

第 XXVIII 部

ストリーム配信技術

第 28 部

ストリーム配信技術

第 1 章 2004 年度 Streaming WG 活動内容まとめ

Streaming WG は、2003 年度の WIDE 研究会の Streaming BoF を前身とし、2004 年 4 月から WG として設立され活動を行っている。WG では、インターネット上でのストリーム配信を行う際の運用ノウハウや配信技術について、実装・実験を通じて議論を行うことを目的にしている。昨年度の BoF ベースの活動も含め、今年度の WG としての活動は以下の通りであった。

1. ストリームアプリケーションにおけるレート変更タイミング決定アルゴリズムの提案
2. 2004 年度における Comet Regulator の研究
3. ストリーム平滑化装置のラボ実験およびフィールド実験による評価
4. 2004 年度秋の WIDE 研究会での実験：衛星回線への Comet TCP の適用

1. では、ストリーム配信中のネットワークの状態の変化に適応し、再生レートを変更する際にバッファの枯渇による再生停止を防止するための手法について議論を行った。受信バッファ残量が枯渇する原因の 1 つには、レート変更のリクエストタイミングが遅いため受信データ量よりも再生による消費データ量が上回ることが挙げられる。本研究では、ストリーミングアプリケーションのバッファ残量に着目し、再生が中断しないストリーム配信をより少ないレート変更回数で実現することを目標とした。パケット到着間隔とバッファ残量の減少レベルから、バッファの枯渇タイミングを予測し、適切なレート変更リクエスト送信タイミングの算出を行うことで、従来方式と比べ、より早いタイミングでバッファ枯渇を予測しレート変更リクエストを行なうことにより、クライアントアプリケーション側で再生を継続したままレート切り替えを実現する手法を提案した。次年度以降は、本方式をクライアントアプリケーションに実装し、既存のアプリケーションと比較した場合

に再生停止時間をどの程度短縮できるかという評価を行う予定である。

2. では、バースト性の高い動画ストリームが、途中の回線やルータなどでパケットロスを起こし画像が再生できないような状況を回避するために、個々のストリームごとに帯域を平滑化するようなシステムを、Comet ハードウェアを利用して開発した。この Comet Regulator では、ストリームのセッションの識別に IP アドレスを使用していたが、2004 年度の実装では、IP アドレスに加えポート番号でも識別を行うようにすることで、より適用範囲の広いシステムとなった。

3. では、Comet Regulator を、ラボ(実験室)で評価を行った結果と、インターネットでのライブ中継に適用した結果を報告した。ラボ実験では、Windows Media 9(WM9)のサーバを使用して、VoD 映像およびライブ映像について、Comet Regulator の有無でパケットのバースト性の評価を行った。WM9 サーバからの平均数百 kbps の映像ストリーム(UDP)には、100 Mbps ものバースト性のパケットが含まれるが、Regulator を通すことで平均送信レートに近いパケット間隔になることが確認できた。また、クライアントとサーバの IP アドレスごとにストリームのセッションが認識され、それぞれ独立して平滑化されることも確認できた。インターネットでのフィールド実験は、2003 年 11 月に行われた南極皆既日食インターネットライブ中継の配信システムに Comet Regulator を組み込み、ストリームデータの帯域平滑化の有用性を実際のインターネット上で実験した。結果として、Comet Regulator を介することによって、パケットが間隔をおいてインターネットに送出され、バースト的な帯域使用が抑えられることが確認できた。ただし、350 kbps という比較的低いビットレートということが影響したのか、クライアントの状態・喪失パケット数・映像品質には大きな差が観測できなかった。

4. では、2004 年度秋の WIDE 研究会の会場を利用して、TCP の長距離高速回線での高速化に効果が得られている Comet TCP の有用性の実験を行った。Comet TCP で当初想定していた数百 Mbps という

高速な長距離回線ではなく、数 Mbps ~ 数 10 Mbps という低速な衛星回線においても、TCP に比較して有用であることを実証することを目的とした。TCP のベンチマークアプリケーションを使用した定量的実験では CometTCP の効果は認められた。しかし、大人数での Web 閲覧では、利用者の主観的な観測での有意な効果は見られないという結果になった。

第 2 章 ストリームアプリケーションにおける再生レート変更タイミング決定アルゴリズムの提案

2.1 背景と研究目標

インターネットにおけるストリーミング配信では、動画データを複数のビットレートでエンコードし、配信ネットワークの状態に合わせ適正レートを選択、配信するマルチビットレート方式が広く用いられている。この技術により、視聴者はレートを自ら選択することなく、配信ネットワークの状態に適したレートでストリームを受信できるようになった。しかし、クライアントがストリームを受信している途中で受信レートが切り替わる場合に再生が中断するという現象が見られる。この現象は、ストリーミングアプリケーションが持つ受信バッファ残量がなくなることにより引き起こされる。受信バッファ残量が枯渇する原因の 1 つには、レート変更のリクエストタイミングが遅いために受信データ量よりも再生による消費データ量が上回ることが挙げられる。この問題は、レート変更リクエストをできるだけ早くサーバへ送ることにより解決するよう見えるが、頻繁にレートを変更することによるサーバ負荷増大といった問題や、動画像のレート切り替えによる視聴感への悪影響といった問題がある。そこで本研究では、ストリーミングアプリケーションのバッファ残量に着目し、再生が中断しないストリーミング配信をより少ないレート変更回数で実現することを目標とする。

2.2 既存研究

クライアントの受信状態把握に関する既存研究には、視聴者の体感視聴品質をリアルタイムで収集し、その結果に基づいてクライアントでの受信状態を向上させるもの [365, 368] がある。この研究では得られた視聴品質情報をサーバの負荷分散に利用するこ

とで、クライアントの受信状態を改善する手法が提案されている。この手法はユーザからの視聴品質情報を常にサーバ側に送信して、その情報の推移からユーザの視聴状態を決定する。提案されている手法により、ユーザの視聴品質を時系列的に収集し、品質が悪化した場合は、受信状態を変更することで品質を改善することが可能である。しかし、サーバが収集したデータを元に配信状態の変更をするため、クライアントの受信状態が悪化した後にしか、配信状態を変更することができない。このため、再生状態などの情報から受信状態の悪化を予測するアルゴリズムが必要となる。

