多様な通信環境に適用可能なデータ転送高速化技術

亀山裕亮[†1]　佐沢真一[†1]　橋間正芳[†1]

近年，企業における業務システムのクラウド化やデータセンタ(DC)へのサーバ集約により，複数の DC 間や，端末と DC 間におけるデータ転送が急増している．これらのデータ転送を高速化するための一般的な解決策として，回線増強や高速化専用ハードウェアの導入が考えられるが，コストの増加や，クラウド環境や仮想環境，モバイル環境へ柔軟に適用することが困難であるなどの課題があった．これに対して我々は，重複除去と圧縮により転送データ量を削減し，ソフトウェアだけでデータ転送速度を大幅に高速化する技術を開発した．本技術はソフトウェアとして実現しているため，既存のサーバや OS 上に搭載することが可能で，ノート PC，タブレット端末，スマートフォンなどのモバイル端末でも利用可能である．重複除去では一度送信したデータを送信側と受信側の双方で保存しておく必要があるため，モバイル端末のような記憶容量が限られたデバイスを利用する場合には重複除去率が下がるという課題があったが，出現頻度を利用した容量削減技術により高い重複除去性能を維持することが可能となった．本技術をモバイル端末に適用し共有サーバからのファイル転送評価の結果，従来技術と比較し，重複除去の成功率をほとんど劣化させることなく，メモリ使用量を 1/3~1/4 に削減できることを確認した．本論文では我々が開発したデータ転送高速化技術について述べる．

WAN acceleration technology applicable to various network environments

HIROAKI KAMEYAMA[†1]　SHINICHI SAZAWA[†1]　MASAYOSHI HASHIMA[†1]

1. はじめに

業務システムのクラウド化やデータセンタへのサーバ集約により，複数のデータセンタ(DC)間や，端末と DC 間におけるデータ転送が急増しており，このようなデータ転送を高速化することが重要な課題となっている．この課題を解決するため，データ転送を高速化する WAN 高速化技術が注目されている．WAN 高速化技術には大きく 2 つの技術がある．

1 つはトランスポート最適化技術である．データ転送に一般的に用いられる TCP(Transmission Control Protocol)では帯域が十分あっても遅延やパケットロス率が大きい回線では輻輳制御の影響で帯域を有効利用できていない．そこで輻輳制御方式の改良や，輻輳制御を行なわない UDP(User Datagram Protocol)を利用することで帯域を有効利用するトランスポート最適化技術が提案されている [1].

2 つ目はデータ最適化技術である．トランスポート最適化は遠距離通信のように帯域を有効利用できていない場合には効果が大きいが，近距離や帯域が細いなど，既に帯域を有効に利用できている場合には効果的ではない．そこで圧縮や同一データの転送を抑止する重複除去により，転送データそのものを削減するのがデータ最適化技術である．インターネットのトラフィックの重複は 15%-60%からあるという調査結果もあり [2]，データ最適化による転送データ量の削減およびそれに応じたデータ転送速度の高速化が期待できる．

我々はデータ最適化技術として，ネットワーク上を流れるデータの中で統計的に出現頻度の高いデータだけを選択し，優先的に保存する重複除去技術 DMA(Deduplication with Memory Adaptive method)，送信データサイズ，ネットワークの利用可能な帯域，端末の CPU 性能などの情報を定期的に収集することで高速化効果を予測し，その結果から重複除去や圧縮処理の実施有無を自動的に切り替える DNA(Deduplication with Network Adaptive method)を開発した．

本技術をモバイル端末に適用し，共有サーバからのファイル転送評価を実施した結果，従来技術と比較し，重複除去の成功率をほとんど劣化させることなく，メモリ使用量を 1/3~1/4 に削減できることを確認した．本技術を用いることで，今後，ますます普及が見込まれるクラウドや仮想環境，モバイル環境における，様々な通信アプリケーションのデータ転送の高速化が期待できる．また，ソフトウェアとして実現しているため，既存のサーバや OS 上に搭載可能で，ノート PC，タブレット端末，スマートフォンなどのモバイル端末でも利用可能である．

2. 従来技術

ネットワークトラフィックの重複を削除する技術は従来からいくつか提案されているが，方法論として大きく 2 つに大別される．1 つはオブジェクトベースのものであり，1 つはコンテンツベースのものである．

†1 株式会社富士通研究所
FUJITSU LABORATORIES LTD.

