

不揮発性ネットワークのための順序制御機能の実装と評価

今野 翔太^{1,a)} 今井 信太郎¹ 北形 元² 新井 義和¹ 猪股 俊光¹

概要: 災害時の情報収集に Web ページへのアクセスは重要である。しかし、災害時にはネットワークやサーバに障害が発生するケースが多く、それらに対して利用者が頻繁にアクセスを繰り返すことによりさらに障害が発生する悪循環が起こる (以降、災害輻輳と呼ぶ)。この利用者による頻繁なアクセスは、「通常の Web サーバの応答がない場合にリクエストが消えてしまう」ことに起因している。この問題を解決するために、Web サーバへのリクエストが消えない仕組みである不揮発性ネットワーク [1] が提案されている。本研究では、不揮発性ネットワークの持つ順序制御機能についてプロトタイプの実装、評価実験を行った。その結果、ネットワークの状態が悪い場合や、利用者の頻繁なアクセスを繰り返す場合に、順序制御機能が効果的に動作することを確認することができた。

1. はじめに

災害時の情報収集に Web ページへのアクセスは重要である。しかし、災害時にはネットワークやサーバに障害が発生するケースが多く、それらに対して利用者が頻繁にアクセスを繰り返すことによりさらに障害が発生する悪循環が起こる (以降、災害輻輳と呼ぶ)。この利用者による頻繁なアクセスは、「通常の Web サーバの応答がない場合にリクエストが消えてしまう」ことに起因している。この問題を解決するために、Web サーバへのリクエストが消えない仕組みである不揮発性ネットワーク [1] が提案されている。

不揮発性ネットワークは、大規模災害時における通信需要の極端な増加、電力供給の停止、通信路の切断など、極めて厳しい条件下において円滑な通信を実現する通信方法である。大規模災害時には、極端な通信需要の増加状態の継続により輻輳が緩和されず、大規模な通信障害が長時間に渡り続く。このような不安定な通信品質の下で、中継ノードでデータを一時蓄積しながら中継していく技術として DTN(Delay Tolerant Networking)[2] が提案されている。また、DTN を災害時の情報共有に利用する研究 [3][4] も提案されており、利用者自身の安否についてや、家族や知人へのメッセージを中央サーバへ送信し、情報を地域内の利用者同士で共有するような仕組みを構築し、実際の地

域での運用を行っている。しかし、DTN がその性質上、送信のみの単方向通信であるという点から、それを利用した手法では、サーバが情報を収集し、利用者同士で情報を共有するような用途には使用できるが、サーバから特定の利用者へ情報を伝達することは困難である。災害時には、知人の安否情報やインターネットを通じた避難情報などを入手することは重要であるが、既存手法ではこのような要求に十分に対応することは難しい。

不揮発性ネットワークには、Web サーバへのリクエストが消えない仕組みを実現するための以下の機能がある。

- 利用者からの Web サーバへのリクエストを抽出・分離するためのセッション分離機能
- 抽出・分離したリクエストを保存し、利用者へ再接続時に応答結果を提供するためのセッション永続化・復元機能
- 保存されたリクエストを適切な順序や速度で送信する順序アクセス制御機能

本研究では、順序アクセス制御機能を対象とする。この機能は、災害輻輳時により多くの利用者に Web ページ等の情報を伝達することを目的としている。この目的を実現するため、利用者からの頻繁な再リクエストやネットワーク帯域を圧迫するようなリクエストに対処する仕組みを構築する。

本稿では、2 節で関連研究について述べ、3 節では不揮発性ネットワークの概要と本研究の目的について述べる。4 節で提案手法を述べ、5 節で実装した提案手法の評価実験と実験結果について述べる。6 節で本研究のまとめを述べる。

¹ 岩手県立大学
Iwate Prefectural University Graduate School of Software and Information Science