受信状態の悪化を予測する研究としては、受信バッファにパケットが到着しない期間から再バッファリングが始まるタイミングを関連付けた研究 [51] がある。この手法によって、パケットの受信状態から再バッファリングの開始を予測することが可能である。しかし、現在の再生レートを再バッファリングの予測に反映し、再生レートの動的な変更をするについては言及されていない。また、現在のストリーミングサービスの中にはパケットをパース的に送信する特性を持った実装形態が存在する。したがって、パケットがパース的に送信されていることを考慮して予測アルゴリズムを改良する必要がある。

2.3 提案手法の概要

前述した通り本研究では受信バッファの枯渇を予測し、適切なタイミングで再生レートを変更することでストリーミング配信の再生が途切れることを防止する。

2.3.1 再生レートを下げる判断基準

前述の通り、受信バッファの残量が減少し、かつ、受信バッファの残量が回復する見込みがないと判断された場合は再生レートを切り替える。以下はその場合の判断基準である。

図 2.1 に、提案手法において受信バッファの残量が再生レートを切り替える前後にどのように変化するかを示す。ここで、図中の B_{\min} は再生中に再生レートを切り替えても、再生レートの切り替え中に再生が停止しない、受信バッファの残量のしきい値とする。したがって、受信バッファの残量が B_{\min} になる前に、受信バッファの枯渇が予想された場合は、その時点で再生レートを切り替える必要がある。す

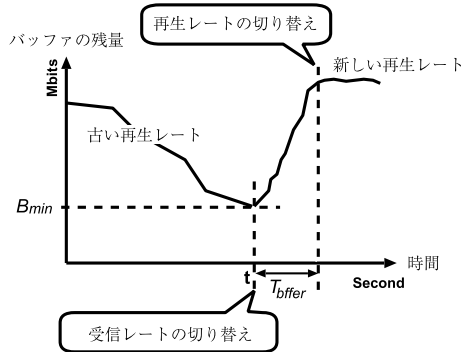


図 2.1. 受信バッファの残量 (再生レート切り替え前後)

なわち、再生を途切れさせることなく再生レートを切り替えるためには、受信バッファが B_{min} になる時刻 t になるまでに受信バッファの枯渇を予測しなければならない。

ここで、再生が途切れずに、再生レートを切り替えるためには、切り替え後のレートで再生が可能になるまで、切り替え前の再生レート P Mbps で再生を継続しなければならない。したがって、再生レートの切り替え開始から再生が開始するまでに要する時間を T_{buffer} 秒とすると、 B_{min} は以下の式を満たす必要がある。

$$B_{min} > P * T_{buffer} \quad (23)$$

また、閾値 B_{min} を決定する際には、前述した受信パケットが到達していない期間と受信バッファの枯渇の相関関係から導かれた値についても考慮する必要がある。

- 受信バッファの残量が減少傾向にある
RTCP の RR パケットに含まれている積算パケット損失率から受信レートを算出し、現在の再生レートと比較することで、受信バッファの残量の状態を推測する。
ある時の受信レートが R Mbps である場合、その時の受信バッファの減少レート A Mbps は以下の式で表すことができる。

$$A = R - P < 0 \quad (24)$$

- 受信バッファの枯渇のしきい値
前述の研究 [51] では、受信バッファの枯渇をパケットの受信の途切れた期間を元にして予測していた。本研究では再生レートやバースト的に送信されるパケットについても考慮する必要がある。

ある。その際に、上記のパケットの受信が途切れた期間も予測のために有用な情報である。このため、Windows Media Technology 9 (以下 WMT9) において、受信パケットが途切れた期間と受信バッファの枯渇の相関関係について検証する必要がある。

- サーバから送信されるパケットがバースト的でない
WMT9 ではサーバがバースト的にパケットを送信しているが、受信バッファの枯渇の予測をする場合、このバースト的に送られてくるパケットを無視することができない。したがって、バッファの残量が少なくなってきた場合でも、サーバがバースト的にパケットを送信し、再生レートを切り替える必要がないと判断できる場合は再生レートを切り替えるべきではない。

以上の条件を満たした場合にクライアントはサーバに対し再生レートを下げるための手続きを開始する。

2.3.2 再生レートを切り替える手法

ここでは X Mbps から Y Mbps に再生レートを切り替えとする。

1. X Mbps の転送を一時停止 (図 2.2(2) 参照)
転送の一時停止を要求する RTSP パケットのほかに、一時停止をするコンテンツの位置もサーバに対して送信する。
2. Y Mbps の送信を開始 (図 2.2(3) 参照)
転送を開始するコンテンツの位置は上で一時停止をした次のデータからになるように、RTSP パケットで要求をする。
3. Y Mbps のバッファリングをしている間は

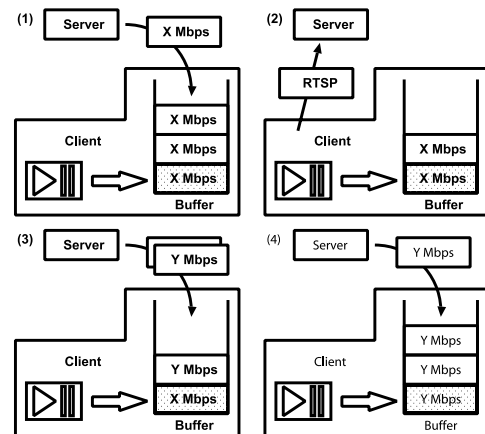


図 2.2. レート切り替えアルゴリズム

X Mbps のデータを再生 (図 2.2(3) 参照)
再バッファリング中は元々バッファで保存して
いたデータを再生する。

4. X Mbps のデータがなくなったら Y Mbps の データを再生 (図 2.2(4) 参照)

サーバに対し RTSP で再生を停止する直前の
図 2.2 の (1) の時点での受信バッファで確保さ
れているデータが再生を終わった時点で再生レ
ートを Y Mbps に変更する。

以上のようにして、再生を切り替えている途中は
 X Mbps のデータを再生することで再生途中にコン
テンツが一時的停止しないようにする。また、切り替
えた後の受信レート Y Mbps は切り替える時の受信
レートよりも低い再生レートで最大の再生レートを
選択する。

