)

DNS へリクエストが来た順序毎にキャッシュサーバを振り分ける。

- クライアント・キャッシュサーバ間のラウンドトリップタイムが最短

ICMP ECHO パケットを利用し、ICMP ECHO REQUEST パケットを送信した後、戻ってきた ICMP ECHO REPLY パケットからラウンドトリップタイムを計測する。

- ネットワーク地図上の距離

- AS_PATH 情報

BGP4[25] の保持する AS_PATH 情報を利用し、キャッシュサーバの設置されている AS からクライアントの AS がどの程度離れているかを計測する。

備考：Autonomous System

図 2.4 に AS 間の相互接続の概念図を示す。

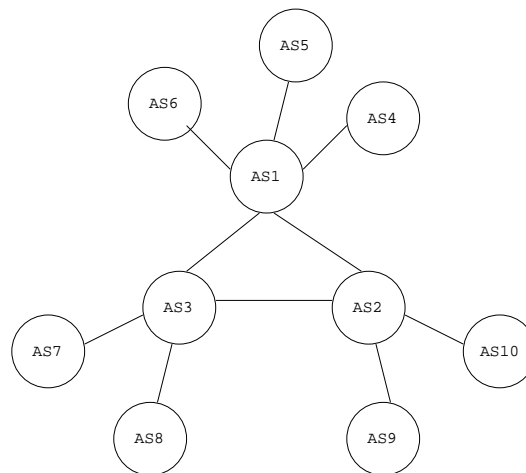


図 2.4: AS の相互接続

インターネットは多くの AS の相互接続によって成り立っている。そして、BGP4 プログラムによって、AS 間の経路制御が行われている。その BGP4 プログラム間で交換される経路情報が AS_PATH と呼ばれるものである。AS_PATH 情報には経由した AS ナンバーが順に記されており、AS_PATH 情報を参照することにより宛先までの経由 AS 数を知ることができる。

- クライアント・サーバ間の経路が最短

traceroute[26] プログラムを使用し、途中ルータの経由数を取得する。

- キャッシュサーバの負荷状況

コネクション保持数、送出パケット量、CPU 使用率

いずれの手法でも問題となるのは、利用者とサーバ間の性能を示す指標の妥当性と、得られた情報から最適なサーバを選択する方法である。経由 AS 数は単独では十分ではないことが示されている [27]。それは、同一 AS 内に複数のキャッシュサーバがあった場合、それらを選択するための指標がないからである。経路情

報をもとにサーバを選択する手法についても、クライアント・キャッシュサーバ間の応答遅延を測定した評価から最適なサーバを確実に選択することは困難である [8]。これらのことからトポロジー情報だけでは十分ではないことがわかる。

2.3 再生アプリケーション、メディアタイプの選択

2.3.1 インターネットの特徴

地上波放送や衛星デジタル放送などの既存の映像・音声配信システムでは、映像・音声の利用者は単一チューナによってデータを受信し、配信者は電波を用いて全ての利用者へ同じデータ帯域を経由してデータを転送する。そのため、利用者は電波の届く限り、同じ質の映像・音声を再生することが出来る。

一方、インターネット上での映像・音声配信では、インターネットのインターオペラビリティの高さから、利用者は異なる受信機器、再生アプリケーションを持ち、様々な種類の回線を利用して映像・音声配信される。そのため、放送型の映像・音声配信を行う場合、多くの利用者が再生可能になるために、送信者は様々な符号化手法を用いてデータを圧縮し、ファイル形式の異なった複数のファイルを提供しなければならない。それらは特に、利用者の持つアプリケーションの種類、送信者・利用者間のデータ転送性能という二つを考慮して作成される。送信者が複数の環境用にメディアを用意できない場合は、メディアを再生する利用者を限定してしまい、放送型配信をすることは難しい。

2.3.2 動的コンテンツを配信する際の問題点

映像や音声データをインターネット上に転送する場合、テキストや画像と比較して非常に広い帯域幅を必要とする。そして、狭い帯域でもより綺麗な音声、映像を楽しむ目的で、各企業によって様々な圧縮手法、配信手法が開発された。企業独自の手法が乱立したため、現在では、動的コンテンツを再生するためにそれぞれの手法に対応した複数のアプリケーションを利用者が持っている必要がある。

また、クライアント・サーバ間の帯域が個々のクライアントによって異なるため、映像を配信する際に個々のクライアント・サーバ間の帯域に適したメディアタイプ（例えば、200Kbps 用等）を作成し、それらを利用者の直感というあいまいな指標で選択している。利用者の直感が外れた時、クライアント・サーバ間の帯域よりも余計なトラフィックが流れる可能性がある。また、利用者が動的コンテンツを再生できない可能性がある。この問題に対し、アプリケーション独自の解決方法として、Real Networks Ltd. の提案する Sure Stream[28] や Microsoft Corporation が提案する Intelligent streaming[29] はパケットロス率を検知しながら帯域情報を取得し、ある一定の帯域幅の範囲に対応した連続メディアの配信レートコントロールが行われている。しかし、異なる種類のアプリケーションも含めたインターネット網全体としてのアプリケーション選択とメディアタイプ選択が出来ないという欠点がある。

アプリケーション選択の関連研究

Generic Media[30] の提案する手法では、動的コンテンツを初めてダウンロードするクライアントに、帯域情報、再生したいアプリケーションの種類を入力させる。その後、そのクライアントが何度アクセスしても設定したアプリケーションで自動的に再生が可能になる。この方法でも、サーバ・クライアント間の実際の帯域は動的に変化するので、クライアントが再生する際の帯域情報を正確に反映できないという欠点がある。また、クライアントに手動で入力させるという部分でも既存の選択方式とあまり変わらない。

第3章 本研究のアプローチ

3.1 映像配信の位置付け

3.1.1 コンテンツ配信の仕組み

インターネットを利用したコンテンツ配信は主に WWW ページを入り口にして行われている。つまり、コンテンツ提供者は WWW サーバに情報を設置することで、コンテンツ利用者に情報を提供している。利用者は自分のマシンにインストールされている WWW ブラウザを使ってそれらのコンテンツを閲覧する。しばしば、提供されるコンテンツには、HTML 言語で書かれたテキストファイルでなく、様々なアプリケーションによって作成されたコンテンツが含まれている。このような場合、独自のアプリケーションで作成されたコンテンツを利用者が再生するため、ブラウザはプラグインという仕組みを提供している。これは、そのコンテンツを再生するアプリケーションを対応づけると同時に、コンテンツファイルの拡張子と再生するアプリケーションのリストをブラウザ内に組み込む。ブラウザは、次回以降そのリストを参照することで自動的に専用のアプリケーションを選択し、利用者は無意識にコンテンツを閲覧できるようになる。また、この機能を利用して、コンテンツを閲覧するために必要なアプリケーションがインストールされていない場合に、そのアプリケーションをインストールするよう促すこともできる。

インターネット上で映像を配信する際も同様に、WWW ページを入り口としてサービスを提供しているケースが多い。すなわち、利用者が HTML で記述されたリンクを参照することで、ブラウザがアプリケーションを起動する。そして、そのアプリケーションがリンクにあるファイルに記述されているメタファイルを読み込むことで、映像配信用のサーバに接続し、指定された映像ファイルをダウンロードし始める。

3.1.2 HTML の構造

WWW が今日のように広く利用されるようになった理由の一つは、テキストで記述された語句とそれに関連する情報を結びつけるハイパーリンクを実装したことにある。ハイパーリンクを記述した文書をハイパーテキストと呼び、HTML 言語はその概念を基に作られた。そして、HTML や情報を転送する WWW サーバが生まれ、それを解釈するブラウザが生まれた。これらの技術を利用することで、ハイパーリンクによって情報を語句に関連付けし、その関連付けを入れ子構造にすることによって世界中の情報を一つの HTML ファイルに関連づけることさえできる。

WWW ページに表示される画像ファイルについても同じことが言える。HTML の IMG タグに画像ファイルのある URI を記述することで、WWW サーバの動いている他のホスト上にある画像を、あたかもそのページにあるかのように表示することができる。つまり、ダウンロードした HTML ファイルに他のホストにある情報をリンクさせるだけで、ブラウザがそれを解釈し、他のホストと新しいコネクションを張り、情報をダウンロードする。利用者にとっては、一つのホームページを一つのホストからダウンロードしたように感じるが、実際は HTML ファイルに記述されている URI にあるホストと新しいコネクションを張り、情報をダウンロードし表示している。図 3.1 にブラウザに表示される画像を他のサーバから取得している概念図を示す。

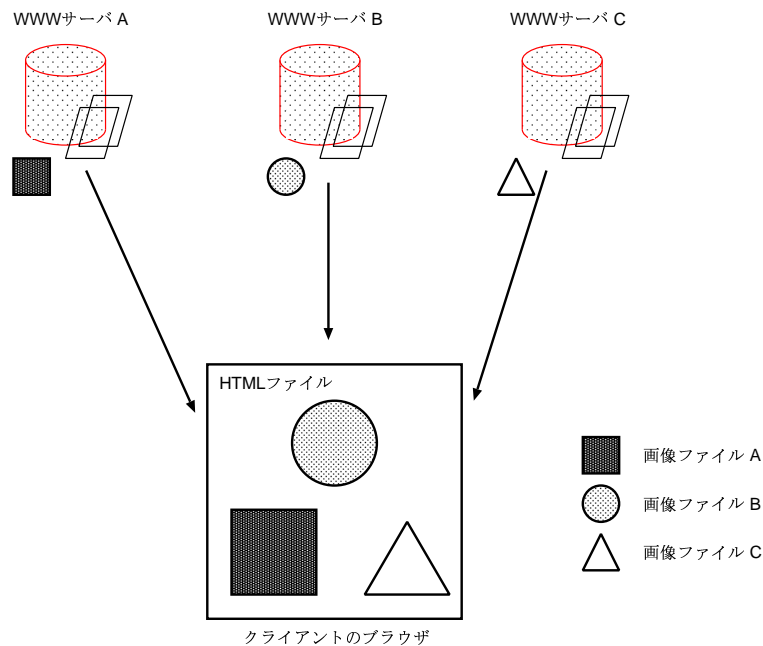


図 3.1: 他のサーバにある画像へのリンク

このように、コンテンツ配信では、利用者がコンテンツを閲覧するためにアクセスする手段が WWW であること、HTML が複数のホストに分散した任意のデータを組み合わせて一つのページとして表現できること、そして、これらが利用者あまり意識されことなく行われている、という3点が特徴と言える。この特徴をうまく利用すると、利用者に意識されずに任意の WWW サーバとクライアント間における転送性能を計測できる。

すなわち、クライアントが最初にアクセスする HTML ファイル内に、データ転送時間を計測する機能を持った WWW サーバに向けたリンクを用意することで、データ転送に要した時間と転送したデータの大きさから、実際にクライアント・サーバ間で転送されたデータ量、つまり実効帯域を算出することができる。その算出された実効帯域を転送性能として利用することで、クライアントは常に適切なサーバからコンテンツを閲覧することができる。

3.1.3 CDN の利用

映像配信をする際に問題となるのは配信するサーバへのアクセスの集中、トラフィックの集中である。第2章で述べたように、その問題を回避するために CDN 技術がある。CDN において、クライアントにとって最適なキャッシュサーバ選択は非常に重要な機能である。しかし、映像配信では継続的なデータ転送が行われる上、品質が転送性能に比例するため、既存の CDN で用いられている応答時間やホップ数などの間接的な情報より、実際にデータ転送を行う2点間の転送性能を用いてキャッシュサーバを選択すべきである。つまり、クライアントが再生できる最高品質の動的コンテンツを選択するために、任意のキャッシュサーバ群から転送性能が最も良好なキャッシュサーバへ誘導すべきである。

この時、クライアント・キャッシュサーバ間の転送性能をどのように計測するかが問題になる。本研究では、利用者が WWW ページを介し動的コンテンツにアクセスする仕組みを考慮し、同時に HTML によるハ

ハイパーリンク機能を利用することで、ネットワークに負荷をかけない形で計測する手法を提案する。それは、最初にアクセスする WWW ページの HTML に複数のキャッシュサーバへのハイパーリンクを張り、それぞれのキャッシュサーバからクライアントへのデータ転送時間を計測するという方法である。これによって同時に複数のキャッシュサーバからクライアントに対する実効帯域を計測できる。

