社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－MBL－36（13）
2006－UBI－10（13）
2006／2／16

Scribeにおける効率的なマルチキャスト木を構築するpushdown手法の提案

八重倉智 †，松尾啓志 †，
† 名古屋工業大学大学院情報工学専攻

分散ハッシュを用いた Application-level Multicast(ALM) の手法の 1 つに Scribe がある．Scribe での従来の pushdown では，木の高さは低く出来るが，ノードだけを評価しているためにネットワーク資源を浪費してしまう問題点がある．そこで本研究では，ノードの評価だけではなくリンクも評価することでネットワーク資源の消費を少なくする Hybrid pushdown を提案する．4 種類の評価方法を用いてシミュレーションを行なった結果，従来の pushdown と同じ程度に木の高さを抑えつつネットワーク資源の消費を少なくすることができた．

Pushdown method to build an effective multicast tree in Scribe

Satoshi Yaekura †　Hiroshi Matsuo †
† Department of Computer Science and Engineering,
Graduate School of Engineering, Nagoya Institute of Technology

Scribe is one of the Application-level Multicast(ALM) using Distributed Hash Table(DHT). Conventional pushdown can build a low tree. But, conventional pushdown has problems to waste network resources on because it evaluates only a node. Therefore, by this paper, we suggest Hybrid pushdown which reduces consumption of network resources by evaluating a link as well as a node. We did simulation with four kinds of valuation methods. As a result, we were able to build the tree which reduced consumption of network resources while keeping height the same as conventional pushdown.

1　はじめに

近年，インターネットの広帯域化や計算機の高性能化に伴い，マルチメディアコンテンツのストリーミング等，インターネットを用いて複数のクライアントへ動画配信を行なうネットワークアプリケーションの需要が高まっている．一般に，これらのアプリケーションでは動画像のような大量のデータを扱うため，多くのネットワーク資源を消費する．従って，これらのアプリケーションでは，ネットワーク資源の消費を少なくすることが重要となる．マルチキャスト通信では，各ホストへの経路上の分岐点においてパケットを複製することで，ネットワーク資源の消費を少なくすることが可能である．また，マルチキャスト通信では全クライアントがサーバへ接続する必要がないため，サーバの負荷も下がる．

現在，マルチキャスト通信は大きく分けて IP マルチキャストと Application Level Multicast(以下 ALM とする) の 2 つがある．IP マルチキャストは，マルチキャスト通信に必要なパケットの複製を IP マルチキャストに対応したルータが行なう．それに対し，ALM ではパケットの複製をマルチキャストに参加したノードが行なう．IP マルチキャストはパケットの複製をルータが行なうため，ネットワーク資源の利用効率が高いと言う利点があり，ALM はパケットの複製をノードが行なうため対応ルータが不要となり，現状のインフラでの実現が容易という利点がある．これまでに，ALM に関する多くの研究がなされてきているが，本研究では分散ハッシュを用いた ALM 手法の 1 つである Scribe を基礎とする．

現在の Scribe では，ノードの性能がヘテロな環境において，様々な問題点があることが指摘されている [10]．この中でも特に，帯域の制限によって発生する pushdown は，構築される木の性能を大きく左右する．そこで本研究では，ヘテロな環境での Scribe において，従来の pushdown よりもネットワーク資源の消費が少なく，効率の良いマルチキャスト木を構築する新しい pushdown を提案する．

2　関連研究

既存の ALM として，様々な手法が提案されている [1][2] [3][4] [5][6][7][8] ．

End System Multicast[1] は，マルチキャストメンバーの情報を集めてメッシュ型のオーバーレイネットワークを構築し，そのオーバレイネットワーク上でマルチキャスト木を構築する，小規模向けのマルチキャストである．Scattercast[2] は，SCX と呼ばれるノードをネットワーク上に配置し，この SCX

