不揮発性ネットワークのための セッション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場合が多く，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．被災者のモバイル端末は生命線となりうるが，大規模災害時には電力の供給が途絶えることが想定される．そのため，できる限りモバイル端末のバッテリーを節約し，且つ情報収集は可能であることが望ましい．この問題を解決するために，利用者が一時的にネットワークから切断でき，かつ 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を提案する．本稿では，不揮発性ネットワークの持つセッション制御機能について，より現実に近い実験環境と実験プログラムを使用したプロトタイプの評価実験を行った．その結果，ネットワーク品質が劣悪な環境や，頻繁なアクセスを繰り返す利用者が存在する場合に，セッション制御機能が有効に動作することを確認できた．

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場合が多く，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．また，被災者のモバイル端末は生命線となりうるが，大規模災害時には電力の供給が途絶えることが想定される．そのため，できる限りモバイル端末のバッテリーを節約し，且つ情報収集は可能であることが望ましい．この問題を解決するために，筆者らは，利用者が一時的にネットワークから切断でき，かつ 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][2] を提案している．

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．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)[3] 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 [4][5] も提案されており，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，利用者が一時的にネットワークから切断でき，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に以下の機能が求められる．

(F1) 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

(F2) 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

(F3) 保存されたリクエストを適切な順序や速度で送信する，セッション制御機能

先行研究 [2] では，(F3) セッション制御機能のプロトタイプを実装し，再リクエストの抑止の効果を評価する実験を行い，その結果，用意した実験環境において，再リクエストの抑止が有効であることが分かった．

[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

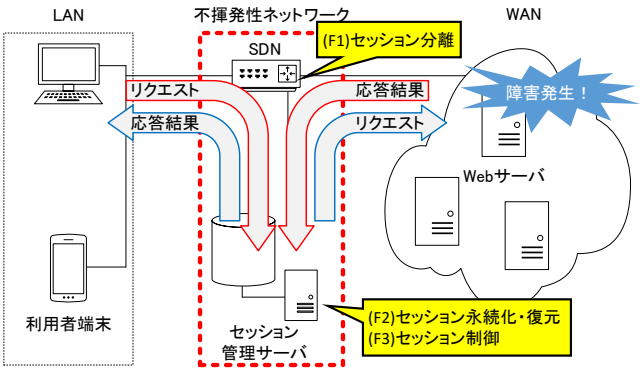

図 1　不揮発性ネットワークの概要

本稿では，不揮発性ネットワークの機能の概要と，(F2) セッション永続化・復元機能と (F3) セッション制御機能の詳細設計，および実装について述べる．また，同機能の評価実験について述べる．

2. 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 1 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)　セッション制御機能

利用者端末からリクエストが発行され，Web サーバの情報を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．この機能は，Software-Defined Network(SDN) である OpenFlow スイッチを利用して実現する．

(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，利用者端末から発行されたリクエストをセッションストレージに保存する．保存されたリクエストは定期的に復元され，Web サーバへ要求発行が行われる．これは，Web サーバから応答が得られるまで継続される．このリクエストの復元と発行は (F3) セッション制御機能によって制御される．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．この仕組みによってセッションは永続化され，利用者端末のネットワークからの切断が可能となる．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，前回と同じ Web ページへのリクエストを発行することにより，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．

(F3) セッション制御機能では，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．

(F2) と (F3) の機能は，図 1 のセッション管理サーバによって提供される．本研究では，不揮発性ネットワーク実現機構よりも WAN 側に構成されているネットワークや Web サーバに障害が発生しており，LAN 側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．

3. セッション永続化・復元機能と制御機能の詳細設計

不揮発性ネットワークでは，(F2) セッション永続化・復元機能は，得られた Web サーバからの応答を対応したリクエストと紐付けて保存する．これは，利用者がセッションを一時的に離脱したり，再開するための機能である．

(F3) セッション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 2 にセッション管理サーバのリクエスト受信時のフロー (以下 1〜4) を示す．