² 東北大学 電気通信研究所
Tohoku University Reserch Institute of Electrical Communication

a) g231m016@s.iwate-pu.ac.jp

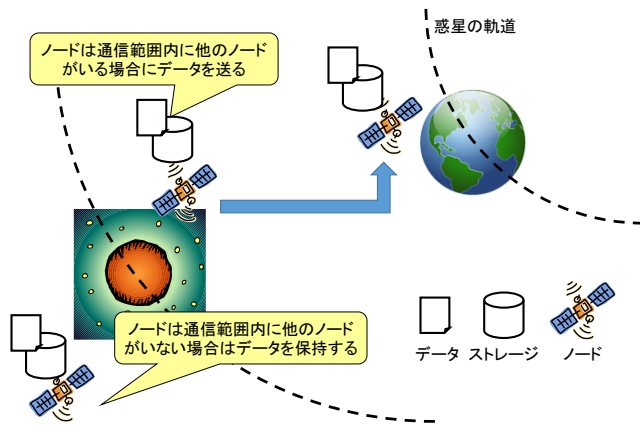


図 1 Delay Tolerant Networking

2. 関連研究

不揮発性ネットワークの関連研究として、DTN(Delay Tolerant Networking)[2]がある。惑星間インターネット通信の手段として研究されているこのネットワークは、大きな伝送遅延やネットワーク切断の多発する、通信環境が非常に悪い場合でも信頼性のあるデータ転送を実現するための通信方式である。普段我々が使用している TCP/IP は、エンド・ツー・エンドの物理的なリンクが安定しているという前提で成り立っているため、このリンクが少しでも途切れただけで通信は失敗となる。これに対し DTN は、物理的なリンクが常に存在しているとは限らないことを前提とし、バンドルと呼ばれるプロトコルをトランスポート層の上に乗せたオーバーレイ・ネットワークを構成する。このバンドルは、中継地点でデータを保持しながら、通信可能となった時点で、URI により指定された宛先ノードにデータを転送する。

一方、DTN の回線断に対する耐性の高さに着目し、DTN を用いて災害時の情報共有を行う研究 [3] や、移動端末を中継ノードに見立てた安否確認システムに関する研究 [4] など、DTN を災害時の情報通信に活用する研究が行われてきた。しかし、DTN が性質上、単方向通信のみの通信であり、また利用者端末に固定的な URI (宛先) が付与されていることが大変稀であるため、サーバから特定のノードへ情報を伝達するような利用をすることは難しい。このため、これらの既存研究では災害時の通信要求を満たす円滑な通信を実現することは困難である。

