

## 論 文

## ファイル転送残量を考慮した P2P ネットワークにおけるファイル共有方式の性能評価

川原崎雅敏<sup>†</sup> 川原 亮一<sup>††</sup> 大沼 智洋<sup>†††</sup> 鈴木 慧<sup>†††</sup>

## Efficient File Sharing Method in P2P Network Based on Residual File Transfer Load

Masatoshi KAWARASAKI<sup>†</sup>, Ryoichi KAWAHARA<sup>††</sup>, Tomohiro OHNUMA<sup>†††</sup>, and Kei SUZUKI<sup>†††</sup>

あらまし P2P ネットワークにおけるコンテンツ取得は一般にダウンロード完了までに長時間を要する。本論文では Gnutella 型の P2P ネットワークにおいて、最初に発見したコンテンツ保有ピアからファイルを取得するのではなく、コンテンツを保有するピアのファイル転送残量に着目し、転送残量が最も少ないピアから優先的に取得することにより、ファイル取得時間を短縮する方式について、シミュレーションによりその有効性を評価する。その結果、評価した方式は負荷を分散させることによってファイル取得時間を短縮できること、また、この方式による短縮効果は、ハブ的なノードが多いネットワーク形態ほど、ネットワークが大規模であるほど、コンテンツサイズの偏りが大きいほど高くなることを示す。

キーワード P2P ネットワーク, フラッディング, 転送時間, ファイル転送残量, 負荷分散

## 1. ま え が き

近年におけるパソコンの高性能化と ADSL・光ファイバといったブロードバンド化の普及は著しく、インターネットの利用者数やトラフィック量は急激に増加している。インターネットの主流であるクライアント/サーバ型の配信モデルはサーバにアクセスが集中するため、最悪の場合サーバがダウンし処理ができなくなってしまう問題がある。そうした問題点を解決する手段としてサーバ機能を分散させてスケーラビリティを確保できる Peer-to-Peer (以降 P2P) ネットワークが注目されている [1], [2]。P2P ネットワークとは、各端末が各自で情報を管理し、他の端末と直接通信を行うネットワークである。

P2P ネットワークにおけるファイル共有は、欲しいファイルを保持しているピアを検索し、そのピアからファイルを取得するという二つのフェーズで実現されるが、現在主流となっている P2P ネットワークではいずれのフェーズも P2P を構成するピアの共同作業により分散環境で実行される。ここで問題となるのが検索効率や取得効率である。検索方法にはフラッディングに代表される非構造型と、DHT (分散ハッシュテーブル) を利用する構造型がある [3]。フラッディングは、順次隣接ピアに問い合わせる方式であるが、問合せメッセージが到達できるノードの範囲が限定されるとともに、各メッセージがノードに到着するまでのホップ数が大きくなるという問題がある。これらの問題を解決する手段として登場したのが構造型であり、DHT を利用することで所要ホップ数は大幅に改善される。しかし DHT 方式は、ファイルのアクセス頻度に関係なく同じ方法で検索するため、人気のあるファイルの検索効率は必ずしも高いとはいえない。最近では、非構造型で検索・取得効率を高める方法が模索され、処理能力の高いピア (Super Node) がすべてのピアのファイル情報や IP アドレスを管理する FastTrack/KaZaA や [4]、コンテンツを一定サイズのピースに分け、ピー

<sup>†</sup> 筑波大学大学院図書館情報メディア研究科, つくば市  
Graduate School of Library, Information and Media Studies,  
University of Tsukuba, Tsukuba-shi, 305-8550 Japan

<sup>††</sup> 日本電信電話株式会社 NTT サービスインテグレーション基盤研究所, 武蔵野市  
NTT Service Integration Laboratories, NTT Corporation,  
Musashino-shi, 180-8550 Japan

<sup>†††</sup> 筑波大学図書館情報専門学群, つくば市  
School of Library and Information Science, University of  
Tsukuba, Tsukuba-shi, 305-8550 Japan

スの所有情報をトラッカーが集中管理し、複数のノードからピースを集める BitTorrent [5] が注目を集めている。

本論文では、非構造型 P2P ネットワークを対象に、フラッディング方式の問題点を改善するファイル共有方式について評価する。その方法は、クエリをフラッディングさせて目的ファイルを所有するノードを検索する際に、最初に発見したノードからファイルを取得するのではなく、発見したノードの中でファイル転送残量が最も少ないノードからファイルを取得することで、特定のノードにリクエストが集中するのを避け、負荷を分散させることによってファイルの転送効率を高めるものである。この方式を、トポロジー生成ツールを利用して大規模な P2P ネットワークでシミュレーション評価し、従来のフラッディング方式に比して転送効率が向上することを示す。

P2P ネットワークにおいて、ファイルの検索・取得効率を向上させる研究は既にいくつか行われている。文献 [6] は、Gnutella [7] の単純さを保持しつつスケラビリティを改善する方式として、能力の高いノードにクエリを誘導する方式や、トークン型のフロー制御によりクエリの集中を緩和する方式を提案している。文献 [8] は過去のダウンロード経験を参考に、機械学習の手法により複数のピアを渡り歩いてファイルを取得する adaptive peer selection を提案している。これらに比べ、本論文で評価する方式は極めて簡便なものである。

最近では、P2P で構成されるオーバレイ網と下位にある物理網とでトポロジー上の不整合による非効率性があることに着目し、物理網を意識した P2P の性能向上方式が提案されている。文献 [9] は、ネットワークの状況をアプリケーションに通知する P4P の試みである。プロバイダが P4P ディスタンスと呼ばれる「仮想的なピア間の距離」を計算してアプリケーションに提供し、アプリケーションとの協業でパフォーマンスを最適化できることを示している。文献 [10] は AS 間距離を重視し、QueryHit したピアの中から同一 AS 内のピアを優先的に選ぶように改良することで、従来は他 AS から取得していたファイルの 40% を同一 AS 内から取得できるようになり、効率が向上したとしている。

P4P のように物理網のトポロジーや負荷状態を意識した方式に対して、本論文で評価する方式は、ピアの転送残量のみに着目したものであり、異なるアプロー

チになっている。

一方、本論文で評価する手法は抽象的にとらえると、P2P ネットワーク上で、ある方法により求めるサービスを提供するノード群を発見し、発見したノード群から定められた基準に従って最適なノードを選択する方法の評価であるといえる。この意味においては、例えば P2P ストリーミングなど実時間性を重視するサービスで、最適な供給ノードを選択する手法として類似の研究が見られる。映像配信や音楽配信など実時間のメディア配信サービスでは、再生に間に合うようにファイルを収集する必要がある。特にオンデマンド型の P2P-VoD (Video-on-Demand) では、視聴者が映像の異なる部分を見るため、参加開始までに配信が終わっている部分や、ファイル取得中のピアが離脱した場合の残り部分は他ピアやサーバから追加で受け取る必要がある。P2Cast [11] や P2VoD [12] は P2P ライブストリーミングを拡張する形で、BASS [13] や PPLiveVoD [14] は BitTorrent を拡張する形で、これらの方法を提案・実現している。文献 [15] では ALM (Application Level Multicast) ルーティングにおいて、ファイル転送の残りの通信量をメトリックに加えたピア選択により、再経路探索の回数を抑制した効率的な通信ができるとしている。しかし、これらは本論文で評価するファイル共有とは対象が異なっている。

本論文は以下のように構成する。2. では現状の P2P ネットワークにおけるファイル共有方式を概観する。3. では評価する方式の概要とアルゴリズムを述べ、4. でそれを検証するための P2P ネットワークトポロジーの生成方法を説明する。5. でシミュレーションによる検証・評価結果を述べ、6. でまとめを行う。