2.4 実装

本研究では、Helix Streaming Server のプロトコ
ルに対応し、UNIX 上で動作するストリーミング再
生アプリケーション Xine を対象に実装を進めてい
るところである。

2.5 まとめ

マルチビットレートストリーミング環境において
レート切り替え時に再生が停止するという問題に対
し、本研究ではアプリケーションバッファ残量に着
目し、レート切り替えタイミングを改善することに
より解決できると考えた。パケット到着間隔とバッ
ファ残量の減少レベルから、バッファの枯渇タイ
ミングを予測し、適切なレート変更リクエスト送信
タイミングの算出を行う。従来方式と比べ、より早い
タイミングでバッファ枯渇を予測しレート変更リク
エストを行うことにより、クライアントアプリケー
ション側で再生を継続したままレート切り替えを実
現する手法を提案した。

今後は本方式をクライアントアプリケーションに
実装し、既存のアプリケーションと比較した場合に
再生停止時間をどの程度短縮できるか評価を行う必
要があると考えられる。

第 3 章 2004 年度における Comet Regulator の研究

3.1 Comet Regulator

Comet Regulator はストリーム通信のパケット間
隔を均一化 (ストリーム平滑化) し、バースト通信を
抑制することにより、ネットワーク機器でのパケット
欠落を防止して通信品質を保つことを目的とした研
究である。本研究は 2003 年度に開始し、2003 年度は
測定器を使用した実験および LIVE!Eclipse (11 月)
でのフィールド実験を実施した。

Comet Regulator の特徴は以下の通りである。

- 1) ネットワーク的には L2 機器として存在し、IP
機器の設定変更を必要としない。
- 2) 富士通研究所がネットワーク処理のオフロード
を行う研究で開発した Comet ネットワークア
ダプタで処理を行い、10 μ 秒の精度で実時間制
御が可能である。
- 3) アダプタ上の ARM プロセッサに対するファ
ームウェアで実現しており、ソフトウェアでプロ
グラミング可能である。

2004 年度に行われた DVTS の相互接続実験では、
バースト性の高いストリームによって画像が再生で
きない状況になることが報告された。これはデータ
通信速度とネットワーク速度に差があると、実装に
よってはネットワークの速度でデータを連続して送
出してしまいバースト状態になるからである。一般
的な送信マシンの接続ネットワークが 1 Gbps へ移
行しつつある現状ではストリーム平滑化の需要は増
大すると考えられる。

2004 年度は以下のように、機能拡張および測定器
を使用した実験を行った。

3.2 新ハードウェア対応

Comet ネットワークアダプタの世代交代に対応し
て、2003 年度に使用していた Comet i-NIC-2R から
Comet X2b (いずれも開発コードネーム) に動作プ
ラットフォームを変更した。単純なブリッジ性能を
比較して、旧版の 700 Mbps から 900 Mbps へ
の高速化を達成した。

3.3 機能拡張

2003年度の実装では、ストリームの識別をIPアドレス単位で行っていた。これは大多数のマシンでは高々1ストリームしか受信しないので、処理の軽減を狙ったものである。しかし、サービスの内容でストリーム平滑化を行うか否かを変化させたり、NATの利用など1つのIPアドレスで複数のストリームを受信することに対応させるため、ポート単位のストリーム識別を行うよう変更した。

ポート単位のストリーム識別を行うためには、IPフラグメントに対応する必要が生じる。もともと、2003年度の実験の際にはIPフラグメント対応処理の計算コストと効果から判断してIPアドレス単位の識別を行った経緯がある。今回、フラグメント対応処理を行うためにデータの流れを刷新し、プログラムの大幅な変更を行った。その結果、後述するように懸念していた処理速度の低下は小さく抑えることができた。

また、ストリームごとの動作を設定ファイルで記述することができるようにし、運用を容易にした。なお、パケット間隔制御の算法は2003年9月のWIDE研究会で発表したPID制御方式から変更は行っていない。

3.4 測定器による実験

新しく実装したComet Regulatorで単一ストリームを平滑化させ、望む動作を実現していることを確認した。

アジレント社のRouterTesterという測定器では任意のバースト長で、任意のIPアドレス、ポートを持ったストリームを複数発生させることが可能であり、この測定器を使用して性能を測定した。その結果、3000ストリーム、合計750Mbpsを平滑化することが可能であることが確認できた。これはLIVE!Eclipseの際に処理した400ストリーム、50Mbpsの最大流量

から見て十分に実運用に供せる性能であると判断できる。

3.5 今後の予定

Comet Regulatorの研究は一応の収束段階にある。現在採用しているストリーム帯域計測法は長いバースト周期の場合、振動するおそれがある。現在のところとくに不都合がないが、今後問題となる場合には測定法および制御法を見直すことになる。

第4章 ストリーム平滑化装置のラボ実験およびフィールド実験による評価

4.1 Comet Regulatorのラボ評価

2003年8月に、(株)インターネット総合研究所と(株)富士通研究所では、ストリームデータの平滑化を行う装置「Comet Regulator」を共同開発し、ラボ環境で検証を行った。

4.1.1 背景

いくつかのストリーミングサーバでは、パケットをバースト的に送出してしばらく休むという動作をしている。これは、細かくして送出すると、サーバのCPUに負荷がかかるため、バーストに送出する実装になっているようである。主要なストリーミングフォーマットである「Windows Media」¹、「Real」²、「QuickTime」³についてパケットの送出間隔を調べたところ、下記のことが分かった。

- Windows Media (図 4.1) では、30個～50個のUDPをまとめて送信して、一定間隔休むという挙動をしている(帯域によってパケットサイズと個数は異なる)。
- Real (図 4.2) では、1個ずつ等間隔に送信している。