間で End System Multicast を構築する．受信ノードは自身の近くにある SCX に接続することでコンテンツを受信し，大規模なマルチキャストを実現する．ALMI[3] は，各マルチキャストメンバーが，自分の情報及び他のメンバーとのユニキャストリンクの情報 (遅延時間等) を session controller と呼ばれるノードに送信し，全てのメンバーの情報を受信した session controller は，その情報から最も効率的な木を中央集権的に構築する．Overcast[4] は，ソースノードから各受信ノードまでの帯域が最大となるマルチキャスト木を構築する．このうち，End System Multicast と Scattercast はメッシュ状の制御トポロジを先に構築する Mesh-first 型のマルチキャスト手法であり，ALMI と Overcast はメッシュ型のオーバレイネットワークを構築せずにマルチキャスト木を構築する，tree-first 型のマルチキャスト手法である．

Internet Indirection Infrastructure(i3)[5] は，分散ハッシュである Chord を用いてマルチキャストを行なう．i3 では，ID と受信者アドレスのペア (ID,R) を trigger と呼び，マルチキャスト受信ノードはセッションに対応した ID を持つノードに trigger を送信し，送信ノードはセッションに対応した ID を持つノードにマルチキャストパケットを送信する．その結果，セッションに対応した ID を持つノードが中継ノードとなり，マルチキャストが行なわれる．Application-level Multicast using Content-Addressable Networks[6] は，分散ハッシュである CAN を用いてマルチキャストを行なう．この手法では，CAN 上でマルチキャストに参加するノードをメンバーとした小さい CAN を構築し，その小さい CAN 内でフラッディングを行なうことでマルチキャストを実現する．Scribe[8] は，分散ハッシュである Pastry[9] を用いてマルチキャストを行なう．Scribe では，Pastry の検索経路を逆にたどるようにマルチキャスト木を構築する．詳細は次節で述べる．SplitStream[7] は，Scribe でのマルチキャスト木を複数用いることで，ノードの転送負荷を平均化する．

分散ハッシュを用いた手法では，制御トポロジを構築する必要がないが，マルチキャスト木の構築がノード ID によって決まり，任意のマルチキャスト木が構築できないという制限がある．

3 Pastry/Scribe

本研究で対象とした Scribe[8] は，分散ハッシュの 1 つである Pastry[9] を用いた ALM である．Pastry 上の各ノードは，2^b(b は Pastry のパラメータを表す) を基数とした文字列で表される一意な 128bit の

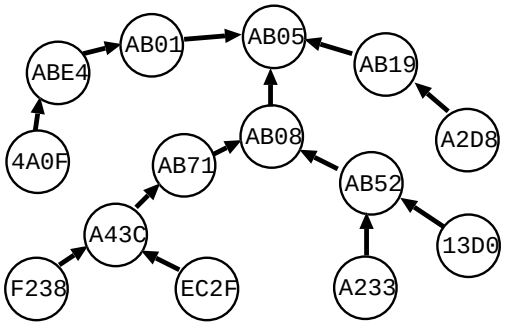

Fig. 1　Pastry 上のノードが，ID=AB05 のノードを検索する際の経路例

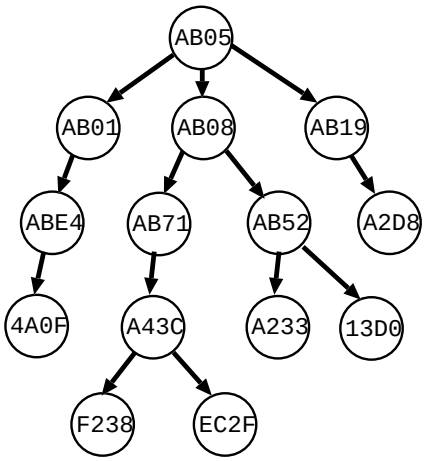

Fig. 2　Scribe 上での multicast key=AB05 のマルチキャスト木の例

ノード ID を持つ．Pastry 上の各ノードは，ID 空間的に遠いノードは大まかな情報，近いノードは詳しい情報を持つルーティングテーブルを維持する．Pastry での検索はこのルーティングテーブルを用いて行なう．ある ID のノードを検索する場合，自分のルーティングテーブルの中から検索する ID と最もプレフィックスが一致する ID を持つノードに検索を依頼する．そして，検索依頼を受けたノードは同様に，自分のルーティングテーブルの中から検索する ID と最もプレフィックスが一致する ID を持つノードに検索を依頼する．以下この操作を繰り返すことによって，最終的に目的のノードに到達する．