計測された転送性能は、キャッシュサーバ選択の指標として用いられ、最終的にクライアントは最も適したキャッシュサーバに誘導される。また、測定された転送性能を基に、アプリケーション、メディアタイプを動的に指定することで、利用者は既にインストールされているアプリケーションを利用し最もその時の状況に合致したメディアタイプにアクセスできる。

3.2 データ転送時間を利用したリクエストナビゲーション機構

3.2.1 目的

映像配信を行う際、CDN 上に大量に存在するキャッシュサーバの中から、サーバ側で適切なキャッシュサーバを選択し、クライアントをキャッシュサーバへ誘導するためのフレームワーク作りを行う。

3.2.2 システム概要

第2章で議論したように、キャッシュサーバ選択にはトポロジー情報だけでは十分でないという理由から、データ転送時間を計測し、転送性能を求めることで、キャッシュサーバ選択を実現する。また、転送性能情報だけを利用するのではなく、トポロジー情報と連携することで最適なキャッシュサーバ選択を実現する。

図 3.2 に転送性能情報を利用したリクエストナビゲーション機構のシステム概要を示す。

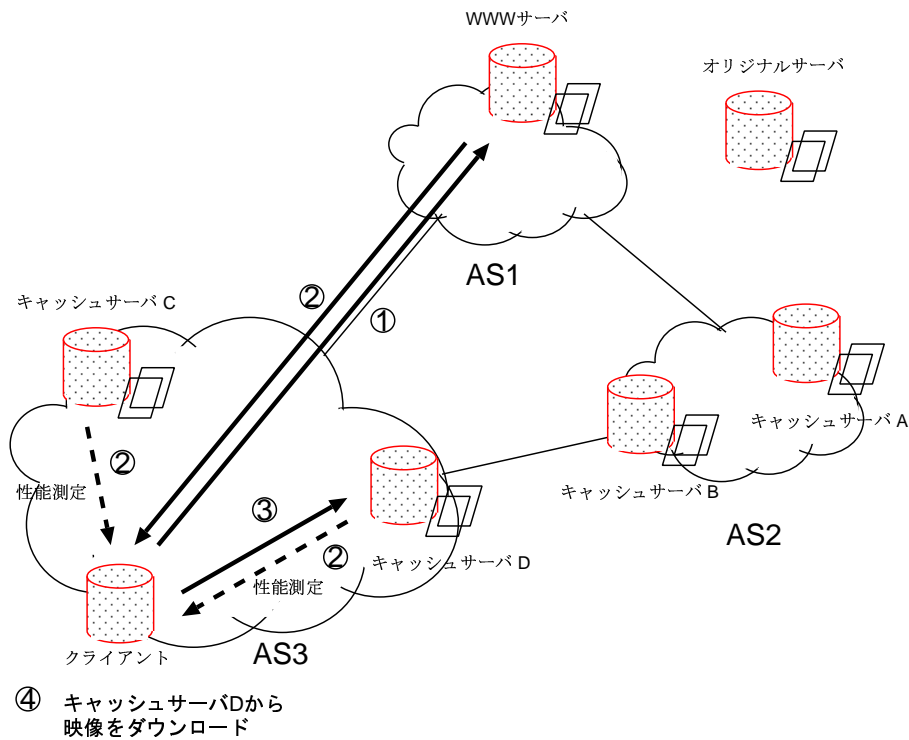


図 3.2: システム概要図

前提条件としてキャッシュサーバはオリジナルサーバが提供している動的コンテンツのデータを共有しているものとする。

サーバ選択までの流れ

第一に、最初に WWW サーバにアクセスのあったクライアントの情報からネットワーク地図上でクライアントに距離が近い位置に設置されている、少数のキャッシュサーバ群を選択する。図 3.2 で言えば、クライアントにとってはキャッシュサーバ C とキャッシュサーバ D がまず選択されることになる。その後、選択されたキャッシュサーバ群がクライアント・キャッシュサーバ間の転送性能を測定する。測定した転送性能情報を利用して、サーバ側でクライアントにとって最適なキャッシュサーバを選択し、クライアントが利用する。図 3.2 で言えば、キャッシュサーバ C との転送性能よりも、キャッシュサーバ D との転送性能の方が優れていたため、クライアントはキャッシュサーバ D から動的コンテンツを受信する。

3.2.3 転送性能の測定

様々なトラフィック測定に関するツールが開発されている。様々なツールを概説しながら、本研究で採用する転送性能測定の位置付けを述べる。

本研究のシステムでは、クライアント・キャッシュサーバ間実際に転送することができる帯域幅を測定することが求められる。また、サーバ選択は短い時間で行われるため、測定にかかる時間は短い方が望ましい。そして、クライアント・キャッシュサーバ間で利用できる帯域の時間的な変化を計測できることが望ましい。

関連研究

- tcpdump[31]

tcpdump プログラムはネットワーク上のある一点でのトラフィックを測定することが出来る。tcpdump プログラムを利用することで、ホストが接続している回線に流れているデータを取得することができ、実際のトラフィックから帯域情報を取得できる。しかし、このツールはネットワーク上の任意の一点でのみ利用できることから、本システムの要求とは合致しないと判断した。

- Pathchar[32]

パケットペアアルゴリズム [33] を利用して、End to End の回線帯域を計測できる。ICMP を利用して応答時間を何度も計測することで、End to End 間の回線の定常的なボトルネックを計測する。このツールは、計測に1時間以上かかる場合もあり、情報の即時性がない。また、回線上に計測用のデータを流すため、途中回線に流れる他のトラフィックに対して負荷がかかり、トラフィック負荷分散機構を構築するために不必要なトラフィックを流すことは矛盾するため、本システムの要求と合致しないと判断した。

- Netperf[34]

実際にトラフィックを End to End で転送し、実効帯域を測定する。この手法は Pathchar 同様、他のトラフィックに影響を与える。また、両端のエンドノードにこのプログラムを起動している必要があるため、利用者透過性がない。よって、この測定方法も本システムの要求と合致しないと判断した。

- ADSL 測定ツール

ADSL 回線の性能を測るため、画像ファイルやテキストファイルをダウンロードした時間を計測し、ダウンロードされたファイルの大きさを除算することで、実効帯域を計測する。インターフェースが WWW であるため、利用者は気軽に利用できる。しかし、性能測定のためだけに不必要なデータを転送しなければいけないため、流れるデータの内容などに工夫が必要であるという欠点がある。

本研究では、ADSL 測定ツールと同様の転送性能測定手法を用い、転送データとして画像ファイルを用いる。なぜなら、動的コンテンツのほとんどが WWW ページと連携しているおり、必ず利用者は WWW ページにアクセスし、何かしらの画像データやテキストデータをダウンロードするからである。データをダウンロードする際の転送時間を測定することで、途中回線に余計な負荷をかけずに実効帯域を測定でき、また、比較的短い時間で測定を終えられる。ただし、この手法では動的コンテンツを最初に受け取る際に転送性能を測るため、リクエストが来た時の情報しか取得できないという問題点がある。

3.2.4 利用可能帯域の時間変化

ネットワークの特性は時間帯によって変動する。そのため、統計的な手法によるキャッシュサーバ選択手法が有効であると考えられる [35]。本研究では、コンテンツの転送性能を最適化することを目的とした、動的に最適なサーバを選択する手法を提案する。実際の振り分けは、サービス利用者が映像をダウンロードする時に参照するメタファイルを書き換えることで実現する。これは映像配信というサービスでしか成り立た

ない方法であるという欠点がある。しかし、DNS を利用した手法と比べると、クライアントが映像をダウンロードする直前の帯域情報を利用した方が明らかに有効である。なぜなら、DNS はそれ自身の規模性を維持するためにキャッシュを利用しているからである。つまり、たとえ DNS サーバ内のデータを動的に更新しても、DNS キャッシュにそのデータが存在する間は実際に更新されていない状態と同じである。したがって、最適なキャッシュサーバ選択には向かないと言える。一方、帯域情報からメタファイルを生成する方法では、全てのクライアントが転送性能データベースを基に生成されたメタファイルを WWW サーバから参照するので、映像をダウンロードする直前の帯域情報を利用することができる。

3.3 データ転送時間を利用した再生アプリケーション、メディアタイプの選択

3.3.1 目的

映像の品質は動的コンテンツの転送データ量に比例する。つまり良好な品質の動的コンテンツを再生するためにはクライアント・キャッシュサーバ間に良好な転送性能が必要である。しかし、実際の転送性能と転送される動的コンテンツのデータ量が一致しなければならない。もし一致していない場合は、必要以上のデータが転送されたり、途中回線でパケットロスが起き、クライアントが動的コンテンツを再生できない可能性がある。また、インターネットでは、動的コンテンツ利用者のメディア再生環境が異なることから、動的コンテンツを多くのクライアントが再生可能になるためには、個々のクライアント毎にクライアントに適したアプリケーション、メディアタイプを選択する必要がある。今現在は、動的コンテンツを受信する際のアプリケーション、メディアタイプ選択が利用者の直感に頼っている。直感というあいまいな指標で選択するのではなく、選択のための定量的な指標を利用者に与えることで、クライアントにとって適した動的コンテンツを提供する。

3.3.2 手法

第 3.2 節のシステムで選択されたキャッシュサーバから、計測した転送性能情報を利用し、アプリケーション、メディアタイプの選択を行うことで、実際の転送性能に適した動的コンテンツを再生することができる。図 3.3 にメディア選択の概念図を示す。

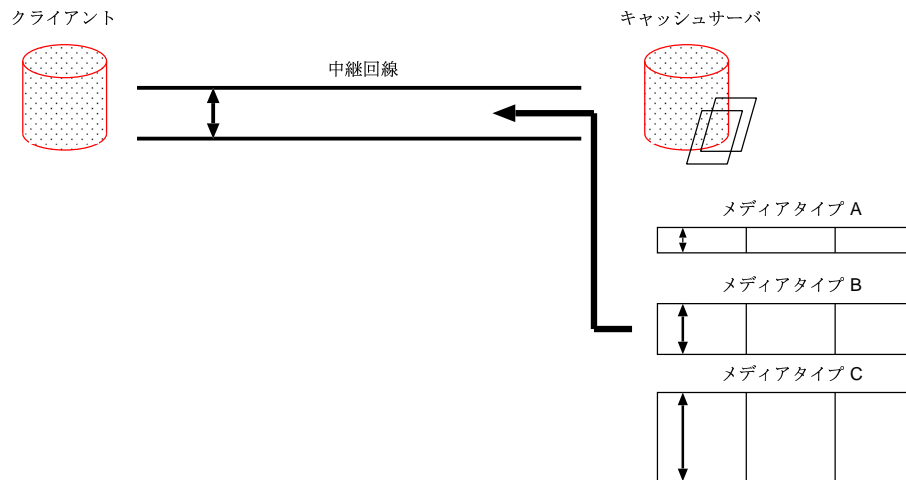


図 3.3: メディアタイプ選択

図 3.3 にあるように、クライアント・キャッシュサーバ間で測定された転送性能に比べ、メディアタイプ A をダウンロードした場合は転送性能を使い切ることができない。また、メディアタイプ C をダウンロードした場合は、転送性能を超えた量のデータが流れ込み、遅延やパケットロスを起こす。よって、クライアントは測定した転送性能に適したメディアタイプ B を選択するべきである。そうすることで、データ量と動的コンテンツの品質が比例するという事実から、クライアントに適した品質のデータをダウンロードすることが可能になる。また、利用するアプリケーションによっても品質が異なったり、配信できるデータ量が限定されている。アプリケーション選択に関しても第 3.2 章でのべた方法で計測した転送性能を利用し、今まで直感に頼っていたアプリケーション、メディアタイプの選択に定量的な指標を与える。

3.3.3 様々なアプリケーション

映像メディア

- Windows Media Technology(ASF)

マイクロソフト社が開発した技術で、比較的低帯域用の映像配信技術である。この技術は比較的多くのユーザが利用している。Inteligent Stream という、自動帯域制御機構を備えている。

- Real Media

Windows Media Technology と同様に、比較的低帯域用の映像配信技術である。比較的多くのユーザが利用している。圧縮手法は公開されていない。SureStream RTCP による自動帯域制御機構を備えている。

