(σ, ρ)-calculus によるストリーミングサーバの配信レートの評価

Evaluating Delivery Rate of Streaming Server on (σ, ρ)-calculus

田中浩之, 豊泉洋 *

概 要

ストリーミングは遅延に対する要求が厳しい。本論文では、(σ, ρ)-calculus を用いて、厳格でかつ、確率に依存しないネットワークの待ち行列遅延を求める。また、CBR で圧縮された動画データを、ひとつのストリーミングサーバから複数のクライアントにオンデマント配信する際、待ち行列遅延を小さく抑えることのできる配信レートを提示する。この配信レートを使用することによって、待ち行列遅延の上限を保障した上で接続台数を増やすことが可能になる。

1 はじめに

現在、インターネットテクノロジーにおいてストリーミングサービスが注目を浴びている。通常、ネット上に存在する音楽ファイルや動画ファイルは、全てをダウンロードしなければ再生できない。しかし、ストリーミングサービスでは、RTP(Real-time Transport Protocol)[1] や RDT(RealNetworks Data Transport)[2] といったプロトコルによって、データは小分けにされてクライアント側に送られる。その為、利用者は、待ち時間を最少にして視聴することができる。

ストリーミングサービスを実現する為には、サーバから送信されるパケットが、想定される時間以内にクライアントに到着することが必要である。もし、想定される時間以上の遅延が発生すると、利用者の再生している動画や音楽が細切れになってしまう。ライブ配信 (リアルタイム配信) のときには、IP マルチキャストを使用することによって、ネットワークの負荷を軽減できる。しかし、オンデマンド配信は IP マルチキャストに適しておらず、利用者数に比例してネットワークの負荷が高まり、遅延が発生しやすい。よって、サービスの実現においては、ネットワークの遅延を評価することが重要である。しかしながら、マルコフ型の待ち行列理論 [3] では、システムや利用者の確率的な振舞いに依存してしまい、確定的な遅延を求めることが難しい。また、ネットワーク上での評価も困難である。そこで、deterministic analysis もしくは (σ, ρ)-calculus ([4],[5]) と呼ばれる評価法が提案されている。この評価法を用いることによって、厳格でかつ、確率に依存しないネットワークの遅延の上限を求めることが可能である。

本論文では、(σ, ρ)-calculus を使用して、CBR (Constant Bit Rate) で圧縮された動画データを、ひとつのサーバから複数のクライアントにオンデマンドで配信する際、待ち行列遅延の上限をある一定値以内に抑えながら、より多くのクライアントにサービスを提供できるストリーミングサーバの配信レートを示す。

2 サーバの配信レート

LAN 上に、Web サーバ、ストリーミングサーバ、クライアントを配置し、CBR で圧縮された動画データをストリーミングサーバからクライアントに配信する。サーバには Helix Universal Basic Server[2]、クライアントには RealOne Player[2] を使用した。なお、動画データは、Helix Producer[2] を用いて以下の設定でエンコードした。

Stream : RealVideo 9 - single rate
Constant bit rate : 450 kbps
Total lenght : 4:26.579
Max startup latency : 4.00 sec
Max time between key frames : 10.00 sec

RealOne Player には、TurboPlay と呼ばれるバッファリング時間を短縮する機能がある。図 1 に、TurboPlay 機能を ON、または OFF にした時の Helix Server からの累積アウトプットプロセスを示す。TurboPlay 機能が ON の場合、開始から 14 秒あたりまでは最大 1855Kbps で配信していたが、その後、460Kbps に配信速度を下げた。一方、TurboPlay 機能が OFF の場合は、90 秒あたりまでは最大 620Kbps で配信していたが、その後、TurboPlay 機能が ON の場合と同様に 460Kbps まで配信速度を下げた。クライアントは、TurboPlay 機能が OFF の場合は約 4 秒間バッファリングを行うが、ON の場合はすぐに動画の再生が始まった。このアウトプッ