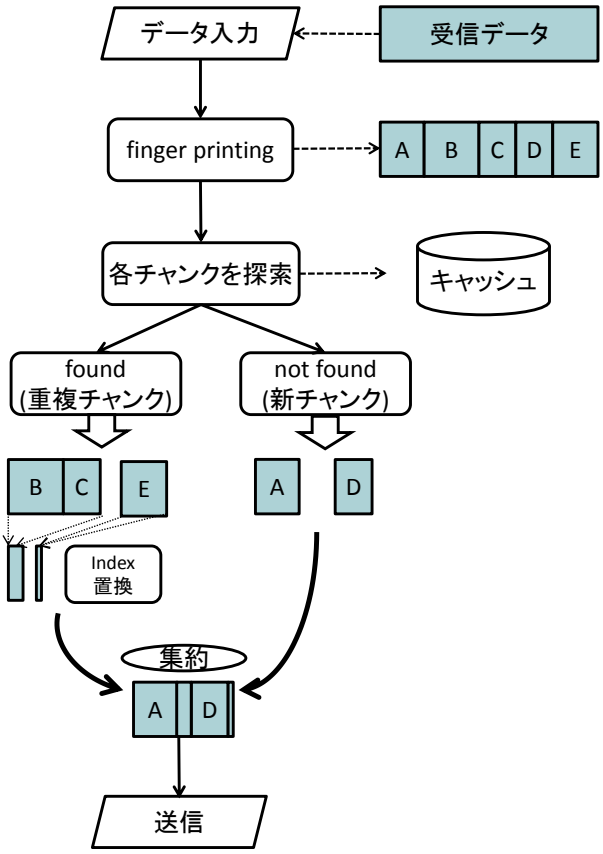

図 1 コンテンツベースの重複除去手法

オブジェクトベースの重複除去技術は，アクセス対象のオブジェクトのファイル名がわかっている場合に，そのファイルに更新がないかどうかを判定し，重複したデータの転送を抑止するものである．更新の有無はファイルの更新日付，サイズ，ファイル全体のチェックサムなどを指標として判断する．たとえば WEB の Proxy [3]や P2P のメディア共有システム [4]がある．これらのシステムはファイル全体として重複を検知しているため，ごく 1 部が更新された場合であっても全体が更新されたと判断してしまう．これに対して，Rsync [5]はデルタコーディングとチャンク分割とハッシュによりファイル内の更新された部分を検知することができる．しかし，Rsync でもファイルごとの更新を対象としているため，ファイル名が変更された場合には重複除去ができず，同じファイルを送ることになる．さらに，これらのオブジェクトベースの重複除去はプロトコルからファイル名を取り出すことが必要であり，必然的にプロトコル依存になる．

これに対してコンテンツベースの重複除去はプロトコルに関する情報なしに通信データのコンテンツのみから重複を検知し除去する方法である．この方法は Spring [6]によって提案されて以来，様々な応用が提案されている．
図 1 にこの手法の概略を示す．

コンテンツベースの手法では，重複を高速に検知するために元のデータをチャンクと呼ばれる可変長ブロックに分割する．このため，データを固定長のウィンドウでスキャンし，ウィンドウ内にあるデータのハッシュ値を Rabin fingerprint と呼ばれる方法で計算する [7].