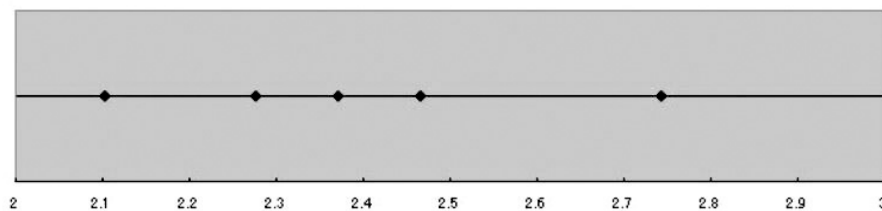


図 4.1. Windows Media のパケット送信間隔

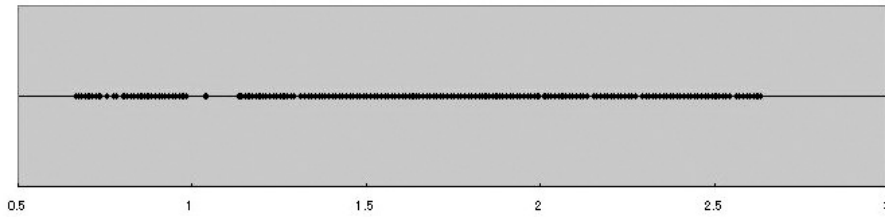


図 4.2. Real のパケット送信間隔

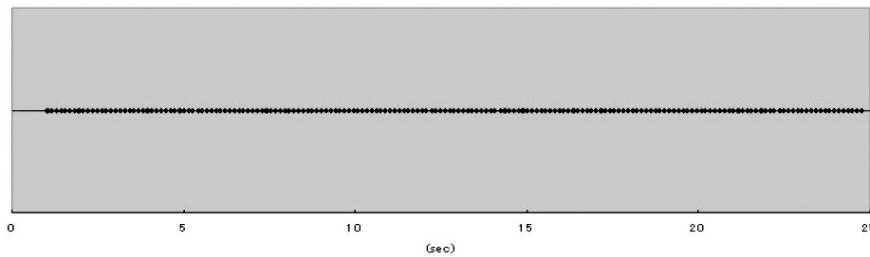


図 4.3. Quick Time のパケット送信間隔

- Quick Time (図 4.3) では、1 個ずつ等間隔に送信している。

Windows Media サービス 9 (以下、WM9) のようにバースト的にパケットが送信されると、経由するネットワーク機器のバッファがあふれ、パケットロスが発生し、ノイズやフレーム落ちの原因となってしまう。特に ADSL の DSLAM (局側集合モデム) や、パフォーマンスの悪い (バッファが少ない) ブロードバンドルータで問題となる。バーストロスを回避するには、トラフィックのシェイピングが有効である。しかし、多くのシェイピング装置は、指定した帯域以上のトラフィックを廃棄する実装となっている。

そこで、Comet Regulator を使って送出パケット

のレギュレーション (平滑化) を行い、その結果を検証した。この Comet Regulator は、IP ネットワーク経由でデジタルビデオを転送するボード「Comet DVIP」(<http://www.comet-can.jp/>) にストリームデータを平滑化するためのファームウェアを載せた構成になっており、Linux のマシンに組み込むことができる。

4.1.2 検証構成

図 4.4 のように、Comet Regulator を経由しないデータ (cap①) と、Comet を経由したデータ (cap②) を比較した。検証には、WM9 のストリームデータを使用した。

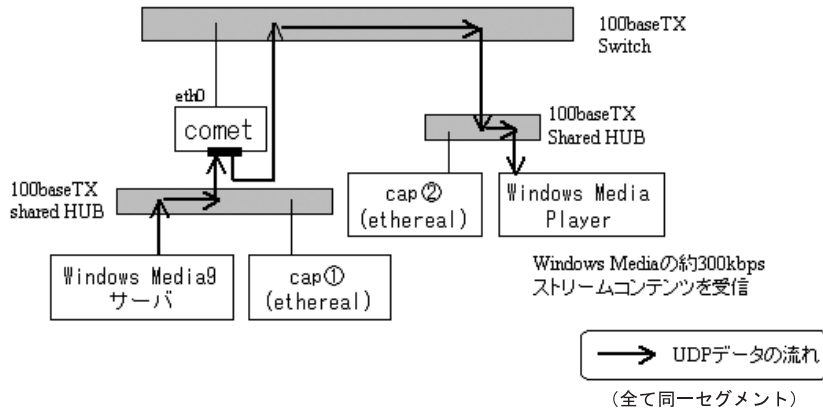


図 4.4. 検証構成

4.1.3 検証結果

・パケットの送信間隔について

296 kbps のライブコンテンツの視聴を行った際の
 パケット送出間隔 (1 packet = 780 byte) を測定し
 たところ、Comet Regulator なし (図 4.5) では、約
 100 msec 間隔に、50 パケット (約 39 kbyte) ずつ固

まっていた。一方、Comet Regulator あり (図 4.6)
 では、1 パケットごとに、ばらばらになっていた。

また、244 kbps のオンデマンドコンテンツの視聴を
 行った際のパケット送出間隔 (1 packet = 1082 byte)
 を測定したところ、Comet Regulator なし (図 4.7) で
 は、約 100 msec 間隔に、30 パケット (約 33 kbyte) ずつ

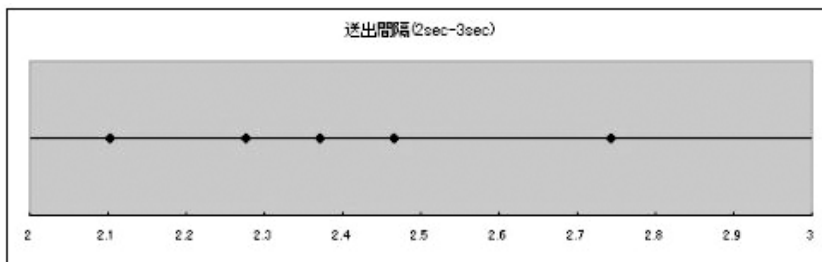


図 4.5. 296 kbps のライブコンテンツの視聴を行った際のパケット送出間隔 (Comet Regulator なし)

