

フロー特定による選択的パケットキャッシュを用いた P2P トラフィック抑制手法

東京大学大学院 学際情報学府総合分析情報学コース
佐々木 健吾, 中尾 彰宏

1 背景と目的

近年, P2P ネットワークモデルを採用したアプリケーションのトラフィックが増加の一途を辿っている. 計測結果 [2] によると, インターネット上の 50-80% のトラフィックは P2P ファイル共有アプリケーションが生成しており, その多くは冗長*1である. 『オブジェクトレベルキャッシング [1]』や『リダイレクション [3]』のような P2P トラフィック抑制手法が存在するものの, P2P アプリケーションで利用するためには, P2P アプリケーションごとに変更を加える必要があり, 汎用的にトラフィックを抑制できないという問題がある.

本研究では, P2P トラフィックが時間的, 空間的局所性を持つという特徴に着眼し, パケットキャッシュを用いた P2P トラフィック抑制手法を提案する. 同時に, パケットキャッシュを効率よく行うために, キャッシュを行うフロー*2を BloomFilter (以下, BF) を用いて特定し, データをキャッシュする手法も提案する.

2 提案手法

2.1 パケットキャッシュを用いた P2P トラフィック抑制手法

本研究で提案するトラフィック抑制手法は, ルータ上で実現される. ルータに P2P アプリケーションのパケットが転送されると, Prefix*3を基準にペイロード部を分割し, データピース*4を構成する. そのデータピースが初めて転送されるデータピースなら, データピースとそれに対応するハッシュ値のマッピングをルータのキャッシュに構成する.

キャッシュにマップが存在するデータピースなら, ハッシュ値に置換される. データピースがハッシュ値に置換されたパケットを用いて ISP 間通信を行うことで, P2P トラフィックを抑制できる. ルータに, データピースがハッシュ値に置換されたパケットが転送されると, キャッシュ上のマップを用いてデータピースに復元する.

本手法は, Prefix さえ分かれば, P2P アプリケーションだけでなく, どのようなサービスにも適応可能である. 例えば, クラウドプラットフォームにてホスティングされたネットワークサービス (動画配信, SaaS など) ように, 冗長トラフィックを発生する可能性が高いサービスなら, トラフィックの抑制が可能である.

*1 同じデータ転送を繰り返し行っている

*2 sourceip, port, destination ip, port から構成されるデータ転送の流れ

*3 アプリケーションの持つ固有のビット列

*4 分割によりできるデータの断片

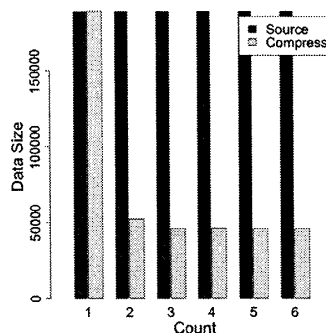


図 1: パケットサイズの変化

2.2 BF を用いたフロー特定による選択的なキャッシュ手法

BF は, 要素が集合のメンバであるかを判断するために用いられるビット列であり, ビット演算を用いた高速な処理が可能で, 少ないメモリ消費でメンバの有無を判断できる.

BF を用いたフロー特定では, まず全フローのデータピースを書き込むことができるサイズ (*bfs*ize) の大きい BF を用意する. あるフローのデータピースが BF を参照したときに, 参照先のビットが TRUE だったら, 他のフローが同じデータピースを転送したと判断する. 参照先のビットに FALSE が含まれている場合, TRUE を書き込む.

異なる n 個のデータピースで, 他のフローが同じデータピースを転送したと判断されたフローは, 同じデータを送っているフローが他に存在すると判断され, データピースのマッピングのキャッシュを開始する.

フロー特定を行うことで, 同じデータを転送していないフローのデータピースはキャッシュされないため, メモリ消費を抑制できる.

3 実験パケットレベルキャッシングを用いたトラフィック抑制に関する実験

3.1 実験概要

BitTorrent を用いて, 同じデータを複数回転送するシミュレーションを行った. データピースのサイズは 128 バイトで, ハッシュ関数には MD5 を用いたため, ハッシュ値のサイズは 16 バイトである.

3.2 実験結果

データ転送時に発生するトラフィック量の変化の様子を, 図 1 に示す. x 軸がデータ転送を行った回数 (count) であり, y 軸がデータ転送に必要なトラフィック量である. Source, Compress はそれぞれ圧縮前, 圧縮後のトラフィック量である.

