

デマンド型配送方式のネットニュースシステムへの適用と評価

菅野浩徳[†] 曽根秀昭^{††} 根元義章^{†††}

現在のネットニュースシステムでは、隣接するニュースサーバ間で記事データの転送と蓄積を繰り返すことで、最終的にすべてのサーバに対して記事の配送が行われる。しかし各ニュースサーバに配送された記事データはすべてが有効に利用されているとは限らず、なかにはまったく利用されることなく時間の経過とともに削除されるといった記事データも存在し、無駄な配送と蓄積を生じている。さらにインターネット利用者数の増加や利用形態の多様化にともなって、サーバ間の通信トラヒックの増加やサーバにおける蓄積領域の大容量化など、システムの維持・運用が困難となりつつある。そこで、これら現行のシステムにおける問題を解決するためには、ユーザからのリクエストに応じて目的とする記事データを配送するデマンド型が有効であると考えられる。デマンド型では、素早い応答を得るためにはサーバとの通信回線が十分に高速である必要があるが、ネットワークの広帯域化が進行しており、今後データ伝送遅延時間は著しく改善されていくものと考えられる。本稿では、このような状況をふまえ、利用者の要求に応じた配送を行うことで高効率な配送を可能とする、デマンド型配送方式を提案する。そして、当方式について実装を行い実験によって評価する。実験によれば、記事の利用効率が向上しサーバ間の不要な配送トラヒックが抑制されることが確認でき、提案方式の有効性が示された。

Application of an On-Demand Delivery Method to NetNews System and Its Evaluation

HIRONORI KANNO,[†] HIDEAKI SONE^{††} and YOSHIAKI NEMOTO^{†††}

In the present NetNews system, articles are delivered to all servers by repeating transmission and accumulation of article data between the adjoining news servers. However, the article data delivered by each news server is not necessarily used effectively. Some article data is deleted, without being used once. It has produced useless delivery traffic and useless accumulation. Furthermore, the maintenance and management of the system are becoming difficult, because the network traffic between servers and the storage area on server are increasing. To overcome these problems, an On-Demand delivery system which delivers required article data according to the request from a user is effective. In this system, in order to obtain a quick response, the network speed between servers needs to be high-speed. But, network is growing to a broadband network, therefore data transmission time will be improved remarkably. In consideration of such situation, in this paper, we propose an On-Demand NetNews delivery system. This method delivers the articles according to the demand of a user and enables highly efficient delivery. We designed and implemented a method to NetNews system, and we evaluated it by experiment. From the result of this experiment, the use efficiency of article has improved and the unnecessary delivery traffic between servers was able to be stopped.

1. はじめに

ネットニュースは、ネットワークを介して情報の共有や特定の話題に関するディスカッションを行える分散型掲示板システムであり、従来よりインターネット

における主要なアプリケーションの1つとして利用者も多い。しかし近年のインターネット利用者数の増加や利用形態の多様化にともなって、配送記事の数やサイズがともに増加しており、サーバ間で分散蓄積を行うための通信トラヒックの増加やサーバにおける蓄積領域の大容量化など、その運用コストが肥大化しシステムの維持・運用が困難となりつつある。

最近ではサーバ間の配送データ量が1日250GB(受信)を超える場合もあり、さらに増加傾向にある^{1)~3)}。このようなサーバ間の配送トラヒックの増加は、配送に必要な回線帯域の増強のみならず、サーバシステム

[†] 株式会社富士通東北システムエンジニアリング
Fujitsu Tohoku Systems Engineering Ltd.

^{††} 東北大学情報シナジーセンター
Information Synergy Center, Tohoku University

^{†††} 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

の増強 (CPU や DISK) など必要となる場合が多く、運用コストをいっそう増加させる。しかも各ニュースサーバに配送された記事データは、すべてが有効に利用されているとは限らず、なかにはまったく利用されることなく時間の経過とともに削除されるといった記事データも存在する。たとえば、文献 4)、5) によれば、観測サーバにおける利用率はいずれも 13% 以下にとどまっている。利用率とは、読まれた記事数とニュースサーバ上の配送記事数との比率である。つまりせっかく配送されたにもかかわらず一度も読まれることのない記事データが多数存在し、無駄な配送と蓄積を生じている。

現行のネットニュースシステムでは、隣接するニュースサーバ間で記事データの転送と蓄積を繰り返すことで、最終的にすべてのサーバに対して記事の配送が行われる。記事の配送選択はニュースグループ単位に可能であるが、グループ数が多くなるにつれ、細かな制御は困難となっている。また、記事数の増加によりすべての記事について目を通すことが難しくなっており、結果として利用率の低下を招く。利用率の低さは、ネットワークシステムにおける効率性から見て好ましくなく、利用のないデータを配送し蓄積することから、先述した運用コスト問題の一因にもなっており、改善を要する。

ニュースサーバ間の配送は、クライアントからのアクセスとは独立に行われ、配送の際にその記事に対する需要を知る術がない。すなわち、サーバ利用者の需要を反映しておらず、また利用者の需要を反映する配送の仕組みを有していない。これが、利用率を低下させ、読まれることのない無駄な記事配送と蓄積を生じる大きな要因の 1 つと考えられる。