(1) リクエスト受信　セッション管理サーバは，利用者からのリクエストを受信する．
(2) 再リクエスト判定　受信したリクエストが再リクエストかどうかを判定する．再リクエストの場合 (3) を実行する．
(2.1) リクエストを保存　受信したリクエストを，通し番号を付与して保存する．
(2.2) リクエストキューへ追加　受信したリクエストをリクエストキューに追加する．
(2.3) リクエスト受付完了レスポンスを生成する　利用者にリクエストが受け付けられた事を伝えるレスポンスを生成する．

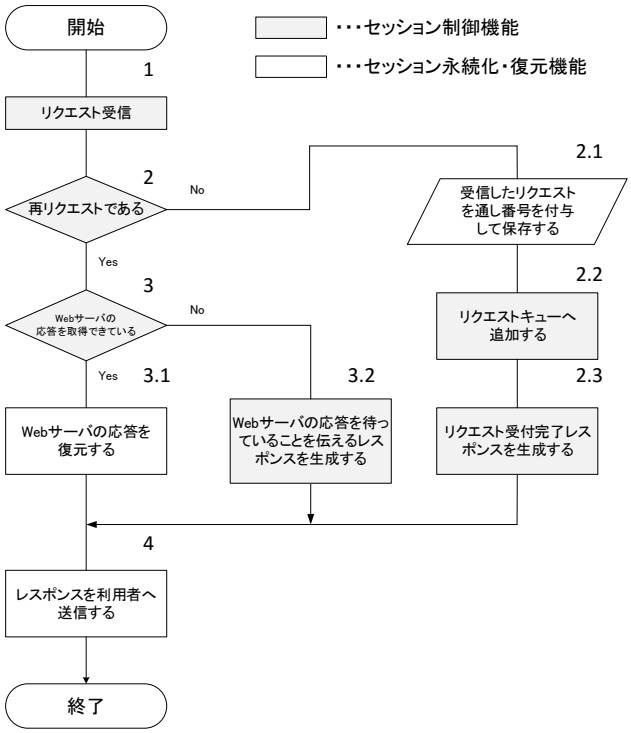

図 2　リクエスト受信のフロー

(3) Web サーバの応答が取得済みか判定　受信したリクエストが再リクエストである場合，Web サーバから応答を取得できているかどうかを確認する．

(3.1) 応答を取得できている　取得できている Web サーバの応答結果を復元する．

(3.2) 応答を取得できていない　受け付けたリクエストがどの程度時間がかかるかを通知するレスポンスを生成する．この結果，ネットワークの負荷が減少される．

(4) レスポンス返却　上記フローで生成または復元したレスポンスを利用者へ返却し，セッションを終了する．

また，セッション制御機能にはこのフローとは別に，リクエスト待ち行列からリクエストを取り出し，Web サーバへタイムアウトでない応答を得られるまでリクエストを発行する処理がある．

4. 評価実験

セッション制御機能の再リクエスト抑止が有効であるかどうかを評価するために実験を行った．

4.1 実験プログラム

評価実験のために，プログラムは HTTP 通信をトラップし，リクエストおよびレスポンスを受信するプログラムを作成した．

この実装では，3 節および図 2 の機能のうち，(2.3) を除く部分を実装した．(2) 再リクエストの判定では，利用者のクライアント IP と要求している URL を用いた．(2.3) のリクエスト受付完了レスポンスの生成については，再リクエスト抑止の効果の検証を優先したため，今回は実装していない．また，リクエスト待ち行列からリクエストを取り出し，Web サーバへタイムアウトでない応答を得られるまでリクエストを発行する処理も省いている．そのため，この実験では，利用者はセッションを一時的に離脱することは無いものとする．

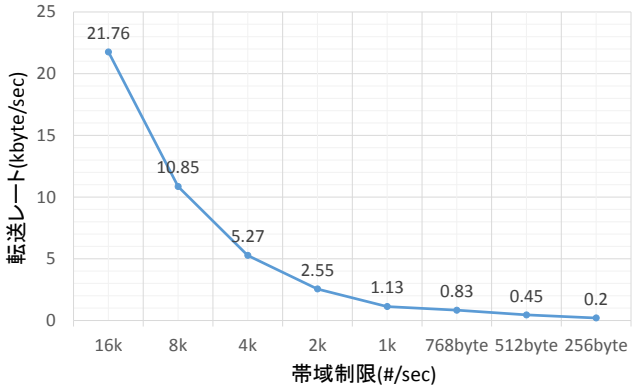

図 3　Web サーバのベンチマークテスト

4.2 実験環境

実験環境の構築を行った．不揮発性ネットワークは大規模災害時のような劣悪な通信環境での動作を想定しているため，遅延や輻輳が発生しやすいネットワークを構築をした．

4.2.1 Web サーバの構築

まず，取得したい Web ページを公開する Web サーバの構築を行った．これは，Cent OS 6 上に Apache2 をインストールし，デフォルトの設定で使用することとした．取得する Web ページは，230byte のプレーンテキストの HTML とした．