- MPEG1

機能が公開されているので、多くの企業や学生が開発可能である。

- MPEG2

DVD で利用されているように、高品質な映像配信技術である。差分圧縮をしているので差分の間引きをすることで、圧縮比率を変更することが出来る。

- MPEG4

最近開発された、映像配信技術である。MPEG1,2 とは異なり、低帯域用に作られた。

図 3.4 にアプリケーションが使用する帯域の範囲を示す。

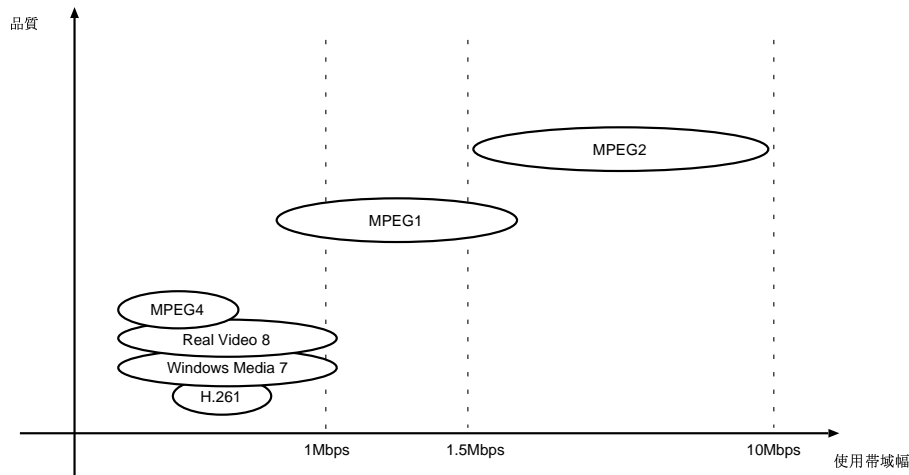


図 3.4: アプリケーションが使用する帯域の範囲

図 3.4 にあるように、様々な映像の符号化手法があり、それぞれ固有の特色を持っている。つまり、一つのアプリケーションで様々な回線品質をもったクライアントに対して動的に適応するのは難しい。

本システムを利用することにより、今まで利用者の直感というあいまいな指標で選択されてきたアプリケーション、メディアタイプを転送性能という指標を用いることで、利用者がより正確に選択を行うことができるようになる。また、アプリケーション固有の解決ではなく、網全体としてアプリケーション、メディアタイプ選択に指標を与えることで、様々なアプリケーションの乱立した状況でのアプリケーション、メディアタイプ選択の問題を回避する。

第4章 本研究の設計

第3章に述べたように、動的コンテンツを提供する側と利用する側の双方の要求を満たすためには、利用者・提供者間のトラフィック負荷分散をすることである。また、利用者・提供者間の転送性能を最大限に利用できるメディアタイプをクライアントが選択できることである。前者については、利用者にとって WWW ページを入り口としたコンテンツ配信の仕方が一般的であるため、利用者に透過的であり、また途中回線に余計なトラフィックを流さないために、必然的にダウンロードする画像のデータ転送時間を計測し、サーバ選択の指標とすることで、CDN 上のキャッシュサーバ選択を最適化する。

また、後者については、前者のシステムで計測した転送性能情報を利用して、これまで利用者の直感に頼っていた、アプリケーション、メディアタイプ選択に定量的な指標を提供することで、利用者は利用者・提供者間の転送性能に適したアプリケーション、メディアタイプの動的コンテンツをダウンロードすることが可能になる。

4.1 システム全体像

本システムでは、キャッシュサーバにある画像をクライアントがダウンロードした時間である、データ転送時間を計測することによって帯域情報を取得し、取得した帯域情報を基に、サーバ側で、選択されたキャッシュサーバ、アプリケーション、メディアタイプの URI を WWW サーバにあるメタファイルに記述する。そのメタファイルをクライアントが参照することによって、CDN 上のキャッシュサーバ選択、キャッシュサーバが持つ動的コンテンツのアプリケーション、メディアタイプ選択を実現する。

クライアントが動的コンテンツをダウンロードする直前にアクセスする HTML に含まれる画像のデータ転送時間を計測することでクライアント・キャッシュサーバ間の帯域をクライアントにとって透過的に計測し、クライアントを適切なキャッシュサーバへ誘導する。

帯域情報を利用してクライアントが適切なキャッシュサーバ、アプリケーション、メディアタイプ選択を行うまでのシステム概要を図 4.1 を利用しながら説明する。図 4.1 に本研究のシステム概要を示す。

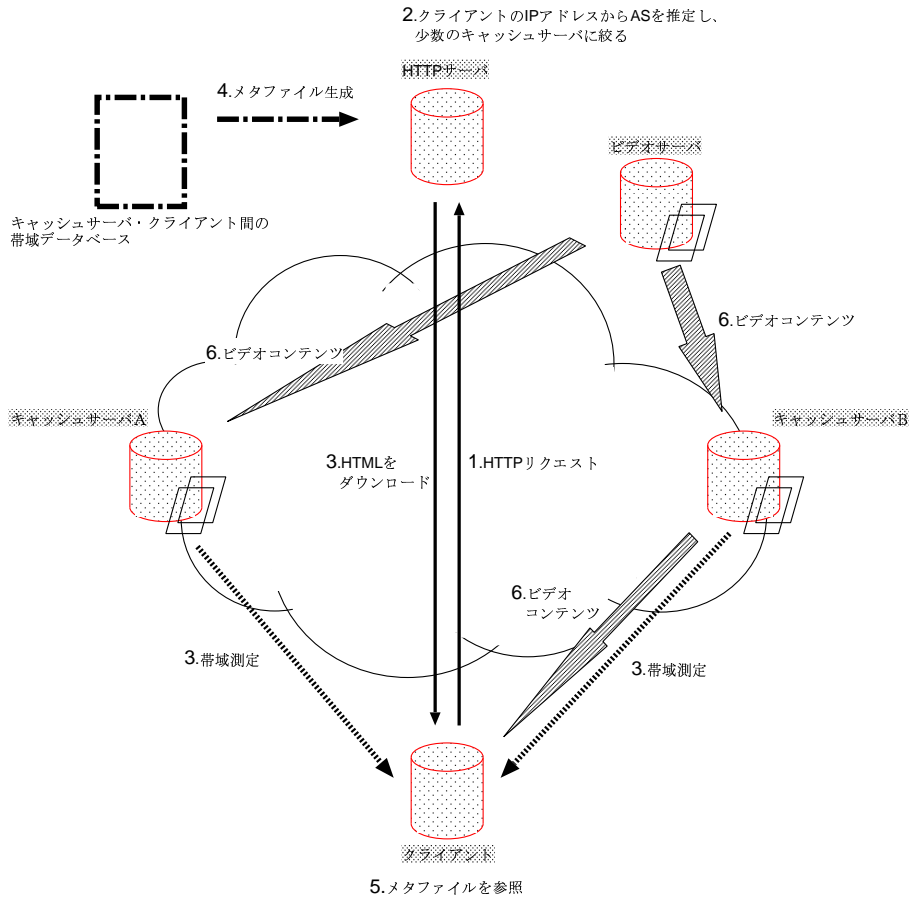


図 4.1: システム概要図

キャッシュサーバ選択までの流れ

1. クライアントが HTTP デモンへリクエストを送出

現在のインターネットにおけるコンテンツ配信では、テキスト情報や画像情報と共に WWW ページを入り口として動的コンテンツを再生する仕組みになっているため、映像を受信する前にクライアントは必ず最初に WWW ページにアクセスする。

2. クライアントの IP アドレスから数台のキャッシュサーバ選択候補を決定

クライアントの IP アドレスと AS 間経路制御プログラムである BGP4 が保持する AS_PATH 情報を用いて、ネットワーク地図上でクライアントに距離が近い、あるいは同一の AS にあるキャッシュサーバを数台選択する。この手法によって、クライアントに大量の画像ファイルをダウンロードさせることを避ける。

3. 帯域データベースを作成し、転送性能の優位を判別

(a) キャッシュサーバからの画像をダウンロード

キャッシュサーバにある画像はクライアントが最初にアクセスした WWW ページの HTML の中でリンクが張られている。その画像をクライアントは HTML のコンテンツの一つとして、キャッシュサーバに接続し、ダウンロードする。そして、キャッシュサーバの HTTP デーモンがデータ転送時間を計測し、送出したデータ量を除算することで帯域を算出する。その結果得られた帯域情報を帯域データベースへ受け渡す。

(b) クライアントのアプリケーションを検出

クライアントのアプリケーション、メディアタイプ選択に定量的な指標を与えるために、クライアントが持っているアプリケーションを検出するプログラムを HTML に記述しておく。

(c) キャッシュサーバの負荷状況を検出

複数あるキャッシュサーバの使用帯域情報、CPU 使用率等を検出する。

4. 帯域データベースからメタファイルを生成

帯域データベースを利用し、クライアントが参照するメタファイルを作る。メタファイルの中には選択されたキャッシュサーバのアドレスとクライアントの持っているアプリケーションの特定のメディアタイプの組が記述される。

5. クライアントがメタファイルを参照

ダウンロードした WWW ページにリンクの張られたメタファイルをクライアントが参照することで、帯域データベースによって選択されたキャッシュサーバへクライアントを誘導する。また、クライアントは、クライアント・キャッシュサーバ間の転送性能に適したメディアタイプを、ネットワーク地図上の距離が近いキャッシュサーバの中でも転送性能が一番優れたキャッシュサーバからダウンロードする。

6. クライアントが映像ソースをダウンロード

クライアントは帯域データベースにより選択されたキャッシュサーバから再生可能なアプリケーションで作成されたメディアタイプをダウンロードする。クライアントが WWW サーバにアクセスし、映像をダウンロードするまでの操作は、本システムを利用していない場合と同様の操作で、本システムを利用することができる。

4.1.1 データ転送時間を指標として用いたサーバ選択の先行研究との比較

参考文献 [35] では主に画像ファイルやテキストファイルを配信した場合を対象として議論されている。それに対して、本研究では映像配信に主眼を置き議論を進める。また、クライアントのプロキシサーバとコンテンツ配信のキャッシュサーバの間で定期的に帯域を測定しているが、本研究では各クライアントがリクエストする毎に帯域を計測する。これにより、計測結果が非常に膨大な量になる可能性があるが、クライアントとキャッシュサーバ間の正確な帯域情報を得ることができる。また、プロキシサーバを用いる場合に必要なクライアント側の設定が不要となり、利用者に意識させることなくキャッシュサーバへ誘導できる。

計測はクライアント側からではなくキャッシュサーバ側から行う。利用者側からの帯域情報を測定する手法は、計測結果を何らかの方法で集積しない限り再利用できず、規模性がなくなる。利用者が帯域情報を測定するのではなく、サーバ側で測定することで、負荷状況など他の状況と協調してキャッシュサーバを選択することができるようになる。

計測に用いる転送データは、画像ファイルやブラウザに表示されない文字列を用いる。転送データの大きさは固定長(例: 1Mbyte の画像ファイル)ではなくクライアントの帯域に合わせられるようにする。

計測によって得られた帯域情報は、IPアドレスの構造に即した形で保持すると共に分析することで、AS_Pathを用いたものより細かな粒度で帯域の状態を抽出できると考える。

4.2 転送性能の優位を判別

4.2.1 キャッシュサーバの画像へのリンク

キャッシュサーバ、あるいはキャッシュサーバと同じネットワーク上にあるホストにデータ転送時間を計測するための HTTP デモンを起動する。クライアントが動的コンテンツをダウンロードする直前に接続する WWW サーバの HTML に IMG タグを利用して、トポロジー情報で選択された少数のキャッシュサーバ群が持っている画像にリンクを張る。図 4.2 に画像ファイル埋め込みの例を示す。

```
<IMG src=' 'http://cacheserver1.sfc.wide.ad.jp/image1.jpg' '><BR>  
<IMG src=' 'http://cacheserver2.sfc.wide.ad.jp/image2.jpg' '><BR>  
<IMG src=' 'http://cacheserver3.sfc.wide.ad.jp/image3.jpg' '><BR>
```

図 4.2: 画像ファイルの埋め込み(例)

この IMG タグを埋め込むことで、サーバ側はクライアントに意識させることなく、トップページをダウンロードさせ、トップページをダウンロードした時に、キャッシュサーバはクライアント・キャッシュサーバ間の帯域情報を計測することができる。図 4.3 にクライアントがキャッシュサーバから画像をダウンロードする仕組みを示す。