これら現行のシステムにおける問題を解決するためには、ユーザからのリクエストに応じて目的とする記事データを配送するデマンド型が有効であると考えられる。一方、従来のシステムと比較して、デマンド型では記事の要求から配送完了までのデータ伝送時間が加算されるため、素早い応答を得るためにはサーバとの通信回線が十分に高速である必要がある。これについては、近年ネットワーク技術の発展によってインターネットに利用されるネットワークについてもその広帯域化が急速に進行しており、今後、データ伝送時間は著しく改善されていくものと考えられる。

本稿では、このような状況をふまえ、利用者の要求に応じた配送を行うことで高効率な配送を可能とする、デマンド型配送方式を提案する。さらに、提案方式を実際に実装して、実験により評価し、その有効性を検

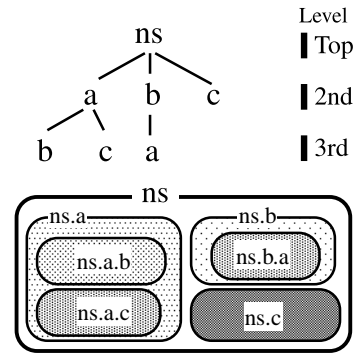


図 1 ニュースグループの木構造と模式図
Fig. 1 A structure of newsgroup.

証する。

2. ネットニュースシステムの概要

ネットニュースでは、記事 (Article) が情報の単位であり、その標準メッセージフォーマットなどは RFC1036⁶⁾ で規定されている。そして、話題や用途によって記事を集約するためのニュースグループ (News Group) という枠組みがある。このニュースグループは、あるカテゴリ名をトップレベル名とする木構造の名前空間を持ち、このニュースグループを単位として記事の配送制御や管理が行われている。ニュースグループの名前空間構造を図 1 に示す。上が木構造図、下が模式図である。この図では例としてカテゴリ名 ns をトップレベル名としている。

ニュースグループ名は、英小文字と数字、およびプラス '+', ハイフン '-', アンダースコア '_' の記号から構成され、各階層はドット '.' で区切られる。各階層は 14 文字までである⁷⁾。この模式図から分かるように、各ニュースグループはそれぞれに閉じた名前空間で記事を管理する。ニュースグループには、たとえばビッグエイト (Big Eight) と呼ばれる comp, humanities, misc, news, rec, sci, soc, talk や、日本語の利用が主体である fj, japan などのトップレベル名がある。

記事は、記事のプロパティを示しニュースソフトウェアで使われる制御情報の集合であるヘッダ部と、記事の内容であるボディ部から成る (図 2)。ヘッダ部には、Date:, From:, Message-ID:, Subject:, News-groups:, Path: の各ヘッダ行が必須とされる。各ヘッダの詳細については RFC1036⁶⁾ を参照願いたい。

図 3 に、ネットニュースの配送モデルを示す。図中の S0~S4 はニュースサーバを、C1~C4 はニュースリーダが動作するクライアントを示す。

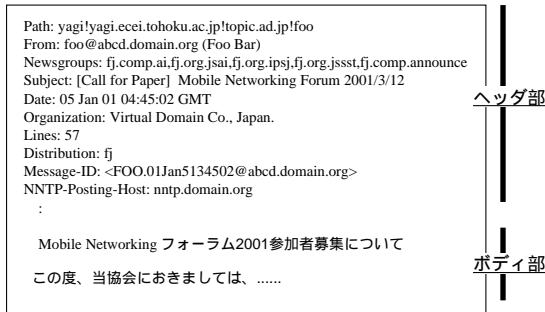


図 2 記事の構造

Fig. 2 A structure of article.

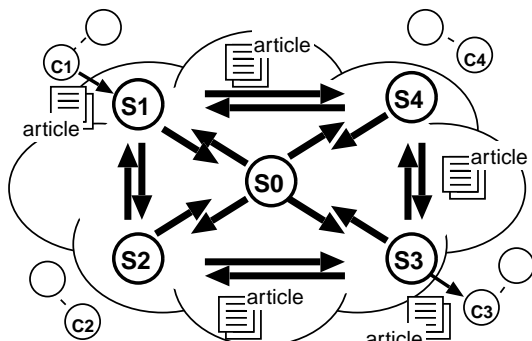


図 3 ニュース配送モデル

Fig. 3 A NetNews delivery system model.

ネットニュースでは、隣接するニュースサーバ間で、記事の転送と蓄積を繰り返すフラッドアルゴリズム (Flooding Algorithm⁷⁾) によって配送が行われる。たとえば、C1 から新たに投稿された記事は、S1 に蓄積され、そこから隣接関係にある S0, S2, S4 などに配送され、さらに S3 にもいずれかのサーバから配送される。このようにして配送された記事は、それぞれのサーバが用意しているスプール上に一定期間保持され、その期間を過ぎると削除される。この保存期間パラメータの決定は、サーバのスプール容量や利用者の興味度合いなどに依存する 경우가多く、サーバによって購読可能な期間と量に差異がある。

このように現状のネットニュースシステムにおいては、ニュースサーバ間の配送はプッシュ型(データ提供側主導)であり、ニュースサーバとクライアント間の配送はデマンド型(データ取得側主導)である。したがって、クライアントからのアクセスとは独立に行われ、配送の際にその記事に対する需要を知る術がない。

3. デマンド型配送方式

3.1 概要

記事の利用率を高め、効率的な配送を実現するためには、ユーザからのリクエストに応じて目的とする記事データを配送するデマンド型が有効であると考えられる。

このとき、特定のサーバにすべての記事を集約して、クライアント側からはそのサーバに直接アクセスし記事を取得する方式では、記事データの蓄積領域の肥大化やサーバへのアクセスの集中などといった問題がある。そこで記事データを複数のサーバに分散蓄積し、クライアント側からは選択的に個々のサーバにアクセスして記事を取得する方式を考えた。そのためには、どのサーバでこういった記事を保持しているかといった情報と利用者が記事を取捨選択するための情報を管理する仕組みも必要である。