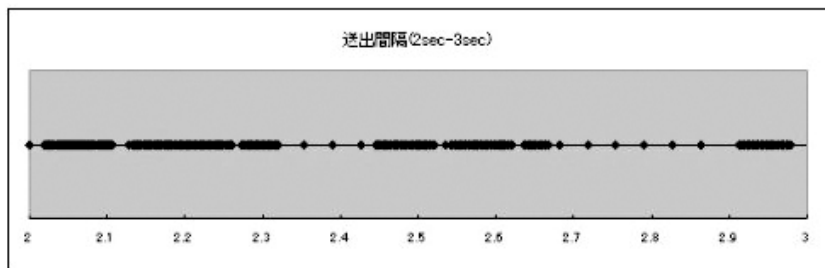


図 4.6. 296 kbps のライブコンテンツの視聴を行った際のパケット送出間隔 (Comet Regulator あり)

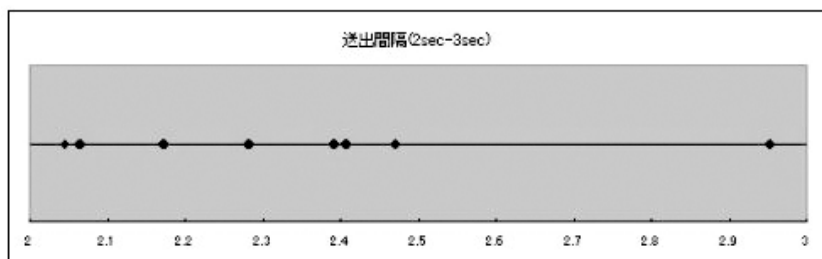


図 4.7. 244 kbps のオンデマンドコンテンツの視聴を行った際のパケット送出間隔 (Comet Regulator なし)

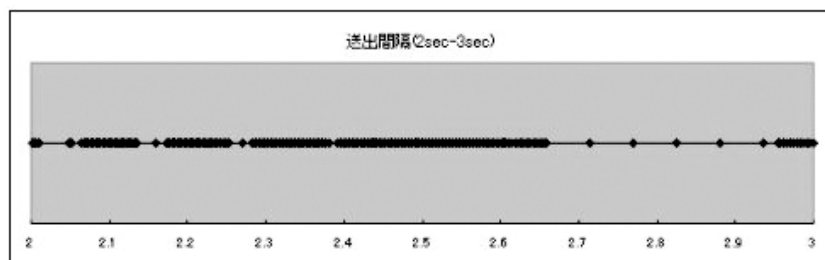


図 4.8. 244 kbps のオンデマンドコンテンツの視聴を行った際のパケット送出間隔 (Comet Regulator あり)

固まっていた。一方、Comet Regulator あり(図 4.8)では、1 パケットごとに、ばらばらになっていた。

・コンテンツの送信帯域について

296 kbps のライブコンテンツの視聴を行った際の帯域(帯域=パケットのサイズ÷直前のパケットとの送信時刻の差、で算出)を測定したところ、Comet Regulator なし(図 4.9)では、平均送信レート(296 kbps)を大幅に超えていた。一方、Comet Regulator あり(図 4.10)では、平均送信レートに抑えられていた。

また、244 kbps のオンデマンドコンテンツの視聴を行った際の帯域を測定したところ、Comet Regulator なし(図 4.11)では、平均送信レート(244 kbps)を大幅に超えていた。一方、Comet Regulator あり(図 4.12)では、平均送信レートに抑えられていた。

つまり、WM9 のライブ、オンデマンドともに Comet Regulator を介すことによって、パケット送出間隔が平均化され、帯域のバースト性が抑えられた。さらに、複数クライアントからのアクセスに対しても、ストリームごとにそれぞれ平滑化された。

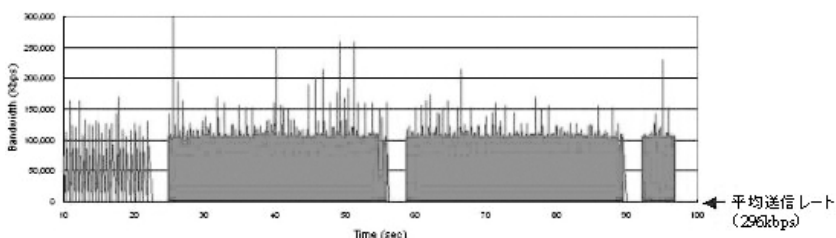


図 4.9. 296 kbps のライブコンテンツの視聴を行った際の帯域 (Comet Regulator なし)

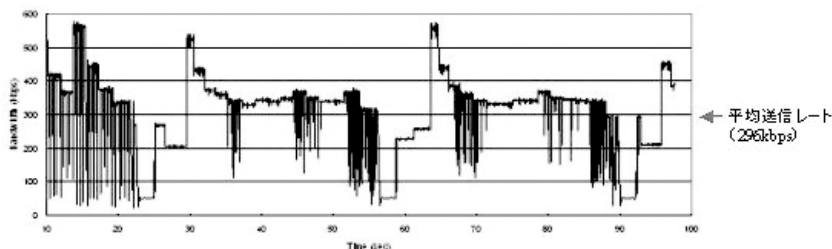


図 4.10. 296 kbps のライブコンテンツの視聴を行った際の帯域 (Comet Regulator あり)

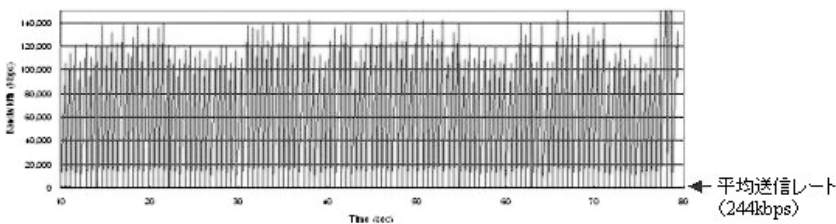


図 4.11. 244 kbps のオンデマンドコンテンツの視聴を行った際の帯域 (Comet Regulator なし)

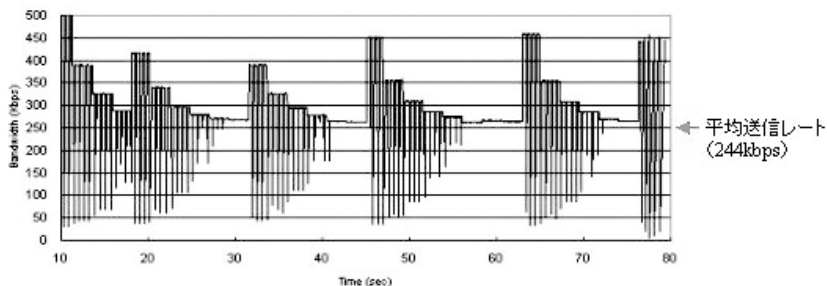


図 4.12. 244 kbps のオンデマンドコンテンツの視聴を行った際の帯域 (Comet Regulator あり)