* H. Tanaka and H. Toyoizumi are with Performance Evaluation Laboratory, University of Aizu, Aizu-Wakamatsu, Japan 965-8580. E-mail: m5061222@u-aizu.ac.jp, toyo@u-aizu.ac.jp

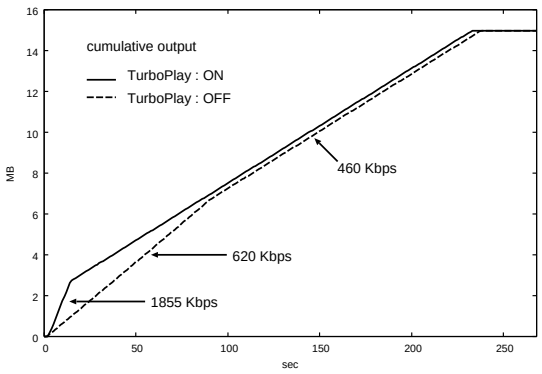

図 1: Helix Server の累積アウトプットプロセス

トプロセスは 1 台のマシンに対してのものであるが、同時に多くのマシンに配信する時に、これらのプロセスが待ち行列遅延に与える影響を評価する。

3 ネットワークモデル

大規模なコンテンツ配信を計画する場合、CDN(Contents Delivery/Distribution Network) が利用されることがよくある。CDN には、いくつかの形態が存在するが、本論文では、ISP が自社ネットワーク内に CDN を構築し、自社ユーザーに対してブロードバンドコンテンツを配信するモデルを想定する。この場合、メインのストリーミングサーバから ISP に設置されたキャッシュサーバに事前にコンテンツがコピーされており、利用者にはキャッシュサーバからコンテンツが配信される。図 2 に、ISP に設置された 1 台のキャッシュサーバから、その ISP に接続している k 台のクライアントに動画を配信するネットワークを示す。

上部のネットワークモデルを使用したストリーミングサービスについて、以下のような条件を与える。

1. 全てのリンクを work conserving link とする。
2. リンクにおけるストリーミングデータのパケットスケジューリングは、FIFO (First In First Out) に従う。
3. キャッシュサーバを接続しているルータからアクセスキャリアまでのリンクでは、常時 50Mbps の帯域がストリーミングサービスの為に予約されている。
4. アクセスキャリア内の回線速度が十分に速いものとする。つまり、速度を ∞ Mbps とおく。
5. k 台のクライアントは、CBR で圧縮された動画ファイルを好きな時間に視聴する。
6. クライアントの再生ソフトが保持しているバッファサイズを十分に大きいものとする。
7. パケットロスが非常に小さいとし、パケットの再送は行わない。

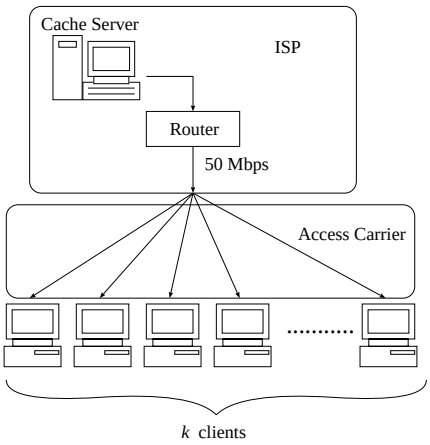

図 2: ネットワークモデル

したがって、キャッシュサーバを接続しているルータの待ち行列遅延を評価する。

4 (σ, ρ)-calculus

この章では、(σ, ρ)-calculus について簡単に紹介する。(σ, ρ)-calculus については、[4] に詳しく説明されている。[4] では、離散時間を対象にしているが、連続時間のときも同様に取り扱うことが可能である。サーバからのアウトプットプロセスを $A \equiv \{A(t), t \geq 0\}$ と記述し、$A(t)$ を時刻 t までにサーバから出力されたデータ量の総和とする。一般に、サーバからのアウトプットプロセスは複雑で、評価するのが難しい。そこで、(σ, ρ)-upper constrained を用いる ([4] pp. 3-5)。