3. 不揮発性ネットワーク

3.1 不揮発性ネットワークの概要

不揮発性ネットワークの概要を図 2 に示す。不揮発性ネットワークは、以下の 3 つの機能からなる。

- (F1) セッション分離機能
- (F2) セッション永続化・復元機能
- (F3) 順序アクセス制御機能

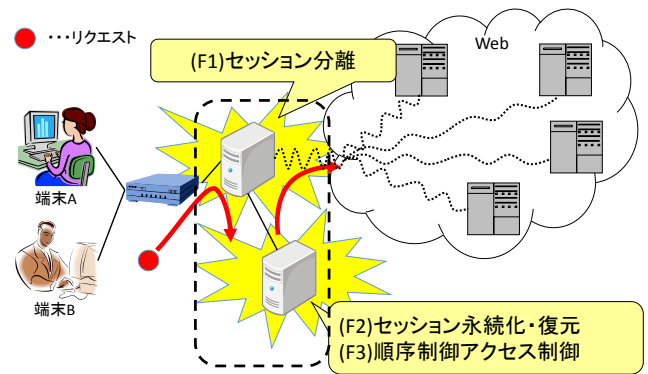


図 2 不揮発性ネットワークの概要

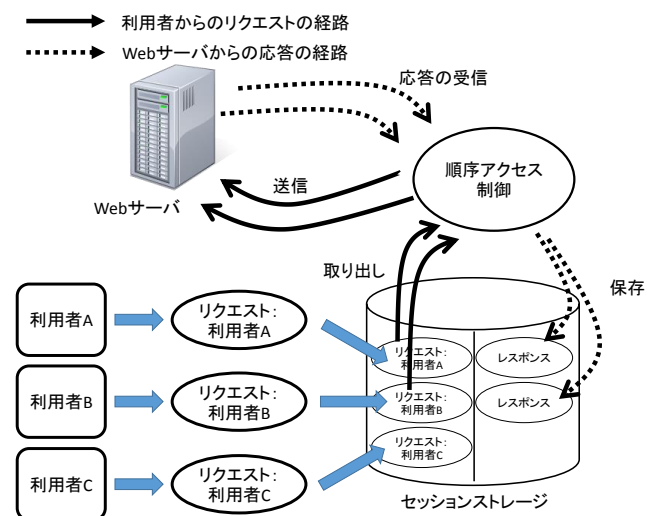


図 3 セッション永続化

利用者端末からリクエストが発行され、Web サーバの情報閲覧できる状態になるまでの過程を「セッション」とする。利用者端末をネットワークから切断することを可能にするには、セッションが永続化されている必要がある。そのために、ネットワーク中に流れるトラフィックから、永続化させるセッションのトラフィックのみを抽出、分離する必要がある。これを実現するのが (F1) セッション分離機能となる。またこの機能は、平常時と緊急時でトラフィックの流れを変更し、緊急時のみ不揮発性ネットワークを利用することを可能にする。

(F2) セッション永続化・復元機能では、セッションを永続化させるために、図 3 のように利用者端末から発行されたリクエストがセッションストレージに保存される。保存されたリクエストは定期的に Web サーバへと接続され、Web サーバから応答が得られるまで継続される。Web サーバから得られた応答は、利用者から発行されたリクエストと紐づけて保存される。この時点で永続化が完了し、利用者端末のネットワークからの切断を可能にする。また、一度ネットワークから切断された利用者端末は、再度ネットワークに接続し、同じリクエストを発行した時、セッションストレージに保存された Web ページの応答を得ること

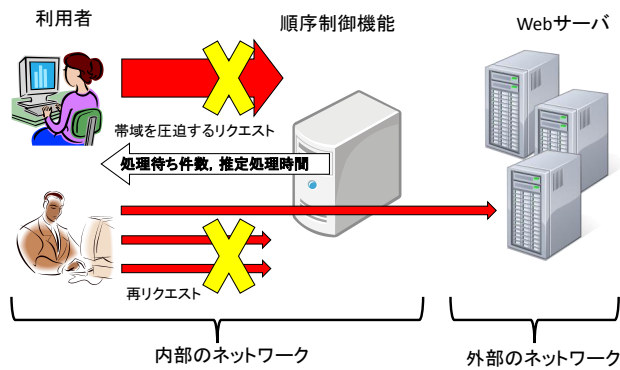


図4 順序制御機能の概要

ができる。利用者はWebサーバからの応答を待つ間通信を維持する必要がなくなり、端末をネットワークから切断することが可能になる。したがって、通信の維持で消費される電力を節約できるため、災害時において残量が限られたモバイル機器のバッテリーを効率よく使用することができる。

3.2 本研究の対象

本研究では、不揮発性ネットワークの3つの機能のうち、(F3) 順序アクセス制御機能（以降、順序制御機能と呼ぶ）について手法の提案、評価を行う。この機能は、災害輻輳時により多くの利用者にWebページ等の情報を伝達することを目的とし、頻繁な再リクエストやネットワーク帯域を圧迫するようなリクエストを抑止する。頻繁な再リクエストは、ネットワークのトラフィックを増やし、輻輳の状況をさらに悪化させてしまう可能性があるため、対処する必要がある。ネットワーク帯域を圧迫するようなリクエストは、利用者の数が少なければ問題にはならないが、利用者の数が多い場合、他の利用者が応答を受け取るための通信帯域を狭めてしまうため、対処する必要がある。