これらの要件を満たす方式として、記事のインデックス情報を管理するディレクトリサーバと、オリジナルの記事データを管理する一次サーバと、クライアントと関係してディレクトリサーバや一次サーバへのアクセス中継とデータキャッシュを行う二次サーバから成る、デマンド型配送方式を提案する。本方式では従来のシステムと比較して、記事の要求から配送完了までのデータ伝送時間が待ち時間として加算されるため、素早い応答を得るためにはサーバとの通信回線が十分に高速である必要がある。近年ネットワーク技術の発展によってインターネットに利用されるネットワークについてもその広帯域化が急速に進行しており、今後、データ伝送時間は著しく改善されていくものと考えられ、このような広帯域ネットワークの利用によれば、記事データはより高速に伝送され、この問題は解決される。

3.2 構造

提案方式は、ディレクトリサーバ、一次サーバ、二次サーバ、クライアントの4種類のノードから構成される。以下に各ノードの主な役割を説明する。

ディレクトリサーバ： ニュースグループ、記事インデックスの管理を行う。二次サーバからの要求に応じてそれらのデータを配送する。

一次サーバ： クライアントからの投稿記事を蓄積し、二次サーバからの要求に応じて記事の配送を行う。新たな記事が投稿された際には、当該記事の記事インデックスをディレクトリサーバに送信する。

二次サーバ： クライアントからのリクエストに応じてディレクトリサーバや一次サーバにアクセスし、

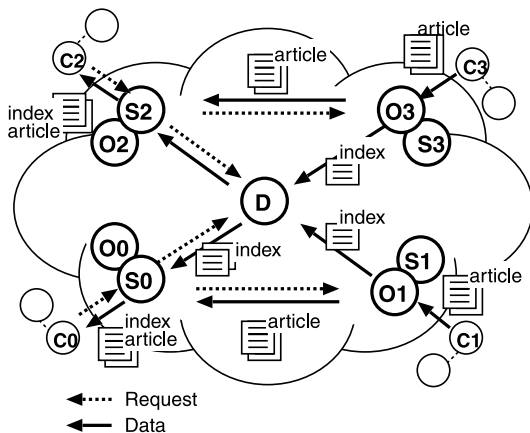


図4 デマンド型配送方式モデル

Fig. 4 An on-demand delivery system model.

記事インデックスや記事の配送を中継する。二次サーバでは、一度配送されたデータを一定時間保持する機構（キャッシュ）を持つことで、ディレクトリサーバや一次サーバとの重複する配送を抑制する。

クライアント： ネットニュースの記事を読み書きできるニュースリーダーと呼ばれるソフトウェアが動作する端末装置。二次サーバを自サイトのニュースサーバとして動作する。

一次サーバ機能と二次サーバ機能は、同一サーバ上での動作が可能である。各サーバは広域なネットワーク上に配置される。

デマンド型配送方式のモデルを図4に示す。O0～O3は一次サーバを、S0～S3は二次サーバを、Dはディレクトリサーバを、C0～C3はクライアントを示す。図4のネットワークやサーバ構成は論理的なものであり、物理的な構成を示すものではない。

たとえば図4において、C1からの投稿記事はO1に蓄積され、同時にO1は投稿記事の記事インデックスをDに送る。C0が記事を購読する場合、C0はS0に記事インデックスを要求する。S0はDにその要求を中継し、Dから記事インデックスを受け取り蓄積するとともにC0に配送する。C0は得られた記事インデックスの情報を参照し、読みたい記事を選択してS0に記事の要求を送る。仮にその記事データをO1が保有しているとする、S0はO1に要求を中継し、O1から記事を受け取り蓄積するとともにC0に配送する。

3.3 記事データの管理方式

新たに投稿された記事には、一次サーバ内で一意な識別名を与え、その識別名によって記事データへのアクセスが可能であることとする。すべての一次サー

バは、たとえば o1.foo.com といった FQDN (Fully Qualified Domain Name) によって、その一意性が保証されることとする。そこで、これらの条件から、一次サーバ上に蓄積される記事データのロケーションは、URL (Uniform Resource Locators) によって示すことが可能となる。本提案方式では新たに “n3” URL スキームを定義する。この URL は次のような構文をとる。

n3 : // host / path

‘host’ は一次サーバの FQDN によるホスト名で、‘path’ は一次サーバ内で与えられた記事データの識別名である。たとえば、ある記事データが認識名 ‘15’ として一次サーバ o1.foo.com に蓄積されるとすると、そのロケーションは n3://o1.foo.com/15 と示すことができる。

記事データの識別名はあくまで記事データを特定するためのポインタであり、そのままに実際のファイルのパスを示すものではない。実装にあたっては、使用する OS やファイルシステムの特性に依りて、最適な格納方式を選択する必要がある。また、その格納方式に応じた識別名と記事データとのマッピングの方法もあわせて検討を要する。