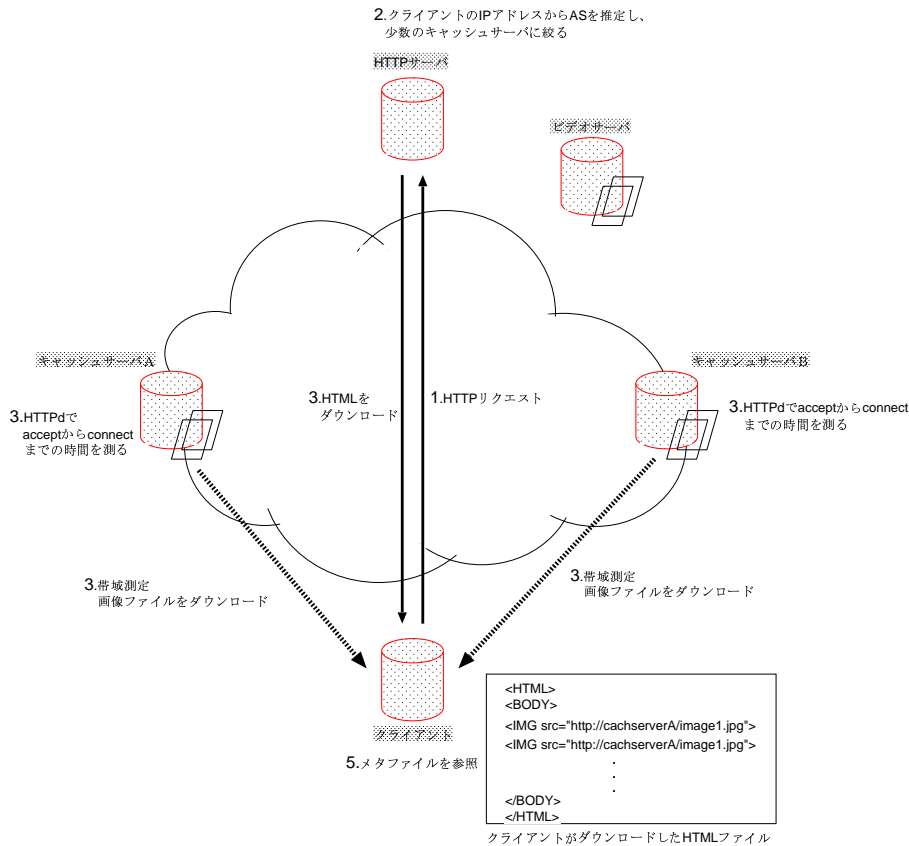


図 4.3: キャッシュサーバから画像をダウンロード

表示される画像

多くのキャッシュサーバがネットワーク上に存在するので、IP アドレスからクライアントの所属する AS 番号を取得し、BGP4 の持つ AS_PATH 情報から、その AS に近い、あるいは同一 AS 内にある少数のキャッシュサーバの画像ファイルに限定する。その際、クライアントの負荷が大きくなることに注意する。

クライアントがダウンロードに待てる時間は約 8 秒といわれている [36]。8 秒以内に画像がダウンロードされるために、画像の大きさを考慮する。28.8Kbps で接続されているクライアントは、8 秒間に約 230Kbit 転送可能である。56Kbps は約 448Kbit。128Kbps は約 1024Kbit。つまり、28.8Kbps のクライアントに少なくとも二つのキャッシュサーバから選択させるためには、最高でも 115Kbit の画像ファイルでなければならない。

4.2.2 データ転送時間の計測方法

図 4.3 にあるように、クライアントがキャッシュサーバで起動している HTTP デーモンへ connect し、キャッシュサーバ側で accept した時間から close した時間までを計測し、データ転送時間とする。また、ダウンロードされた画像ファイルの大きさをデータ転送時間で除算することによって帯域情報とする。図 4.4 以下に帯域情報の計算式を示す。

$$\text{帯域情報} = \frac{\text{画像サイズ}}{\text{転送終了時刻} - \text{転送開始時刻}}$$

図 4.4: 帯域情報の計算式

本研究では計測用の HTTP デーモンを作成し、`gettimeofday` 関数を用い、画像ダウンロード時間を計測した。計測したダウンロード時間は IP アドレス、サブネットワークレベル毎にデータベースに保持され、キャッシュサーバ選択の指標として利用される。

4.2.3 ダウンロード時間データベース

転送性能が優れたキャッシュサーバを判別するために、以下の情報を帯域データベースが保持する。

- キャッシュサーバの IP アドレス
- クライアントの IP アドレス
- クライアント・キャッシュサーバ間の帯域情報
- クライアントの IP アドレス毎の帯域情報の平均値
- クライアントの IP アドレス毎の帯域情報の標準偏差
- サブネットワークレベルの帯域情報の平均値
- サブネットワークレベルの帯域情報の標準偏差

複数のキャッシュサーバで計測されるこれらの情報を利用し、クライアントが参照するメタファイルを生成する。

AS 内部のネットワーク状況

IP アドレス情報と帯域情報を用いて、帯域情報がある一定の閾値以内であれば、粒度は粗いものの、その情報を基に大きなネットワークとして見ていたものを、サブネットワークとして区切って扱うことができる。サブネットワークの情報を持っていることによって、新たに接続してきたクライアントに対しても、帯域の予測をすることができる。

4.2.4 優位の判別方法

データベースサーバに保存されたデータから、キャッシュサーバを選択する。また、帯域情報とアプリケーション、メディアタイプ情報からメディアタイプを選択する。

4.3 リクエストナビゲーション機構

キャッシュサーバの画像をダウンロードすることで計測した転送性能情報を利用して、クライアントが参照するメタファイルを生成する。それぞれのクライアント毎に異なるメタファイルを生成し、そのメタファイルを参照できるように、入り口のWWWページにアクセスした際にクライアント毎にメタファイルへのハイパーリンクを変える。

図4.5にあるように、メタファイルを自動生成し、それをクライアントが参照することによって、適切なキャッシュサーバへクライアントを誘導することができる。図4.5にメタファイルによるクライアントの誘導の概念図を示す。

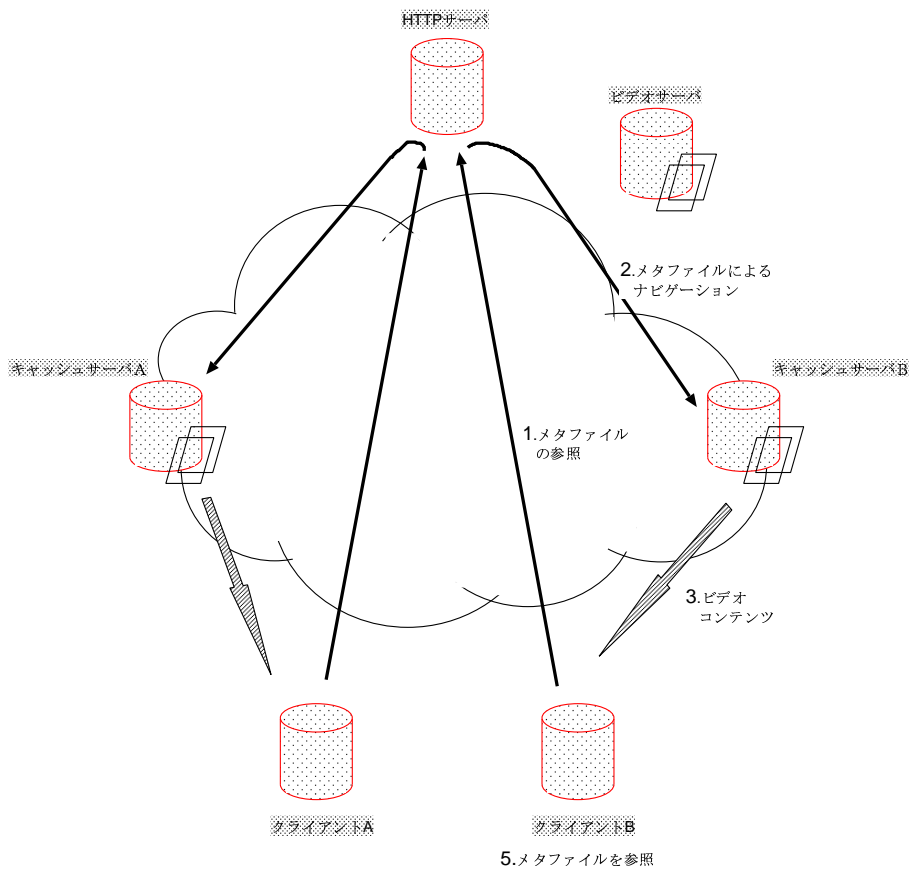


図 4.5: メタファイルによる誘導

4.4 メタファイルの生成

図4.6、図4.7にメタファイルの記述例を示す。Real Media の場合は図4.6の内容を、ram という拡張子のテキストファイルとして作成する。

```
rtsp://cacheserverB.sfc.wide.ad.jp/live.rm
```

図 4.6: メタファイルの記述例 1(Real Media)

Windows Media Technology では図 4.7 に示す内容を拡張子が asx というテキストファイルに記述する。

```
<ASX version =''3.0''>
<Entry>
<Ref href='''http://cacheserverB.sfc.wide.ad.jp/live.asf'' />
</Entry>
</ASX>
```

図 4.7: メタファイルの記述例 2(Windows Media Technology)

これらのメタファイルをクライアント毎に生成することで、クライアントを適切なキャッシュサーバに誘導でき、またアプリケーション、メディアタイプを帯域情報から選択することができる。

4.5 映像品質

4.5.1 データ転送時間を利用したアプリケーション選択機構の設計

放送型映像配信機構では、動的コンテンツを再生する個々のクライアントの再生環境に適したアプリケーション、メディアタイプを選択する仕組みを実現する必要がある。それは、クライアント・サーバ間の転送性能に適合したメディアタイプを選択することで、必要以上のトラフィックを流すことがなく、また、個々のクライアントの再生環境に適しているため、より多くの利用者に動的コンテンツを再生してもらうことができる。

第 4.2 節で計測したデータ転送時間から算出された帯域情報を利用することでクライアントに適したアプリケーション、メディアタイプを選択する仕組みを提案する。

4.5.2 前提条件

映像、音声の配信を行うオリジナルサーバには、複数のアプリケーションに対応したデータを設置する。また、複数の帯域に適したデータを持つことを前提とする¹。また、オリジナルサーバにあるデータはキャッシュサーバも持っていることを前提とする。そして、本システムで提案する帯域情報を利用したリクエストナビゲーションと共に用いる。

4.5.3 システム概要

図 4.8 にアプリケーションを選択するまでのシステム概要を示す。

¹Generic Media[30] が開発した一つの圧縮手法で圧縮されたファイルを様々な圧縮手法へ自動生成するソフトウェアと協調することが望ましい。

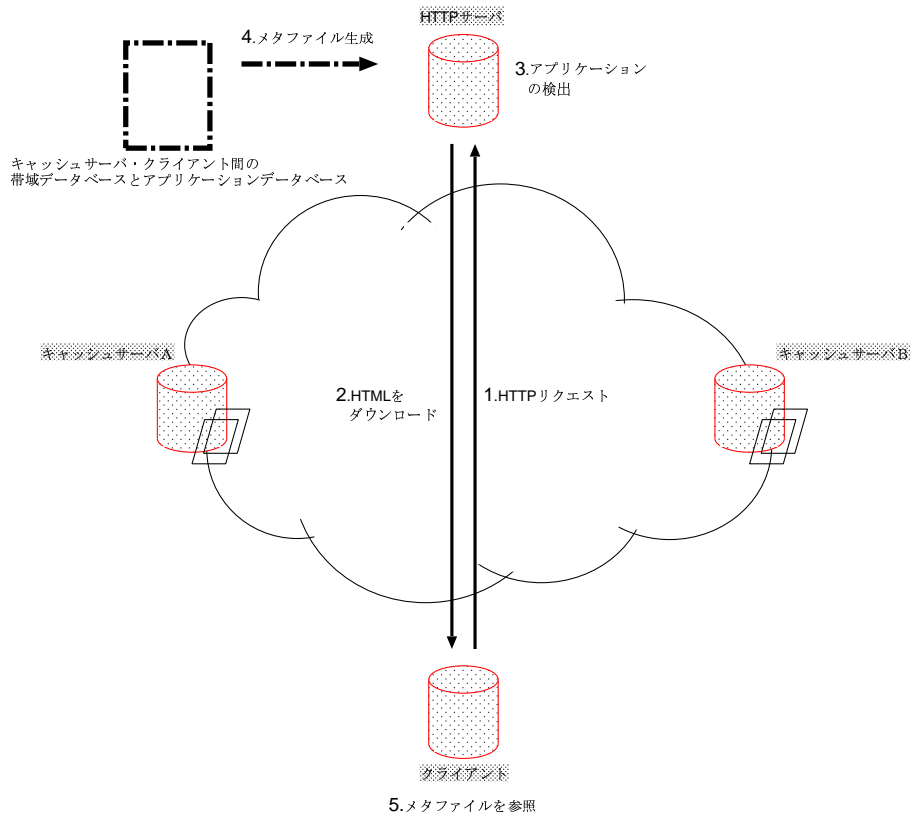


図 4.8: システム概要図

アプリケーション選択までの流れ

1. クライアントの保持するアプリケーションのリストを取得

クライアントが HTML をダウンロードした際に、その HTML に Javascript を埋め込むことによってクライアントのブラウザにインストールされているプラグイン情報を取得する。持っているアプリケーションによって、表示する HTML を変える。