4.2.2 通信速度の制限

次に，ネットワーク遅延の再現のため，この Web サーバに対して通信帯域の制限を行った．帯域制限には The dummynet project[6] の ipfw モジュールを使用した．また，制限速度の決定をするため，事前にベンチマークテストを複数回行った．

ベンチマークテストには Apache Bench を使用し，設定値はクライアント数を 10，合計発行リクエスト数を 100 として計測を行った．図 3 にベンチマークテストの結果を示す．

帯域制限が 768byte/sec 以上の時，転送レートは帯域制限値よりも大きな値を得ていることがわかる．しかし，帯域制限が 512byte/sec 以下のとき，帯域制限値よりも下回っている．これは，ネットワークの輻輳が発生したためであると考えられる．

この結果から，本実験では，帯域を転送レートの低下が発生した 256byte/sec に制限した．

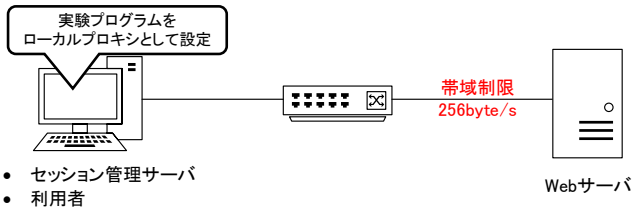

図 4　構築した実験環境

表 1　実験機器の性能

	セッション管理サーバ	Web サーバ
OS	Windows7 64bit pro	CentOS 6
CPU	Core i5	Core i5
RAM	8GB	4GB

図 4 にこのネットワークを使用して構築した実験環境を示す．また，表 1 に使用した実験機器の性能を示す．簡単のため，セッション管理サーバと利用者を同一機器上に構成した．利用者は，ローカルプロキシサーバとして設定された実験プログラムを通して，HTTP 通信を行う．

4.3　実験内容

構築した環境とプログラムを用いて実験を行った．本実験では，2 節で述べたように，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．そのため，輻輳が発生している状況で，ネットワークの悪化を防ぎ，より多くのリクエストに対応できるかどうかで提案手法を評価した．具体的には，提案手法を用いた場合と用いない場合で，以下の項目で評価を行う．

- リクエスト受信から Web サーバから応答を受信するまでの時間 (以降，セッション時間とする)
- 再リクエストではないリクエストの数
- セッション成功数

再リクエストではないリクエストの数を評価する理由は，再リクエストは他の利用者の通信を妨げ，得られる応答の数が増加することはないため，災害時の情報収集には不要な通信であるためである．セッション時間を評価するのは，ネットワーク負荷の悪化が発生しているかどうかを検証するためである．平均時間が長くなれば，ネットワーク負荷が悪化していると判断できるからである．また，リクエストが応答を得られるまでの時間が長くなれば，通常のリクエストはタイムアウトし，セッションは失敗する．そこで，セッションの成功数を検証することで，通信品質を検証した．

実験内容として，利用者は Internet Explorer 11 を使用し，容量 230byte のプレーンテキストページへのアクセスを要求しているものとし，実験を簡潔にするため，この利用者が 1 分間ページ更新をし続けるものとした．

表 2　実験結果

評価項目	提案手法	通常動作
セッション平均時間 (秒)	21	242
受け付けたリクエスト数 (個)	25	97
非再リクエスト数 (個)	25	2
セッション成功数 (個)	25	25

