

マルチキャストストリーミングの最適マーキング法

豊泉 洋[†] 田中 浩之[†]

[†] 会津大学 性能評価研究室 〒965-8580 福島県会津若松市一箕町鶴賀

E-mail: [†]toyo@u-aizu.ac.jp

あらまし マルチキャストとユニキャストを組み合わせることにより、ストリーミングサーバーからのダウンロードレートを効率的に削減する方法を示した。ストリーミングサーバーへのリクエストを Poisson Process としてモデル化し、その性質を利用することにより、ダウンロードレートを最小化する最適なマルチキャストとユニキャストの組み合わせ方法を導出した。この方法を使うことにより、大規模なストリーミングサービスを行う場合でも、ストリーミングサービスの規模に対してルートのオーダーでダウンロードレートを押さえることが可能となる。

キーワード マルチキャスト、ストリーミング、Poisson Process、再生過程、映像配信、CDN

Optimal Merging Scheme of Multicast Streaming

Hiroshi TOYOIZUMI[†] and Hiroyuki TANAKA[†]

[†] Performance Evaluation Lab. University of Aizu Tsuruga, Ikki-machi, Aizu-wakamatsu, Fukushima, 965-8580
Japan

E-mail: [†]toyo@u-aizu.ac.jp

Abstract Combining multicast and unicast, we show the download rate from the streaming server can be effectively reduced. By modeling the request from users as a Poisson arrival, we can derive the optimal combination of unicast and multicast. Even in the large-scale streaming service, the download rate will be reduced by the order of $n^{1/2}$, where n is the number of users in the streaming service.

Key words multicast, streaming, Poisson Process, renewal process, video distribution, CDN

1. Introduction

ブロードバンド環境による常時接続が一般家庭にも浸透し、より多くの人たちがインターネットを経由して、気軽に映像サービスを受けることができるようになった。インターネットでの映像サービスの実現には、さまざまなサービス方法が提案されている。本論文では、On-demand でユーザーにストリーミング映像を提供するサービスを考える。

ユーザーからのリクエストに対して、コンテンツのオブジェクトを遅延せずに配送する最も容易な方法は、ストリーミング配信を行うサーバーとユーザーごとに unicast のリンクをオンデマンドで設定することである。しかし、この方法で映像データを配送すると同時に接続するユーザーの数 n に対して、線形のオーダーで映像配信サーバーからの通信帯域が必要となる。一人あたり $2Mbps$ の速度 (NTSC 並映像品質) で 1 万人のユーザーが同じサーバーにアクセスする場合を考えると、映像配信サーバーは、 $20Gbps$ のスピードでネットワークと接続する必要がある。コンテンツデリバリーネットワーク (CDN) のような仕組みで、アクセスの ISP 毎に映像サービスを置いた場合に

は、インターネット全体にかかる負荷を減らすことができるが、それでも、映像サーバーは同じデータパケットを繰り返し転送しなくてはならず、非効率的である。

このような on-demand 型のストリーミングサービスにおける配信サーバーの帯域を効率的に使う方法が、精力的に検討されている。[5] では、Skyscraper と呼ばれる配送スケジューリングにより、リクエストから再生までの遅延を許容しながら、帯域を一定以下に抑える方法が提案されている。[2] では、冗長なパケットを転送するアプローチにより、転送エラーの再送処理が不要な方法が Digital Fountain という名前で提案されている。

また、本論文と同様に、リクエストから再生までの遅延を許容しない場合のダウンロードレートの削減量の評価も行われている。[3], [4] では、将来のリクエストの到着時点まで予測できることを前提とした場合に到達可能な平均帯域の下限 b_0 を導出している。その他のマルチキャストストリーミング関連の文献は、[6] に詳しい。

本論文では、マルチキャストストリーミングサーバーのダウンロードレートを削減するために、マルチキャストとユニキャストを組み合わせた簡易な方法について数学的にモデル化し、そ

の最適化の手法について述べる。

2. Modeling of Multicast Streaming

本論文では、以下のようなマルチキャストとユニキャストを組み合わせたストリーミングサービスの実現方法を考える。ユーザーはストリーミングサービスを受ける場合には、あらかじめ決められたストリーミングサーバーへの接続を行う。

視聴者からのリクエストがあった時、ストリーミングサーバーには、次の二つの選択肢がある。

(1) 他のユーザーも聴取可能なマルチキャストストリームとしてコンテンツを配信する。

(2) リクエストのあったユーザーのみが視聴できるユニキャストとしてコンテンツを配信する。

どちらの場合も、コンテンツはその内容の先頭部分から配信される。

ユーザーは、リクエスト時点で聴取可能なマルチキャストストリームがない場合には、新たに、マルチキャストストリームを配信を要求する。一方、リクエスト時点でマルチキャストストリームがある場合には、このマルチキャストストリームを聴取し、同時に、配信済みのためマルチキャストストリームとして受信できなかった部分をユニキャストとして受信する。ただし、到着時点で聴取可能なマルチキャストストリームがあったとしても、あえてマルチキャストストリームの配信を選ぶことができる。ストリーミングサーバーでは、ダウンロードレートを最小にするように、この二つの選択肢を最適に選ぶことが必要となる。