2. 帯域データベースの情報を利用

帯域情報データベースを参照することで、クライアント・キャッシュサーバ間の帯域情報を得る。

3. メタファイルの生成

クライアント・キャッシュサーバ間の帯域情報とアプリケーションのリストから、キャッシュサーバ選択によって選択されたキャッシュサーバと同時にメディアタイプをメタファイルに記述する。図 4.6 と図 4.7 にあるように記述する。

4. メタファイルの参照

図 4.5 にあるように、クライアントが生成されたメタファイルを参照することによって、帯域情報によって選択されたキャッシュサーバから適切なアプリケーション、メディアタイプを選択し、動的コンテンツを再生する。

このシステムによって、クライアントが意識することなく、帯域情報を利用したメディアタイプ選択、また、アプリケーション選択が実現する。

次に上記のアプリケーション選択における個々の機能について述べる。

4.5.4 アプリケーションのリストを取得

コンテンツ配信の入り口となる WWW サーバからダウンロードする HTML の中に Javascript で記述されたアプリケーション情報取得プログラムを記入する。そこで取得されたアプリケーションの情報をデータベースに保持する。図 4.9 にクライアントが持つプラグインを確認し、メタファイルを切り替える例を示す。

```
<HTML><HEAD><TITLE></TITLE>
<script language="JavaScript1.1">
<!--
function CPlug(CP){
    for(i=0; i < navigator.plugins.length; i++){
        if(navigator.plugins[i].name.indexOf(CP) != -1)
            return true
    }
    return false
}
// --->
</script>
</head>
<body bgcolor="#FFFFFF">

<script language="JavaScript1.1">
<!--
if(CPlug("RealPlayer")){
    document.write("<a href=\"metafile.ram\">Click here</a>")
}
else if(CPlug("Windows Media")){
    document.write("<a href=\"metafile.asx\">Click here</a>")
}
//--->
</script>

</body>
</html>
```

図 4.9: Javascript によるプラグイン検出

図 4.9 は Windows Media Player と Real Media のどちらかを持っているクライアントを対象にし、Real Media を優先した場合の例である。

4.5.5 メディアタイプの選択

4.2 節で計測した個々のクライアント毎の帯域情報を帯域データベースから取得し、Javascript によって検出したプラグイン情報を組み合わせることによって、メディアタイプの選択を行う。

4.5.6 メタファイルの生成

メディアタイプの選択が行われた後、オリジナルサーバがあらかじめ用意しておいた、個々のクライアントの再生環境に適応する動的コンテンツのファイルの URI を記述したメタファイルを生成する。それぞれの帯域、アプリケーションに応じたメディアタイプのデータをメタファイルに記述することでクライアントに適切なアプリケーション、メディアタイプ選択を実現する。

第5章 評価

5.1 定性評価

本節では、これまで述べてきたデータ転送時間を利用したリクエストナビゲーション機構について、他の機構との定性評価を行う。

5.1.1 データ転送時間を利用したリクエストナビゲーション

まず本機構の比較対象となる機構を示す。その上で、どのような項目について、本機構と比較対象の機構を比較し、評価を行うかを説明する。その後、本機構と比較対象との比較結果を示す。

本機構の比較対象となる機構は、大きく分けて以下の二つである。

- DNS を利用

様々な指標によって得た情報を DNS データベースと連携させることにより、サーバ選択を利用者に依存せずにサーバ選択を実現できる。

- 単一 IP アドレスアーキテクチャ

この手法はネットワークの経路情報によるキャッシュサーバ選択を行う。複数のキャッシュサーバに一つのアドレスをつけることで実現する。

本節の評価で基本となるものは、参考文献 [19] のリクエストナビゲーションへの要求事項である。本研究は既存に広く利用されているメタファイルを参照するという方法で利用者を適切なキャッシュサーバへ誘導する機構を実現した。よって、要求事項の利用者透明性がある。また、実装非依存性についても DNS を利用したリクエストナビゲーション機構と同様に、基本的にはキャッシュサーバに HTTP サーバを設置するだけでよく、実装やプラットフォームには依存しない。そして、規模適応性についても、AS 内部に限られたキャッシュサーバからの選択に利用することから規模適応性がある。他の二つの手法もこれらを満たしている。

表 5.1 にリクエストナビゲーションにとって必要な要求に照らし合わせ、今まで挙げた項目以外の要求事項を用いて評価する。DNS を利用した手法を「DNS」、単一 IP アドレスを利用した手法を「単一 IP」と略して記す。

表 5.1: リクエストナビゲーションの機能評価

	サービス非依存性	サービス提供者非依存性	選択ポリシーの柔軟性	即時性	現実性
DNS				×	×
単一 IP		×	×		×
本研究	×				

これまでのリクエストナビゲーションにおいて、実帯域を計測するといった現実性や計測した結果を素早く反映させるような機構は存在しなかった。本研究は映像配信というサービス依存性があるものの、映像配信の特徴を利用し、現在まで行われてこなかった即時性や現実性のあるリクエストナビゲーションを実現した。

5.2 各機能の動作証明

2台のキャッシュサーバと一台のWWWサーバを用い、データ転送時間を利用したキャッシュサーバ選択、メディアタイプ選択が行われることを証明する。

5.2.1 実験環境

図 5.1 に実験環境を示す。

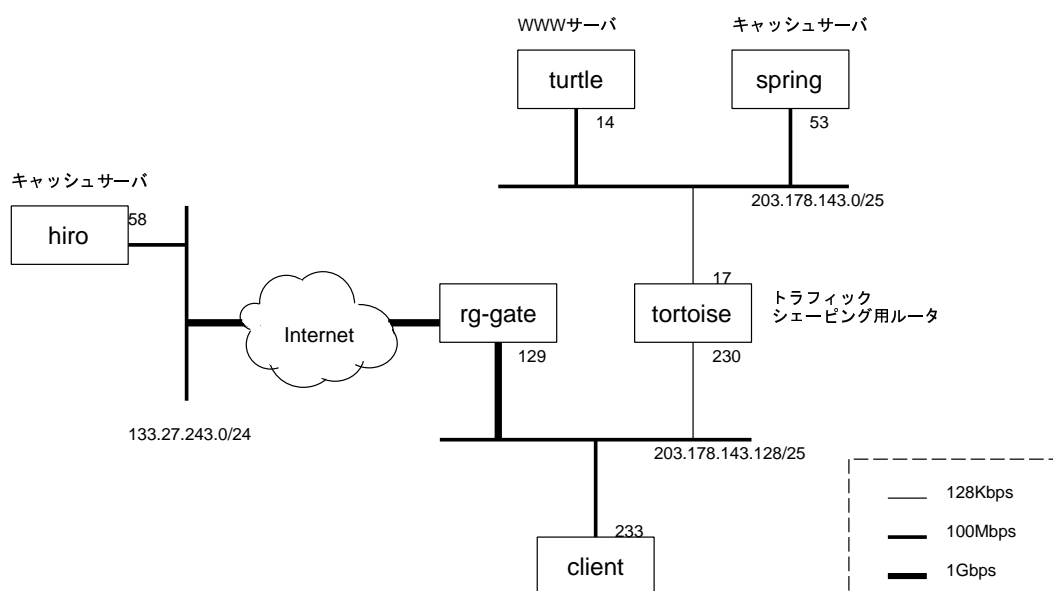


図 5.1: 実験環境

5.2.2 実験概要

図 5.1 にあるように、クライアントはホスト turtle の WWW サーバにコンテンツ配信の入り口としてアクセスする。その後、キャッシュサーバである、ホスト spring と hiro から画像をダウンロードする。最後に turtle で生成されたメタファイルを参照することによって、動的コンテンツを再生することを証明する。クライアント・キャッシュサーバ間の帯域に差異をつけるため、ホスト tortoise において dummynet プログラムを利用してトラフィックシェーピングを行い、client・spring 間の回線を 128Kbps に設定した。

5.2.3 ハードウェア環境

表 5.2 に本実験に用いたホストのハードウェア環境

表 5.2: ハードウェア環境

ホスト名	spring	hiro	turtle	tortoise	client
機種	PC 互換機	PC 互換機	PC 互換機	PC 互換機	PC 互換機
プロセッサ	PentiumIII 850MHz	PentiumIII 700MHz	PentiumIII 900MHz	Celeron266MHz	PentiumIII 850MHz
メモリ	128MB	384MB	512MB	192MB	512MB
NIC	100BaseTX	100BaseTX	100BaseTX	10Base-T/10Base-T	100BaseTX

5.2.4 ソフトウェア環境

表 5.3 に本実験に用いたホストのソフトウェア環境を述べる。

表 5.3: ソフトウェア環境

ホスト名	spring	hiro	turtle	tortoise	client
OS	FreeBSD 4.2-RELEASE	FreeBSD 4.1-RELEASE	FreeBSD 4.4-RELEASE	FreeBSD 4.4-RELEASE	Windows2000
使用言語	C	C	C,Perl		
コンパイラ	gcc 2.95.2	gcc 2.95.2	gcc 2.95.3		
WWW サーバ			Apache 1.3.22		
Traffic shaping			Apache 1.3.22	dummynet	

次に、本実験における各機能の動作をログファイルを参考にしながら検証する。

5.2.5 アプリケーションを選択

ホスト client がホスト turtle に接続し、HTML をダウンロードする際、Javascript によってプラグイン情報を取得し、Real Media 用の HTML に切り替わる。図 5.2 にホスト client がダウンロードした HTML ファイルのブラウザ表示を示す。

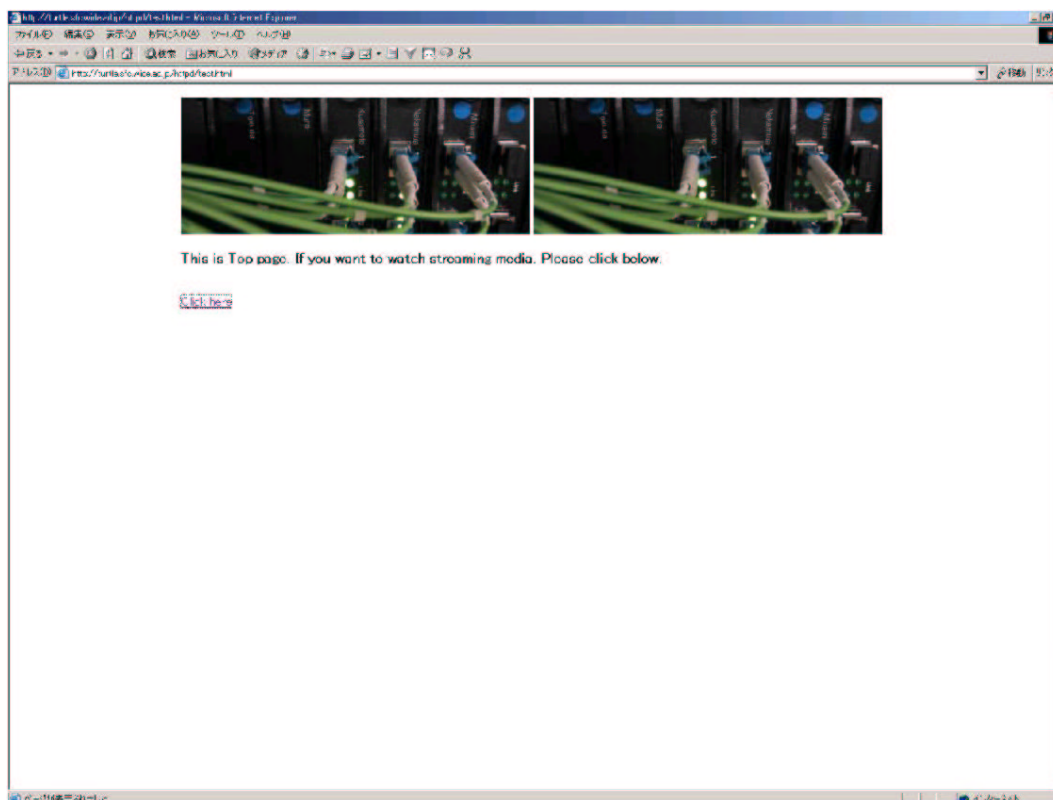


図 5.2: ダウンロードした HTML の表示画面

左方にある画像がホスト hiro の画像にリンクが張られている。また右方の画像はホスト spring の画像にリンクが張られている。そして、画像の左下部にある”Click here” という語句にメタファイルへのリンクを張った。クライアントは”Click here” を参照することで、動的コンテンツをダウンロードすることができる。