一度割り当てた URL はその記事データだけに有効であるよう、一意性を保証するものとする。たとえば n3://o1.foo.com/15 の記事データが削除されたとしても、新たな記事データのために使われることはないものとする。このことによって、ディレクトリサーバではインデックスデータの管理を確実かつ容易なものとし、二次サーバ側では WWW などのキャッシュシステムのようなリロード処理などの実装が不要となり簡素化できる。ディレクトリサーバや一次サーバが管理するインデックスデータや記事データへのアクセスは、キャッシュデータが削除された場合を除けば、同一データについてたかだか 1 回に抑えられる。

3.4 記事インデックス

記事インデックスは、図5に示すように、投稿記事のヘッダから利用者が記事の選択を行うために用いられている項目を切り出し、管理番号を X-Seq: ヘッダとして、そして一次サーバに蓄積された記事のロケーションを X-Ur1: ヘッダとして付与したものを加えたりリストとする。

ディレクトリサーバは複数設置して、ニュースグループ単位で分担させることも可能であると考えている。これによって、ニュースグループの管理を容易にし、アクセスを分散させる効果が期待できる。

```
X-Seq: 0000000001
Newsgroups: ns.rec.tv
Message-ID: <03sp.siw@foo.com>
Date: 17 Feb 2000 21:54:38 +0900
From: bar <bar@foo.com>
Subject: Information
Lines: 41
X-Url: n3://news.foo.com/82
```

図 5 記事インデックスの例

Fig. 5 An example of article index.

newsgroup	high	low	flag
ns.rec	0000000020	0000000001	m
ns.rec.radio	0000000010	0000000001	y
ns.rec.tv	0000000023	0000000002	y
ns.rec.tv.talk	0000000111	0000000001	y

図 6 グループファイルの例

Fig. 6 An example of group file.

3.5 ニュースグループの管理

ニュースグループは、グループファイルによって管理する。グループファイルは、図 6 に示すようなデータベースであり、ニュースグループ名、蓄積している記事範囲（最大番号、最小番号）、フラグのフィールドから成る。これは従来のニュース配送システムである Cnews や INN などの active ファイルと同等な仕様である。グループファイルは、ディレクトリサーバ、二次サーバそれぞれに配置される。ディレクトリサーバに配置されるグループファイルは、そのディレクトリサーバが管理するニュースグループについての情報を持つ。二次サーバに配置されるグループファイルは、その二次サーバが購読対象とするニュースグループについての情報を持つ。

3.6 ディレクトリサーバの分散方式

配送規模の拡大（サーバ数や利用の増大）にともなって、広域に分散配置される一次サーバや二次サーバと比べてディレクトリサーバへのアクセス集中が特に懸念される。そこで、提案するデマンド型配送方式では、すべてのニュースグループを単一のディレクトリサーバに集約しなくともよく、複数のサーバで分散管理できるようにする。これにより、図 7 に示すように、ディレクトリサーバへのアクセスを複数のサーバに分散し、1 台あたりの負荷を低減させる効果を期待できる。D および D1 ~ Dn はディレクトリサーバを、S1 ~ Sn は二次サーバを示す。

複数のディレクトリサーバによる分散管理を可能とするために、どのニュースグループがどのディレクトリサーバで管理されるかというロケーション情報を二次サーバ側に与える。ここでは、ニュースグループ

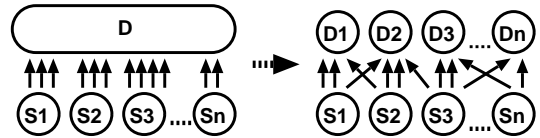


図 7 ディレクトリサーバの分散とアクセス負荷の低減

Fig. 7 Distributing of directory server and reduction of the access load.

newsgroup	server
ns.rec	d1.bar.net
ns.rec.radio	d1.bar.net
ns.tv	d2.bar.net
ns.tv.talk	d2.bar.net:d3.foo.net

図 8 ディレクトリサーバファイルの例

Fig. 8 An example of directory server file.

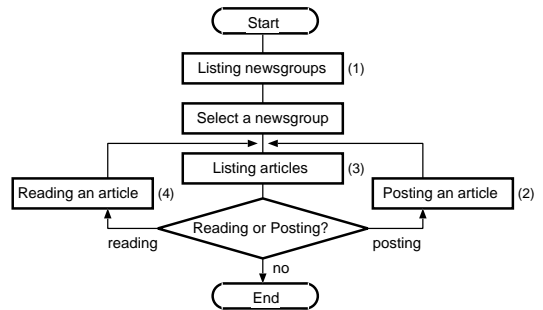


図 9 記事の投稿および参照フロー

Fig. 9 Flowchart for the posting and reading of article.

名とディレクトリサーバ名からなるテーブルを保持するためのディレクトリサーバファイルを新たに用いる（図 8）。これにより、目的とするニュースグループ名からそのグループと記事を管理するディレクトリサーバを特定できる。

ディレクトリサーバ名は記号 ‘:’ を挟んで複数記述できるものとする。先頭はプライマリサーバで、オリジナルデータを持つサーバとする。それ以降のサーバを分散サーバとする。アクセスの順番については、通常はプライマリサーバにアクセスして、故障などによって応答がない場合に次の分散サーバにアクセスを切り替える方式や、ラウンドロビン方式によって順次アクセス先を切り替える方式などが考えられるが、これは運用に応じて最適な方式を選択できるものとし、本稿ではその詳細までは規定しない。

3.7 システムの動作とプロトコル

クライアントからの記事の投稿および購読は、図 9 に示すような一連の手続きにより行われる。このとき、図 9 中の (1) ではグループファイルが (3) では記事インデックスファイルが (4) では記事ファイルが

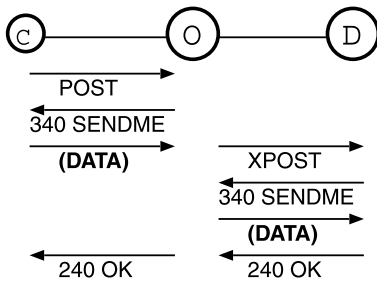


図 10 記事の投稿
Fig. 10 Process of posting an article.