図 1 に示すように，Pastry では目的の ID に少しずつ ID が近づいていく形式で検索経路が選ばれる．Scribe では，Pastry の検索経路を利用し，検索経路の次の中継ノードに join 要求を出すことでマルチキャスト木を構築する (図 2)．

一般的にノードの上り帯域には制限があるため，各ノードがマルチキャスト木上で保持することの可能な子ノード数には限界がある．保持することの可能な最大の子ノード数を Degree と呼ぶ．Scribe では，マルチキャスト木構築時に Degree 以上の join

要求を受けた場合，pushdown と呼ばれる操作を行なうことにより，子ノード数を Degree 以下に抑える仕組みを有する．以下に pushdown の動作を示す．(1)join 要求を受けた親ノードは，自分の子ノードと join 要求をしてきたノードの中から，特定のポリシーに従って 1 つのノードを pushdown ノードとして選択する．(2) 親ノードは選択した pushdown ノード以外のノードの中からもう 1 つのノードをリダイレクト先ノードとして選択する．(3) 親ノードは pushdown ノードにリダイレクト先ノードの情報を送信し，その情報を受信した pushdown ノードはリダイレクト先ノードに再 join する．

この pushdown 操作によって，親ノードが選択した pushdown ノードは親ノードから見て孫の位置に移動し，その結果，子ノード数は Degree 以下に抑えることが可能となる．

Scribe での pushdown 手法として Bharambe ら [10] によって，以下の 2 種類の pushdown が提案され，Preempt Degree pushdown が，ノードの性能がヘテロな環境でも高さの低いマルチキャスト木が構築可能と報告されている．

- Preempt ID pushdown
  マルチキャスト木の ID である，mutlticast key と一致するプレフィックスが多い ID を持つノードが優先される
- Preempt Degree pushdown
  Degree が多いノードが優先される

しかし，Preempt Degree pushdown では Degree のみしか考慮していないため，親ノードからどんなに離れているノードであっても，Degree が大きければ pushdown 時に優先されることとなる．その結果，親子ノード間の距離が大きいマルチキャスト木が構築され，ネットワーク資源の消費が大きくなる問題点がある．

4 提案手法

本研究では，pushdown 時にノードとリンクの評価値を両方とも用いる Hybrid pushdown 及び，Hybrid pushdown におけるパラメータを動的に変更する Dynamic Hybrid pushdown を提案する．

4.1 Hybrid pushdown

Hybrid pushdown では，ノードとリンクの両方を評価するために以下の評価式を用いて pushdown を行なう．

$v = \alpha \times Degree + \beta \times \frac{1}{Hop}$ (α,β：パラメータ)