ウィンドウサイズを n，ウィンドウ内のデータを $w_0, w_1, \ldots w_{n-1}$ とすると，Rabin fingerprint は以下のように表せる．

$$f_p(0,n) = \sum_{k=0}^{n-1} w_k \, p^{n-1-k}$$

ここで，p としては比較的大きな素数が使われる．この計算は本来 n 回の掛け算を必要とするが，1 回計算すれば次のバイトに移った時に前回の計算結果を利用して以下のように計算できる．

$$f_p(1,n+1) = f_p(0,n) \times p - w_0 p^n + w_n$$

この計算は 3 回の掛け算で実行でき効率的である．この計算で得られたf_pのうち，先頭(あるいは後ろ)の m ビットが決められたパターンになっているかのチェックを行う．パターンは基本的にどのようなパターンでもよく，m ビットが一致する確率は，f_pがランダムであるとすれば $1/2^m$ の確率なので，例えば m=10 とすれば平均して 2^{10}=1KB に 1 回の割合でデータのバウンダリが得られる．

次に個々のチャンクに対して SHA1 などの実用的に衝突確率を無視してよいハッシュを計算し，これをキーとしたハッシュテーブルから重複を探索する．

ネットワークトラフィックの重複除去においては，送信側では送信するデータから重複が検出された場合はそれを適切なメタ情報（重複データの SHA1 もしくは過去の送信を特定できるインデックス）で置き換えてから送信する．一方，受信側では受信したデータをキャッシュしておき，メタ情報を受信した場合は自分のキャッシュから元のデータを復元する.コンテンツベースの重複除去はプロトコル非依存であるため，自分の通信だけでなく，他のアプリの通信や，逆方向の通信，さらに別のプロトコルで行われた通信に対しても重複を除去することが可能である．

上述の方式は，送信側と受信側でキャッシュの同期がとれていなければ信頼性のある転送を実現できない．ところが，キャッシュの容量は有限であるため，適当なキャッシュ置換ポリシーに基づき，古いチャンクをキャッシュから削除する必要がある．これは通常 Least Recently Used(LRU)を通常用いる．これはネットワークのトラフィックについての統計的な調査によれば [8]，チャンクの重複は近い時間内に発生する可能性が高いと考えられるためである．

以上の重複除去技術は重複したデータを送ることに起因するコストが，重複を検知，管理，復元するコストよりはるかに高いことを前提としている．したがって，デバイスの性能やネットワークの速度によっては，重複除去は逆に性能を劣化させてしまう. このため, 上述の Fingerprinting やキャッシュの読込・書込と性能の関係については [9]において詳しくベンチマークがされている．

また，上記の重複除去技術は主にルータなどに搭載した形態をとっていたため，End-To-End での高速化ができなか

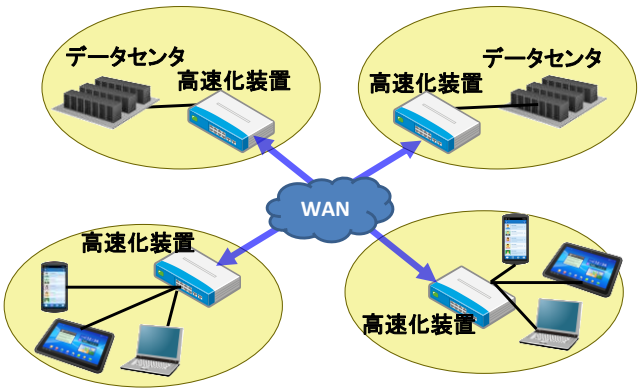

図 2 多様な通信環境の例

った，最近になって，例えば EndRE [10]はこのような課題に取り組み，End-To-End でのデータ最適化を目指している．EndRE は端末で動作させるいために軽量な SAMPLEBYTE 法を提案し，CPU コストの削減を行っている．

3. 多様な通信環境における重複除去

これまで重複除去技術は，主にストレージシステムやデータ転送高速化の専用ハードウェアなどに実装されてきた．しかし，今後 DC 間だけではなく DC と端末間のデータ転送に適用した場合，多様な通信環境での利用が想定される（図 2）．