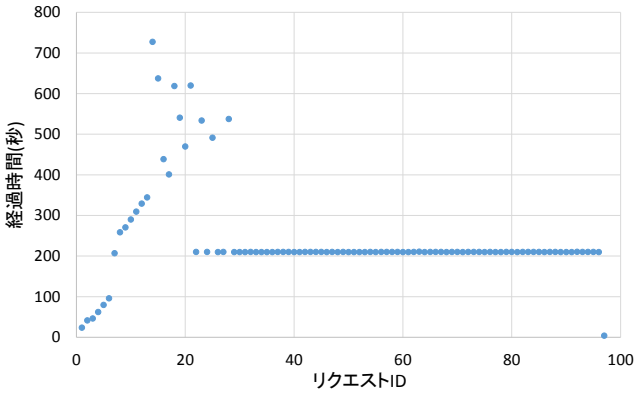

図 5　提案手法を用いない場合のセッション時間

4.4　実験結果

実験結果を表 2 に示す．提案手法を用いた場合，セッションの平均時間が大きく減少していることがわかる．これは，再リクエストを排除したことにより，Web サーバへの接続数が減少したため，ネットワーク負荷の悪化を抑えられたためと考えられる．

また，受け付けたリクエストのうち，セッション成功数の割合に注目すると，通常動作の場合では約 25.5%のセッション成功率である．さらに，受け付けたリクエストのうち，非再リクエストの数に注目すると，わずか 2 個であるため，セッションが成功してもほとんどの応答が不要な通信となっていることがわかる．

一方で，提案手法を用いた場合，受け付けたリクエストのうち 100%のセッションが成功し，全てのリクエストが非再リクエストである．これは，ネットワーク負荷の悪化を抑えられ，セッション時間の増加が防がれたため，セッションがタイムアウトにより失敗することなく完了したからだと考えられる．

また，個々のセッション時間の詳細を図 5，図 6 に示す．

図 5 では，Web サーバリクエスト数が増加するごとに，セッション時間が増加していることがわかる．さらに，一定の時間が過ぎると，およそ 200 秒程度で一定となっているセッションが連続している．このセッションは全て失敗という結果になっており，タイムアウトしていることがわかる．

一方，提案手法を用いた図 6 では，セッション時間は実験開始から終了まで安定している．これは，Web サーバへの再リクエストの抑止に成功したために，Web サーバへの負荷を抑えられ，安定した通信を行えたと考えられる．

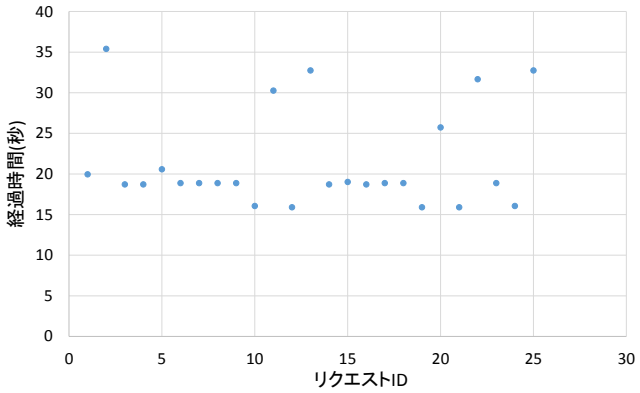

図 6　提案手法を用いた場合のセッション時間

5. おわりに

本研究では，不揮発性ネットワークの機能の概要と，セッション永続化・復元機能と制御機能の詳細設計とその実装を行った．さらに，提案手法による再リクエストの抑止の効果を検証するため，より現実に近い実験環境と実験プログラムを使用したプロトタイプの評価実験を行った．その結果，ネットワーク品質が劣悪な環境や，頻繁なアクセスを繰り返す利用者が存在する場合に，セッション制御機能が有効に動作することが確認できた．

今後の課題として，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の検討があげられる．

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キングの提案”, 信学技報, Vol. 113, No. 168, MoNA2013-24, pp. 63-66 (2013).

[2] 今野 翔太，今井 信太郎，北形元，新井 義和，猪股 俊光，“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”, 情報処理学会研究報告，マルチメディア通信と分散処理研究会報告 2014-DPS-160(3)，1-6，2014-07-17.

[3] 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

[4] 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).

[5] 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).

[6] The dummynet project, “http://info.iet.unipi.it/ luigi/dummynet/”