必要となる．記事インデックスファイルは (2) の処理によって生成される．

クライアントと一次サーバ，二次サーバ間の通信プロトコルは，RFC977 (Network News Transfer Protocol⁹⁾) に準拠したコマンドとメッセージを使用する．このためこれまでのニュースリーダがそのまま利用可能である．また，一次サーバ，二次サーバ，ディレクトリサーバ間の通信には，RFC977 準拠コマンドのほか，以下のコマンドを追加している．

- XPOST
一次サーバからディレクトリサーバに記事インデックスを送るために用いる．
- XARTICLE
二次サーバが一次サーバから記事データを取得するために用いる．

当システムの動作について，記事の投稿の場合と購読の場合を，それぞれ図 10 と図 11 に示す．

まず，記事の投稿の場合について説明する．図 10 において，D はディレクトリサーバを，O は一次サーバを，C はニュースリーダが動作するクライアントを示す．矢印はコマンドおよびデータの送受信方向を示す．

C は記事の送信のため O に POST コマンドを発行する．O は記事の受け入れが可能な状態であればメッセージコード (340) で返答する．そのメッセージを確認して C は記事データを O に送信する．続いて，記事インデックスを O に送信するために XPOST コマンドを D に送信する．O は記事インデックスの受け入れが可能な状態であればメッセージコード (340) で返答する．そのメッセージを確認して O は，受け取った記事データから記事インデックスデータを作成し D に送信する．正常にデータ送信が完了すれば，D は O に転送完了メッセージコード (240) を返し，O は同様に C にメッセージコード (240) を返すことで記事の投稿処理が完了する．

次に記事を購読する場合について説明する．図 11

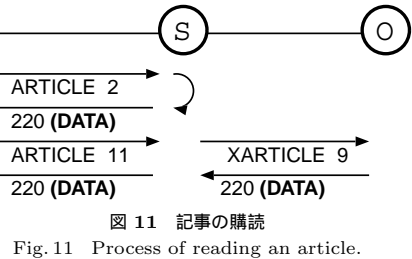
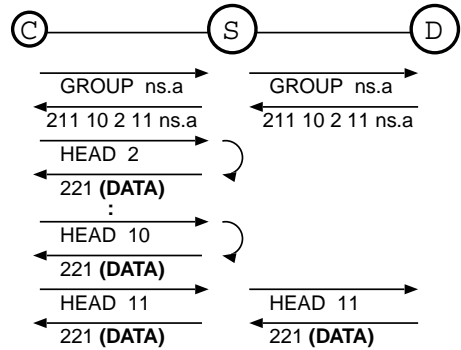


図 11 記事の購読
Fig. 11 Process of reading an article.

において，D はディレクトリサーバを，O は一次サーバを，S は二次サーバを，C はニュースリーダが動作するクライアントを示す．矢印はコマンドおよびデータの送受信方向を示す．ここで二次サーバ S は，ニュースグループ ns.a について，記事番号 2 から 10 までの記事インデックスと記事をすでにキャッシュ領域に保持しているものとする．

C はニュースグループを選択するため S に GROUP コマンドを発行する．S は GROUP コマンドのオペランドで指定されたニュースグループが利用可能であれば，D にその GROUP コマンドを中継する．D は GROUP コマンドで指定されたニュースグループが利用可能であれば，メッセージコード (211) に最小記事番号 A_{min} ，最大記事番号 A_{max} ，蓄積記事数 A_{num} を示して返答する．S はそのメッセージコード (211) を C に中継する．次に C は S に対して HEAD コマンドを A_{min} から A_{max} まで順に送信する．S は HEAD コマンドで要求された記事インデックスをすでにキャッシュ領域に保持していれば，D にアクセスすることなく，保持する記事インデックスデータをメッセージコード (221) に続いて送る．要求された記事インデックスを保持していなければ，D に対して HEAD コマンドを中継する．D は要求された記事インデックスデータをメッセージコード (221) に続いて送る．S は届いた記事インデックスデータをキャッシュ領域に蓄積するとともに，C にメッセージコード (221) と記事インデックスデータを中継する．この一

連の処理により、 C は A_{num} 個の記事インデックスを得る。

図 11 の場合、 $(A_{min}, A_{max}, A_{num}) = (2, 11, 10)$ であり、 S は記事番号 2 から 10 までの記事インデックスについては D にアクセスすることなくキャッシュ領域に保持する記事インデックスデータを C に返し、記事番号 11 については D にアクセスして記事インデックスデータを得て、それをキャッシュ領域に蓄積するとともに、 C に記事インデックスデータを送る。

続いて C は得られた記事インデックスの情報を参照し読みたい記事を決定して、 S に ARTICLE コマンドを発行する。 S は要求された記事データをキャッシュ領域に保持していれば、 O にアクセスすることなく、メッセージコード (220) に続いて保持する記事データを送る。要求された記事データを保持していなければ、 O に対して XARTICLE コマンドを発行する。 O はメッセージコード (220) に続いて要求された記事データを送る。 S は届いた記事データをキャッシュ領域に蓄積するとともに、 C にメッセージコード (220) と記事データを中継する。