## 2. P2P ネットワークにおけるファイル共有方式

ここでは代表的な非構造型 P2P である Gnutella と BitTorrent について説明し、それぞれの取得効率について論じる。

### 2.1 Gnutella [16]

Gnutella はピア型 P2P システムを用いた代表的なファイル交換ソフトである。ピア型 P2P とは、ファイルの探索・発見・取得機能を各ピアが分散して受け持つ、極めて純粋な P2P ネットワークである。Gnutella に参加するためにはエントリポイント (最初にアクセスするピア) を探す必要がある。仲間やインターネットなどを介して取得しネットワークに接続

される。新規参入したピアは、エントリポイントへ向けて Ping メッセージを送信する。Ping を受け取ったピアは、自分の知っている他のピアへ Ping を転送すると同時に Pong を送信元に返す。Pong には、IP アドレスやポート番号など通信に必要な情報が含まれている。こうして新規ピアは、エントリポイントを介して Pong をかき集めてピアグループを形成することで Gnutella ネットワークが構築される。次にほしいファイルの探索・発見のために、Gnutella ネットワークに向けて Query をブロードキャストする。Query を受け取ったピアは、Query を他のピアに転送すると同時に、要求されているファイルがあれば QueryHit を返す。Query の転送により探索範囲が広がっていく過程はフラッディングと呼ばれるが、無制限のフラッディングを抑制するために、Query の転送ホップ数に TTL (Time to Live) という制限を設けている。要求元は QueryHit に含まれる IP アドレスのホストに向けて直接 TCP 接続され、ファイルを要求する Get コマンドを送る。以上の経緯でファイルは流通していく。

## 2.2 BitTorrent [17]

BitTorrent は、ユーザのダウンロードを管理するために所蔵情報を 1 箇所に集中させるハイブリッド型 P2P システムに近いファイル交換ソフトである。Gnutella と異なり、データファイルはファイルを配布する場合に必要な情報が記された torrent ファイルと「ピース」と呼ばれるファイルの断片に分解され、トラッカーと呼ばれる仲介役のコンピュータが一括して管理する。ネットワークの最初は通常一つのシーダー(ピースを 100%もつピア) から始まり、他のピアはトラッカーを介してシーダーからピースをダウンロードするとともにそのピースを同じ torrent ファイルをもった別のピアに配布し、交換を行う。ネットワーク上のピアはこれを繰り返すことでピースを収集し、すべてのピースが集まると再びもとのデータファイルのように合成される。アップロードを全くしないクライアントに対しては、ダウンロードのみのクライアントを BitTorrent のネットワーク上から排除するようになっている。

## 2.3 両方式の比較

BitTorrent はファイルをピースに分割して共有する方式をとっている。ピアが一つでもピースを入手すれば、そのピースをアップロード可能となり、ダウンロード中であるピアもアップロードに参加できる。このため、ファイルの流通度という観点では BitTor-

rent の方が効率的である。そうした性質から、近年では BitTorrent の技術に注目が集まっている。しかし、人気度の少ないコンテンツでは、ピースの所有ノードが少なくファイルの取得効率には上がらない。一方、Gnutella はすべてのノードが対等で完全に自律して機能するため、BitTorrent に比べ耐障害性に優れるという利点もある。

両方式の取得速度が同程度になれば Gnutella によるファイル共有はいっそう有用なものとなる。

## 3. 評価対象とするファイル共有方式

### 3.1 評価対象方式の概要

Gnutella モデルにおける従来の探索方式では、コンテンツを取得したいノード(リクエスタ)が隣接するノードにクエリを転送し、一番最初に保有のレスポンスがあったノードをダウンロード元に決定している(図 1(a))。しかし、この方式ではファイル転送の要求が過剰なノードからダウンロードを行う可能性があり、コンテンツの取得に長時間を要するおそれがある。こうした状況を避けるため、本研究では各ノードのファイル転送残量に着目し、探索の範囲内で最も転送残量が少ないノードからダウンロードを行う取得方式でコンテンツの取得時間の短縮を試みる(図 1(b))。従来方式と比較してどの程度時間短縮できるかを様々なネットワーク状況下において評価する。

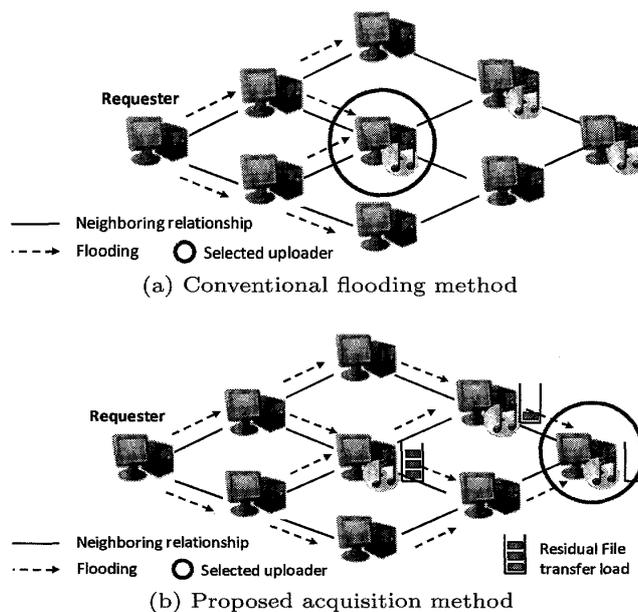


図 1 従来のフラッディング方式及び評価する取得方式  
Fig. 1 Conventional flooding method and proposed acquisition method.

### 3.2 各ノードの属性及び動作パラメータ

方式を評価するにあたり、ノード属性と動作パラメータを明確化する必要がある。条件設定としては、探索コンテンツを  $Y_1, Y_2, Y_3$  とし、これらは少数が P2P ネットワーク上に初期配置されているとする。探索アルゴリズムは上述のとおり Gnutella 方式を用いる。探索コンテンツを保有していないノードは、ポアソン生起 [18] に基づいてリクエストを発生させ、コンテンツの取得を試みる。各ノードには回線速度を設定し、ファイル転送残量との商をとることで転送時間の見積りを行う。

[パラメータ]

- コンテンツサイズ： $Y_1, Y_2, Y_3$
- TTL の初期値： $TTL_{init}$
- ダウンロード先の ID： $downloader$
- 現在時刻： $T_{now}$
- リクエスト生起率： $\lambda$

[各ノードの属性]

- コンテンツ保有フラグ ( $Y_1 \sim Y_3$ ):  $flag_{c1}, flag_{c2}, flag_{c3}$  (保有: 1, 未保有: 0)
- 残余ファイル転送残量： $B$
- ファイル転送完了時刻： $T_{fs}$
- ファイル受信完了時刻： $T_{fr}$
- 隣接ノードリスト： $next[]$
- 回線速度： $C$
- 受信フラグ ( $Y_1 \sim Y_3$ ):  $flag_{r1}, flag_{r2}, flag_{r3}$  (受信: 1, 未受信: 0)

### 3.3 アルゴリズムの詳細

(1)  $flag_{c1} == 0$  or  $flag_{c2} == 0$  or  $flag_{c3} == 0$  のノードは、生起率  $\lambda$  のポアソン生起によりコンテンツ要求を発生させる。これをリクエストとする。コンテンツ保有フラグが 0 であるコンテンツを要求しフラグを立てる。  $Y_1$  を要求する場合は  $flag_{r1} = 1$  (以後  $Y_1$  を要求する場合で説明する)。

(2) リクエストは  $next[]$  を見て隣接ノードの ID を調べ、該当するノードのうち  $flag_{c1} == 1$  のノードを検索し  $Y_1$  を保有するノードを検索。見つからない場合、隣接ノードの  $next[]$  を見て再度  $flag_{c1} == 1$  のノードを検索する。同時に TTL の値を  $-1$  する。

(3)  $flag_{c1} == 1$  のノードを発見したら、暫定的にそのノード ID を  $downloader$  に書き込む。

(4) 引き続き (3) の動作を繰り返し、新たに発見したノードの  $B$  が  $downloader$  の  $B$  よりも小さければ、そのノード ID を  $downloader$  に書き換える。

