Peer to Peer ネットワークシステム
Freenet の耐障害性に関する研究

丁 寧[1] , 江崎 浩[2]

東京大学工学部電子情報工学科[1] 東京大学大学院情報理工学系研究科[2]

１．背景と目的

P2P（Peer-to-Peer）システムとは、従来からのクライアントサーバシステムにおいて専用に設けられたサーバを介して行われていたユーザ間の情報のやり取りを、直接ユーザ(ノード)同士でやり取りができるようにする技術やシステム、サービスを指す。サーバに依存していた情報のやり取りから解放されて、必要な情報を必要なときに手に入れたり、またほしい人に提供したりすることができる。

Freenet[1]-[6]はサーバを持たない分散型の情報蓄積配信収集システムであって、誰が情報の発信/蓄積を行ったか、誰がシステム上の情報を閲覧したかということを完全に隠蔽した匿名性の高い情報共有を目的として開発された。 人(ユーザ)が求める情報だけが結果的にシステム上に残り、求められない情報は自動的に消えていくという設計思想を持っている。 本研究の目的は Freenet の耐障害性を評価検討し、その問題点を改善することにある。

２．Freenet システムの概要

２．１．機能要素

Freenet では、以下のような機能が備えられている：

①情報はネットワーク上でその所在が特定できるところに蓄積しない。Freenet では、検索の過程で経由したサーバントのルートをたどって情報ダウンロードする。検索とダウンロードを中継するサーバントは情報を中継するときに自分の情報保存領域にも同じ情報のコピーを保存する(キャッシング)。

②情報の発信は発信者を特定できる情報が残らないようにネットワーク上のどこからも発信できる。Freenet では、特定の保存場所を指定して情報の発信するのではなく、Freenet という匿名性が確保された仮想的な空間上に保存することで、他人による発信者への追跡を防ぐことを実現する。仮想的な空間に保存することで、パス（アドレス）による情報蓄積場所の特定ができなくなるので、情報を取り出すための新たな仕組みが必要となる。

③情報の受信は受信者を特定できる情報を残さずに、ネットワーク上のどこからでも必要な情報を受信することができる。情報を送出するサーバントや情報を中継するサーバントは、最終的に誰がその情報を要求したかを知ることができないように設計されている。

④需要の多い(参照頻度が高い)情報は、Freenet システムとして情報の蓄積(情報が Freenet 上に存在すること)を確保しなければならない。Freenet では、1 回の KEY 情報検索で複数のサーバントに同じ情報が保存(キャッシュ)されるので、検索要求が多い情報は多数のサーバント上に保存されることになる。

⑤需要の少ない(参照頻度が低い)情報は Freenet システム上から消去されなければならない。Freenet では、各サーバントで蓄積する情報量の上限を個々に設定している。情報はこの上限値内でサイクリックにキャッシュされるため、需要が多い情報は多くのサーバント上で何度も蓄積されるが、需要が少ない情報はキャッシュから削除されてしまうので、結果的に Freenet システム上から消滅していくことになる。

⑥システムが、能動的/明示的にに、情報を削除する仕組みを持たない。

２．２．情報の発見と探索方法

Freenet では、200,000 ノードまでのスケーラビリティ（拡張性）とフォルトトレラント性（耐故障性）が実証されている。

図２．１（出典［1］）は、その動作例を示している。

① 最初に左端の Peer から、リクエストメッセージ(検索要求)が出る。
② 受け取った Peer は、該当ファイルが無いためメッセージを他の Peer に転送する。
③ 転送を受けた Peer は、要求されている KEY 情報が無くかつ転送可能な Peer も無いため、フェイルメッセージを返す。