図1では、前にストリームがないリクエスト c1 に対しては、マルチキャストが適用され、続くリクエスト c2 は、c1 がスタートしたマルチキャストとともに、オブジェクトの先頭部分をユニキャストで視聴する。ユニキャストの通信は、c1 のマルチキャストの視聴に追いついた時点で打ち切られる。c3 については、再度マルチキャストが配信されている。

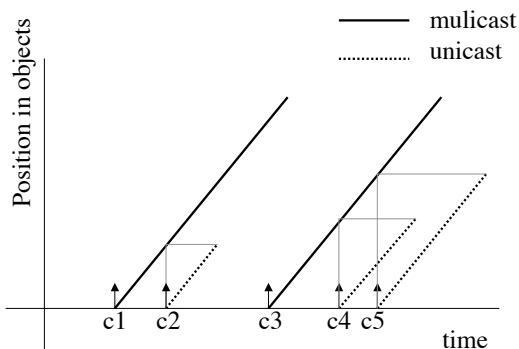


図1 マルチキャストとユニキャストを組み合わせたストリーミング配信：↑はリクエストの到着を表す。

リクエスト時点で、目標のコンテンツを配信するマルチキャストストリームがあったとしても、そのマルチキャストストリームの配信するよりも、後続のリクエストのために、新たにマルチキャストストリームを配信する方が、効率的な場合があ

る。この場合には、ストリーミングサーバーは、ユニキャストではなく、マルチキャストのストリームの配信を開始する。図1では、c3 に対してユニキャストで配信するよりも、マルチキャスト配信を行った方が後続の c4 と c5 の帯域を削減できる。本論文では、ストリーミングサーバーの使用帯域の効率化の観点から、どの時点でのリクエストに対してマルチキャスト配信を行うのが最適な戦略となるかを検討する。

3. Modeling of Multicast Streaming

簡単のため、ストリーミングサーバー上のコンテンツオブジェクトは、長さ s で、そのプレイレート（再生レート）を 1 とする。異なる長さやプレイレートのコンテンツがある環境では、それぞれのコンテンツ毎に別のストリーミングサーバーがあるとして、以下のモデル化を適用することができる。

視聴者からのストリーミングサーバーへのリクエストは、レート λ の Poisson 過程に従うと仮定する。実際のストリーミングサーバーへのリクエストは、コンテンツの内容等により、Poisson 過程とは異なることも予想されるが、ここでは最も一般的な確率過程として、Poisson 過程を選ぶことにする。実際に、[1] では、マルチメディアサーバーへのリクエストが Poisson 過程となることが確かめられている。

時刻 0 で最初のリクエストが到着し、このリクエストに対してマルチキャストストリームの配信が行われたとする。時刻 x までに到着したリクエストについては、この「親」マルチキャストストリームに対する「子」となり、マルチキャストによる未受信分のデータをユニキャストにより受信する。時刻 x 以降に最初に到着したリクエストは、時刻 0 で到着したリクエストと同様に、親マルチキャストストリームとなる。この x をユニキャスト限界時間とよび、以下では、 x の最適値を求めることを目標とする。

$N(t)$ を $(0, t]$ 間のリクエスト数、 n 番目の視聴者からのリクエスト時点を T_n とする ($T_0 = 0$)。ここで、 $[0, x]$ に到着したリクエストによるストリーミングサーバーからのダウンロード量 R を考える。時刻 0 に到着したリクエストにはフルマルチキャストストリームを配信するが、それ以降、時刻 x までの到着したリクエストは、自分が到着する以前にマルチキャストストリームとして配信され、未受信の部分をユニキャストとして受信する必要がある。したがって、

$$R = \sum_{i=1}^{N(x)} T_i + s. \quad (1)$$

ここで、 R の期待値を考える。 $N(x)$ について条件づけをする。

$$\begin{aligned} E[R] &= s + E \left[\sum_{i=1}^{N(x)} T_i \right] \\ &= s + E \left[E \left[\sum_{i=1}^{N(x)} T_i \middle| N(x) \right] \right]. \end{aligned}$$

一方、 $N(x) = n$ で条件付けすると、Poisson Process の到着時点列である $T_1, T_2, \dots, T_{N(x)}$ は $[0, x]$ 上の独立で同一な一様分布の列 U_1, U_2, \dots, U_n の順序統計量と同値となることが知られている