(5)  $TTL == 0$  になるまで探索を続け、終了時点での  $downloader$  を最終的なダウンロード先に決定する。

(6) 決定したダウンロード先の  $B$  には  $Y_1$  を加算し  $T_{fs}$  を  $B/C$  で見積もる。

(7)  $T_{now} > T_{fr}$  ならば、ダウンロードが完了しているので、 $flag_{c1} = 1$  とし、コンテンツ保有フラグを立てる。

(8) 以上の動作を  $Y_1 \sim Y_3$  のいずれかがネットワーク全体の 30% のノードが保有するまでリクエストを発生させ続け、各リクエスト時の平均転送時間を算出する。

本評価方式で求める転送時間は以上のとおりである。通常のフラッディング方式による算出方法は (3) で探索を終了させ、同様に (6) の手順で転送時間を算出する。二つの方式にて得られた転送時間の商が短縮倍率になる。

## 4. P2P ネットワークトポロジー

P2P ネットワークにおける検索効率にはネットワークトポロジーに依存するため、実社会のネットワークの法則性について考慮した上で構成しなければならない。

### 4.1 現実世界のネットワークの性質

社会に存在するネットワークは巨大で複雑な構造を有しているが、一定の共通する性質をもっていることが研究により明らかにされている [19], [20]。

一つ目の性質はスケールフリー性である。これは、一部のノードが他の多くのノードとリンクでつながっており、大きな次数をもっている一方で、大多数のノードはごくわずかなノードとしかつながっておらず、次数は小さいという性質である。

この性質を数学的な規則性によりモデル化したのがバラバシ/アルバートモデル (以降、BA モデルとす

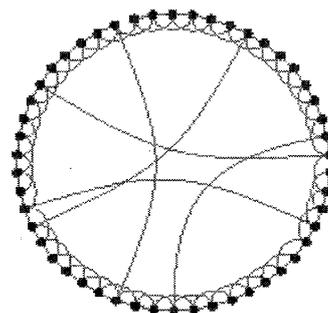
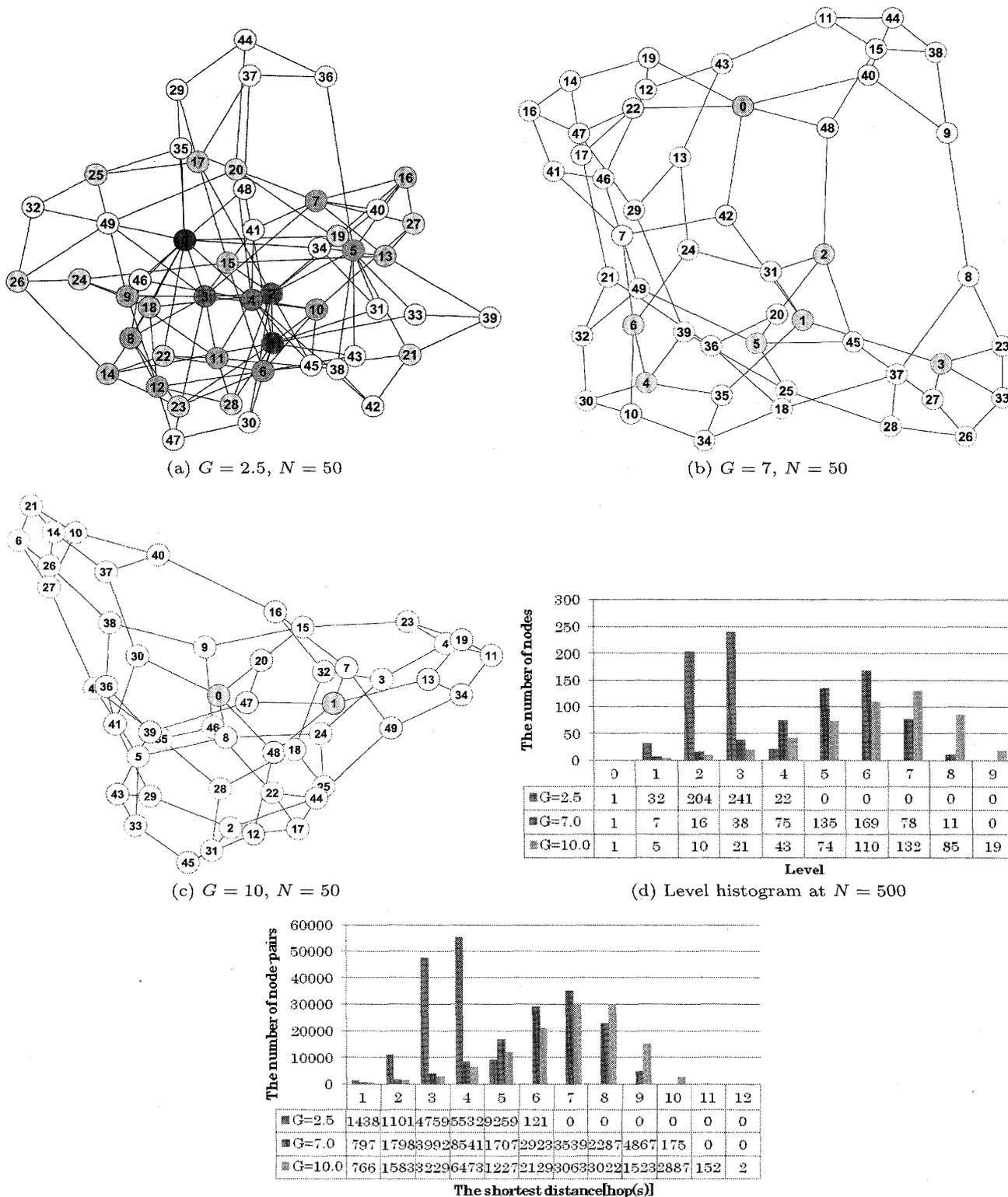


図 2 WS モデルの基礎トポロジー  
Fig. 2 Basic topology of WS model.



(e) Minimum hop distribution at  $N = 50$

図3 使用したトポロジーのイメージ  
Fig.3 Network topology overview.

る)である。BAモデルの重要な要素は二つあり、一つはネットワークが成長していくこと、そしてもう一つは既に多くの枝をもっているノードに対し優先的に

接続することである。次のようなアルゴリズムでネットワークを生成する。

- (1) まず、 $m$  個のノードからなる完全グラフ (す

べての他ノードに対してリンクしているグラフ) から始まり, 新しいノードを 1 個追加する.

(2) そのノードから既に存在している  $m$  個のノードに対してリンクを張る. このとき, リンクが張られる確率は, それぞれのノードのその時点での次数  $k$  に比例するものとする.

この法則で構成されたネットワークはノードのもつリンク数がべき乗則に基づいて分布する [21]. 一般に, 実社会で観測されるネットワークはべき乗則に基づくことが知られている.

二つ目の性質はスモールワールド性である. これは, 言葉の意味どおり「世間は狭い」という性質で, ネットワークの平均経路長  $L$ , つまり平均何ホップで各ノードに到達できるかがノード数  $N$  に比べて小さいという性質である. 信号や情報, 物質などの伝達効率を考えるとネットワークの  $L$  は小さい方が望ましいとされている.

三つ目の性質はクラスタ性である. 身の回りの知人関係を例に挙げて説明すると, この性質は各人の友達同士がまた互いに友達関係である度合を表している. DHT 方式のように各ノードがクラスタリングされたネットワークは, 自ノードと趣向が近いノード同士が密集しているため, 探索効率が高く, 負荷の分散にもつながる.

高いスモールワールド性とクラスタ性を両立させた数学モデルはワッツ/ストロガツツモデル (以降, WS モデルとする) と呼ばれる (図 2).