4. 順序制御機能

本研究では、順序制御機能サーバ以降のネットワークやWebサーバに障害が発生しており、順序制御機能サーバ以前のネットワークには障害が発生していない環境を想定している。図4に順序制御機能の概要を示す。

順序制御機能は頻繁な再リクエストや、ネットワーク帯域を圧迫するリクエストを抑止する。また、投入した特定のリクエストが処理されるまであと何件待ちかを返答する機能と、リクエストの処理に必要な時間がどのくらいかを推定して返答する機能を持つ。これは、(F2) セッション永続化・復元機能と連携して、利用者に再度アクセスする時刻の目安を提示するための機能である。

図5に順序制御機能の処理のフローを示す。

1.1 リクエスト発行 順序制御機能サーバは、利用者からのリクエストを受信する。この受信したリクエストは、

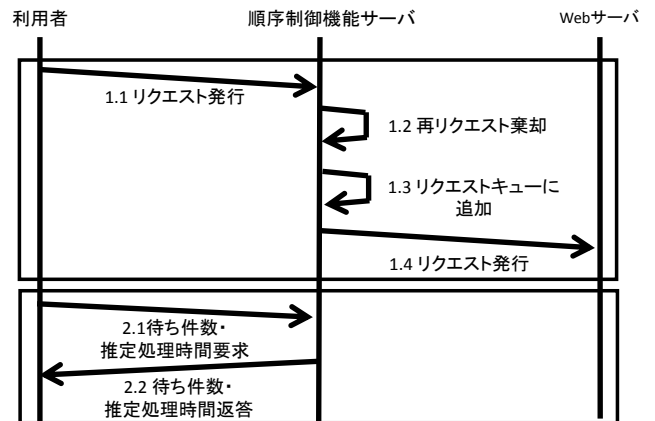


図5 順序制御機能のフロー

災害時は(F1)セッション分離機能から送信される。

- 1.2 再リクエスト棄却** 順序制御機能サーバは、受信したリクエストがリクエスト待ち行列内に既に存在する場合、再リクエストとして棄却する。この結果、ネットワークの負荷及びネットワークアクセスが減少される。
- 1.3 リクエストキューに追加** 受信したリクエストをリクエスト待ち行列に追加する。
- 1.4 リクエスト発行** 順序制御機能サーバは、Webサーバにリクエストを送信する。送信するリクエストの数は、ネットワークの状態に基づいて、順序制御機能サーバによって決定される。
- 2.1 待ち件数・推定待ち時間要求** 順序制御機能サーバは、リクエストの処理時間を常に計測し、推定待ち時間や、あと何件待ちかの要求に対応する。
- 2.2 待ち件数・推定処理時間返答** 順序制御機能サーバは、受信したリクエストがどの程度時間がかかるかを利用者へ通知する。

5. 評価実験

順序制御機能の再リクエスト抑止が有効であるかを評価するために実験を行った。

5.1 実験環境

構築した実験環境を図6に示す。順序制御機能サーバは実験用Webサーバと利用者が操作するクライアントの間に設置した。実験環境はすべてLANケーブルによる有線接続とした。本実験では、10BASE-TのHUBを使用し、実験用Webサーバへのリンク速度を10Mbpsに制限している。

順序制御機能サーバで動作させる実験用サーバプログラムならびに利用者PCで動作させるクライアントプログラム、負荷用PCで動作させる負荷用プログラムの作成には、Java 1.7を使用した。実験用Webサーバの構築には、CentOS 6とApacheを使用した。