図 11 の場合、 S は記事番号 2 の記事データについては O にアクセスすることなくキャッシュ領域に保持する記事データを C に返し、記事番号 11 については O にアクセスして記事データ (ここでは O 上の記事番号 9) を得て、それをキャッシュ領域に蓄積するとともに、 C に記事データを送る。

4. 配送効率の評価

配送効率の指標となる記事の利用効率と配送量について、実験により評価を行う。

4.1 準備

まず、提案システムの実装を行った。開発したシステムは C 言語により記述し、UNIX プラットフォーム上で動作させた。トランスポート層のプロトコルは TCP を使用し、一次サーバ、二次サーバ機能を行うプログラムとして “n3d”、ディレクトリサーバ機能を行うプログラムとして “n3dsd” を、それぞれデーモンとして常駐させる。実験には図 12 のようにディレクトリサーバ (D) と、一次サーバ (O) を兼ねたサーバ 1 台と、二次サーバ (S) 1 台によって構成される実験ネットワークを構築した。 O/D のサーバは Sun Ultra10 で OS は Solaris2.6 を、 S のサーバは Sun Enterprise450 で OS は Solaris2.6 を使用した。

O/D のサーバに、既存のニュースサーバから配送された記事を注入する。 S には本来の二次サーバ機能のほか、クライアントプロセスを動作させる機能を持た

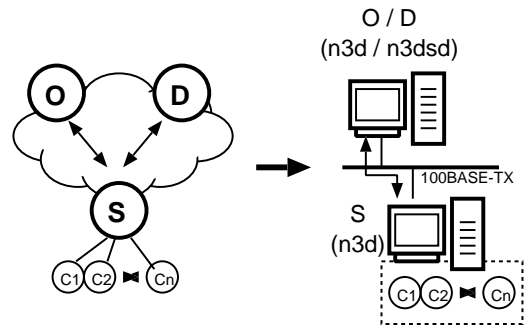


図 12 想定モデルと実験装置
Fig. 12 Experimental environment.

表 1 実験データの内容
Table 1 Overview of the experimental data.

name	value
対象日数	6
ニュースグループ数	21,503
総記事数	611,737
総配送量	30,149 MB
平均利用率	8.2% (4.2%)

せ、運用中のニュースサーバのアクセスログを入力としてクライアントのアクセスをシミュレートする機能を可能にした。実験ネットワーク上を流れるトラフィックの測定には tcpdump⁹⁾ を用いた。

実際の運用データを解析するのではなく、既存のニュースサーバからそのログを基に実験システムに記事を注入する方法で実験を行った理由は、ネットワークやニュースサーバの利用状態が刻々と変化する運用システムでは、配送量などのトラフィックを同じ条件で比較するのは困難であると判断したためである。

本実験において使用した配送データおよびアクセスログは、実験ネットワーク TRIX¹⁰⁾ に接続する商用 ISP のニュースサーバのものをを用いた。なお、使用したデータ項目は、ニュースグループ名、アクセスコマンド、転送量、記事ファイル名、メッセージ ID だけであり、データの匿名性に注意を払い、アクセス元アドレスなどアクセス者が特定できる項目はいっさい使用していない。

まず、2000 年 4 月 11 日～16 日までの 6 日間の配送記事を実験システムに注入した。実験は 1 日ごとのデータを O/D に注入し、 S からアクセスを行いネットワークを流れるトラフィックを tcpdump によって計測するという手順を繰り返した。

実験に用いたデータの内容を表 1 に示す。平均利用率とは、日ごとの総記事数 Tr と記事の総参照回数 Ar との割合 (Ar/Tr) の平均値である。 Ar は、同

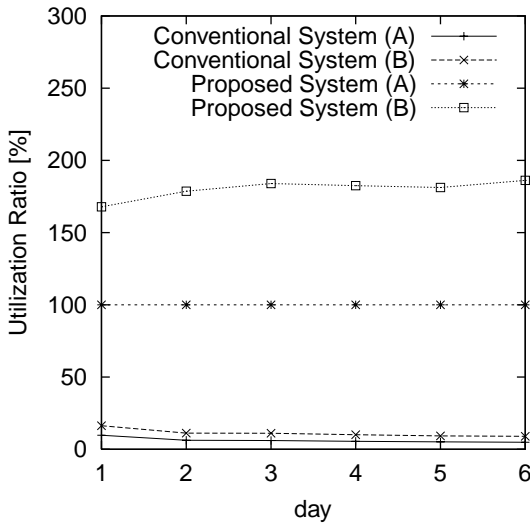


図 13 記事の利用率

Fig. 13 Utilization ratio of articles.