WS モデルは次のようなアルゴリズムでネットワークを生成する.

(1) すべてのノードを, 近隣の  $a$  個のノードと格子状につなぐ.

(2) それらのリンクを確率  $p$  でランダムに張り替える.

本研究でのシミュレーションに用いるネットワークトポロジーは単純なアルゴリズムでありながら, 「スケールフリー性」「スモールワールド性」「クラスタ性」という現実世界のネットワークの三つの性質すべてを満たしたトポロジー生成ツールを使用する.

このツールで生成されるトポロジーはべき乗則に従う Power-Law Network [21] である. 文献 [16] は, 現実の Gnutella ネットワークのトポロジーがべき乗則で近似できることを実測により検証している.

#### 4.2 トポロジー生成方法

トポロジーの生成に使用したツールは Sun Mi-

croSystems Laboratories が開発した Java プログラム, Node.java, Graph.java, Node.java, PLGraph.java [20] である. このプログラムは, トポロジー全体のノードの総数  $N$ , 隣接ノード数の分布指数  $G$ , 最低隣接ノード数  $k$ , ネットワークタイプを使用者が設定することで目的のトポロジーを生成することができる.

隣接ノード数が  $k$  になる確率を  $k^{-G}$  と定義し,  $G$  の値が大きいくほど隣接ノード数が  $k$  であるノードの比率が多くなり, 小さい場合は  $k$  よりも多く隣接ノードをもったノードの比率が多くなる. 図 3 は本論文で使用するネットワークタイプのトポロジーイメージである. 図 3(a)~(c) は 50 個のノードで構成されており, 枝の本数が多くなる, すなわち  $G$  の小さいトポロジーほど, フラッディングが密に行われる. 図中では, ノードの色の濃いノードほど, 次数が高い.  $G = 2.5$  の場合は図 3(a) のように次数の高いノードがかなり多く存在しているが,  $G = 10$  の場合では, 図 3(c) のようにほぼ均等になり, ほとんどのノードの次数は, 最低隣接数である  $k$  (図中では 3) になっている.

図 3(d) は,  $N = 500$  の場合のレベルヒストグラムである. レベルとは, 中心ノード (最も枝の本数の多いノード) から各ノードへの最短距離のことである. 中心ノード自身のレベルは 0, 1 ホップで到達できるノードはレベル 1 である. また, 図 3(e) に  $N = 500$  の場合のノードペアのすべての組合せ 124,750 通りの最短ホップ数分布を示す. これらのグラフから,  $G$  が大きければ大きいほど, ノードの探索のためにより大きな TTL が必要であることが分かる.

## 5. シミュレーションによる評価

1 台の物理 PC 上にトポロジー生成ツールを用いて大規模仮想 P2P ネットワークを構成し, 方式評価を行う.

### 5.1 実装条件

OS : Windows XP Professional

使用言語 : Borland C++ Compiler 5.5

CPU : Pentium® D 2.80 GHz

### 5.2 評価尺度

3.1, 3.2 で述べた方法に基づき

- トポロジー構造を変化させたとき
  - ネットワークの規模を変化させたとき
  - 3 種類のコンテンツのサイズ比率を変えたとき
- 以上の条件で生起率  $\lambda$  を変化させたときのファイル転送の平均時間を評価する.

なお、ファイル転送時間には本方式固有の時間として、転送残量が最も少ないノードを選択するという処理に要する時間がかかる。この処理時間は、TTLの範囲内をすべて検索し、リクエストが各ピアから転送残量データを取りまとめ、最も少ないピアを発見するまでの時間となる。従来方式ではヒットすればすぐに転送されるので、本方式とはファイル転送時間に差が生じるが、この時間差を構成するのは付加的な探索時間と Query/QueryHit メッセージの送受時間、更にリクエストでの計算時間である。この内、メッセージの送受時間や計算時間はファイルを転送する時間に比べて十分小さく無視できる。付加的な探索時間については、従来方式でもリクエストであるユーザが目的コンテンツを確定するまで QueryHit メッセージは到着し続ける一方、各ノードが実行する PC 内のファイル検索は TTL 範囲内のノードで同時並行的に進むので、従来方式と評価方式で探索時間に顕著な差が生じることはないと考えられる。したがって、シミュレーション上は考慮していない。

### 5.3 シミュレーション条件

シミュレーションのデフォルト値を表1に示す。リクエスト生起率 $\lambda$ は、まだ目的のコンテンツをもっておらず、かつダウンロード中でもないノード1個当りのリクエスト生起率であり、平均 $\lambda$ のポアソン分布に従う。既に目的のコンテンツをもっている、またはダウンロード中であるノードは、そのコンテンツに対して新たなリクエストを発生させないものとする。

表1 シミュレーション条件  
Table 1 Simulation conditions.

| Parameter                                     | Value                               |
|---|-------------------------------------|
| Exponent(G)                                   | 2.5                                 |
| Least number of adjacent nodes (k)            | 3                                   |
| The number of nodes (N)                       | 500                                 |
| File Size                                     | Y1=1[MB],<br>Y2=5[MB],<br>Y3=10[MB] |
| The number of initial arrangement of contents | Y1, Y2, Y3:<br>1% of N              |
| TTL   | 4                                   |
| Access link speed                             | 8[Mbps]                             |

シミュレーションの完了条件は、3.3で述べたとおり Y1, Y2, Y3 のいずれかが  $N$  の 30% のノード ( $N = 500$  であれば 150 ノード) に保有された時点としている。

### 5.4 トポロジー構造を変化させたとき

#### 5.4.1 $\lambda$ を変化させたとき

図4(a), (b) は生起率 $\lambda$ を変化させたときの転送時間特性を、 $G$  をパラメータとして示したものである。図4(b)以降、転送時間 (Transfer time) とは、各ファイル転送における平均の転送所要時間である。すべてのトポロジーにおいて、評価方式は従来方式に比して転送時間が短くなっており、この傾向は容量が大きいコンテンツほど顕著である。図4(c) に生起率に対する転送時間の短縮倍率 (評価方式の従来方式に対する比; Reduction rate) を示す。

$G$  が小さいトポロジーほど、評価方式により転送時間の改善が図られ、短縮倍率は大きい。しかし、 $G$  が大きくと、図4(b)のように従来方式と評価方式の差がほとんどなくなり、短縮倍率の変化も小さくなる。この理由は以下のように考えられる。第1は、 $G$  が大きくなると次数が高いノードが少なくなるため、従来方式では  $G$  が小さいときにリクエストが集中しやすかったハブ的なノードが存在しなくなり、その結果、特定ノードへの集中が起りにくく、転送時間は大きくなならない。第2は、 $G$  が大きくなると TTL の範囲内で探索に失敗する (発見できない) リクエストが増えるが、失敗時はそのリクエストは破棄され、次のリクエストが発生するまでに転送残量が消化される。そのため、転送残量は大きくなり、従来方式でも転送時間はそれほど伸長しない。

図4(d) はリクエストに失敗する確率を測定したもので、各トポロジーでの探索成功率のグラフである。 $G$  が大きくなると探索成功率が低下している。これは、ノード次数が低いトポロジーでのフラッディング過程において、TTL の範囲内で目的のコンテンツを見つけられず、接続要求ができなかったリクエストが時折発生してしまうことを示している。