Definition : 全ての $0 \leq s \leq t$ に対して、

$$A(t) - A(s) \leq \rho \cdot (t - s) + \sigma \qquad (1)$$

が常に成立するとき、A は (σ, ρ)-upper constrained であると言う。

(σ, ρ) による特徴付けは、プロセスに対して制約を課す。パラメータ ρ は sustainable rate、σ は burst size と考えることもできる。

A が (σ, ρ)-upper constrained であるとき、式 (1) より、全ての $0 \leq s \leq t$ に対して、$\sigma \geq A(t) - A(s) - \rho \cdot (t - s)$ が常に成立している。ここで、

$$\sigma' = f(\rho) = \max_{0 \leq s \leq t} [A(t) - A(s) - \rho(t - s)] \qquad (2)$$

とすると、σ' は、ある ρ に対して式 (1) を満たす最小の σ である。今後は、この σ' を単に σ と書く。

サーバからクライアント $i(i = 1,2,\ldots,k)$ に対する個別のアウトプットプロセスを A_i とすると、k 台のクライアントに対するアウトプットプロセス A は、$A(t) = \sum_{i=1}^{k} A_i(t)$ である。また、A_i が (σ_i, ρ_i)-upper constrained なら、A は $(\sum_{i=1}^{k} \sigma_i, \sum_{i=1}^{k} \rho_i)$-upper constrained となる ([4] pp. 5-6)。

サービスポリシーが FIFO で、速度 c のリンクに進むプロセス A が (σ,ρ)-upper constrained であると仮定する。d を全てのデータのなかでの待ち行列による最大遅延とすると、$\rho \le c$ ならば、$d \le \sigma/c$([4] pp. 8-10)。

5　遅延を発生させにくい配信レート

図 2 のネットワークモデルに対し、(σ,ρ)-calculus を適用することによって、待ち行列遅延の上限をある一定値以内に抑えながら、より多くのクライアントにサービスを提供できるキャッシュサーバの配信レートを示す。

ここで、全てのクライアントが同じ動画ファイルを視聴していると仮定する。あるクライアントのデコードプロセスを $P \equiv \{P(t), t \ge 0\}$ と記述し、$P(t)$ を時刻 t までにデコードされるデータ量の総和とする。動画の再生時間は p であるが、その前にバッファリングが時間 b 行われる。CBR なので、時刻 b から $p+b$ 間の P の傾きはほぼ一定であり、その傾きを α とする。また、キャッシュサーバからあるクライアントに対する個別のアウトプットプロセスを A する。ただし、$A(0) = 0$、$A(p) = \alpha p$ とする。特に、全ての $0 \le t \le p$ に対して、常に $A(t) = P(t+b)$ を満たすプロセスを $\bar{A}$ とする (図 3)。

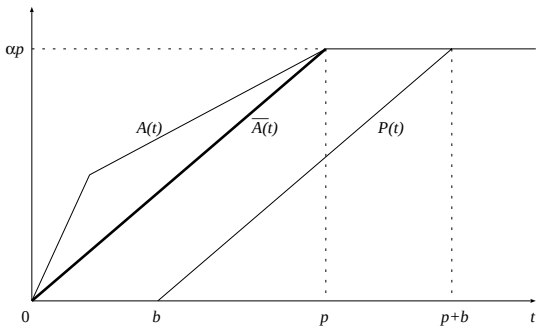

図 3: 遅延を発生させにくい配信レート $\bar{A}$

アクセスキャリア内の回線速度を ∞ Mbps と仮定しているので、待ち行列遅延は発生しない。したがって、待ち行列遅延はキャッシュサーバを接続しているルータでのみ発生する。ここで、個別のアウトプットプロセス A が (σ,ρ)-upper constrained であるとすると、k 台のクライアントが同じ動画ファイルを視聴するので、全てのストリーミングデータのなかでの待ち行列による最大遅延 d は、$k\rho \le c$ ならば、$d \le k\sigma/c$ (c=50Mbps) となる。ここで、最大遅延 d に対して最も tight な上限を与える (σ,ρ) の組み合わせについて考える。上限を小さくする為には、σ が可能な範囲内で小さくなれば良い。式 (2) より、$f(\rho)$ は減少関数なので、ρ が一番大きいとき、つまり、$\rho = c/k$ のとき上限が最小になる。よって、待ち行列遅延の上限を求める為の適切な組み合わせは、$(\sigma,\rho) = (f(c/k), c/k)$ である。