(例えば、[7] の Theorem 2.3.1 参照)。したがって、

$$\begin{aligned} E \left[\sum_{i=1}^{N(x)} T_i \mid N(x) = n \right] &= E \left[\sum_{i=1}^n U_i \right] \\ &= \frac{nx}{2}. \end{aligned}$$

よって、 R の期待値は、

$$E[R] = s + E \left[\frac{N(x)x}{2} \right] = s + \frac{\lambda x^2}{2}, \quad (2)$$

となる。

さらに、 X_m を m 番目の親マルチキャストになったリクエストの到着時点とし、 R_m をその親マルチキャストが作る子ユニキャストを含めた全ダウンロード量とする。リクエストの到着過程が Poisson Process なので、 (X_m, R_m) は独立、同一分布を持つ。 $X(t)$ を時刻 t までに到着したリクエストの全ダウンロード量とする^(注1)と、 $X(t)$ は

$$X(t) = \sum_n^{N(t)} R_m, \quad (3)$$

とあらわされ、Renewal reward process (例えば [7], [8] 参照) となる。次の section では、ダウンロードプロセスが Renewal Reward process となることを利用して、最適な戦略を考える。

4. Optimal Strategy

マルチキャストを効果的に使うことにより、ユニキャストでのストリーミングサービスに比べ、劇的にサーバーからのダウンロードレートを減らすことができる。ここでは、特に平均ダウンロードレートを最小にする最適なユニキャスト限界時間 x_0 を求めることを考える。ここで、

$$b(x) := \lim_{t \rightarrow \infty} \frac{X(t)}{t}, \quad (4)$$

とすると、 $b(x)$ はユニキャスト限界時間を x とした場合のサーバーからの平均ダウンロードレートと考えることができる。

Theorem 1 (最適ユニキャスト限界時間). 大きさ s のコンテンツに対するリクエストがレート λ の Poisson Process にしたがってストリーミングサーバーへ到着するとき、ユニキャスト限界時間を x とした場合のサーバーからの平均ダウンロードレート $b(x)$ は次のように表される。

$$b(x) = \frac{2\lambda s + \lambda^2 x^2}{2(\lambda x + 1)}. \quad (5)$$

また、平均ダウンロードレートを最小にするユニキャスト限界時間を x_0 とすると、

$$x_0 = \frac{(1 + \lambda s)^{1/2} - 1}{\lambda}, \quad (6)$$

となる。

(注1)：時刻 t でダウンロードが終了していないリクエストについては、時刻 t 以降に必要なダウンロード量を含める

Proof. Section 3. より、時刻 t までのダウンロード量 $X(t)$ は、Renewal reward process としてモデル化できることがわかっているため、再生定理 ([7] Theorem 3.6.1 参照) を適用できる。Poisson 過程の性質より、ユニキャスト限界時間が過ぎた後で、最初のリクエストが到着するまでの時間は、リクエストの到着間隔と同様に、平均 $1/\lambda$ の指数分布となる。したがって、 $X(t)$ の再生間隔の期待値は、 $x + 1/\lambda$ となる。再生定理より、確率 1 で、

$$\frac{X(t)}{t} \rightarrow \frac{E[R]}{x + 1/\lambda} \text{ as } t \rightarrow \infty, \quad (7)$$

が成立する。(2) と (7) より (5) が成立する。また、この結果から (6) も容易に導くことができる。□

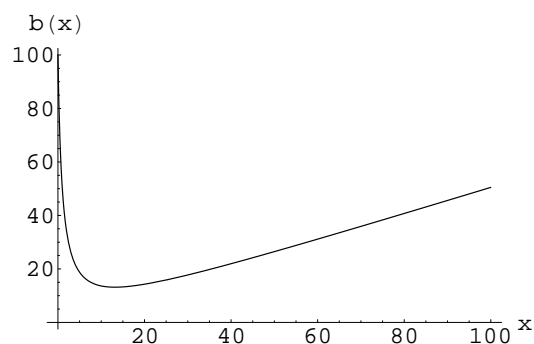


図2 平均ダウンロードレート $b(x)$ とユニキャスト限界時間 x の関係：但し、リクエストレート $\lambda = 1$ 、コンテンツの大きさ $s = 100$ とした。