Research on protection from DoS attack on Freenet

1 Nei Tei / Department of Information and Communication Engineering, University of Tokyo.

2 Hiroshi Esaki / School of Information science and Technology, University of Tokyo.

④ フェイルメッセージを受け取った Peer は、まだ転送していない Peer に向けてリクエストメッセージを転送する。

⑤⑥⑦⑧ 同様なシーケンス(リクエストおよびファイルのやり取り) で続く

⑨ 最終的にリクエストメッセージは要求された KEY 情報と一致する KEY 情報を保持している Peer に転送される。

⑩⑪⑫ その結果として要求された KEY 情報に対応したファイルとその KEY 情報が、リプライメッセージと一緒に送信元 Peer に転送される。

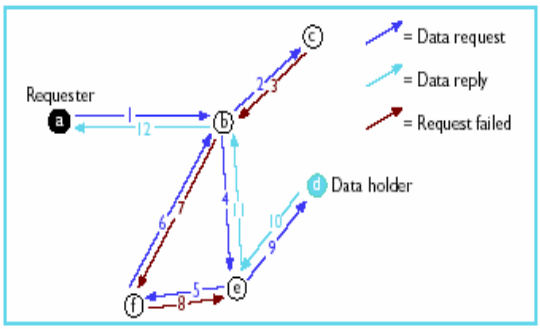


図２． 1 Freenet 発見・探索の動作例(出典[1])

２．３．ファイル(情報)の挿入

まず、たとえば Peer A が新たなファイルを登録すると、GUID である KEY 情報が生成され、その Peer A が持つキャッシュ上におなじ KEY 情報がほかに存在するかどうかが調べられる。存在しなければ、先ほどの発見・探索の場合と同じルーチングに基づき、同じ KEY 情報が存在しないかどうかが調べられる。このとき、KEY 情報と一緒にファイルも送られる。そして、TTL がゼロになった時点でも KEY 情報の衝突が無かった場合 (Freenet P2P ネットワーク上に同じ情報がすでに存在しないこと) にファイルを新規に登録(保存)しようとしている Peer に衝突がなかったことをあらわす「all clear メッセージ」が伝えられる。これら一連のメッセージに KEY 情報とファイルが一緒に送られるので、all clear メッセージが要求元の Peer A に戻るパスの途中で、その GUID に近い Peer がそのファイルを管理することになる。その時点で、各 Peer のルーチング情報が書き換えられることになる。

３．Freenet の耐障害性に関する検討

３．１．small world phenomenon

Gnutella や Freenet などのピュア型 P2P ネットワークの１つ耐障害性として、P2P ネットワークにおいて small world phenomenon が挙げられる。 少数のパワーのある(有力な)Peer によって、システムの大部分が構成および特徴付けられるということが実際のシステムにおいて観測されている。ピュア型 P2P システムでは、ランダムな Peer に関する障害の発生や外部からの攻撃が行われた場合には、耐障害性が高いということがいえる。 しかし、Power のある Peer をまとめて攻撃するような計画的な攻撃に対して、非常に弱いという事実が、経験的に実証されている。図３． 1 (出典［1］)はその実験を示している。

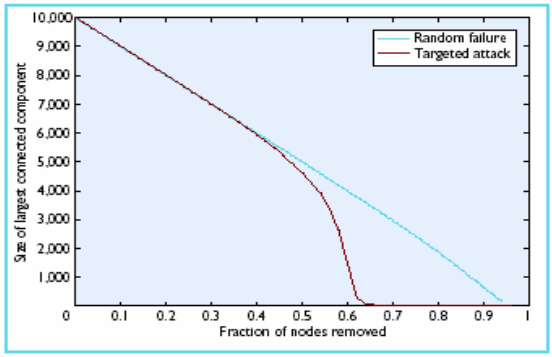


図３． 1 Freenet の耐障害性(出典[1])

３．２．ジャンクデータへの対応

Freenet の独特なファイル転送とファイル保存アルゴリズムが原因となり、ジャンクデータによる DoS 攻撃による被害が非常に大きくなってしまう。ジャンクデータの大量な挿入や要求により、経路上にある Peer は Flood され、有用なデータをプッシュアウトされてしまう。 本研究では、このような大量のジャンクデータの挿入による DoS 攻撃に対する対応法の検討と提案を行っている。