この結果より、評価方式は  $G$  が小さい、すなわちハブ的なノードが多いときに短縮効率の向上が期待できるといえる。

図4でもう一つ特徴的なことは、リクエスト生起率の増加に対する転送時間の伸長が指数関数的ではなく線形に近いことである。この現象は以下のように解釈できる。

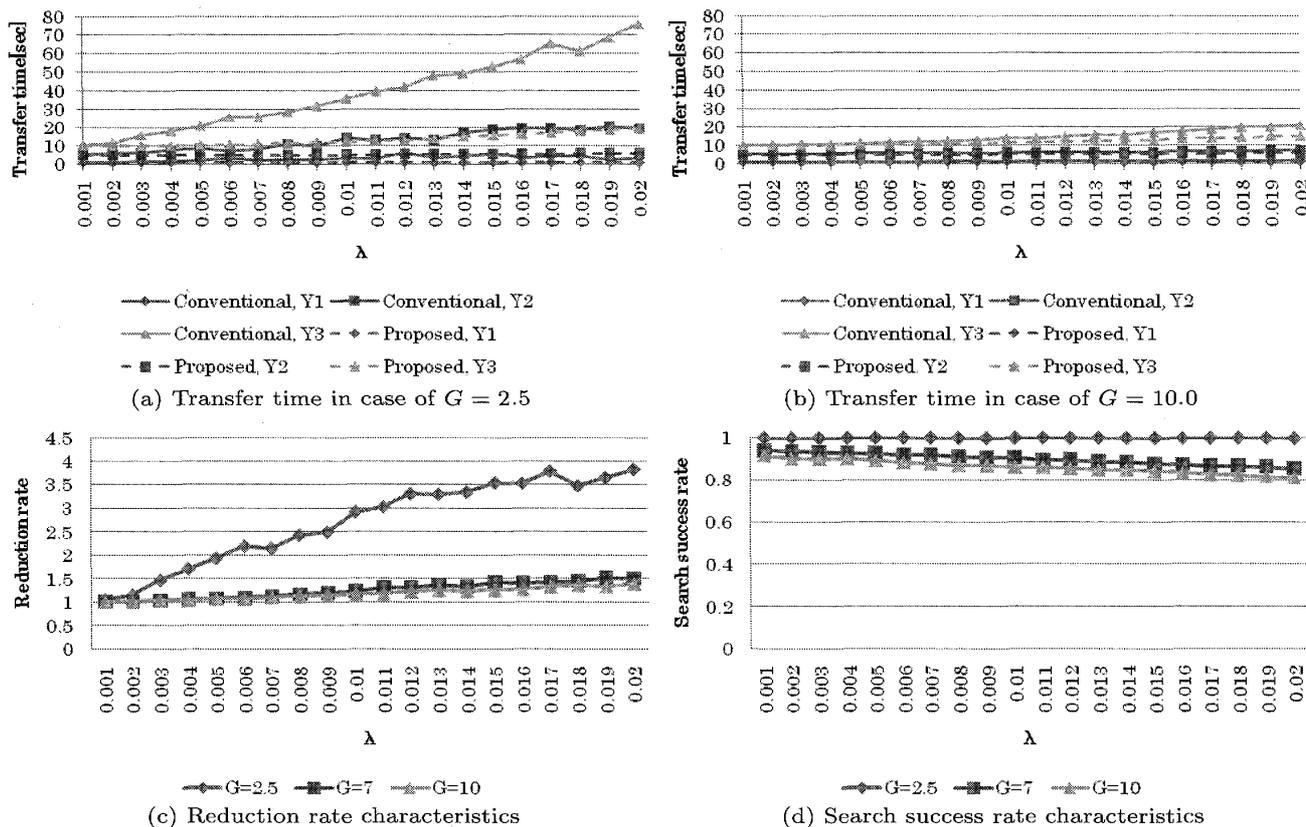


図 4 各トポロジー構造における推移  
Fig. 4 File transfer characteristics in each topology network.

従来方式では、ハブノードに転送要求が集中して転送時間が伸張するが、転送要求を出してしまったノードはそれ以上要求を出すことはなく、以降は蓄積した転送残量は次第にはけていく。一方、評価方式では負荷分散が図られるため、転送要求の集中は軽減されるが、これは転送待ち状態のノードが減ること、すなわちリクエストとなり得るノードが減らないことを意味し、ノード当りのリクエスト生起率が同じであれば評価方式の方が相対的に負荷が大きくなるためである。

#### 5.4.2 TTL を変化させたとき

TTL はコンテンツの探索範囲を決めるパラメータである。本項では、TTL が特性に及ぼす違いを考察する。図 5(a), (b) は、 $\lambda$  を 0.01 に固定し、TTL を変化させたときの転送時間特性を示したグラフである。 $G$  の大小にかかわらず、評価方式により転送時間は大きく改善されている。従来方式では、TTL を大きくしていくと転送時間は増大するが、やがて飽和するようになる。この傾向は、コンテンツ容量が大きいほど顕著である。一方、評価方式では TTL を増加させても転送時間はほとんど変化せず、良好な特性を示す。

図 5(a), (b) を比べると、 $G$  の大きいトポロジーは

ど、転送時間の増加が飽和する TTL の値が大きいことが分かる。これは、 $G$  が大きくなると図 3(a)~(c) のようにトータルの枝数は少なくなり、各ノードの次数も低下して平滑化する結果、図 3(e) のようにノード間の距離が遠くなり、探索に必要な TTL も増えることによる。図 5(c), (d) に示した探索成功率を比べても、 $G$  の大きいトポロジーほど、探索には大きな TTL が必要であることが分かる。探索に必要な TTL は、 $G = 2.5$  で 4、 $G = 10$  でも 6 あれば十分であり、実際、図 5(a), (b) においても、これ以上の TTL では転送時間の増加は飽和している。

TTL が小さい領域では、従来方式と評価方式とで転送時間の差は小さい。これは、TTL が小さい場合は 5.4.1 で述べたようにリクエストに失敗している間に従来方式でも転送残量がはけ、転送残量がたまらないために起こる現象である。

#### 5.5 ネットワーク内のノード総数を変化させたとき

図 6 は 5.4 と同様に生起率  $\lambda$  を変化させたときの転送時間特性 (図 (a)~(c)) と短縮倍率特性 (図 (d)) を示し、ノード総数をパラメータとして比較したものである。ノードの総数によらず、短縮率の変化傾向は