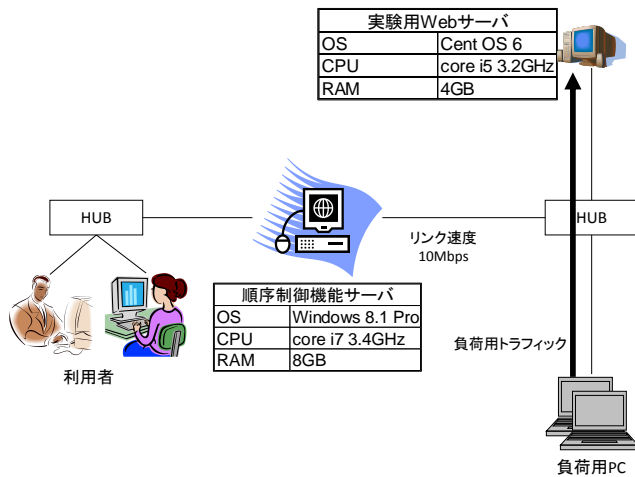


図 6 実験環境

5.2 クライアント

本研究では、クライアントとして一台の PC を使い、プログラム側で複数スレッドをたて、1 スレッドを一人の利用者としてカウントする。たとえば利用者を 50 人としたとき、実際には 50 スレッド (+メインスレッド) が一台のクライアント PC で動作していることになる。このとき、スレッドごとにユニーク ID を割りあて、これを再リクエストかどうか判定するためのキーのひとつとしている。本研究では、実験ごとに利用者数やリクエスト発行の間隔、再リクエストまでの待ち時間を変動させている。

5.3 順序制御機能サーバ

順序制御機能サーバは、4 節で述べたように、クライアントから発行されたリクエストを、Web サーバへの送信待ち行列に追加する。ここで、待ち行列に追加する際、追加しようとしているリクエストが再リクエストかどうか判定する。再リクエストである場合、リクエストは待ち行列に追加されずに破棄される。判定条件は、リクエストに含まれている利用者のユニーク ID と、宛先 URL の両方が一致するリクエストが、すでに待ち行列内に存在するかどうかとした。以降、リクエスト全体のうち、再リクエストではない元々のリクエストを「元リクエスト」と呼ぶ。また、本実験では、Web ページのキャッシュ機能を働かないようにしている。

順序制御機能サーバは、同時に複数のリクエストを Web サーバへの送信待ち行列から取り出す。取り出されたリクエストに対して、Web サーバへの送信、Web サーバからの応答の受信、Web サーバからの応答結果をクライアントへ送信、という操作を行う。実験では Web サーバへの送信待ち行列から取り出されるリクエストの数を変動させることで、利用者が Web サーバからの情報を得られるまでの待ち時間の評価を行う。

表 1 実験 1: クライアント・順序制御機能サーバ設定値

| | |
|----------------------|---------------------|
| 利用者数 | 10, 20, 30 … 100(人) |
| リクエスト発行間隔 | 5000(msec) |
| 再リクエスト待ち時間 | 5000(msec) |
| 順序制御機能サーバのリクエスト同時処理数 | 1(個) |

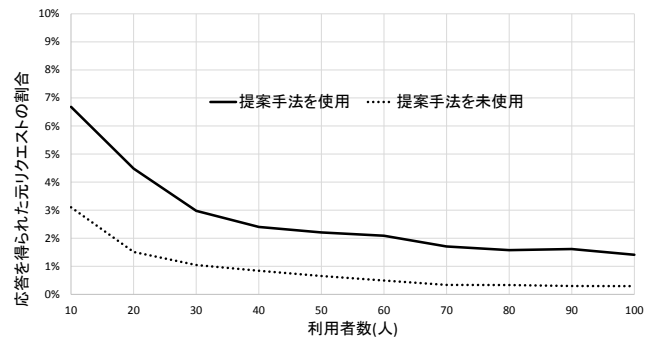


図 7 実験 1: 情報を得られた元リクエストの比較

5.4 ネットワーク負荷

ネットワークに負荷がかかっている状況の再現のために、負荷用 PC から実験用 Web サーバに対し、常に UDP プロトコルで送信可能な最大容量の packets を送信し続けることで、ネットワークに大きな負荷が掛かっている状況を再現した。このとき負荷用 PC のネットワーク使用率は 10Mbps 中、約 98% であった (約 9.8Mbps を負荷用 PC からの通信が使用している状態)。