４．DoS 攻撃

４．１．DoS 攻撃の種類

① 負荷をかける

インターネットプロトコルの脆弱性を攻略して、ネットワークに接続されたコンピュータに過剰な負荷をかけて、サービスを提供することを妨害する。 膨大な数のアクセスを実行することにより、通信帯域、ディスク容量、あるいは CPU 資源を圧迫したり動作不可能な状態にしたりする。

② プロトコルやプログラムの脆弱性を攻撃する

サーバーアプリケーションの脆弱性を攻略し、サービスプロセスに多量の例外処理を発生させることにより、サービス提供を妨害したりサービス提供不能にする攻

撃などである。OS のバグなどを利用してサーバをフリーズさせる。

４．２．DoS 攻撃の形態

① 単独で攻撃を仕掛けるもの

負荷型の攻撃を単独で行っても効果が薄いため、現在では非主流となりつつある。

② ウイルスやトロイの木馬を使用して複数マシンからターゲットに対し一斉に攻撃を行うもの

分散協調型サービス停止攻撃 （DDoS 攻撃）複数の攻撃元が一斉にひとつの標的に対して負荷型の DoS アタックをしかける。攻撃者が事前に標的以外のコンピュータに攻撃プログラムを仕掛けておいて、一斉に DoS 攻撃をしかける手法。攻撃に参加するコンピュータが多いほど攻撃能力が増大される。攻撃元となる踏み台が多数用意されるため、犯人の追跡が困難。現在の DoS アタックの主流。

５．Freenet の耐障害性改善方法の一提案

５．１．提案方法の方針

Freenet の独特なファイル転送・保存システムを保ちながら、ジャンクデータへの対応を行う方策を提案する。以下が、提案方法の設計方針である。

① ジャンクデータの定義・検出の時に多数の Peer の承認が必要にする（正当性が難しい）ジャンクデータの発信歴のあるユーザのブラックリストを誰か持つ（システムはユーザ認証を要求しエンドユーザ間は匿名性が要求される）。

② ジャンクデータだとわかったらすぐ発信したユーザのサービスを停止し、ジャンクデータの KEY 情報をブロードキャストし、削除する。

③ エンドユーザに対するファイルやユーザの匿名性は保障する

④ 基本的には、スーパーユーザに対しても 匿名性を保障したい。しかし、いざというときのトレース機能が必要。外部の信頼できる組織(RCA) を利用して、Freenet の運用に関わる人には、スーパーユーザ（CA）を含めて、匿名性を保障する

５．２． 提案システムの動作概要

Freenet のユーザが外部の信頼できる組織 RCA に認証し、認証された ID をもらう。ファイルを挿入する前にスーパーユーザ（CA）に RCA からもらった認証 ID を使って、認証してシーケンス番号をもらう。シーケンス番号と一緒にファイルをファイルサーバに挿入する。ほかのユーザがそのファイルをダウンロードしてもファイルのシーケンス番号がわかるだけで、匿名性が保たれる。もしこの挿入されたファイルがジャンクデータの挿入による DoS 攻撃のファイルだと、普通のユーザが判断し、それをスーパーユーザに報告したら、スーパーユーザがシーケンス番号からファイルの送り主の認証 ID を割り出し、これ以上のシーケンス番号の発行を拒否し、ブロックする。そして当ユーザが挿入したファイルのシーケンス番号を Freenet 上にブロードキャストし、削除を命令する。