Hybrid pushdown 時の親ノードはこの評価式を用いて pushdown ノードとリダイレクト先ノードを選択する．具体的には以下のように選択する．

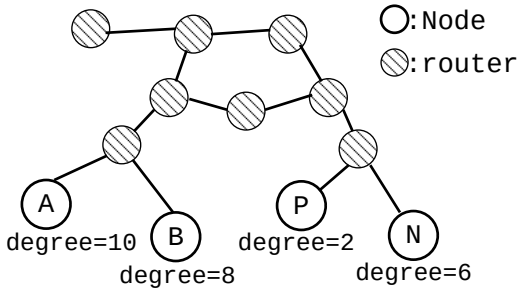

Fig. 3　物理ネットワーク例

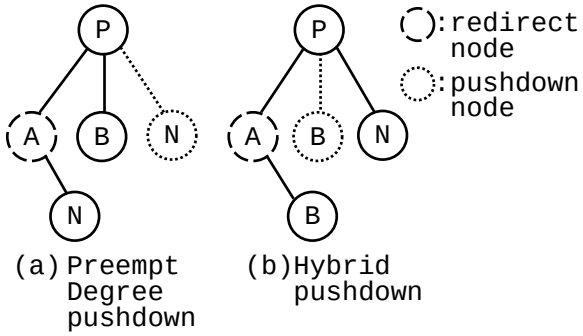

Fig. 4　Preempt Degree pushdown と Hybrid pushdown の比較

pushdown **ノード**：自分の子ノードと join 要求をしてきたノードの中から評価値が最も低いノードを選択する．

リダイレクト先ノード：残りのノードの中から評価値で重み付けした確率に従ってランダムに選択する．

このような評価式を用いて pushdown ノードを選択することにより，親ノードはノードの評価である Degree とリンクの評価である Hop の両方を評価することができ，ネットワーク資源の消費を少なくすることが期待できる．また，評価値で重み付けした確率に従ってランダムにリダイレクト先を選択することで，評価値の高いノードほど多くの子孫ノードを持つこととなり，Degree を重視した評価値を用いた場合は木の高さが低くなり，Hop を重視した評価値を用いた場合はネットワーク効率が良いマルチキャスト木の構築が可能となる．

以下，具体例により説明する．図 3 に示す物理ネットワークにおいて，ノード P がノード A，B を子ノードとしたマルチキャスト木を仮定する．そこに，ノード N が join してきた場合，Preempt Degree pushdown では図 4(a) のようにマルチキャスト木が構築される．この木において物理ネットワーク的に考えた場合，ノード P-A-N の接続が一度遠いノードにパケットを送信してから戻ってくる接続であり，ネットワーク利用効率の悪いマルチキャスト木となる．それに対し，Hybrid pushdown(パラメータは $\alpha = 0.1$,$\beta = 1.0$ と仮定する) では，図 4(b) のように

マルチキャスト木が構築される．このマルチキャスト木では，図 4(a) のノード P-A-N のように，遠いノードにパケットを送信してから戻ってくるような接続は無いため，ネットワーク利用効率が良く，ネットワーク資源の消費が少なくなる．

4.2 Dynamic Hybrid pushdown

Hybrid pushdown 時に子ノードの平均 Degree が低い場合には，Degree を優先したパラメータにすることで低い木を構築することが可能である．逆に，子ノードの平均 Degree がある程度以上に高い場合には，どの子ノードであっても Degree が十分にあるために，どの子ノードを pushdown ノードとして選択しても木の高さは低くなる．そこで，子ノードの平均 Degree が高い場合には，Hop を重視したパラメータにすることで，木の高さを低くしたままネットワーク資源の消費を減少させることが可能である．

Hybrid pushdown におけるパラメータを子ノードの平均 Degree を用いて動的に変化させる Dynamic Hybrid pushdown を提案する．Dynamic Hybrid pushdown では，閾値を用いて，子ノードの平均 Degree が閾値より小さい場合には Degree を重視したパラメータとし，子ノードの平均 Degree が閾値より大きい場合には Hop を重視したパラメータとする．

5 シミュレーションによる性能評価

提案手法の有効性を計算機シミュレーションによって評価する．評価方法としては以下の 4 種類を用いた．

- Median Depth
  マルチキャスト木の高さの中央値
- Link stress
  マルチキャスト木が使用した任意の物理リンクにおいて，同一のデータがそのリンク上を通過した回数
- Resource Usage
  マルチキャスト木上の全受信ノードに対してデータを送信した際に消費されるネットワーク資源の総量で，以下の式で表される

  Resource Usage= $\sum_{i=1}^{L} D_i \times S_i$
  L:マルチキャスト木が使用した総物理リンク数
  D_i:リンク i の遅延
  S_i:リンク i の Link stress
- Relative Delay Penalty(RDP)
  任意のノードにおいて，ソースノードからマルチキャスト木上のリンクを経由した場合の遅延と，ソースノードからユニキャストを使用した場合の遅延の比