5.5 Web サーバ

実験では、読み込み対象の Web ページはテキストと画像で構成された、約 2MB の一般的な容量の Web ページを使用した。全ての利用者はこの Web ページへのリクエストを発行するものとした。

5.6 実験 1

順序制御機能を使用する場合と使用しない場合で一定時間システムを稼働させ、その結果を比較することで機能の評価を行った。本研究ではより多くの利用者に情報伝達することを目的としているため、評価基準を「一定時間内に発行されたリクエスト中、Web サーバから情報を得ることができた元リクエスト数の割合」とした。

表 1 に、実験 1 のクライアントと順序制御機能サーバの設定値を示す。実験では一回の試行時間を 10 分とし、利用者数を 10 人から 100 人まで、10 人ずつ変化させたときの Web サーバから情報を得ることができた利用者数を計測した。利用者の再リクエストは、5000 ミリ秒経過しても情報が得られない場合に一度だけ発行するものとした。また、実験 1 では順序制御機能サーバはリクエストを同時に複数処理しないものとした。

図 7 に実験結果を示す。縦軸は、時間内に発行された元

表 2 実験 2：クライアント・順序制御機能サーバ設定値

| | |
|--------------------------|--------------|
| 利用者数 | 50(人) |
| リクエスト発行間隔 | 5000(msec) |
| 再リクエスト待ち時間 | 2500(msec) |
| 順序制御機能サーバの リクエスト同時処理数 | 2, 10, 50(個) |

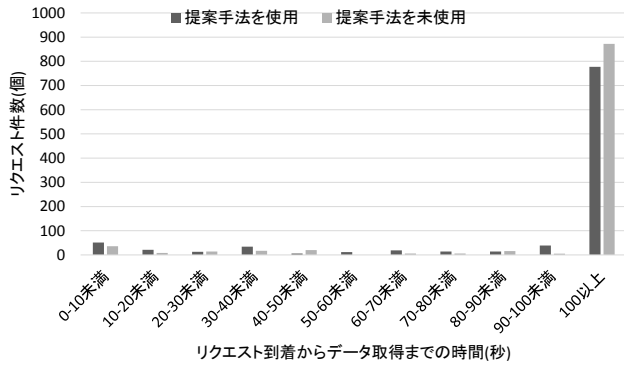


図 8 実験 2：リクエストの同時処理数 2 個の処理時間の分布

リクエストのうち、応答を得られたものの割合である。同図から、利用者数にかかわらず Web サーバから情報を得ることができた元リクエスト数の割合が、順序制御機能を使用する場合の方が上回っている事が分かる。

5.7 実験 2

実験 2 では、発行されたリクエストが順序制御機能サーバに到着してから、Web サーバの応答を得られるまでの時間の分布を比較した。表 2 に、実験 2 のクライアントと順序制御機能サーバの設定値を示す。この実験では利用者数を 50 人とし、一人当たり元リクエストを 20 個発行するものとする。したがって、発行される元リクエストの数は 1000 個となる。実験 2 では、順序制御機能サーバのリクエスト同時処理数を変動させる。また、再リクエストの発行までの間隔を 2500 ミリ秒とした。その他、再リクエストの回数や元リクエストの発行間隔などは実験 1 と同一である。

図 8 にリクエストの同時処理数が 2 個の場合の実験結果を示す。横軸は、元リクエストが順序制御サーバに到着してから、Web サーバから応答を得るまでの時間である。同図から、短時間で応答を得られた元リクエストは少ないことが分かる。また、提案手法の使用・未使用による違いは見られない。

同様に、図 9 にリクエストの同時処理数が 10 個の場合の実験結果を示す。同図では図 8 と比較して、短い時間で処理が終了している元リクエストがやや増加しているが、図 8 と同様に、提案手法の使用・未使用による違いは見られない。これは、順序制御機能サーバのリクエストの同時処理数が十分でなかったため、順序制御機能の有無に関わらず、実験結果に差が生じなかったためであると考えられる。

