

# TCP 並列接続を用いたプログレッシブダウンロード における順序制御方式の実装

## Implementation of Sequence Control Method in Progressive Download Using Parallel TCP Connections

1420180 平城 光雄 Mitsuo Heijo  
指導教員 舟阪 淳一

### 1 はじめに

近年、効率的なコンテンツ配信を目的とした CDN 等でのコンテンツの分散配置などがすでに運用されている。このような同一のファイルが様々な場所に配置されている状況を利用して複数のサーバと同時に通信を行うことで、より高速なファイルのダウンロードを実現する方式が提案されている [1][2]。また既存の提案ではプログレッシブダウンロードにおける動画再生等を考慮した順序制御は目的とされていない。本研究では、性能の異なる複数の TCP 接続を利用して異なるサーバから同一のファイルを分割取得する場合を想定し、リクエスト送信時に各 TCP 接続間の性能を比較することで到着順序逆転の発生を抑制する順序制御方式を提案する。

### 2 関連研究

既存研究では、複数のサーバからファイルを分割して並列ダウンロードし、順序通りにクライアントに提供する代理サーバが提案されている [2]。既存の提案である重複再要求方式では、非有効ブロックの個数がある閾値を超えた場合は別の接続に要求を再送信することで事後的に順序逆転に対処していた。この提案では、要求送信時における予防的な順序逆転への対策は行われていない。また、動画等の再生を考慮した評価も行われていない。

### 3 提案方式

確立した TCP 接続群の中で最高性能の接続には、最若番のブロックを要求する。最高性能の定義は使用回数が最多の接続とする。低性能接続には前回の要求の送信からブロックの到着までのブロック送受信回数の差分  $D$  を計測し、 $D$  に基づいて最若番でないブロックを要求する。 $D$  は式 1 から求める。 $T$  はその時点でのブロックの総受信回数、 $P[X]$  は接続  $X$  の前回の受信完了時におけるブロックの総受信回数、 $N$  は接続の個数である。図 1 に 2 つの接続間に 2 倍の性能差がある場合の模式図を示す。図 1 では低性能接続の要求ブロック ID をこの性能差を考慮した要求を送信することでブロック番号 0-5 が順序通りの到着が期待できる様子を示している。

$$D = T - P[X] - N \quad (1)$$

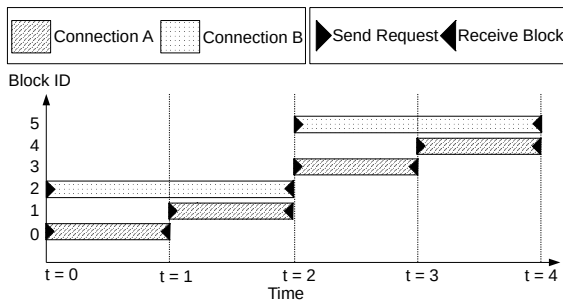


図 1: 遅延要求の模式図

### 4 実験と評価

実装したプログラムでテストベッドと公開ネットワークで評価を行なった。本稿では紙面の都合上、公開ネットワークでの結果のみを掲載する。公開ネットワークでの実験には利用のしやすさから Ubuntu のイメージファイルのパブリックミラーを使用した。サーバは 9 つ使用し、2 つ

は国内での残りの 7 つは国外のものを使用した。図 2 に事前に調査したミラーサーバにおける帯域性能の 24 時間変化を示す。2 より国内のサーバと国外のサーバの遅延および帯域の性能差はおよそ 5-10 倍程度であり、15 時から 18 時の間で帯域の変化が大きいことがわかる。よってその時間帯に 754Mbyte のファイルを各方式 10 回試行し、平均値を比較した。評価対象は、先行研究で提案されている重複再要求のみを実装したもの (NORMAL) と、重複再要求に加えて提案方式である差分計測を用いた遅延要求方式を実装したもの (DIFF) の 2 種類である。評価項目は、ブロックの理想到達時刻からの遅れを表す遅延時間の最大値である初期バッファリング時間、バッファ内の非有効ブロックの数、ブロックの理想到達時刻からの正の遅れの平均である平均遅延時間である。実験結果を表 1 に示す。

表 1: 実験結果

方式	初期バッファリング時間 [s]	非有効ブロック数	平均遅延時間 [s]
NORMAL	6.203	6.435	1.550
DIFF	3.130	3.730	0.866

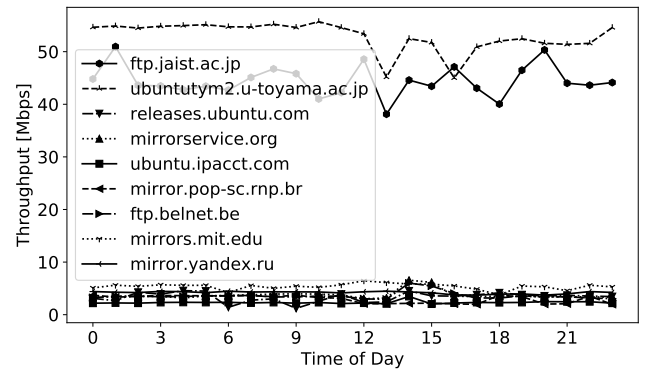


図 2: ミラーサーバにおける帯域性能の 24 時間変化

表 1 より、公開ネットワークでの評価において差分計測を用いた遅延予測方式は重複再要求の場合と比較して、約 50% の初期バッファリング時間の短縮、約 30% の非有効ブロック数の削減、約 30% の平均遅延時間の短縮できたことが確認できる。

### 5 まとめ

複数のサーバから同一のファイルを並列に分割ダウンロードする際の順序逆転の発生を抑制する要求方式を提案し、実装した HTTP クライアントをテストベッドと公開ネットワークで評価し優位性を確認した。今後の課題としては、プロキシとして組み込んだシステムでの実際の動画の視聴体験まで含めた評価などが考えられる。

### 参考文献

- [1] Juhoon Kim, Yung-Chih Chen, Ramin Khalili, Don Towsley, Anja Feldmann, "Multi-source Multipath HTTP (mHTTP): A Proposal," ACM SIGMETRICS Performance Evaluation Review, vol.42, no.1, pp.583-584, 2014.
- [2] Junichi Funasaka, Atsushi Kawano, and Kenji Ishida: Adaptive Parallel Downloading Method for Proxy Systems, IEICE Trans., Vol.E90-B, no.4, pp.720-727, 2007.