一記事に対しての複数回の参照もそのまま参照回数として計上している。また、括弧内に示した値は、同一記事に複数回の参照があった場合にもその参照回数を1回と計上し、 T_r に対する割合を求めた値(1回でも参照されたファイルの割合)である。

4.2 結果と考察

実験による記事の利用率の推移を図 13 に、累積配送量の推移を図 14 に示す。利用率は、日ごと ($d = 1..6$) の累積記事数 $\sum_{i=1}^d Tc(i)$ と、累積参照回数 $\sum_{i=1}^d Ar(i)$ の割合 ($\sum_{i=1}^d Ar(i) / \sum_{i=1}^d Tc(i)$) である。このとき、同一記事に複数回の参照があった場合には、参照回数を1回と計上し、それを利用率 A とする。一方、同一記事に対しての複数回の参照をそのまま参照回数として計上し、それを利用率 B とする。

図 13 において Conventional System (A), Conventional System (B) は、それぞれ従来方式における利用率 A と利用率 B を示し、Proposed System (A), Proposed System (B) は、提案方式における利用率 A と利用率 B を示す。また図 14 において Conventional System は従来方式による配送量を示し、Proposed System は提案方式による配送量を示す。

利用率について見ると、利用率 A では、配送記事数は参照回数(参照記事数)に等しいため、提案方式によれば 100% の利用率が得られる。そして、利用率 B についても約 180% 前後と、より高い利用率が得られている。これに対して、従来方式では利用率 A, 利用率 B とともに 10% 前後と低く、提案方式によって記事の利用効率を向上できることが分かる。

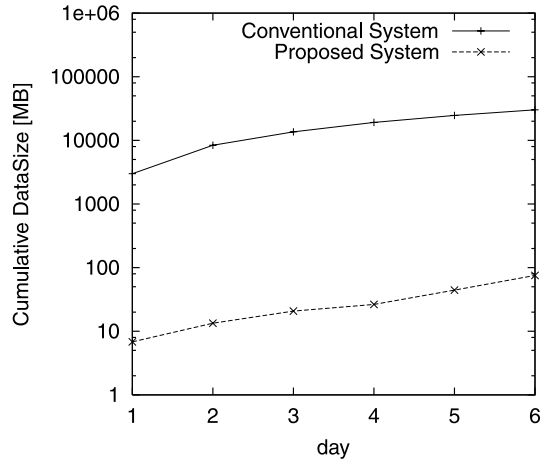


図 14 累積配送量

Fig. 14 Cumulative datasize of articles.

また配送量については、従来方式による配送量と比較して提案方式による配送量が大きく下回っている。たとえば、6日目の時点で従来方式が約 30GB であったのに対して、提案方式によれば約 80MB である。すなわち、提案手法によってサーバ間の不要なトラフィックを抑制できることが示された。これは同時に分散サーバのコンテンツの蓄積領域も削減されることになり、運用コストの削減につながる。

5. 他の方式との比較

ネットニュースシステムにおける配送効率を改善する方式としては、これまで、自サーバにおけるニュースグループの利用統計から配送を受けるニュースグループを決定し、記事の流量を抑える方式の研究¹¹⁾(方式 1)や、末端のニュースサーバをキャッシュサーバとし、クライアントからの要求によって記事の配送を行うとともに、一度取得した記事の再利用を行うことで効率化を図る方式の研究^{4),12)}(方式 2), 方式 1 と方式 2 を組み合わせたハイブリッド方式の研究^{5),13)}(方式 3)などが行われている。方式 2 については、たとえば Delegate¹⁴⁾ や DNAS¹⁵⁾, NNTPCache¹⁶⁾, DNEWS¹⁷⁾ といったいくつかの実装がある。

ここで、他サーバへの中継配送機能を持つサーバをバックボーンノードサーバ、中継配送機能を持たないサーバをリーフノードサーバと呼ぶことにすると、上であげたような各方式はいずれもバックボーンノードサーバとリーフノードサーバ間の配送効率を改善することを目的としており、データ流量の多いバックボーンノードサーバ間の配送については考慮されていない。

これに対して、本稿で提案するデマンド型配送方式

によれば、オリジナルの記事を広域に分散された一次サーバが管理し、二次サーバが必要に応じた記事の取得とキャッシュによる再利用を行うので、ネットニュースシステムの配送系全体についての配送効率の改善に有効である。

各方式について記事の利用率の観点から比較すると、方式1と方式3では実際に利用されない記事も配送される可能性があるのに対して、方式2は実際に利用される記事のみが確実に配送されるという特性から、方式1や方式3と同等もしくはそれ以上の高い利用率が得られる。

方式2は、本稿で提案するデマンド型配送方式と同様にクライアントからの要求によって記事の配送を行うものである。そこでこの方式2を用いて、ある特定のニュースサーバで記事を管理し、クライアント側からはそのサーバに直接アクセスして記事を取得するシステムの構築が考えられる。しかし、特定のサーバに記事データを集約し管理することは、実際に利用されるかどうか分からない記事データも多数蓄積されることとなり、効率的とはいえない。

6. おわりに

本稿では、現行のネットニュースシステムの配送方式に起因する問題を明らかにし、その問題を解決するために、ネットワークの広帯域化をふまえ、より高効率な配送が可能なデマンド型配送方式を提案した。提案するデマンド型配送方式については、サーバの機能と構成、配送プロトコルの拡張などの設計を行った。そして、新たに設計されたデマンド型ニュース配送システムについて、実装を行い、実験による評価を行った。実験によれば、記事の利用効率が向上し、サーバ間の不要な配送トラヒックが抑制されることが確認でき、提案方式の有効性が示された。

当システムの実際の運用を考えた場合、一次サーバや二次サーバと比べてディレクトリサーバへのアクセス集中が特に懸念される。本稿では、ディレクトリサーバをニュースグループごとに分散し、そのアクセス負荷を低減する手法について示したが、ディレクトリサーバに対して新たな記事が届いているかどうかの確認(GROUPコマンドによる)のための通信と処理によるアクセス負荷の問題があり、これを効果的に抑制する手法が求められる。筆者らはこの問題への対処について、記事インデックスの更新時間の統計的予測に基づくアクセス回数の低減手法を提案している¹⁸⁾。また、マルチキャストによって事前にディレクトリサーバ上のインデックスデータを二次サーバ側へ配送する