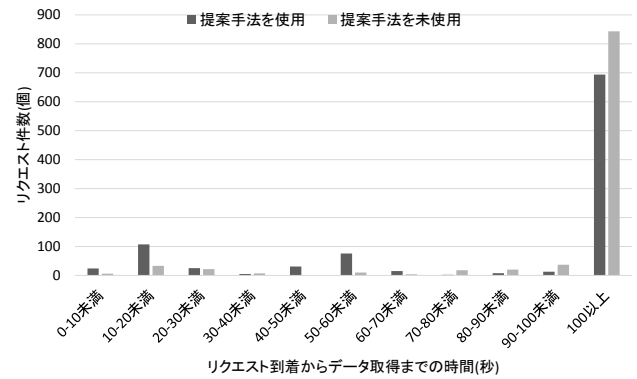


図 9 実験 2：リクエストの同時処理数 10 個の処理時間の分布

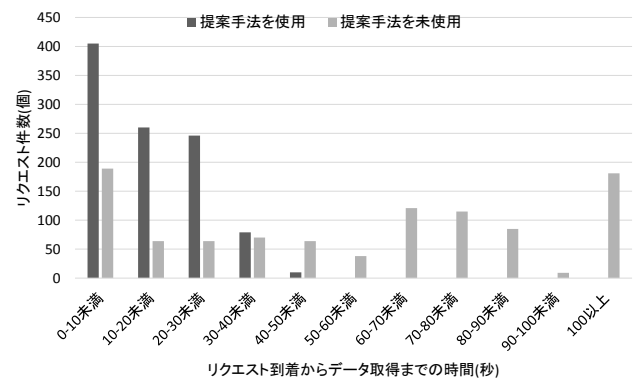


図 10 実験 2：リクエストの同時処理数 50 個の処理時間の分布

表 3 実験 3：クライアント・順序制御機能サーバ設定値

| | |
|--------------------------|----------------------|
| 利用者数 | 50(人) |
| リクエスト発行間隔 | 10, 20, 30 … 60(sec) |
| 再リクエスト待ち時間 | 2500(msec) |
| 順序制御機能サーバの リクエスト同時処理数 | 10(個) |

図 10 にリクエストの同時処理数が 50 個の場合の実験結果を示す。同図では提案手法を使用した場合は、提案手法を使用しない場合と比較して、短い時間で処理が終了している元リクエストが多いことがわかる。これは、提案手法により再リクエストが棄却され、順序制御機能サーバが Web サーバに送信するリクエストの総数が減少したためであると考えられる。

5.8 実験 3

最後に、利用者が発行する元リクエストの発行間隔を変動させたときの、リクエストが到着してから応答を得るまでの平均時間を比較した。表 3 に、実験 3 のクライアントと順序制御機能サーバの設定値を示す。この実験 3 では、利用者が発行する元リクエストの発行間隔を変動させることで、元リクエストが応答を得るまでの待ち時間を評価する。図 11 に提案手法を使用した場合、図 12 に提案手法を使用しない場合の実験結果を示す。横軸は、利用者が発行