５．３．ユーザ認証・ファイル挿入・ファイル削除・ユーザブロック方式

手順①から⑤まではユーザの認証、⑥,⑦はファイルの挿入、⑧,⑨はジャンクデータの発見・削除。そして手順⑤を拒否することによってユーザをブロックする。

図５．１はアルゴリズムを示している

① ユーザ１が RCA の公開鍵を使って自分の身分を暗号化し、RCA に送る

② RCA がユーザ１の身分を RCA の秘密鍵で解読し、かわりに ID を発行して暗号化し、ユーザ１に送る

③ RCA の秘密鍵で暗号化した ID を RCA は CA に教える、CA は RCA の公開鍵でそれを解読する

④ ユーザ１が CA の公開鍵を使って RCA から送ってきた暗号化した自分の ID をさらに暗号化し、CA に送る

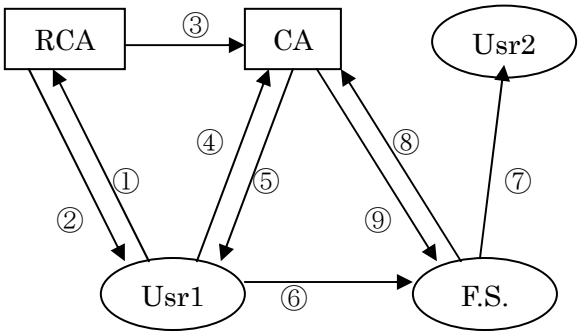
⑤ CA がユーザ１の ID を CA の秘密鍵で解読し、RCA からもらった認証 ID と照合し、合ったらかわりにシーケンス番号をユーザ１に発行して CA の秘密鍵で暗号化して送る

⑥ ユーザ１がシーケンス番号を CA の公開鍵で解読し、挿入したいファイルと一緒にファイルサーバに送る

⑦ ユーザ２がファイルサーバからファイルをダウンロードしてもシーケンス番号しかわからず、匿名性が保たれる

⑧ ファイルサーバがジャンクファイルのシーケンス番号を CA に教える

⑨ CA はジャンクファイルだとわかったら当シーケンス番号がつくファイルの削除命令を Freenet 上にブロードキャストする。削除すべきファイルのシーケンス番号に対応する ID を持ったユーザ１にシーケンス番号を発行する手順⑤を停止し、ブロックする



F.S.：File Server

図５． １新しいシステムの認証方式

５．４．削除(パージ)すべきファイルの定義方法の検討

Freenet に対する DoS 攻撃が発生されたとき、ここではつまり削除すべきファイルが大量挿入されたとき、その削除すべきファイルに該当する基準、定義や摘発などは例えば以下のような方法が考えられるが、個人的なジャンクに対する定義の主観的な違いで決定的な定義方法がなく、それぞれのポリシーになるため、数値に落ちるようなオート解決方法よりもマニュアルによる判断になることが望ましい。

①トラフィックの変化
負荷型 DoS 攻撃が仕掛けられたときのトラフィックの変化パターンを元に、攻撃を察知する方法である。しかし、トラフィックの変化パターンだけでは巧妙なアタックのときには本当に攻撃であるかどうかわからない可能性がある。
②スコアリング
マッチポンプの問題が起こる可能性がある
③BBS や E メールなどによる摘発
10 人以上集まって声を上げるといった方法がある。しかし、その 10 人はのっとられていて間違った情報を伝える可能性があるので、のっとりかどうかを判断するために human relationship が必要である。数値に落ちるでは解決できないので主観的な判断になるしかない所がある。つまりオートではなく、マニュアルで判断すべきである。

６．考察

上の章で議論したように、Freenet の耐 DoS 攻撃性を改善するために、克服しなければならない二つの難問は、①ジャンクデータの定義、②匿名性とセキュリティのトレードオフである。①はかなり主観的な部分があるためポリシーの問題にもなる。②は匿名性を完全に保ちながら、セキュリティ認証を解決できれば Freenet のもともとの思想にも合致するが、現時点ではこれを満足することができる技術と方法が見当たらない。 今後の研究課題である。

参考文献

[1] Ian Clarke, Theodore W. Hong, Scott G. Miller, Oskar Sandberg, and Brandon Wiley, "Protecting Free Expression Online with Freenet," IEEE Internet Computing 6(1), 40-49 (2002)

[2] Andy Oram (ed.), Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly and Associates, Sebastopol, CA (2001)

[3] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System", 2001

[4] Amr Z Kronfol, FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine, 2002

[5] Ian Clarke, "A distributed decentralized information storage and retrieval system" 1999

[6] http://freenetproject.org/