一度目のデータ転送では, ルータ上にキャッシュを作る過程が必要であるため, 圧縮は行われず, 圧縮に必要なパラメー

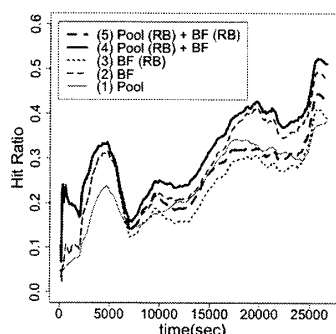


図 2: キャッシュヒット率の変化

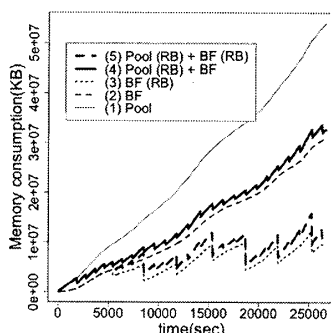


図 3: メモリ消費の変化

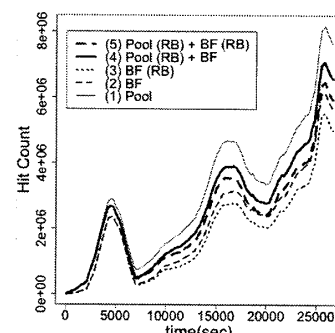


図 4: キャッシュヒット数の変化

タの転送により、多くのトラフィックが発生した。しかし、二度目のデータ転送では、トラフィック量を元の約 27% に圧縮し、三度目以降は、余分な通信も発生せず、トラフィック量も元の約 24% に圧縮できた。二度目、三度目のデータ転送を比べてみると、圧縮されていないデータピースが、二度目のデータ転送には存在している。これは、一度目の通信で分割されていたデータピースである。そのデータピースは二度目の通信でキャッシュされ、三度目の通信では圧縮されているため、三度目以降の通信は二度目よりも圧縮率が高い。

4 フロー特定に関する実験

4.1 実験概要

表 1 の実験パターンについて、キャッシュヒット率、メモリ消費、キャッシュヒット数を計測する実験を行った。計測を行った対象は、BitTorrent のログデータ 100GB 分である。表 1 の (RB) は、参照ビットのことであり、Pool(RB) では 1000 秒、GBF(RB) では 3000 秒毎に使用されていないマップを削除する。(1) は、フロー特定を行うことなく、全てのマップをルータ上にキャッシュする。(2) と (3) は、提案手法の BF を用いてフロー特定を行い、特定されたフローのマップのみキャッシュを行う。(4) と (5) は、Pool と GBF を組み合わせた手法であり、パケットを受信すると、まず、Pool(RB) 方式でキャッシュを開始する。同時に、BF 方式でフローの特定を開始する。フローが特定されたら、そのフローに関するマップは、BF 方式のキャッシュにキャッシュされる。 n は 30 であり、 $bfsiz$ e は 2^{27} バイトである。

表 1: フロー特定のパターン

#	Pattern
(1)	Pool
(2)	BF
(3)	BF(RB)
(4)	Pool(RB)+BF
(5)	Pool(RB)+BF(RB)

4.2 実験結果

時間変化に対する (1)~(5) の手法のキャッシュヒット率、メモリ消費、キャッシュヒット数の変化を、図 2, 3, 4 に示す。x 軸が時間変化を、y 軸がそれぞれキャッシュヒット率、メモリ消費、キャッシュヒット数を示す。

(2) は、(1) と比べてキャッシュヒット率が高く、メモリ消費も約半分であるため、使用される確率の高いデータを選択できている。さらに、Pool 方式と BF 方式を組み合わせた (4)

は、(2) で見逃したキャッシュヒットを補えるため、キャッシュヒット率が大きく向上する。しかし、メモリ消費は、(1) と比べ抑制されているものの、時間の経過に比例して、メモリ消費が増大してしまう。

それに対し、(3) のメモリ消費は一定で、(1) の約 15% であるものの、(1) とキャッシュヒット率が同等であり、キャッシュヒット数は (1) の約 62% に抑えられている。

そこで、(3) に Pool(RB) 方式と組み合わせた (5) を検証する。(5) は、(1) と比べキャッシュヒット率が最大で 19%、平均しても 4% 向上し、キャッシュヒット数も (1) の 84% を補うことができている。メモリ消費は、(3) と同等である。

以上の結果から、(5) が総合的に優れていると考えている。

5 結論

本研究では、第一にパケットキャッシュを行うトラフィック抑制手法を提案した。結果として、ISP 間トラフィックを元の約 24% に抑制することができることを示した。

第二に、BF を用いたフロー特定による選択的キャッシュ手法も提案した。結果として、フロー特定を行わないキャッシュ手法と比べ、キャッシュヒット数を大きく減らすことなく、キャッシュヒット率が最大で 19% 向上し、メモリ消費も 15% に抑制できることを示した。

以上から、本提案手法を用いて、汎用的でメモリ効率の良いトラフィック抑制を実現できる。

今後の課題として、(1) データピースの断片のキャッシュを用いた圧縮効率の向上、(2) 多対多環境での実験、(3) 実環境での実験、(4) 最適なパラメータの自動推定が挙げられる。

謝辞

本研究の一部は、総務省の委託研究『セキュアクラウドネットワーク技術の研究開発』と科研費 基盤研究 (B) 課題番号 21300020 『大規模分散環境を用いた P2P ネットワーク流通ファイル制御システムの研究』の成果である。

参考文献

- [1] Mohamed Hefeeda, Cheng-Hsin Hsu, and Kianoosh Mokhtarian. pcache: A proxy cache for peer-to-peer traffic. *SIGCOMM*, 2008.
- [2] H. Schulze and K. Mochalski. Internet study 2009. *Technical report, ipoque GmbH*, 2009.
- [3] Haiyong Xie, Arvind Krishnamurthy, Avi Silberschatz, and Y. Richard Yang. P4p: Explicit communications for cooperative control between p2p and network providers. *DCIA*, 2008.