例えば，遠隔地にあるファイル共有サーバへのアクセスでは，事務所からデスクトップ PC と有線ネットワークを利用する場合だけではなく，出張先からラップトップ PC と無線ネットワークを利用する場合など，1 つのサーバに対して複数のクライアント端末がそれぞれ異なるネットワーク経由で接続してくる多様な通信環境が存在する．

このような多様な通信環境にソフトウェアのみで重複除去を適用する場合，下記の課題が考えられる．

- 重複除去では一度送信したデータを送信側と受信側の双方で保存しておく必要があり，当然多くのデータを保存した方が重複を発見する可能性は高くなる．しかし，記憶容量は有限であり，特にモバイル端末のように記憶容量が限られている場合には重複除去率が下がってしまう．また，一般的なモバイル端末に搭載された CPU では，重複除去や圧縮のようなデータ最適化の処理による CPU 負荷が高い．
- 高速回線を利用している場合や，転送するデータのサイズが小さい場合，あるいは内容が毎回異なるような状況において重複除去や圧縮を適用すると，処理のオーバヘッドにより逆にデータ転送速度が低下してしまう場合があるが，利用者があらかじめ効果を予測することは困難である．

4. 開発技術

我々は多様な通信環境に適用可能な重複除去技術として，

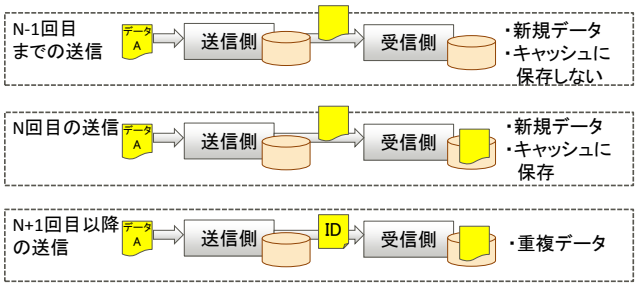

図 3 DMA の概念図

ネットワーク上を流れるデータの中で統計的に出現頻度の高いデータだけを選択し，優先的に保存する DMA(Deduplication with Memory Adaptive method)，および送信データサイズ，ネットワークの利用可能な帯域，端末の CPU 性能などの情報を定期的に収集することで高速化効果を予測し，その結果から重複除去や圧縮処理の実施有無を切り替える DNA(Deduplication with Network Adaptive Method)を開発した．本章では DMA 及び DNA の詳細について述べる．

4.1 DMA (Deduplication with Memory Adaptive method)

重複除去は重複チャンクの管理のためにメモリやディスクを多く必要とする．しかし，割り当て可能なメモリないしディスクはデバイスごとに異なる．このように接続している 2 つの重複除去のキャッシュリソースが非対称な場合，割り当てリソースの小さい方ではキャッシュ置き換えが頻繁に起こるため，重複が発生するまでの間隔が長いときに重複除去ができない問題がある．

この問題を解決するため，双方のキャッシュ容量に大きな非対称性がある場合，キャッシュの大きい方はチャンクの出現頻度を補助的に使用する方法を開発した．キャッシュの大きい方はキャッシュ開始出現頻度が一定値を超えた時点ではじめてキャッシュを行う．ただし，出現頻度をカウントするためにはどちらかが十分な容量を持つ必要がある．このため，この方式は，モバイル端末の容量が少ない端末が，クラウド上のデータセンタなどにアクセスするケースを想定している．図 3 に本技術の概要を，図 4 に処理フローを示す．ここで重要なのは，キャッシュ開始閾値と我々が呼ぶ，キャッシュの開始タイミングを規定する値である．送信側は重複回数がキャッシュ開始閾値より少ない場合，受信側にキャッシュ不要を指示するフラグを付加して，実データを送信する．その後，重複回数がキャッシュ開始閾値に一致した場合にキャッシュ要として，実データを送る．これ以後の重複時はキャッシュ位置を指定する ID 相当のデータを送信し，重複を除去する．キャッシュ開始閾値により，重複除去の削減率と使用メモリのバランスを制御することができる．特にキャッシュ開始閾値を 1 とすれば，通常の重複除去と同じである．しかし，最適なキャッシュ開始閾値の決定は困難な問題である．現状はクライアント側とサーバ側のキャッシュ容量の比率や，同時接続数を考