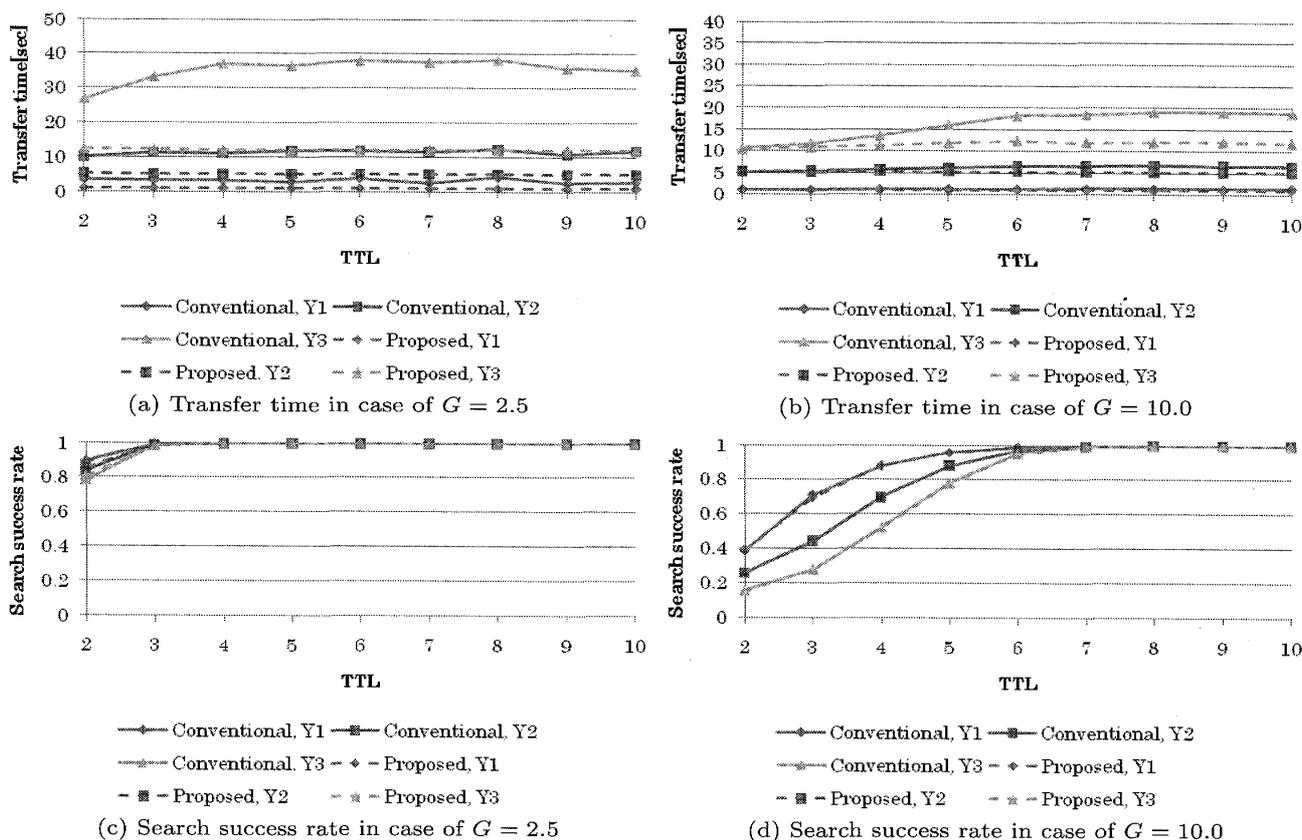


図5 TTLを変化させた場合の推移  
Fig. 5 Characteristics when changing TTL.

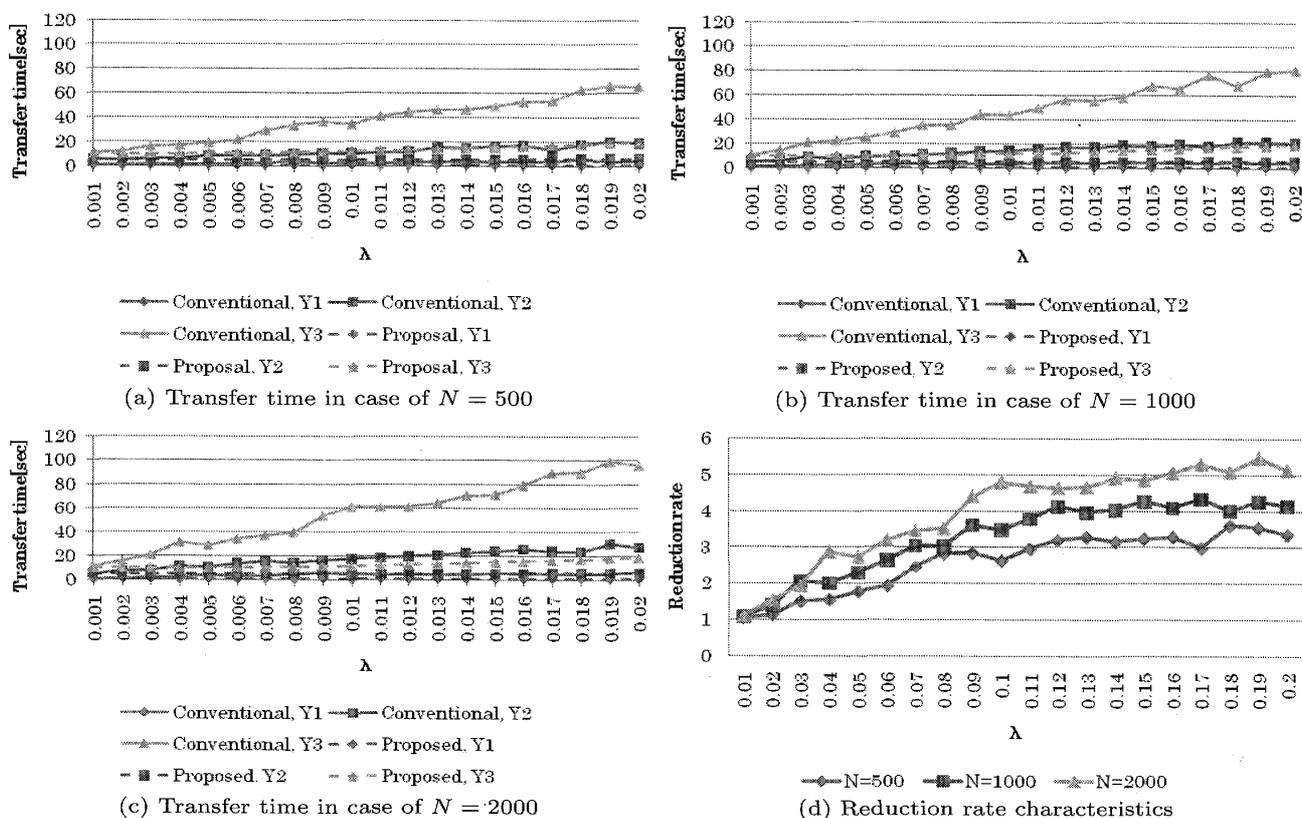


図6 各ネットワーク規模における推移  
Fig. 6 Characteristics in each network scale.

同じであったが、全体的にノード数の多いネットワークほど短縮倍率が高くなった。これはノード数の大きいネットワークほど、従来方式では転送残量が増大し、評価方式では増大が抑制されることによる。ノード数の多いネットワーク ( $N = 2000$ , 図 6(c)) では、 $\lambda$  は 1 ノード当りのリクエスト発生率であるから、 $N$  が増えればネットワーク全体のリクエスト数も増える。その結果、従来方式では、ハブノードへの仕事の集中が起こりやすくなり、転送時間が増える。一方、評価方式では、リクエストの絶対数が増える点は同様であるが、できる限り負荷の集中を避けるようにダウンロードを行う。 $N$  が増えると、初期配置するコンテンツ数もそれに比例して増えるので、 $N$  の数に比例するリクエストの増加があっても、転送時間が増えることはない。更に選択できるノード数も増加するため、評価方式では負荷分散を効率的に進めることが可能となる。この結果、評価方式では転送時間を短縮することができる。

この結果より、今回考案したアルゴリズムは大規模なネットワークであるほど有効であるといえる。

### 5.6 3種類のコンテンツサイズ配分を変化させたとき

図 7 は生起率  $\lambda$  を変化させたときの短縮倍率の推移を示し、コンテンツのサイズ配分をパラメータとして比較したものである。

このグラフでは、コンテンツサイズ配分によらず、リクエストの生起が頻繁になるにつれ、短縮倍率は向上している。これは、従来方式ではハブ的なノードにリクエストが集中してしまうためである。また、3種類のコンテンツサイズにばらつきがあるほど短縮倍率

は高く、コンテンツサイズが均一に近いほど短縮倍率は高くなり、倍率の変化も緩やかである。図 7 では、最も偏りの大きい  $Y1 = 1, Y2 = 2, Y3 = 13$  の場合に短縮倍率が最も高く、最も偏りの小さい  $Y1 = 5, Y2 = 5, Y3 = 6$  の場合は、他の場合に比べて短縮倍率が伸びていない。

コンテンツサイズの配分比率による特性の違いは以下のように解釈できる。今回のシミュレーションでは  $Y1, Y2, Y3$  を要求するリクエストはすべて均等な頻度で発生させており、コンテンツのサイズが均等な場合、3種類のコンテンツは均等にダウンロードされ、ネットワーク内に均等に拡散していく。しかし、コンテンツのサイズに偏りがある場合、サイズの小さいコンテンツはすぐに転送が終わるので先にネットワーク内に広がり、サイズの大きいコンテンツはすぐには広がらない。そのためサイズが小さい  $Y1$  あるいは  $Y2$  がネットワーク全体の 10% 近くまで広がってもサイズの大きい  $Y3$  を保有するノードは数えるほどしかなく、その数少ない  $Y3$  保有ノードにリクエストが集中してしまう。この状態で  $Y3$  がハブ的なノードにあった場合、従来方式ではそのノードに  $Y3$  リクエストが偏ると考えられる。その結果、従来方式では  $\lambda$  が小さくとも  $Y3$  の転送時間が比較的長くなってしまふ。一方で、評価方式では可能な限り仕事量の少ないノードにリクエストを行うことで、ハブノードへのリクエストの集中を避けている。これによって転送時間の増加を避けることができ、評価方式での短縮の効果は大きくなっている。