4.2 ストリーム平滑化のフィールド実験

2003年11月24日にライブ！ユニバースが主催する南極皆既日食インターネットライブ中継の配信システムに Comet Regulator を組み込み、UDP データレギュレーションの有用性を実際のインターネット上で実験した。

4.2.1 Comet Regulator について

Comet Regulator は、通信プロトコル高速処理を実現するボード「Comet i-NIC-2R(GbE × 2 NIC 付き)」にストリームデータを平滑化するためのファームウェアを搭載している。Comet ボードのホストには、Linux を搭載した 1U サーバを使用。事前のラボ検証(詳細は、4.1 節参照)では、Comet Regulator を介すことによって、パケット送出間隔が平均化され、帯域のバースト性が抑えられた。さらに複数クライアントからのアクセスに対しても、ストリームごとにそれぞれ平滑化された。

今回のフィールド実験では、次の事項を確認する。

- Comet Regulator を使って、大規模な実トラフィックと多数のストリームをうまく平滑化し、負荷に耐えられるかどうか。
- Comet Regulator あり・なしを比較して、パケットロス率に違いがあるかどうか。
- Comet Regulator あり・なしを比較して、サーバの負荷に違いがあるかどうか。

4.2.2 実験構成

VECTANT(データセンタ)に、Comet Regulator を介した配信サーバ(WMS-1)と、何も介さない配信サーバ(WMS-2)を設置した(図 4.13)。

オリジンサーバからのストリームデータとクライアントへのストリームデータは、混在してお互いの帯域を圧迫しないように別セグメントに分けた。

4.2.3 実験結果

・アクセス数

各配信サーバには、350 Kbps のライブストリームに延べ 5000 回の視聴があり、最大の同時配信ストリーム数は、500 であった。予想以上のアクセス数であったが、大規模な実トラフィックと多数のストリームに耐えることができた。

・視聴を行った際のパケット送出間隔

クライアント側で取得したパケットキャプチャデータを確認したところ、Comet Regulator ありの場合(図 4.14)では、1 パケットごとにばらばらに送出されていた。一方、Comet Regulator なしの場合(図 4.15)は、固まって送出されてしまっていた。

・パケットロス、品質

配信サーバに出力されるログを集計し、クライアントの状態(図 4.16 と図 4.17) 喪失パケット数(図 4.18 と図 4.19) 品質(図 4.20 と図 4.21)を比較したが、大きな違いは見られなかった。

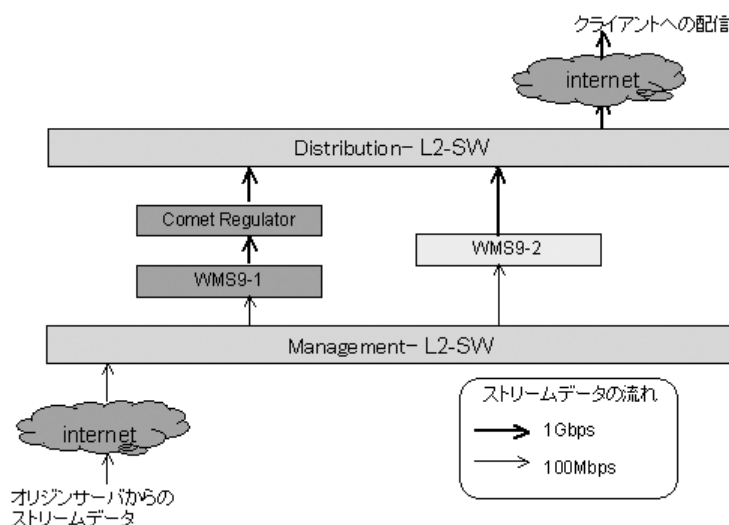


図 4.13. 実験構成図

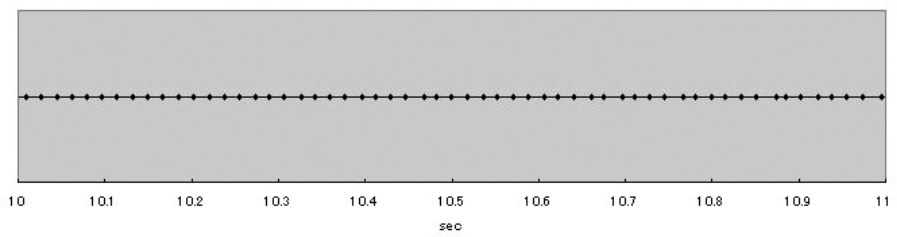


図 4.14. Comet Regulator ありで視聴を行った際のパケット送出間隔

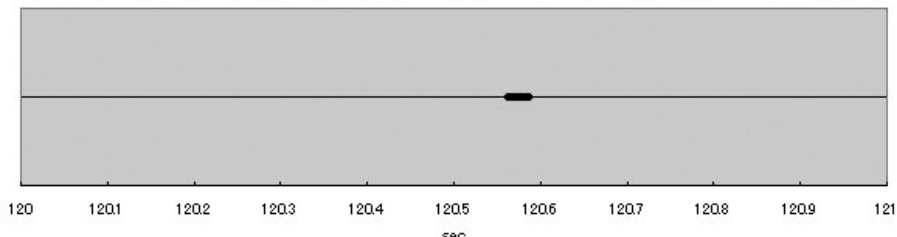


図 4.15. Comet Regulator なしで視聴を行った際のパケット送出間隔

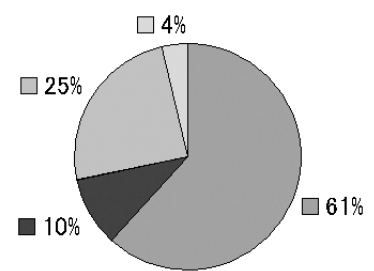


図 4.16. Comet Regulator ありのクライアントの状態

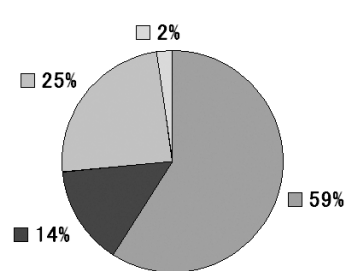


図 4.17. Comet Regulator なしのクライアントの状態

- 200: 正常に接続されました
- 210: 切断されたクライアントが再接続されました。
- 400: 要求された URL が無効です。
- 401: クライアントのアクセスが拒否されました。
- 404: 要求されたコンテンツが見つかりません。
- 408: クライアントが切断されたため、クライアントからログを送信できませんでした。
- 420: クライアントが切断され、再度接続しようとしたが失敗しました。
- 500: Windows Media サーバーが、内部エラーを検出し、ストリーム配信を停止しました。