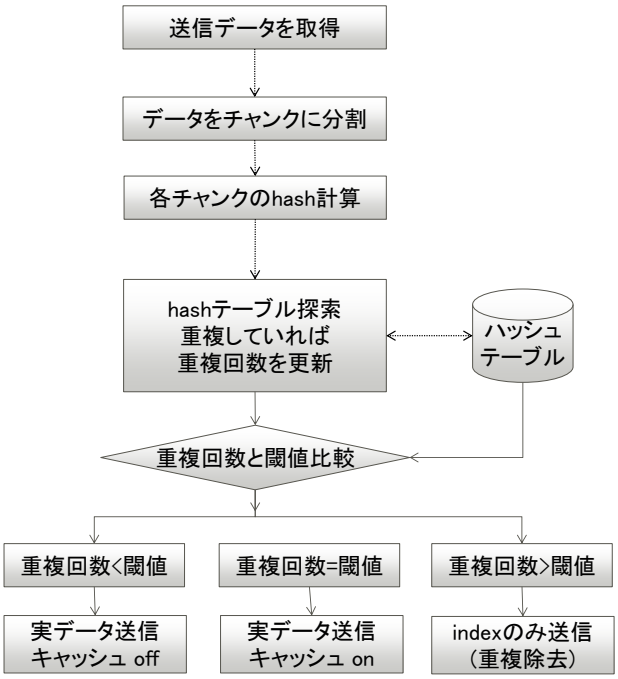

図 4 DMA の処理フロー

慮し，1～3 の値を設定可能としている．

4.2 DNA (Deduplication with Network Adaptive Method)

重複除去の効果はネットワークの空帯域の状況に大きく依存する．このため，我々はネットワークの可用帯域により，重複除去のリソース割り当てを調整する方法を開発した．

可用帯域はテストパケットを定期的に送るなどの手法で推定可能である．一方，各デバイスの重複除去速度を測定し，両者の比較により，可用帯域の方が重複除去速度よりも高い場合は，スループットは向上せず，逆に劣化する．このため，トラフィックの削減を無視してよく，転送速度のみが重要な場合や，可用帯域が十分にある場合は重複除去をキャンセルしている．なお，1 回の送信データが小さいショートパケット的な状態においても重複除去をキャンセルする．これにより，端末の CPU 能力に応じて帯域の変化に追従しながら有効なタイミングでのみ重複除去を行うことが可能である．図 5 に DNA の概念図を示す．

5. データ最適化アーキテクチャ

全章で提案した技術を我々は JAVA 上で WAN 高速化ソフトウェアとして実装した．高速化対象アプリと本ソフトウェア間の TCP 通信を終端し，終端された TCP 通信で受信したデータは可能であれば重複除去された上で，対向する WAN 高速化ソフトウェアに送信される．受信側では，復元されたデータを高速化対象アプリが本来アクセスすべき TCP アプリに転送する．これらの転送のルールは設定ファイルで事前に決められている．図 6 に本実装の概念図を示す．

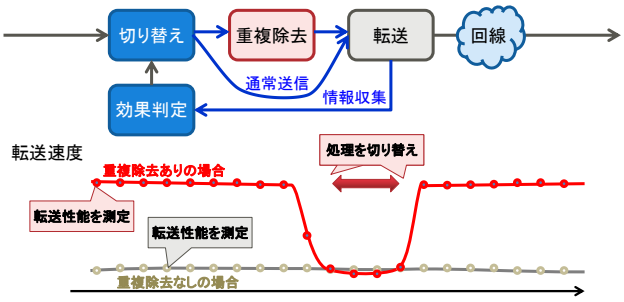

図 5 DNA の概念図

5.1 キャッシュ同期

今回の実装では高速化対象のアプリケーションを容易にカスタマイズできるようにするため，高速化アプリの宛先を本ソフトウェアに変更し，高速化アプリと本ソフトウェア間の TCP 通信を終端する構成をとった．このようにプロキシはアプリケーションのレイヤで行われるため，デバイスドライバなどのインストールが不要で，多様な端末で使用することができる．