よって、評価方式は、コンテンツサイズの偏りが大きいほど短縮倍率が高くなるといえる。

### 5.7 その他の特性

本実験では 3.3 で述べたとおり  $Y1, Y2, Y3$  のいずれかが  $N$  の 30% になる時点でシミュレーション完了としている。本節では、シミュレーション完了時点で保有されている各コンテンツ数の比率と、シミュレーション完了までの所要時間を既存方式と評価方式で比較する。なお、デフォルトである  $N = 500, G = 2.5, Y1 = 1, Y2 = 5, Y3 = 10$  の条件下で実験を行っている。

#### 5.7.1 コンテンツ保有率

図 8 にシミュレーション完了時点におけるノードのコンテンツ保有率を示す。

$\lambda$  が大きくなるとピアの転送残量が増大し、サイズの大きい  $Y3$  が広がりやすくなる。この傾向はリクエ

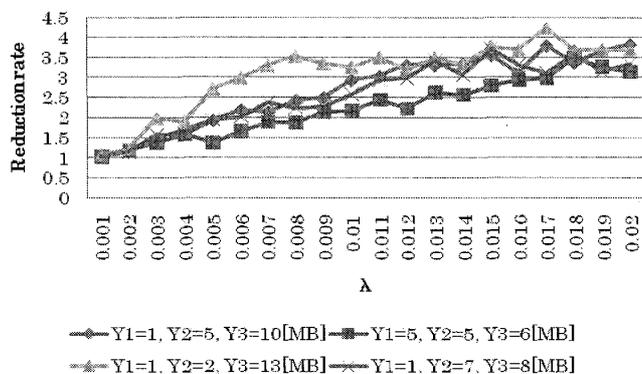


図 7 3種類のコンテンツサイズの各比率における短縮倍率の推移

Fig. 7 Change in the reduction rate in each size ratio of contents.

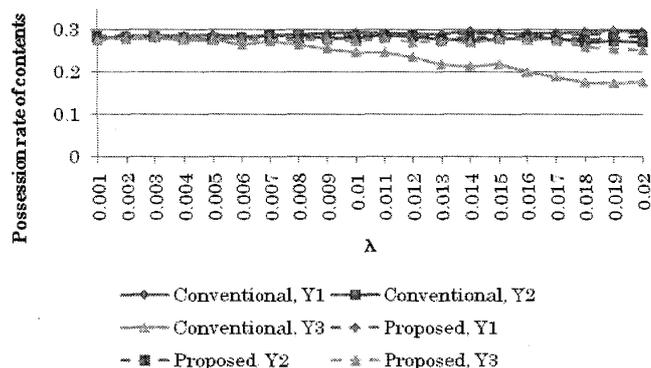


図 8 シミュレーション完了時のコンテンツ保有率  
Fig. 8 Possession rate of contents at the end of simulation.

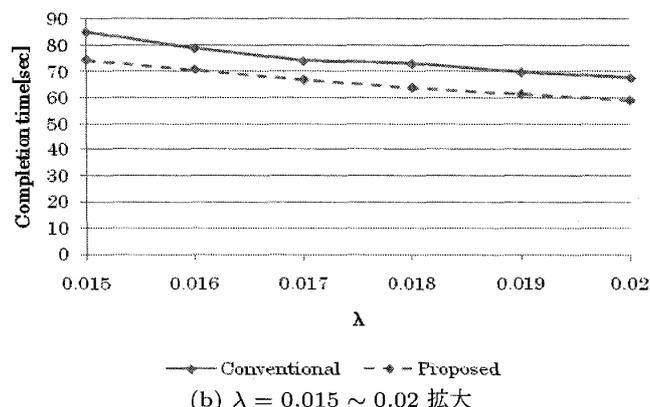
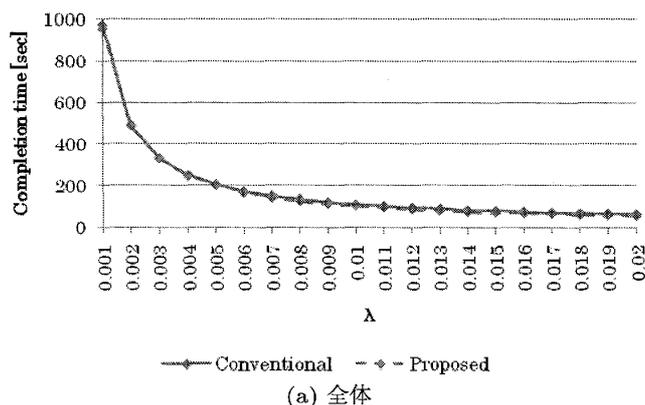


図 9 シミュレーション完了までの所要時間  
Fig. 9 Required time for simulation.

ストの集中が起りやすい従来方式に顕著で、評価方式によっては大幅に改善されている。これは、評価方式がリクエストの分散を行っている効果であると考えられる。

なお、完了条件が Y1, Y2, Y3 のいずれかが 30% 保有された時点であるため、コンテンツ保有率の最大値は 30% である。最もサイズの小さい Y1 が常に先行して完了条件を満たし、Y1 が 30% となるように思われるが、 $\lambda$  が小さい領域では Y1 が 30% 以下になる場合がある。これは、必ずしも Y1 が完了条件を満たすのではなく、Y2 や Y3 が完了条件 (30%) を満たしたことでシミュレーションが完了する場合があることを示している。転送残量の蓄積が大きならない場合は、Y2 や Y3 にリクエストが偏れば、Y2 や Y3 が先に完了条件を満たすことになる。この傾向は、特に  $\lambda$  の小さい領域において強く現れている。

### 5.7.2 シミュレーション完了時間

図 9 にシミュレーション完了に要した時間特性を示す。図 9(a) は全体を、図 9(b) は  $\lambda = 0.015 \sim 0.02$  の領域を拡大して示している。ここでいう「完了時間」とは、シミュレーション上の経過時間で、終了までに要した時間である。なお、転送残量が最も少ないノードを選択するのに要する時間は含まない。 $\lambda$  が同じであれば、時間がかかっているほどコンテンツの拡散が遅いことを表す。コンテンツの拡散が速い転送方式であるほど、帯域の利用効率が良い転送方式であるといえる。

リクエストがまばらである  $\lambda = 0.001 \sim 0.005$  程度では、ほとんど転送残量が貯まらないため、従来方式と評価方式の差はほとんど見られない。図 9(a) に示

すように、この領域で極端にシミュレーション時間が長いのは、 $\lambda$  が小さいためにリクエストの間隔が非常に長いことによる。

$\lambda$  が大きくなると、転送残量がピアに蓄積するようになり、負荷分散のなされない従来方式では、転送時間が長くなる。このため、従来方式ではファイルの拡散が遅くなり、シミュレーション完了までの時間も長くなってしまう。 $\lambda = 0.02$  では、図 9(b) に示すように、従来方式は評価方式に比べ 14% 程度完了時間が長引いている。

また、5.7.1 で述べたとおり、評価方式では重いコンテンツである Y3 の拡散も非常に速いことから、リクエストの集中を避ける評価方式が大幅にコンテンツの拡散速度の向上に貢献していることがうかがえる。