5.2.6 データ転送時間の計測

ホスト hiro とホスト spring において、本研究で作成した計測用の HTTP サーバを起動し、画像がダウンロードされる時間を計測した。計測には `gettimeofday` 関数を用い、HTTP デーモンがクライアントの `connect` を `accept` した時間から `close` した時間を計測した。HTTP デーモンは以下の情報を順に CSV 形式で記録する。

- アクセス時間
- ポート番号
- 帯域情報
- ダウンロード時間
- ダウンロードされたファイルサイズ

図 5.3、図 5.4 に 203.178.143.233 のアドレスを持つホストからのアクセスログを示す。

```
2002/1/15 00:52:36,203.178.143.233,1093,124972.729134,1.191748,18617
```

図 5.3: アクセスログ (spring)

```
2002/1/15 00:52:35,203.178.143.233,1094,34499884.178828,0.004317,18617
```

図 5.4: アクセスログ (hiro)

図 5.3、図 5.4 からクライアントがアクセスし、画像ファイルをダウンロードしたことがわかる。

5.2.7 クライアントに応じたキャッシュサーバ選択

キャッシュサーバ (spring,hiro) で計測された帯域情報を利用しメタファイルに適切なキャッシュサーバを記述する。メタファイルを記述するために、キャッシュサーバ (spring,hiro) はクライアントがホスト turtle に各キャッシュサーバ (spring,hiro) からの帯域情報を受け渡す。ホスト turtle では各キャッシュサーバから帯域情報を受け取るポリシーサーバが起動している。

キャッシュサーバは、測定した帯域情報、過去の履歴からの平均、標準偏差を計算し、そのデータをポリシーサーバへ渡す。受け渡すデータは CSV 形式で記述されている。以下にキャッシュサーバがポリシーサーバに受け渡すデータを順に示す。

- キャッシュサーバの IP アドレス
- 接続したクライアントの IP アドレス
- 帯域情報
- 過去のアクセスから求めた帯域情報の平均
- 過去のアクセスから求めた帯域情報の標準偏差
- 過去のアクセスから求めたサブネットワークレベルの帯域情報の平均
- 過去のアクセスから求めたサブネットワークレベルの帯域情報の標準偏差
- クライアントがキャッシュサーバにアクセスした時刻

図 5.5 に turtle のポリシーサーバがキャッシュサーバ hiro,spring から受け取ったログを示す。

```
133.27.243.58,203.178.143.233,34499884.178828,38460001.604224
,33033298.109855,26431655.388807,36662216.993240,2002/1/1500:52:35
203.178.143.53,203.178.143.233,124972.729134,31629289.857584
,36537013.370049,26431655.388807,36662216.993240,2002/1/1500:52:36
```

図 5.5: 各キャッシュサーバからの帯域情報 (turtle)

図 5.5 のログからホスト turtle がキャッシュサーバ spring,hiro から帯域情報を受け取ったことがわかる。

5.2.8 帯域情報を元に適切なメディアタイプの選択

今回の実験では、計測した帯域情報のみを用いて、メディアタイプの選択を行った。クライアントがメタファイルを参照することで、帯域の優位を判別する CGI が起動する仕組みになっている。その CGI がどのメディアタイプを選択するべきかを決定し、クライアントに対してメタファイルの内容を返す。

今回の実験では、キャッシュサーバに Real Media の Real Server を起動させ、Real Media のメディアタイプの選択を行った。用意したメディアタイプは 64Kbps 用、128Kbps 用、192Kbps 用、1Mbps 用の 4 つである。

この CGI には、ポリシーサーバがホスト hiro,spring から受け取った帯域情報を比較し、クライアント・キャッシュサーバ間の転送性能が良いキャッシュサーバを選択するポリシーを記述した。

また、クライアント・キャッシュサーバ間の帯域情報を基にメディアタイプ選択に関する以下のポリシーを記述した。

- 0-64Kbps の範囲なら 64Kbps 用のメディアタイプを選択
- 64Kbps より大きく 128Kbps 以下なら 128Kbps 用のメディアタイプを選択
- 128Kbps より大きく 192Kbps 以下なら 192Kbps 用のメディアタイプを選択
- 192Kbps より大きい場合、1Mbps 用のメディアタイプを選択

図 5.5 の実験結果を参照すると、クライアント・キャッシュサーバ間の帯域情報から、client・hiro 間の転送性能がより良好であることがわかる。よって hiro から動的コンテンツをダウンロードすることが決定される。次に、上記のメディアタイプ選択のポリシー記述から、クライアント・キャッシュサーバ (hiro) 間の帯域情報は約 3Mbps なので、1Mbps 用のメディアタイプが選択される。hiro で起動している Real Server のアクセスログを図 5.6 に示す。

```
203.178.143.233 - - [15/Jan/2002:00:53:50 +0900]
"GET real8video1M.rm RTSPT/1.0" 200 494602
[WinNT\_5.0\_6.0.9.380\_play32\_NS8J\_ja\_686]
[02cb3f21-0901-11d6-ef4e-b93a181bc2e6]
[Stat1:      220      0      0      0
      0 20\_Kbps\_Stereo\_Music\_High\_-\_RA8]
[Stat2:      20672      0      0      0      0      0
      0      0      0      1      50
20\_Kbps\_Stereo\_Music\_High\_-\_RA8] 1420952 52 68 0 0 15
```

図 5.6: Real Server へのアクセス (hiro)

図 5.6 からクライアントは hiro のキャッシュサーバを選択し、1Mbps 用のメディアタイプをダウンロードしたことがわかる。以上の操作で、クライアントはデータ転送時間を利用し、適切なキャッシュサーバとメディアタイプを選択することができたことが証明される。

以上のことを基に、本システムは利用者に CDN の存在や回線帯域を意識させることなく適切なコンテンツ配信をすることができる。

5.3 サーバが用意すべきメディアタイプの提案

2001年11月3日11:40から2001年12月28日7時16分までの約2ヶ月間、慶應義塾大学環境情報学部村井研究室のWWWサーバであるwww.sfc.wide.ad.jpのトップページへアクセスのあったクライアントに対してデータ転送時間を測定した。そのデータを分析することで、配信するオリジナルサーバがどの種類の帯域用のデータを保持すればよいか提案する。総アクセス数は24955回であった。図5.7にwww.sfc.wide.ad.jpのトップページにアクセスしたクライアント・WWWサーバ間の全ての帯域情報に関する累積度数分布と度数分布を示す。前提として、このデータはクライアント・WWWサーバ間の一般的な帯域情報を調べるために、sfc.wide.ad.jpドメインとkeio.ac.jpドメインからのアクセスは削除した。また、複数回アクセスしたクライアントの帯域情報は平均を取ることによって一つのデータとした。図5.8にあるように、転送性能を50000bpsの単位で区切り、度数分布を調べることで、どの帯域が一番頻度が高いかを調べた。

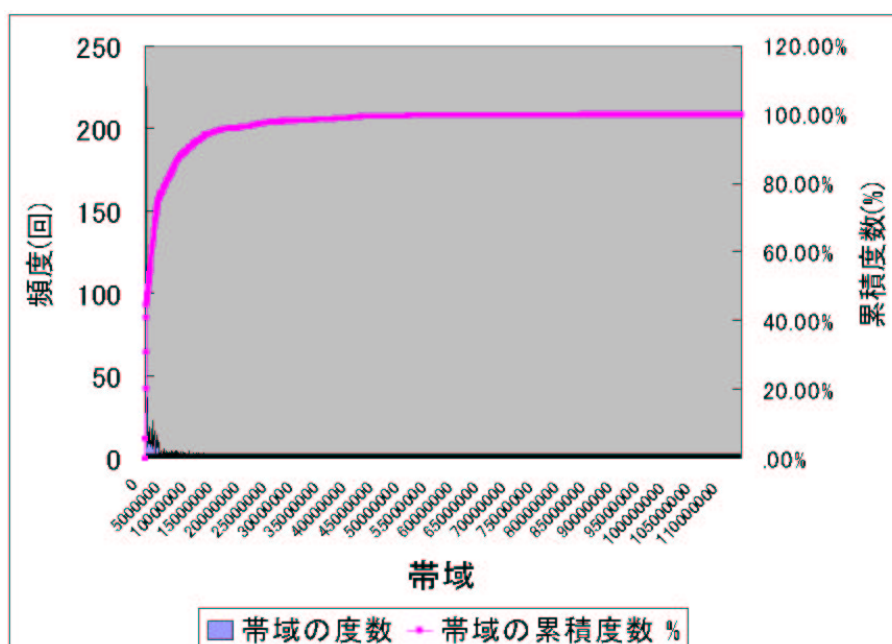


図 5.7: www.sfc.wide.ad.jp にアクセスした全てのクライアントから計測された帯域情報の累積度数分布と度数分布

図 5.7 を見る限り、0bps から 500000bps の間に約 50%のクライアントが存在しているのが推測できる。より詳しく見るために、図 5.8 に 0bps から 500000bps までの情報を 10000bps の単位で区切ったグラフを示す。

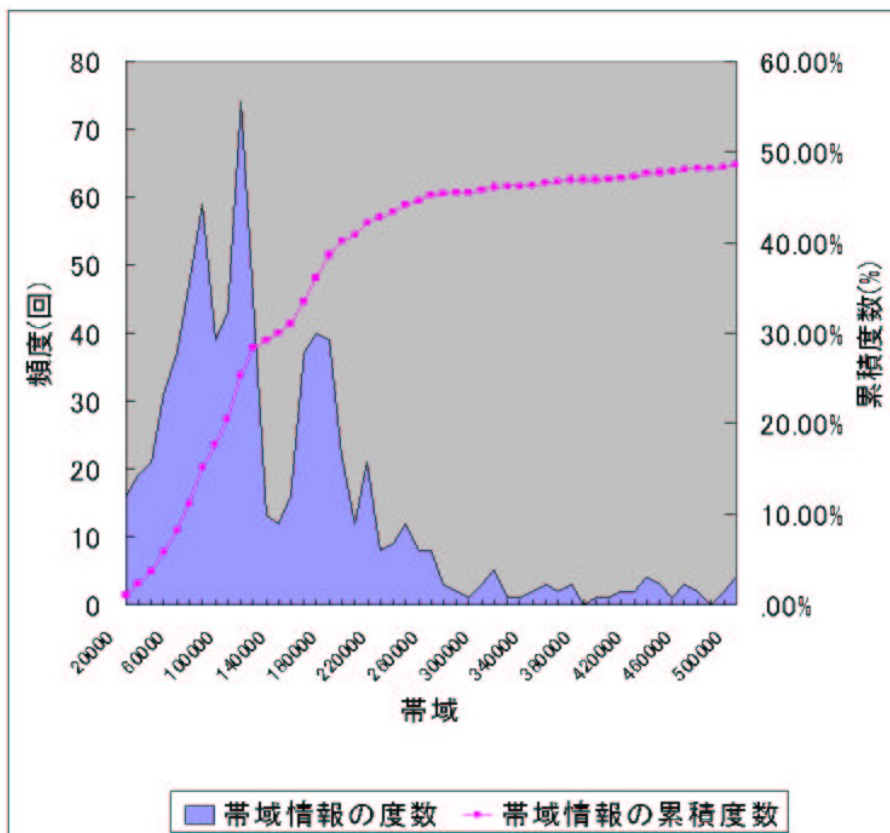


図 5.8: www.sfc.wide.ad.jp にアクセスした中の約 50% のクライアントから計測された帯域情報の累積度数分布と度数分布

図 5.8 の度数分布グラフから、128Kbps,192Kbps,64Kbps あたりへのアクセスが集中していることがわかる。また、0Kbps から 500000bps の間に全体のクライアントの 50% が集まっている。配信するサーバには、多くの利用者に再生可能にするために、これらの帯域用のメディアタイプを作成することが重要である。