Theorem： 各々のクライアントに CBR で圧縮された同じ動画データをオンデマント配信する際、待ち行列遅延の上限をある一定値以内に抑えながら、より多くのクライアントにサービスを提供できるキャッシュサーバのアウトプットプロセスのひとつは、クライアントのデコードプロセス P を時間的に並行移動したプロセス $\bar{A}$ である。

Proof: プロセス $\bar{A}$ が k 台 (k は実数とする) 接続できるとき、A は $k+\varepsilon$ 台 ($\varepsilon > 0$) 接続することができないことを背理法で証明する。

待ち行列遅延を q 以内 ($q < b$) に制限したいとする。$\bar{A}$ が $(\bar{\sigma},\bar{\rho})$-upper constrained であるとき、q 以内という条件のもとで最大 k 台接続できるとすると、次の 2 式を同時に満たす。$k\bar{\rho} = c$、$q = k\bar{\sigma}/c$ (c はリンク速度)。一方、A が (σ,ρ)-upper constrained であるとき、$k+\varepsilon$ 台接続できると仮定すると、少なくとも次の 2 式を同時に満たす。$(k+\varepsilon)\rho = c$、$q = (k+\varepsilon)\sigma/c$。よって、

$$\rho = \frac{c}{k+\varepsilon} < \frac{c}{k} = \bar{\rho}, \quad \sigma = \frac{cq}{k+\varepsilon} < \frac{cq}{k} = \bar{\sigma}$$

$\bar{A}$ は、全ての $0 \le s \le t \le p$ に対して、$\bar{A}(t) - \bar{A}(s) = \alpha(t-s) \le \bar{\rho}(t-s) + \bar{\sigma}$ が成立している。したがって、

$$\bar{\sigma} = \max_{0 \le s \le t \le p} [(\alpha - \bar{\rho})(t-s)]$$

より多くのクライアントにサービスを提供したいので、$\alpha \ge \bar{\rho} = c/k$ となり、$\bar{\sigma} = (\alpha - \bar{\rho})p$。

以上のことから、全ての $0 \le s \le t$ に対して、

$$A(t) - A(s) \le \rho(t-s) + \sigma < \bar{\rho}(t-s) + \bar{\sigma} = \bar{\rho}(t-s) + (\alpha - \bar{\rho})p$$

が成立する。この式に、$t = p$、$s = 0$ を代入してみると、(左辺) $= A(p) - A(0) = \alpha p$、(右辺) $= \bar{\rho}p + (\alpha - \bar{\rho})p = \alpha p$。(左辺) = (右辺) となってしまい、矛盾。よって、A は $k+\varepsilon$ 台を接続することはできない。 ■

なお、上の証明では全てのクライアントが同じ動画ファイルを視聴しているという条件のもとで証明して

いるが、異なる動画ファイルを視聴している場合も、ほぼ同様に証明することが可能である。

6　数値例：接続台数の比較

図 1 で示した TurboPlay 機能を ON、または OFF にした時の Helix Server からの累積アウトプットプロセスに、5 章で求めた遅延の上限を小さく抑えられるアウトプットプロセス $\bar{A}$ を書き加えると、図 4 のようになる。