## 6. むすび

本論文では、非構造型 P2P ネットワークを対象に、発見したノードの中でファイル転送残量が最も少ないノードからファイルを取得するファイル共有方式について、シミュレーションによる評価を行った。その結果、様々なネットワーク状況においてファイル共有の

効率化が可能であることを実証できた。評価方式は、負荷を分散させることでファイル取得時間を短縮することが可能であり、ハブ的なノードが多いほど、ネットワークが大規模であるほど、コンテンツサイズの偏りが大きいほど、評価方式による短縮効果は大きくなる。

なお、本論文では一定の条件下でシミュレーションを行ったが、実際のネットワークに適用する場合は、ファイルの種類も増え、人気コンテンツへのクエリ頻度も高くなると予想される。クエリ頻度が増加すると、既存方式では特定のノードにリクエストが集中すると考えられるが、今回の評価結果でもクエリ頻度が大きいときに評価方式の効果は大きくなるため、コンテンツの人気度に偏りがある場合には、負荷が偏る既存方式に比し負荷を分散させる評価方式によるアプローチは効果的であると予想される。なお、一般にコンテンツのアクセス頻度には偏りがあるとの報告がある [22], [23]。

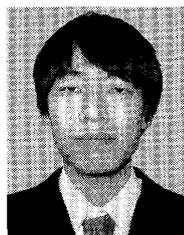
また、ノードのストレージ容量は有限であるため、コンテンツのどれを残すかといったキャッシュの問題がある。今回はコンテンツの入換えのない理想ケースでの評価になっており、ストレージ容量やキャッシュの影響は今後の課題である。

更に今回の評価では、ノードの参加・離脱を考慮していないが、一般に非構造型の Gnutella ネットワークは、構造型の DHT や BitTorrent に比べてノードの参加離脱の影響は少ないといえる。今回の評価においても、新規参加ノードはコンテンツ保持者とはならないのでピアの選択結果は変わらず、評価結果に影響しない。ノードの離脱については、もし各ノードが等確率で離脱するのなら評価結果は変わらないが、等確率ではなく、例えばハブノードはなかなか離脱せず、その他のノードは離脱しやすい場合には、既存方式では離脱ノードの負荷がハブに向けられて負荷集中が進み、評価方式でも他の一般ノードに向けられて転送時間は伸張すると考えられる。詳細な評価は今後の課題としたい。

## 文 献

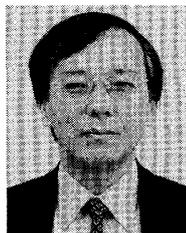
- [1] K. Aberer and M. Hauswirth, "An overview on peer-to-peer information systems," Workshop Distributed Data and Structures WDAS-2002, 2002.
- [2] 岩田真一, 井上誠一郎, 大谷弘喜, 亀井 聡, "特集: P2P 技術の基礎知識," UNIX MAGAZINE, vol.20, no.9, pp.82-107, Sept. 2005.
- [3] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys & Tutorials, Second Quarter 2005, vol.7, no.2, pp.72-92, 2005.
- [4] B. Yang and H. Garcia-Molina, "Designing a super-peer network," Proc. 19th Int'l Conf. on Data Engineering ICDE, March 2003.
- [5] BitTorrent.<http://www.bittorrent.com/>
- [6] Y. Chawathe, S. Ratnasamy, and L. Breslau, "Making Gnutella-like P2P systems scalable," Proc. 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.407-418, Karlsruhe, Germany, Aug. 2003.
- [7] Gnutella Development Forum, "The Gnutella v0.6 protocol," 2001.
- [8] D.S. Bernstein, Z. Feng, B.N. Levine, and S. Zilberstein, "Adaptive peer selection," Lecture Notes in Computer Science, Springer-Verlag, Berlin, vol.2735, pp.237-246, 2003.
- [9] H. Xie, Y.R. Yang, A. Krishnamurthy, Y.G. Liu, and A. Silberschatz, "P4P: Provider portal for applications," ACM SIGCOMM Computer Communication Review, vol.38, no.4, pp.351-362, Aug. 2008.
- [10] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPS and P2P users cooperate for improved performance?," ACM SIGCOMM Computer Communication Review, vol.37, no.3, pp.29-40, July 2007.
- [11] Y. Guo, K. Suh, J. Kurose, and D. Towsely, "P2Cast: Peer-to-peer patching scheme for VoD service," ACM WWW2003, pp.301-309, Budapest, Hungary, May 2003.
- [12] T.T. Do, K.A. Hua, and M.A. Tantaoui, "P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment," 2004 IEEE International Conference on Commun., vol.3, pp.1467-1472, Paris, France, June 2004.
- [13] C. Dana, D. Li, D. Harrison, and C.N. Chuah, "BASS: BitTorrent assisted streaming system for video-on-demand," 2005 IEEE 7th Workshop on Multimedia Signal Processing, pp.1-4, Shanghai, China, Oct. 2005.
- [14] Y. Huang, T.Z.J. Fu, D. Chiu, J.C.S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," ACM SIGCOMM Computer Communication Review, vol.38, no.4, pp.375-388, Oct. 2008.
- [15] 牧村真吾, 三好 匠, "ファイル転送を考慮した ALM ルーティング手法の最適化," 信学技報, NS2004-49, June 2004.
- [16] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," Proc. First International Conference on Peer-to-Peer Computing, pp.99-100, Aug. 2001.
- [17] B. Cohen, "Incentive build robustness in BitTorrent," Proc. First Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, June 2003.

- [18] 和田 維作, “乱数ライブラリ,” 和田 維作のホームページ, (オンライン) <http://www001.upp.sonet.ne.jp/isaku/rand2.html>
- [19] 増田直紀, 今野紀雄, 複雑ネットワークの科学, 産業図書, 2005.
- [20] O. Densmore, “An exploration of power-law networks,” (online) <http://backspaces.net/sun/PLaw/index.html>
- [21] L.A. Adamic and B.A. Huberman, “Power-law distribution of the World Wide Web,” *Science*, vol.287, no.5461, p.2115, March 2000.
- [22] H. Yu, D. Zheng, B.Y. Zhao, and W. Zheng, “Understanding user behavior in large-scale video-on-demand systems,” *Proc. 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, pp.333–344, Leuven, Belgium, April 2006.
- [23] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “YouTube traffic characterization: A view from the edge,” *Proc. 7th ACM SIGCOMM Conference on Internet Measurement*, pp.15–28, San Diego, CA, USA, Oct. 2007.  
(平成 20 年 9 月 30 日受付, 21 年 2 月 9 日再受付)



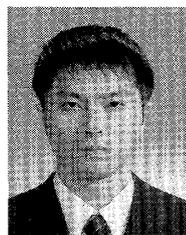
鈴木 慧

筑波大・図情在学中. P2P ネットワークにおけるプログレシブダウンロード方式に関する研究に従事.



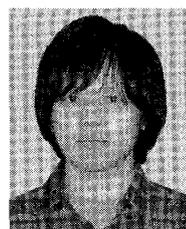
川原崎雅敏 (正員)

昭 50 京大・工・電気卒. 昭 52 同大大学院修士課程了. 同年日本電信電話公社(現 NTT) 入社. 以来, トラフィック制御, B-ISDN/ATM ネットワーク構成法, コンテンツ流通管理技術等の研究に従事. 平 16 より筑波大学大学院図書館情報メディア研究科教授, 現在に至る.



川原 亮一 (正員)

平 2 早大・理工・電気卒. 平 4 同大大学院修士課程了. 同年日本電信電話(株) 入社. 以来, インターネットトラフィック制御・測定・管理・設計技術に関する研究に従事. 現在, NTT サービスインテグレーション基盤研究所特別研究員.



大沼 智洋

平 20 筑波大・図情卒. P2P ネットワークにおけるファイル検索・取得の効率化に関する研究に従事. 現在, トップランフォームズ(株) 勤務.