また，複数のデバイス間でどのように同期をとるかが問題となるが，一般的にキャッシュの同期方法には大きく分けて optimistic なものと pessimistic なものがある．optimistic なアプローチは，キャッシュは基本的に同期していると考え，受信側でキャッシュミスが発生した場合，再送要求を行う．このため，同期がとれている間は無駄な通信は発生しないが，1 度同期がずれてしまうと，通信の往復が発生するため，レイテンシが非常に高くなる．pessimistic なアプローチはキャッシュを同期するために送達/削除確認を行う．これにより同期は保障されるがキャッシュ同期のための通信が必要になる．

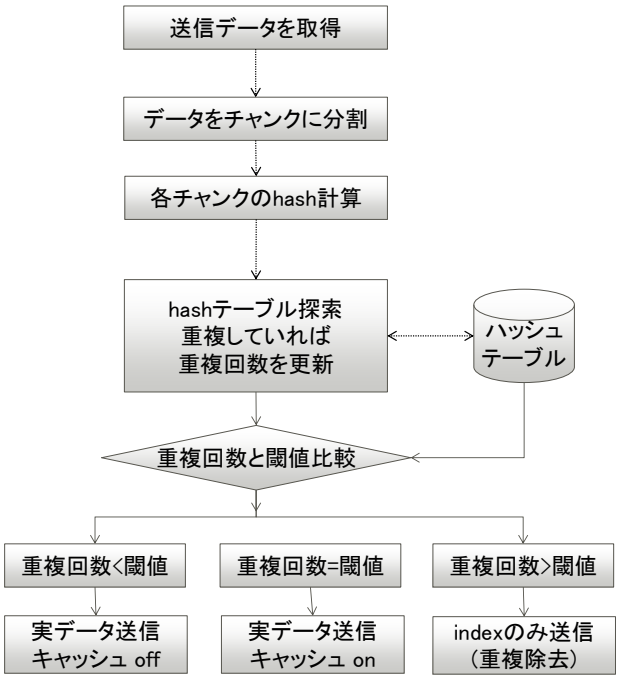

図 6 データ最適化システムの構成

今回の我々の実装はアプリレイヤでプロキシとして動作するので，ファイルのダウンロードキャンセルなどの場合に TCP がリセットされるとキャッシュの同期ずれが起こりやすいという（図 7）．また，モバイル通信を含めた複雑なネットワーク上でかつキャッシュの割り当ても端末ごとに不均等になるため，optimistic なアプローチでは例外的とされる状況が頻繁に発生し，これを回復するための通信が大きなオーバヘッドになると考えられる．このため，我々は pessimistic なアプローチを採用し，重複除去制御専用の TCP セッションを設け，このセッション内で，キャッシュの送達確認，およびキャッシュ削除の確認を行うこととした．

5.2 多拠点間 WAN 高速化

多拠点間の WAN 高速化装置でキャッシュの同期をとる場合，各装置は独立にキャッシュリソースを割り当てる．このため，一方の WAN 高速化装置のキャッシュ容量に余裕があっても対向の WAN 高速化装置によってチャンクが LRU アルゴリズムに基づき削除されることがありうる．このためキャッシュの削除時には重複除去制御用 TCP セッションにより，キャッシュの同期を行う．ただし，ローカルマシンのキャッシュ削除時に直ちにメモリから削除すると，キャッシュ削除通知と重複送信が行き違いになった場合，キャッシュミスが発生する．このため，削除時に完全にはメモリからチャンクを削除せず，trash に移動する．上記のような行き違いによりキャッシュミスが発生した場合は trash から探索する．これにより，多拠点間においてもキャッシュミスを原則なくすことが可能である．