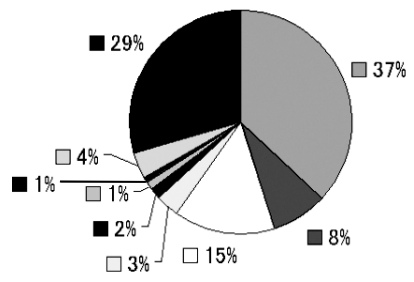


図 4.18. Comet Regulator ありの喪失パケット数

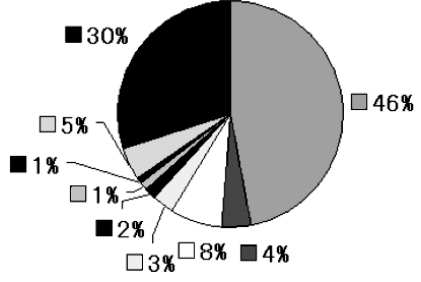


図 4.19. Comet Regulator なしの喪失パケット数

- 0%
- ~1%
- ~10%
- ~20%
- ~30%
- ~40%
- ~50%
- 50%~
- 測定不可

・ 配信サーバの負荷

Windows 2003 Server に付属しているパフォーマンスログを使って、「サーバの CPU 使用率」、「メモリ使用率」を計測したが、大きな違いはなかった。

・ 帯域

クライアントに流れる帯域を計測したところ、Comet Regulator なし (図 4.22) では、パースト的に約 40 Mbps 送出されてしまったのに対し、Comet Regulator あり (図 4.23) では、350 Kbps 付近で帯域が抑えられた。