また、詳しく見るために、全てのデータを 64Kbps で区切り、アクセストップ 9 を算出した。10 位はエントリーされる数が多いので、省略した。表 5.4 にクライアントの帯域状態の多い順番のリストを示す。

表 5.4: 64Kbps で区切った時のクライアントの度数

1 位	128Kbps	301 クライアント
2 位	198Kbps	176 クライアント
3 位	64Kbps	143 クライアント
4 位	256Kbps	70 クライアント
5 位	1.125Mbps	22 クライアント
5 位	1.68Mbps	22 クライアント
7 位	640Kbps	21 クライアント
8 位	1.06Mbps	19 クライアント
9 位	704Kbps	18 クライアント

これらの情報から、逆に言うと、今現在の回線のトレンドに非常に一致する。つまり、データ転送時間を測定し、ある一瞬の実効帯域を算出する手法はある程度の有効性がある。

第6章 結論

本章では、本研究の結論として、本研究の成果と今後の課題について述べる。

6.1 本研究の成果

本研究では、映像配信を行う際に重要な問題点である、アクセス頻度の高いサーバへのアクセスの負荷分散、トラフィックの負荷分散、そしてクライアントに適した品質の動的コンテンツを適応的に配信する技術について論じた。特に、サーバ負荷分散技術の一つである CDN を映像配信に利用し、映像配信をする際のキャッシュサーバ選択へ新しい指標を提案した。

現在、Akamai などの CDN のサービスを推進している企業の多くは、利用者の IP アドレスを参照し、BGP4 の持つ AS_Path 情報、キャッシュサーバの負荷情報を利用し、DNS の A レコードを書き換えることでキャッシュサーバの選択を行なっている。しかし、それらの情報を用いたキャッシュサーバ選択だけではクライアントが最適なキャッシュサーバを選択できていない。キャッシュサーバ選択にはトポロジー情報だけでなく、実効帯域を考慮に入れる必要性があった。

本研究では、動的コンテンツを再生するには必ず WWW サーバにアクセスし、画像やテキスト情報をダウンロードするというコンテンツ配信の特性を利用し、不必要なトラフィックをネットワーク上に流さずにクライアント・キャッシュサーバ間の実効帯域を計測する手法を実現した。それは、HTTP によるデータ転送時間によって計算できる実効帯域をキャッシュサーバ選択の指標として用いることで実現した。そして、クライアントが動的コンテンツをダウンロードする直前の実効帯域をキャッシュサーバ選択に反映させるため、DNS のように情報伝播に時間がかかる方法ではなく、メタファイルをクライアントに参照させる手法を用い、クライアントをキャッシュサーバに誘導させる仕組みを実現した。

また、動的コンテンツ配信において、アプリケーション、メディアタイプ（例 RealMedia 300Kbps 用等）を利用者の直感というあいまいな手法で選択していたが、キャッシュサーバ選択のために計測したデータ転送時間によって求められる実効帯域を指標に用いることで、利用者の直感というあいまいな指標ではなく、定量的な指標を用いてクライアントはアプリケーション、メディアタイプを選択することができた。

6.2 今後の課題

今後の課題として、以下の三つが挙げられる。

- 統計的手法を用いたトラフィック変動の予測
- 帯域情報をサブネットレベルで集約
- キャッシュサーバ選択のための他の指標との協調

本節では、それぞれの課題について説明する。

6.2.1 統計的手法を用いたトラフィック変動の予測

本研究では、平均と標準偏差を求めているが、それだけでは十分ではない。真壁論文 [35] にあるように、最小二乗法を用いたトラフィック変動のベクトルを求めることで、ある程度の帯域変動を予測することが可能になる。このような統計的手法を用いて、キャッシュサーバを選択することで、長時間回線を流れる連続メディアに対して効果的である。

6.2.2 帯域情報をサブネットレベルで集約

本研究の中でも触れたが、帯域情報を推測するために、帯域情報データベースにサブネットレベルでの平均、標準偏差を持つことで、初めてアクセスしたクライアントに対しても何らかの指標を提供することができる。

6.2.3 他の指標と協調

- キャッシュサーバの負荷状況

帯域情報を利用してキャッシュサーバを選択したとしても、キャッシュサーバの処理能力を超えていたとしたら、クライアントは選択された品質の動的コンテンツを再生することができない。帯域情報と共に、キャッシュサーバの負荷状況である、ネットワーク負荷状況、CPU 使用率、クライアントのコネクション数をサーバ選択に反映させることが必要である。

- 経路情報

本研究では、ネットワーク地図を BGP4 プログラムの持つ AS_PATH 情報を用いた。しかし、AS 内部にも大量のキャッシュサーバがあった場合、クライアントにとってストレスを感じさせる量の画像をダウンロードさせるべきではない。そのような場合に、OSPF や RIP のような経路制御機構が持っている AS 内部の経路情報を利用することで AS 内部においてのネットワーク位置に近いキャッシュサーバを選択することができる。

謝辞

本研究を行うにあたり、多くの方のご協力とご指導を賜りました。お世話になった全ての方に心より感謝いたします。

まず、学部時代から引き続き辛抱強く指導していただいた慶應義塾大学環境情報学部教授 村井純博士に感謝いたします。

本研究の過程において、数々のご助言、励ましのお言葉をいただきました、慶應義塾大学環境情報学部助教授の中村修博士、同学部助教授の楠本博之博士に感謝いたします。そして本研究のアイデアについて数多くの相談にのって頂いた、慶應義塾大学環境情報学部専任講師の南政樹氏に感謝致します。

また、快く副査を引き受けてくださった、東京大学大学院情報理工学系研究科助教授 江崎浩博士に感謝致します。

他にも、慶應義塾大学環境情報学部、徳田・村井・楠本・中村・南研究室の皆様、WIDE プロジェクトの皆様、その他数え切れない程多くの皆様のご協力を賜りました。ここに重ねて感謝の意を表させていただきます。

最後に本研究を進めて行くにあたり、精神的に支えてくださった重近範行氏、杉浦一徳氏、土本康生氏、内山映子女史、宮川祥子女史、石橋啓一郎氏、小川晃通氏、石田剛朗氏、折田明子女史、小原泰弘氏、中前周氏、川喜田佑介氏、藤枝俊輔氏、室井比宏氏、江木啓訓氏、林亮氏、牧兼充氏、長橋賢吾氏、海崎良氏、菅沢延彦氏、小川浩司氏、豊野剛氏、小嶋元氏、丹野久志氏、大学院棟 T20 在住の皆様、友人達、そして両親と妹に感謝致します。

2002 年 1 月

廣石透

参考文献

- [1] K. Claffy, G. Miller, and K. Thompson. "The nature of the beast: recent traffic measurements from an Internet backbone". *INET98*, 1998.
- [2] Sean McCreary and K. Claffy. "Trends in Wide Area IP Traffic Patterns". *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, September 2000.
- [3] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext transfer protocol – HTTP/1.0. Request for Comments 1945, Internet Engineering Task Force, May 1996.
- [4] P. V. Mockapetris. Domain names: Concepts and facilities. Request for Comments 882, Internet Engineering Task Force, November 1983.
- [5] Inc. Akamai Technologies. <http://www.akamai.com/>.
- [6] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an autonomous system (AS). Request for Comments 1930, Internet Engineering Task Force, March 1996.
- [7] 唐津登志夫, 高田敏弘, 栗原聡, 菅原俊治. "転送履歴を利用した url resolver における最適経路選択". 第一回インターネットテクノロジーワークショップ (WIT'98), 1998.
- [8] K.L.Johnson, J.F.Carr, M.S.Day, M.F.Kaashoek. "The Measured Performance of Content Distribution Networks". *5th International Web Caching and Content Delivery Workshop*, May 2000.
- [9] Microsoft Corporation. "Windows Media Technologies". available at <http://www.microsoft.com/windows/windowsmedia/default.asp>.
- [10] Real Networks, Inc. "Real Player". available at <http://www.real.com/player/index.html>.
- [11] 総務省. "DSL普及状況公開ページ". http://www.soumu.go.jp/joho_tsusin/whatsnew/dsl/.
- [12] H.Zimmermann. "OSI Reference model - The ISO Model of Architecture for Open Systems Interconnection". *IEEE Transactions on Communications*, April 1980.
- [13] Andrew Odlyzko. "internet growth: Myth and reality, use and abuse". *iMP: Information Impacts Magazine*, November 2000.
- [14] M. Day, B. Cain, G. Tomlinson, and P. Rzewski. A model for content internetworkin(CDI). Internet Draft, Internet Engineering Task Force, November 2001. Work in progress.
- [15] M. Green, B. Cain, G. Tomlinson, S. Thomas, and P. Rzewski. CDN peering architectural overview. Internet Draft, Internet Engineering Task Force, March 2001. Work in progress.

- [16] D. Gilletti, R. Nair, and J. Scharber. Accounting models for CDN peering. Internet Draft, Internet Engineering Task Force, October 2000. Work in progress.
- [17] D. Gilletti, R. Nair, and J. Scharber. CDN peering authentication, authorization, and accounting requirements. Internet Draft, Internet Engineering Task Force, November 2000. Work in progress.
- [18] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck. Known CDN request-routing mechanisms. Internet Draft, Internet Engineering Task Force, November 2001. Work in progress.
- [19] 下川俊彦, 吉田紀彦, 牛島和夫. "多様な選択ポリシーを利用可能なサーバ選択機構". 電子情報通信学会論文誌, September 2001.
- [20] D.Andresen, T.Yang, V.Holmedahl, and O.Ibarra. "sweb:towards a scalable www server on multicomputers". *Proc. 10th International Parallel Processing Symposium*, April 1996.
- [21] V.Cardellini, M.Colajanni, and P.S.Yu. "dns dispatching algorithms with state estimators for scalable web-server clusters". *World Wide Web Journal*, Baltzer Science, July 1999.
- [22] C.Yoshikawa, B.Chun, P.Eastham, A.Vahdat, T.Anderson, and D.Culler. "using smart clients to build scalable services". *USENIX '97*, January 1997.
- [23] 重近範行, 中村修, 笹川信義, 村井純. "長野オリンピックのネットワークと情報提供システム". 情報処理, Febraury 1998.
- [24] Cisco Systems, Inc. "distributeddirector". available at <http://www.cisco.com/warp/public/cc/pd/cxsr/dd>.
- [25] Y. Rekhter and T. Li. A border gateway protocol 4 (BGP-4). Request for Comments 1771, Internet Engineering Task Force, March 1995.
- [26] V.Jacobson. "traceroute". available at <ftp://ftp.ee.lbl.gov/traceroute.tar.Z>.
- [27] H.Kamiya, K.Ohta, N.Kato, G.Mansfield, , Y.Nemoto. "An Improved Content Search Engine - Usage of Network Configuration Informatin-". *Proc. IEEE TENCON '98*, December 1998.
- [28] Real Networks, Inc. Surestream とは. available at <http://service.jp.real.com/help/faq/surestream.html>.
- [29] Microsoft Corporation. インテリジェント ストリーム. available at <http://www.microsoft.com/japan/windows/windowsmedia/features/intellistream/default.asp>.
- [30] Generic Media, Inc. "Gfeneric Media Publishing Service". <http://www.genericmedia.com/pdf/GMcharts.pdf>.
- [31] "tcpdump". available at <http://www.tcpdump.org>.
- [32] V.Jacobson. "Pathchar". available at <ftp://ftp.ee.lbl.gov/pathchar>.
- [33] Keshav.S. "A Control-Theoretic Approach to Flow Control". *Proceedings of ACM SIGCOMM '91*, September 1991.
- [34] "netperf". available at <http://www.netperf.org>.

- [35] 真壁知, 太田耕平, 加藤寧, Glenn MANSFIELD, 根元義章. "ネットワークの負荷変動を考慮した動的なミラーサーバ選択方式". 電子情報通信学会論文誌, March 2001.
- [36] Zona Research, Inc. "the need for speed", 1999.