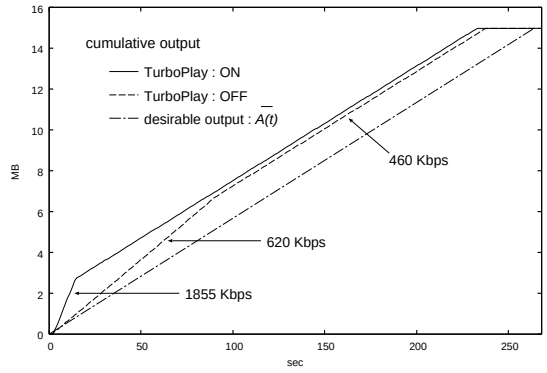

図 4: Helix Server の配信レートと $\bar{A}$ の比較

接続できるクライアント数 k と待ち行列による最大遅延 d の関係は、$d \leq k\sigma/c$ で求められる。c はリンク速度 (ここでは 50Mbps)、$\sigma = f(\rho) = f(c/k)$ である。複数のクライアントに、全て TurboPlay 機能を ON で配信した場合、OFF で配信した場合、5 章で求めたアウトプットプロセス $\bar{A}$ で配信した場合の、クライアント数と待ち行列による最大遅延の上限の関係を図 5 に示す。

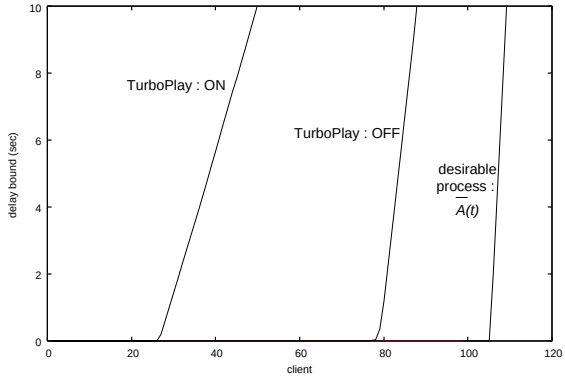

図 5: 接続台数の比較

TurboPlay 機能が ON の場合、クライアントでのバッファリング時間を短縮する為に、開始から 14 秒あたりまでは最大 1855Kbps で配信する (図 4)。これは、帯域が十分に余っているときには問題にはならないが、同時に複数のクライアントがストリーミングサービスを開始すると、一時的にネットワークに負荷がかかり、大きな待ち行列遅延が発生する可能性がでてきてしまう。一方、アウトプットプロセス $\bar{A}$ で配信すると、遅延の上限を小さく抑えたまま多くのクライアントにサービスを提供できる。TurboPlay 機能はバッファリング時間を短縮できるが、待ち行列遅延の上限の保障を大幅に弱めてしまうので、使用には注意しなければならない。

7　まとめと今後の課題

複数のクライアントにオンデマンド配信する際、待ち行列遅延の上限をある一定値以内に抑えながら、より多くのクライアントにサービスを提供できるストリーミングサーバの配信レートを示した。より多くのクライアントにサービスを提供できるサーバのアウトプットプロセスは、クライアントのデコードプロセスを時間的に並行移動したプロセスである。この配信レートを使用することによって、待ち行列遅延の上限を保障した上で接続台数を増やせることがわかった。

今回の評価では、パケットロスの再送、回復を考慮にいれてない。よって、今後は信頼性のある配信方法にも焦点をあてたい。また、接続台数が増加しても、ネットワークの負荷をより小さく抑える配信方法についても検討していきたい。

参考文献

[1] Request for Comments: 1889, RTP: A Transport Protocol for Real-Time Applications. http://www.ietf.org/rfc/rfc1889.txt

[2] RealNetworks Inc. Products and Services. http://www.realnetworks.com/products/

[3] L. Kleinrock, *Queueing Systems Volume 1: Theory*, John Wiley & Sons Inc, 1975

[4] Cheng-Shang Chang, *Performance Guarantees in Communication Networks*, Springer, 1999.

[5] Jean-Yves Le Boudec and Patrick Thiran, *Network Calculus : A Theory of Deterministic Queuing Systems for the Internet*, Springer, 2001.