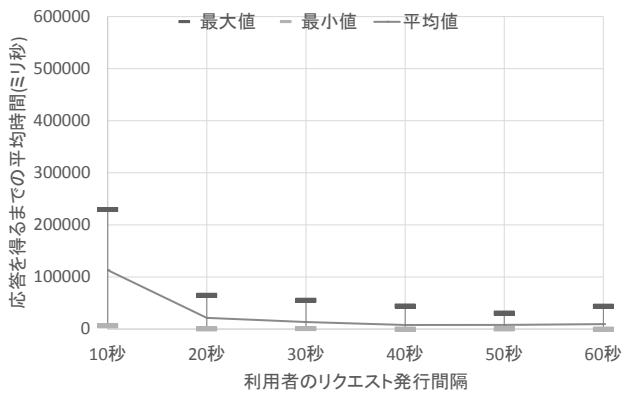


図 11 実験 3：提案手法を使用した場合の元リクエストが応答を得るまでの平均時間

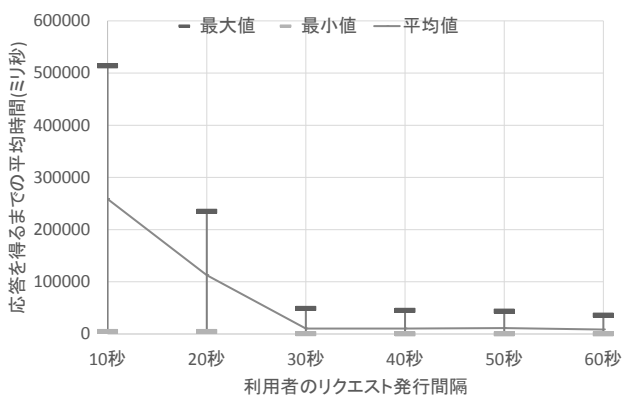


図 12 実験 3：提案手法を使用しない場合の元リクエストが応答を得るまでの平均時間

する元リクエストの発行間隔である。縦軸は、元リクエストが到着してから、応答を得るまでの平均時間である。また、平均値の他に元リクエストが到着してから、応答を得るまでの時間の最大値、最小値を示す。

図 11, 図 12 から、提案手法を使用した場合、利用者がリクエストを頻繁に発行するような状況でも、リクエストが応答を得られるまでの平均時間が短いため、リクエストの処理性能が高いことがわかる。これも、提案手法によって再リクエストを棄却したことで、順序制御機能サーバが Web サーバに送信するリクエストの総数が減少したためであると考えられる。一方で、リクエスト発行間隔が長くリクエスト数の少ないネットワークに余裕のある状況では、提案手法の使用・未使用による差は見られない。

5.9 考察

実験 1 の結果から、順序制御機能が再リクエストを棄却した分だけ、他のリクエストを処理することができたため、Web サーバから情報を得ることができた元リクエストの数を増加させることができたと考えられる。また、実験 2 と実験 3 の結果から、利用者の頻繁なリクエストに対して、提案手法を使用する場合に、提案手法を使用しない場

合よりも多くのリクエストを処理することができると考えられる。

この実験結果は、順序制御機能の目的である災害輻輳時により多くの利用者に Web ページの情報を伝達することを満たすものである。したがって、順序制御機能は有効であると言える。

6. おわりに

本研究では、Web ページの情報を多くの利用者に伝達することを目的とした順序制御機能サーバの再リクエスト抑止について、Web サーバへ繋がりにくい環境を用意し、順序制御機能の有無による評価実験を行った。その結果、想定した環境において順序制御機能は有効であることがわかった。今後の課題としては、ネットワーク帯域を圧迫するリクエストの判定や、リクエストの処理に必要な時間の推定手法などを検討する。

謝辞 本研究は科研費 (24300021) の助成を受けたものである。

参考文献

- [1] 北形元, 笹井一人, 高橋秀幸, 木下哲男, “大規模災害時のための不揮発性ネットワークの提案”, 信学技報, Vol. 113, No. 168, MoNA2013-24, pp. 63-66 (2013).
- [2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall and H. Weiss, “Delay-Tolerant Networking Architecture”, RFC 4838 (Informational), (2007) <http://www.ietf.org/rfc/rfc4838.txt>
- [3] 塚田晃司, 野崎浩平, “災害時孤立集落での利用を想定した地域内情報共有システム”, 情報処理学会論文誌, Vol. 51 No. 1, pp. 14-24 (2010).
- [4] 小山由, 水本旭洋, 今津慎也, 安本慶一, “災害データベース・Twitter と連携する DTN ベース災害安否確認システムの提案”, 第 19 回 マルチメディア通信と分散処理ワークショップ (DPSWS2011), pp. 89-93 (2011).