付録A DSL普及率

図 A.1 に参考文献 [11] にある、DSL 加入者の推移を示す。

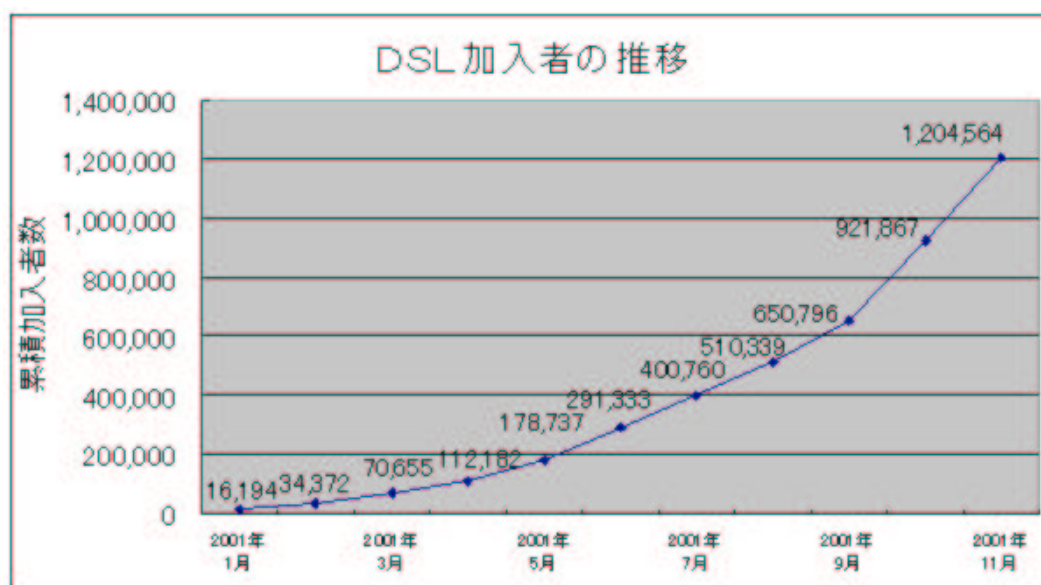


図 A.1: DLS 加入者の推移

付録B インターネットバックボーンにおけるトラフィックの傾向

参考文献 [2] にあるトラフィックの傾向を示す。

表 B.1 に参考文献 [2] にある TCP トラフィックの Top25 を示す。

表 B.1: Top 25 TCP application categories seen during February, 2000

Protocol	Source	Destination	Packets	Bytes	Average Size
HTTP	80	0	140780543	100044030753	710
	0	0	45319842	17319763013	382
NNTP	0	119	17895481	15992967942	893
HTTP	0	80	94578965	7844163850	82
FTP Data	20	0	6728097	6689611587	994
SMTP	0	25	8878925	6071084052	683
NNTP	119	0	8857217	5399672480	609
HTTP/Web Proxy	8080	0	2331669	2327032104	998
Napster	0	6699	3331109	1838804438	552
HTTPS	443	0	3035809	1535037132	505
Napster	6699	0	3377828	1528188686	452
FTP Data	0	20	5498097	1262294037	229
Napster	6688	0	1230335	935883810	760
	1755	0	1182358	908626624	768
POP3	110	0	1820887	798255125	438
Hotline	5501	0	685536	787122008	1148
RTSP	554	0	754508	616087123	816
Napster	0	6688	1149382	348845629	303
SMTP	25	0	6438672	339788020	52
RealAudio	7070	0	445551	298598442	670
Shoutcast	8000	0	353545	296161291	837
Web Cache	3128	0	325447	280739543	862
HTTPS	0	443	2635048	280418901	106
NetBIOS SSN	139	0	312658	264965212	847
	2189	0	294654	174140223	590

表 B.2 に参考文献 [2] にある UDP トラフィックの Top25 を示す。

表 B.2: Top 25 UDP application categories seen during February, 2000

Protocol	Source	Destination	Packets	Bytes	Average Size
	0	0	15108822	2568130721	169
RealAudio	0	7070	4610070	2029483625	440
DNS	53	53	9290872	1064980650	114
DNS	53	0	3444558	638796849	185
Half Life	27015	27005	2199098	452384485	205
DNS	0	53	5286554	332598249	62
	0	6770	619334	230280312	371
Starcraft	6112	6112	4167625	217783755	52
EverQuest	9001	9000	908432	171755388	189
Half Life	27005	27015	2754416	160806176	58
RealAudio	6970	0	532356	154663054	290
Unreal	7777	0	1109005	141327485	127
EverQuest	9005	9000	777485	138637166	178
Unreal	0	7777	1892613	107613279	56
Quake 3: Arena	27960	27960	784258	80146922	102
Half Life	27015	0	424614	77756002	183
Quake II	27901	27910	946674	58842583	62
Half Life	0	27005	317735	58062692	182
	0	22	50337	45995046	913
	28001	0	255081	44789873	175
NetBIOS NS	137	137	509854	43996075	86
Quake II	27910	27901	232865	43942438	188
	0	371	46531	40574996	871
CU-SeeMe	7648	7648	136843	39681727	289
Quake II	27901	0	601427	37327785	62
Half Life	27005	0	498300	29191062	58

付録C 予備実験

予備実験として HTTP によるデータ転送を利用した帯域測定の精度の検証を行った。以下に実験結果と考察を述べる。

C.1 実験概要

2 台の PC を用意し、一台の PC で本研究で作成したデータ転送時間計測用の HTTP デーモンを起動し、もう一台の PC がクライアントとして HTTP デーモンから画像を取得する環境を構築した。その際、HTTP デーモンの起動しているホストで dummynet を利用してトラフィックシェーピングを行った。28.8Kbps,64Kbps,128Kbps,300Kbps,500Kbps,1Mbps,10Mbps,100Mbps といった 8 つの場合を想定し、個々の帯域に対して HTTP のデータ転送による帯域測定の精度を計測した。計測を行うために、10Kbyte,20Kbyte,30Kbyte,50Kbyte,100Kbyte,200Kbyte,300Kbyte,500Kbyte,1Mbyte といった、9 つのサイズの画像を利用し、それぞれの画像サイズに対し 5 回づつ計測を行った。

C.2 実験結果

28.8Kbps に設定した場合

表 C.1 にクライアント・サーバ間を 28.8Kbps に設定した際に計測された帯域を示す。

表 C.1: クライアント・サーバ間を 28.8Kbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	325434599	333887445	339770925	338280701	333887445
20Kbyte	31696	34572	34424	34498	34572
30Kbyte	41899	43034	43034	50618	50619
50Kbyte	17233	20973	20995	21149	20567
100Kbyte	22148	22973	22300	22503	23675
200Kbyte	24663	23989	28115	27692	28268
300Kbyte	28237	28254	28586	28124	28424
500Kbyte	27865	27929	27479	27613	27375
1Mbyte	27616	27196	27560	27245	27596

64Kbps に設定した場合

表 C.2 にクライアント・サーバ間を 64Kbps に設定した際に計測された帯域を示す。

表 C.2: クライアント・サーバ間を 64Kbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	332448275	341274336	331021459	335339130	331021459
20Kbyte	76968	76599	76969	76602	76967
30Kbyte	112474	113007	112476	111426	113006
50Kbyte	65901	65901	65686	66008	65686
100Kbyte	67308	67308	67252	66320	67086
200Kbyte	62356	62072	62095	61977	61766
300Kbyte	63597	62495	63498	63416	63236
500Kbyte	61849	61580	61202	61386	61719
1Mbyte	61312	61097	61363	61239	61253

128Kbps に設定した場合

表 C.3 にクライアント・サーバ間を 128Kbps に設定した際に計測された帯域を示す。

表 C.3: クライアント・サーバ間を 128Kbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	333887445	332448275	335339130	335339130	331021459
20Kbyte	152509	153969	153972	153969	153979
30Kbyte	225000	227139	227145	227148	227128
50Kbyte	132894	146383	132897	132897	146381
100Kbyte	131468	131045	132973	133408	131045
200Kbyte	127541	128647	128748	128748	128748
300Kbyte	126933	124547	126150	125183	124610
500Kbyte	124812	124547	124660	124661	124282
1Mbyte	123142	123087	123124	123105	123346

300Kbps に設定した場合

表 C.4 にクライアント・サーバ間を 300Kbps に設定した際に計測された帯域を示す。

表 C.4: クライアント・サーバ間を 300Kbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	335339130	333887445	333887445	333887445	338280701
20Kbyte	357185	364180	364170	364171	364185
30Kbyte	531886	531701	530331	542195	531978
50Kbyte	323272	310849	342474	310999	342476
100Kbyte	322859	321919	323092	324448	307210
200Kbyte	299014	299014	301778	299101	299562
300Kbyte	296450	296073	293225	293203	293203
500Kbyte	291495	290852	294196	289630	291889
1Mbyte	288784	288796	289525	290114	289090

500Kbps に設定した場合

表 C.5 にクライアント・サーバ間を 500Kbps に設定した際に計測された帯域を示す。

表 C.5: クライアント・サーバ間を 500Kbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	326813559	331021459	316098360	328204255	331021459
20Kbyte	612645	616812	595437	606318	616800
30Kbyte	887267	884614	918064	896416	887588
50Kbyte	518213	574459	577409	518642	569916
100Kbyte	515169	490354	514495	513000	513713
200Kbyte	488836	487752	488835	488072	488839
300Kbyte	495382	488298	481588	490255	501972
500Kbyte	480457	487530	492252	483967	487528
1Mbyte	484651	479453	482844	481692	483984

1Mbps に設定した場合

表 C.6 にクライアント・サーバ間を 1Mbps に設定した際に計測された帯域を示す。

表 C.6: クライアント・サーバ間を 1Mbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	322711297	333887445	288868913	329606837	333887445
20Kbyte	1212729	1235972	1242150	1196886	1189459
30Kbyte	1897995	1838437	1864899	1788937	1838465
50Kbyte	1160258	1145833	1160528	1160845	1121686
100Kbyte	1030774	1030620	988600	1032091	1027783
200Kbyte	1010444	1013989	992410	1026910	1033474
300Kbyte	994943	1005241	1004658	993616	1000066
500Kbyte	966577	978511	970759	975860	976086
1Mbyte	959376	968166	963069	967420	963383

10Mbps に設定した場合

表 C.7 にクライアント・サーバ間を 10Mbps に設定した際に計測された帯域を示す。

表 C.7: クライアント・サーバ間を 10Mbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	335339130	306063492	280465454	263235494	246415335
20Kbyte	14902793	10080997	14969696	14913902	14832360
30Kbyte	9312187	11905313	10164236	15454910	12617338
50Kbyte	12596875	12440146	8874964	9125584	10110479
100Kbyte	3630659	8055675	3879588	3281696	4084753
200Kbyte	5032345	9951525	4030018	9626483	7035706
300Kbyte	5606327	5344584	6041120	5928621	6601019
500Kbyte	6692846	7004214	6021913	8179525	6152992
1Mbyte	7773241	7445706	7467732	7314041	7819085

100Mbps に設定した場合

表 C.8 にクライアント・サーバ間を 100Mbps に設定した際に計測された帯域を示す。

表 C.8: クライアント・サーバ間を 100Mbps に設定した際に計測された帯域

画像サイズ	一回目 (bps)	二回目 (bps)	三回目 (bps)	四回目 (bps)	五回目 (bps)
10Kbyte	258818791	242540880	239527950	241780564	243305993
20Kbyte	14883392	14891700	14839236	14833734	14901405
30Kbyte	21759036	21727305	21753079	19086629	21650349
50Kbyte	10583870	11700043	11771302	11744948	10709297
100Kbyte	3662062	10853865	10947759	9608974	9340082
200Kbyte	4635906	5259151	9348000	10278403	9512668
300Kbyte	10320269	4555373	5344981	9253173	9960921
500Kbyte	9809097	4407682	7062571	4602702	9695693
1Mbyte	5110060	7053214	6572064	6792730	6786737

C.3 考察

実験結果から、3 つのことがわかる。

1. 画像サイズが 10Kbyte では帯域測定が正確に行われない

小さい画像サイズを利用すると、データ転送時間が非常に小さくなるので、計測誤差の影響を大きく受けてしまう。

2. 64Kbps から 1Mbps の帯域測定では、50Kbyte 以上の画像サイズを用いた場合に約 10% から 20% 程度の誤差で計測することができる

3. 10Mbps, 100Mbps の帯域を計測する場合、正確に計測できていない

これは HTTP が TCP を利用してデータを転送しているため、TCP のウィンドウサイズが最大になる前に画像転送が終了してしまい、実際の帯域より狭く計算されてしまっていると予想できる。このようなことを防ぐためには、あらかじめ帯域の太さをある程度予測し、その帯域にあわせた画像サイズを計測に用いることが重要である。

これらのことから、50Kbyte 以上の画像サイズを利用することが有効であることがわかる。また、10Mbps や 100Mbps といった太い帯域幅の場合には、より大きな画像を転送して計測するか、あるいは、別の手法が必要である。