手法についても検討を進めている。

今後の課題としては、これらの改善手法を含めて、本提案方式をより大規模な環境に適用し、スケーラビリティに関しての評価を行うことがあげられる。さらに当方式は、ネットニュースシステムに限らずミラーリングシステムやメールマガシンなどといった他システムへの適用も可能であると考えている。今後、これらへの応用についても検討を進めたい。

参考文献

- 1) NEC: Usenet statistics for newsfeed.mesh.ad.jp. <http://newsfeed.mesh.ad.jp/>.
- 2) BEST INTERNET COMMUNICATIONS: news1.best.com usenet newsfeed statistics page. <http://news1.best.com/>.
- 3) Wirehub! Internet: Wirehub! Internet Newsfeed Statistics Page. <http://doema.wirehub.nl/news/>.
- 4) 知念賢一, 山口 英, 山本平一: NetNewsにおける利用状況を反映する記事配送モデルの提案, 情報処理学会研究会報告, 95-DPS-71, Vol.95, No.71, pp.49-54 (1995).
- 5) 舟阪淳一, 最所圭三, 福田 晃: NetNewsのためのキャッシングアルゴリズム, 電子情報通信学会論文誌, Vol.J82-B, No.5, pp.818-826 (1999).
- 6) Horton, M. and Adams, R.: Standard for Interchange of USENET Messages, IETF RFC1036 (1987).
- 7) Spencer, H. and Lawrence, D. (著), 紀太 章 (監訳), 田中 幸 (訳): Usenet ネットニュース管理, オライリージャパン (1999).
- 8) Kantor, B. and Lapsley, P.: Network News Transfer Protocol, IETF RFC977 (1986).
- 9) The Tcpdump Group: TCPDUMP public repository. <http://www.tcpdump.org/>.
- 10) 東北地域内インターネット相互接続研究会: TRIX Home Page. <http://www.tia.ad.jp/trix/>.
- 11) 渡辺健次, 小串英俊, 近藤弘樹: 読まれているニュースグループのみ自動的に購読するニュースシステムの構築, 情報処理学会研究会報告, 96-DSM-2, Vol.96, No.64, pp.19-24 (1996).
- 12) Gschwind, T. and Hauswirth, M.: A Cache Architecture for Modernizing the Usenet Infrastructure, *Proc. 32nd Hawaii International Conference on System Sciences—1999* (1999).
- 13) 舟阪淳一, 山本和彦, 福田 晃: ニュースグループごとに配送方式を自動変更する NetNews システムの評価, 電子情報通信学会技術研究報告, CPSY97-33, pp.49-54 (1997).
- 14) Sato, Y.: DeleGate Home Page. <http://www.delegate.org/>.
- 15) 梶田将司, 今井祐二: 複数 NNTP サーバへの

統合的アクセスの実現, *Proc. JAIN Symposium 92*, pp.39-47 (1992).

- 16) Assange, J. and Bowker, L.: NNTPCache NNTP usenet news cache software home page. <http://www.nntpccache.org/>.
- 17) NetWin: DNEWS USENET News Server Software. <http://www.netwinsite.com/dnews.htm>.
- 18) 菅野浩徳, 曾根秀昭, 根元義章: デマンド型配送システムにおけるディレトリサーバ分散手法, 情報処理学会研究会報告, 2000-DSM-19, Vol.2000, No.92, pp.13-18 (2000).

(平成 13 年 5 月 8 日受付)

(平成 13 年 10 月 16 日採録)



菅野 浩徳 (正会員)

昭和 59 年茨城大学工学部卒業。同年(株)富士通東北システムエンジニアリング入社。平成 9 年より宮城教育大学教育学部非常勤講師。平成 13 年東北大学大学院情報科学研究科博士前期課程修了。分散システム, 情報ネットワーク等の運用管理技術に関する研究開発に従事。情報処理教育, 地域情報化等にも興味を持つ。



曾根 秀昭

昭和 53 年東北大学工学部電気工学科卒業。昭和 55 年同大学大学院工学研究科修了。同年同大学工学部助手, 平成 4 年同大学電気通信研究所助教授。同大学情報科学研究科, 大型計算機センターおよび総合情報システム運用センター勤務を経て, 平成 13 年より同大学情報シナジーセンター教授および副センター長。工学博士。計算機システムとネットワークシステム等の運用に関する研究と, 電子応用計測, 耐ノイズ性通信方式等の研究に従事。電子情報通信学会, IEEE, 電気学会等の会員。



根元 義章 (正会員)

昭和 43 年東北大学工学部通信工学科卒業。昭和 48 年同大学大学院博士課程修了。工学博士。同年同大学助手。昭和 59 年同大学電気通信研究所助教授。平成 3 年同大学大型計算機センター教授。平成 7 年同大学大学院情報科学研究科教授。平成 10 年より同大学大型計算機センター長, 平成 13 年より同大学情報シナジーセンター長併任。マイクロ波伝送路回路, 情報ネットワーク, 情報伝送システム, 画像処理, 文字認識等の研究に従事。昭和 56 年 IEEE・MTT・Micro Wave Prize 受賞。IEEE, 情報処理学会各会員。