図2に $b(x)$ のサンプルを示す。この場合の最適ユニキャスト限界は $x_0 = 9.04988$ となる。

最適ユニキャスト限界 x_0 は (6) より、 λ の関数と見ることができる。これを $x_0(\lambda)$ とし、 $\lambda \rightarrow 0$ の極限を考える。

$$x_0(\lambda) \rightarrow s, \text{ as } \lambda \rightarrow 0.$$

リクエストが「子」になるためには、 $x < s$ が必要である。したがって、これは、リクエストのレートが低く、後続があてにならないので、いつでも自分は「子」になるというのが最適な戦略となることを示している。一方、 $\lambda \rightarrow \infty$ の極限では、

$$x_0(\lambda) \rightarrow 0, \text{ as } \lambda \rightarrow \infty.$$

となり、自分のすぐ後にリクエストが存在するので、自分はいつでも「親」になるのが最適な戦略となる。

最適ユニキャスト限界時間 x_0 を (5) に代入することにより、最適戦略の場合の平均ダウンロードレート $b(x_0)$ を得ることができる。

Theorem 2 (最適戦略の場合の平均ダウンロードレート). Theorem 1 で得られた最適ユニキャスト時間 x_0 を採用した場合の平均ダウンロードレートは、

$$b(x_0) = (1 + 2\lambda s)^{1/2} - 1, \quad (8)$$

となる。

すべてユニキャストでストリーミングサービスを実現した場合の平均ダウンロードレートは明らかに λs となる。これは、マルチキャストサービスの規模を表すリクエストレート λ に対してサーバーからのダウンロードレートが線形に大きくなることを示している。一方、Theorem 2 より、マルチキャストとユニキャストを組み合わせた場合には、 λ のルートのオーダーでダウンロードレートが押しえられることを示している (図 3 参照)。

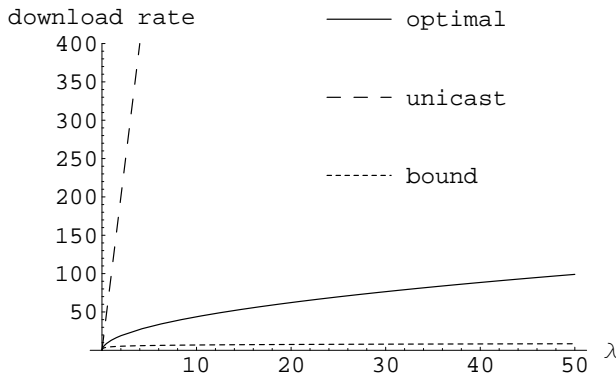


図 3 ストリーミング方式によるダウンロードレートの変化：最適ユニキャスト限界を採用した場合 (Optimal) とユニキャストだけを使ってストリーミングサービスを行った場合 (unicast) を比較している。但し、コンテンツの大きさ $s=100$ とした。また、参考のために、将来のリクエストの到着時点まで予測できることを前提とした場合の平均帯域の下限 (bound) [4] も示した。

Remark 1. Theorem 2 は、過去のリクエスト到着時点は既知として使用した場合のダウンロードレートが削減量を示しているが、[4] では、未来のリクエストの到着時点まで予測できることを前提とした場合の平均帯域の下限 b_0 を導出している。

$$b_0 = \log(1 + \lambda s). \quad (9)$$

未来のリクエスト時点の情報まで使える場合には、さらにダウンロードレートがかなり削減できることがわかる (図 3 参照)。

5. Conclusion

大規模なストリーミングサービスを行う場合に、ユニキャスト限界時間を使って、ユニキャストとマルチキャストを組み合わせることによって、ストリーミングサーバーからのダウンロードレートを効率的削減する簡易な方法を提案した。また、リクエストを Poisson Process として、モデル化することにより、ユニキャスト限界時間を最適化する方法を示した。この方法を使うことにより、ストリーミングサービスを受ける人数のルートのオーダーでダウンロードを押しえることが可能になり、スケーラブルな形でサービスを提供できる。

文 献

- [1] Almeida, Carrico, Maretzek, Noble, and Fletcher. Analysis of genomic sequences by chaos game representation. *BIOINF: Bioinformatics*, 17, 2001.
- [2] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67, 1998.
- [3] Derek L. Eager, Mary K. Vernon, and John Zahorjan. Optimal and efficient merging schedules for video-on-demand servers. In *ACM Multimedia (1)*, pages 199–202, 1999.
- [4] Derek L. Eager, Mary K. Vernon, and John Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. *Knowledge and Data Engineering*, 13(5):742–757, 2001.
- [5] Kien A. Hua and Simon Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In *SIGCOMM*, pages 89–100, 1997.
- [6] Anirban Mahanti, Derek Eager, Mary Vernon, and David Sundaram-Stukel. Scalable on-demand media streaming with packet loss recovery. *Proceedings of SIGCOMM'2001*, page 12, 2001.
- [7] S. M. Ross. *Stochastic Processes*. John Wiley and Sons, 1996.
- [8] R.W. Wolff. *Stochastic modeling and the theory of queues*. Princeton-Hall, 1989.