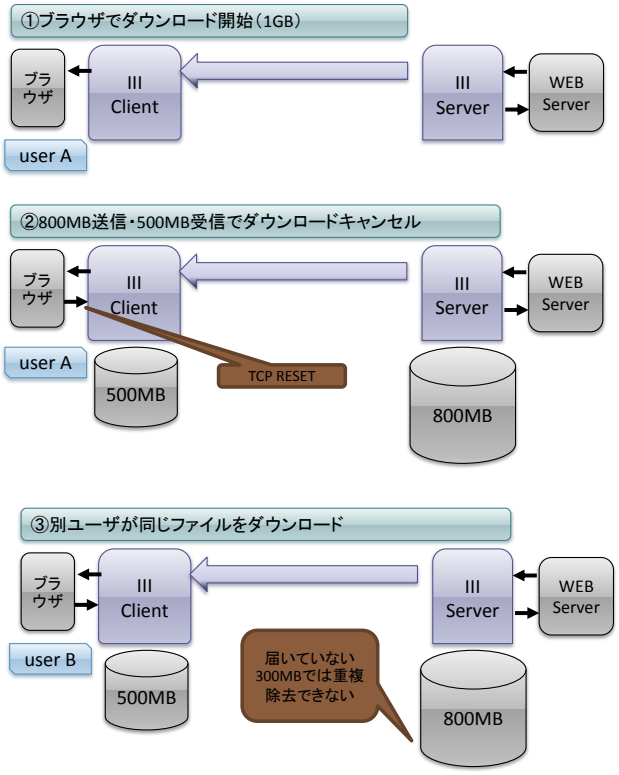

図 7 TCP reset 時のキャッシュ同期ずれ

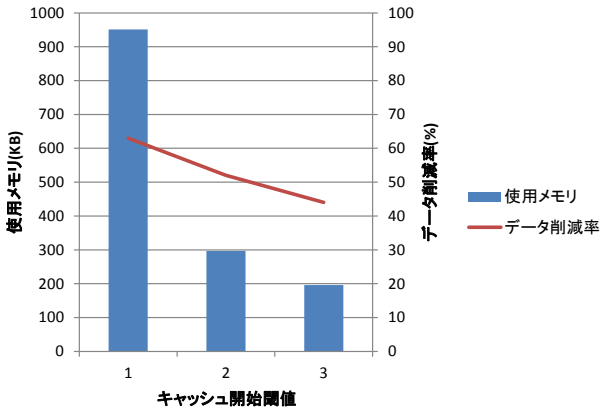

図 8 DNA の使用メモリと帯域削減率

6. 評価

今回開発した重複除去技術を評価するため，我々は企業における典型的なファイル共有のシナリオを設定し，帯域削減効果と使用メモリの相関関係および転送速度について測定を行った．使用した環境は Windows Server 2008 R2 ,Xeon 3.3GHz，メモリは 8GB で，帯域の変動時の評価をするため，ネットワークエミュレータを用いた．

6.1 DMA 評価

ファイル共有の典型的な操作パターンを用意し，DNA 手法の評価を行った．DMA の性能は，キャッシュ開始閾値により決定される．このため，キャッシュ開始閾値を変えながら，使用メモリと帯域削減率の関係を評価した．この評価結果を図 8 に示す．本結果によれば，キャッシュ開始閾値を 1 から 2 に変更した場合，メモリの使用量は約 1/3 に削減できており，さらにキャッシュ開始閾値を 3 に上げるとメモリの使用量は約 1/4 になる．一方，データ削減率の低下は 63%から 53%さらに 44%と順次劣化はするが，データ削減率の低下より，メモリの削減効果の方が高いため，低メモリのデバイスに本 DMA 手法は有効であると考えられる．