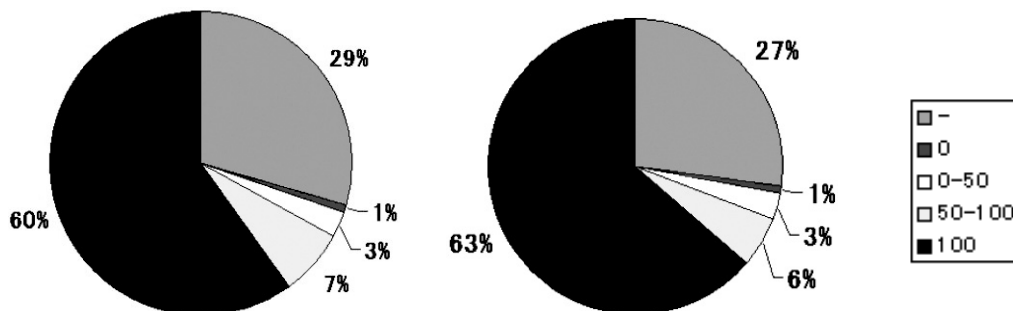


図 4.20. Comet Regulator ありの品質 図 4.21. Comet Regulator なしの品質

ストリームの再生中にプレーヤーから報告されたストリームの品質の最低値で、100 が最高。

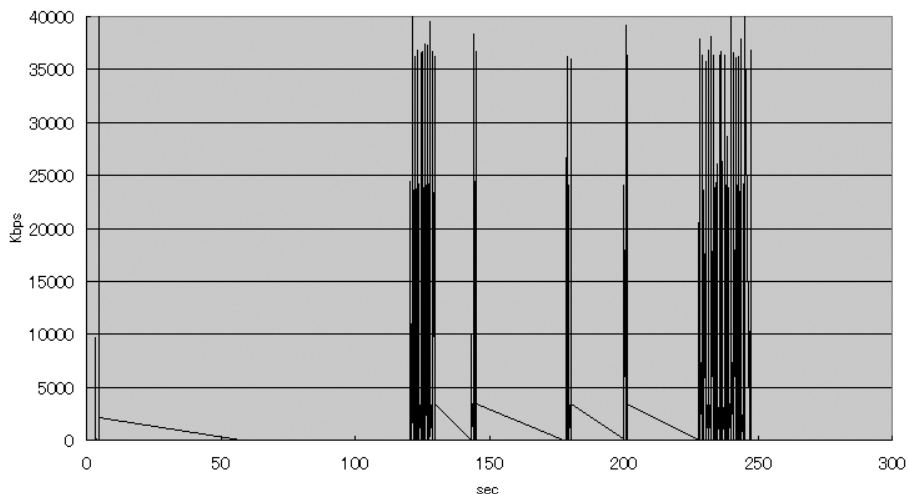


図 4.22. Comet Regulator なしの帯域

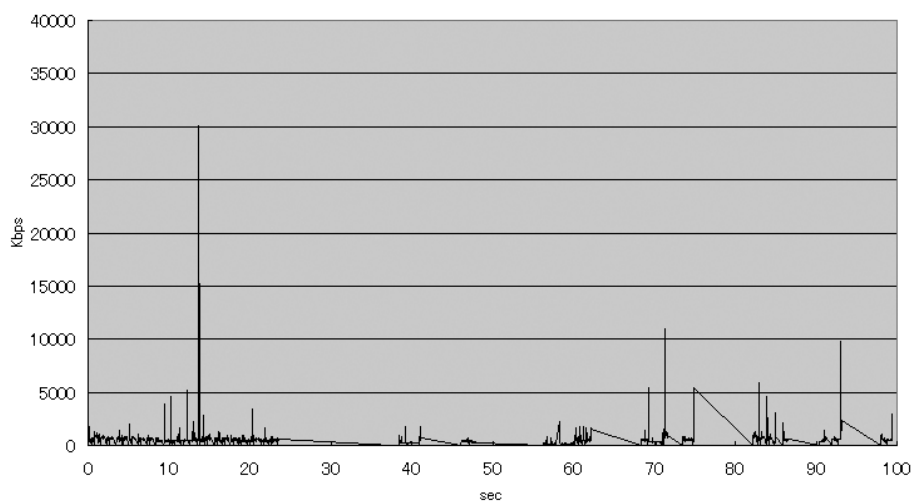


図 4.23. Comet Regulator ありの帯域

4.2.4 考察

実験結果より、Comet Regulator を介すことによってパケットがばらばらに送出され、帯域が抑えられることがわかった。しかし、クライアントの状態、喪失パケット数、品質に差がなく、配信サーバのパフォーマンスにも差が見られなかった。350 Kbps というコンテンツビットレートでは、配信サーバや DSLAM に負荷がかからないのかもしれない。

今後は、レート検出および制御アルゴリズムの改善や、MPEG や DV などの高品質コンテンツを数十本同時に配信できるような環境での評価が必要であると思われる。

第 5 章 2004 年度秋の WIDE 研究会での実験：衛星回線への Comet TCP の適用

5.1 Comet TCP

長距離回線において TCP は実効通信速度が低いことが知られている。Comet TCP 通信技術は TCP の長距離高速回線 (long fat pipe) での高速化を念頭に研究された。その成果として 2003 年には、通常のパケット欠落率をもつ回線で、200 msec の遅延がある環境でも、700 Mbps 以上の TCP 通信を維持できることを実証した。この成果によって、SC2003 の最大帯域距離積賞を受賞している。

5.2 実験の目的

2004 年度の秋の WIDE 研究会では、Comet TCP を衛星回線に適用してその有用性を実験した。通常の TCP 通信では帯域の上限を探るために定期的にパケット欠落と帯域縮小を繰り返すのに対し、本通信方式は割り当られた帯域を 10 μ sec 程度の精度で保持するという特徴を持つ。CometTCP で当初想定していた利用状況と同じく長距離だが、数百 Mbps という高速ではなく数 Mbps ~ 数 10 Mbps の低速な衛星回線においても、この特徴により TCP に比較して有用であることを実証するのがこの実験の目的である。

5.3 実験環境

表 5.1 のような衛星回線を使用した。

表 5.1. 実験に使用した衛星回線

名称	Nexter	DVB-RCS
帯域 (Mbps)	1.5/0.5	10/2
CometTCP 使用帯域	1.5/0.5	4/0.8
RTT (msec)	500-520	600-630
ロス率	少	1-5%

5.4 実験内容

我々が用意した Web proxy を Comet TCP で中継し、実験に賛同した参加者がブラウザを設定することによって利用する形態を採った。また、試験用データとして Windows Media Player 9 (以下 WMP9) の動画ストリーミング(1 Mbps オンデマンド、300 kbps ライブ)を用意した。これらを用いた利用者による主観試験のほか、ネットワーク性能試験プログラム iperf を使用して定量的な TCP の通信速度を測定した。

5.5 実験結果

5.5.1 ブラウザでの web アクセス

被検者の感想は効果がある、ないと意見が分かれた。これは、人間の感覚は性能に対して線型に効果を実感できるものではなく、あるしきい値をもって実感できること、また、使用感は回線状況以外の要素、たとえば接続したサーバの反応や、コンテンツの内容によって大きく左右されるため、効果を実感しにくいことが原因だと思われる。

5.5.2 ストリーミングデータの視聴

有用性は認められなかった。帯域の再設定が頻繁に生じ、その都度データのバッファリングが行われ動画像の再生が停止した。

パケットの欠落が起きても TCP の帯域は無視できる時間で復活する。したがって、再送による遅延がバッファリング時間に対して通常の状況より長時間であることが原因であると仮定して、バッファリング時間を遅延に対して十分に長い 20 秒に設定して実験したが、効果は見られなかった。

WMP9 クライアントにおける統計情報ではデータの欠落、ロスは共に記録されていなかった。これは TCP がパケットの欠落を補償していることを示しているが、それはすなわち、再送で生じる RTT 分の遅延時間の間、データがクライアントに渡されて

表 5.2. iperf による定量的測定結果

	Comet TCP 無	Comet TCP 有	回線速度
Flab 会場	850 kbps	1.4 Mbps	1.5 Mbps
会場 Flab	460 kbps	470 kbps	500 kbps
Flab 会場	456 kbps	3.9 Mbps	4 Mbps
会場 Flab	—	—	800 kbps

いないことを意味する。WMP9 は接続回線状況によって自動的に使用帯域を変化させるが、その変化手法は公開されていない。もし、遅延に敏感な算法を採用していると仮定すると、パケットの欠落で小さい帯域への再設定を行うが、帯域の急激な復活により再度大きな帯域への再設定を行うという繰り返しが発生しているという観測結果に一致する。

今回の実験は WMP9 の画質を向上することが目的ではないため、WMP9 という特定のメーカーの製品に対して動作の追及を行う予定はない。しかし、現在のストリーミングシステムは複雑な動作をしており、単純なモデリングによる改良はさほど効果を得られないという知見を得た。

5.5.3 iperf による定量的測定

iperf による定量的測定（2 分間）を行った結果を表 5.2 に示す。

遅延が大きな環境だったことから、TCP の速度が遅く、Comet TCP を使用することで大幅な改善が見られた。

Comet TCP の処理実績は、最大同時処理セッション数は 60、累計 14000/日。最大帯域は 1.6 Mbps であった。

5.6 実験のまとめ

iperf を使用した定量的実験では 1.5 Mbps の細かい回線でも RTT 500 msec という長い遅延のおかげで Comet TCP の効果が認められた。しかし、Web の閲覧ではそのほかの要素も多く利用者の主観的な観測での有意な効果は見られなかった。また、ストリーミングではパケットロスが生じると TCP 層が RTT の長さ分の時間パケットを止めてしまい、アプリケーションレベルで見た改善はできなかった。