6.2 DNA 評価

DNA 手法を評価するため，端末性能として，約 20Mbps 程度で重複除去できるようにチューニングした端末において，帯域を変えながらスループットの改善度を評価した．このため帯域をネットワークエミュレータで調整しつつ，CIFS ファイル共有にて 100MB のファイルを 2 回転送し，2 回目の転送速度により，実効スループットが物理帯域に対してどれだけ向上しているかを以下の式で算出した．

$$\text{Throughput Gain} = \frac{\left(\frac{\text{転送サイズ}}{\text{転送時間}}\right)}{\textit{物理帯域}(\textit{エミュレータで設定})}$$

DNA の評価結果を図 9 に示す．本評価によれば，DNA

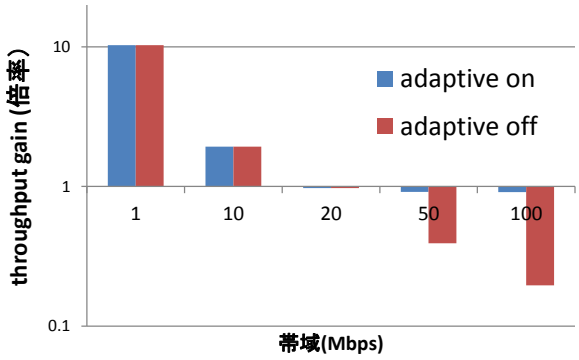

図 9 DMA 適用時のスループット

を組み込まない場合，帯域が高くなると急激に性能がかえって劣化するが，組み込んだ場合は劣化がほとんどない．組み込んだ場合でも残るわずかな劣化はプロキシによるオーバヘッドによるものと考えられる．

7. まとめ

本論文では，我々が開発したネットワークや利用可能なハード資源に自動適応しつつ重複除去を行う WAN 高速化技術について述べた．本技術はサーバからタブレット端末まで，同一のソフトウェアで実現され，通信帯域や自分もしくは接続先の空き容量に応じて，システム全体でほぼ最適なキャッシュリソースの割り当てを行うことが可能である．これにより，従来技術と比較し，重複除去の成功率をほとんど劣化させることなく，メモリ使用量を 1/3~1/4 に削減できることを確認した．また，モバイル端末のような CPU 性能が低い場合でも，重複除去機能の動作を低帯域のような効果がある場合に制限することにより，従来発生していた性能劣化を防ぐことができる．本技術を適用することにより、今後モバイル端末を含めた多様な通信環境においてデータ転送高速化が期待される．

参考文献

[1] 亀山裕亮, 佐沢真一 , 佐藤裕一, “WAN 高速化に適した転送プロトコルの提案,” 電子情報通信学会技術研究報告 情報ネットワーク 112(464), pp.167-172, 2013.

[2] Y. Zhang , N. Ansari, “On Protocol-Independent Data Redundancy Elimination,” Communications Surveys & Tutorials, IEEE (Volume:16 , Issue: 1), pp.455-472, 2014.

[3] “Squid Web proxy cache,” http://www.squid-cache.org/.

[4] “PeerApp,” http://www.peerapp.com/.

[5] “Rsync,” http://rsync.samba.org/.

[6] N. T. Spring , D. Wetherall, “A Protocol-Independent Technique for Eliminating Redundant Network Traffic,” *conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 2000.

[7] M. O. Rabin, “Fingerprinting by Random Polynomials, Center for Research in Computing Technology,” Harvard University, Tech. Rep. TR-15-81, 1981.

[8] A. Anand, “Redundancy in Network Traffic: Findings and Implications,” Proceedings of 11th International Joint Conference on Measurement and Modeling of Computer Systems, pp. 37-48, 2009.

[9] M. Martynov, “ Challenges for High-Speed Protocol-Independent Redundancy Eliminating Systems,” Proceedings of 18th International Conference on Computer Communications and Networks pp. 1-6, 2009.

[10] B. Aggarwal, “ EndRE: an End-System Redundancy Elimination Service for Enterprises,” Proc. 7th USENIX conference on Networked systems design and implementation, pp. 28-28, 2010.