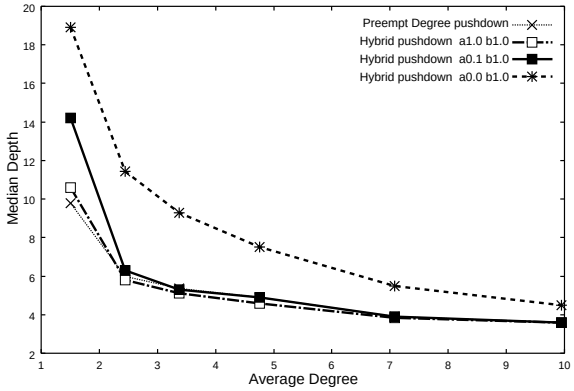

Fig. 5 Median Depth による Hybrid pushdown と Preempt Degree pushdown の比較

ネットワークシミュレータとして ns-2[11]，ネットワークモデルとして GT-ITM[12] を用いた．ノード数は Scribe に参加するノードが 512 台，ルータが 64 台の計 576 台．Hybrid pushdown でのパラメータは，Degree を重視したパラメータとして $\alpha = 1.0,\beta = 1.0$，中間のパラメータとして $\alpha = 0.1,\beta = 1.0$，Hop を重視したパラメータとして $\alpha = 0.0,\beta = 1.0$ の 3 種類でシミュレーションを行なった．Dynamic Hybrid pushdown で用いる閾値は経験的に 7 とし，平均 Degree が 7 より小さい場合には $\alpha = 1.0,\beta = 1.0$ の Degree を重視したパラメータ．平均 Degree が 7 以上の場合には $\alpha = 0.0,\beta = 1.0$ の Hop を重視したパラメータとした．全てのシミュレーションにおいて，Scribe(Pastry) の ID は 128bit とし，Pastry のパラメータは B=4，M=32，L=32 とした．全ての結果は 10 回のシミュレーションの平均値である．

5.1 Hybrid pushdown の評価と考察

図 5 に Median Depth の結果を示す．Hybrid pushdown で，Hop を重視したパラメータの場合は木の高さが高くなるが，中間のパラメータ及び，Degree を重視したパラメータの場合には Preempt Degree pushdown とほぼ同じ程度に低い高さの木を構築することが可能であった．これは，評価式によって Hop だけではなく Degree も評価したため，評価値に従ってランダムにリダイレクト先ノードを選択する場合に，Degree の大きいノードほど多くの子孫を持つ確率が多くなったためである．

図 6 に Average Link stress の結果を示す．Hop を重視するパラメータの Hybrid pushdown が最も低い Link stress となり，Preempt Degree pushdown が最も高い Link stress となった．Preempt Degree pushdown では，Hop を評価していないために親子ノード間の距離が遠くなることが多く，その結果，Link stress が高い，つまりネットワークの利用効率

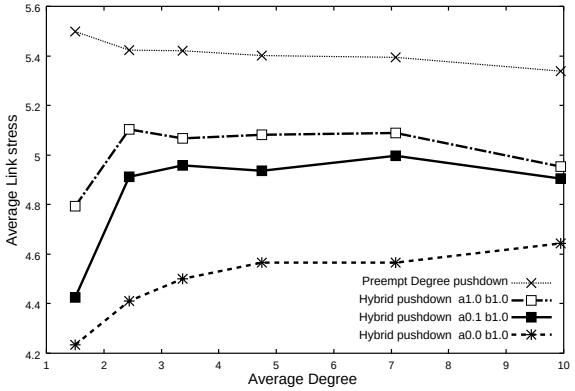

Fig. 6 Average Link stress による Hybrid pushdown と Preempt Degree pushdown の比較

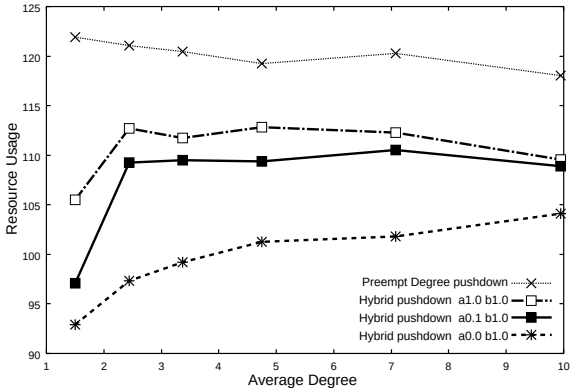

Fig. 7 Resource Usage による Hybrid pushdown と Preempt Degree pushdown の比較

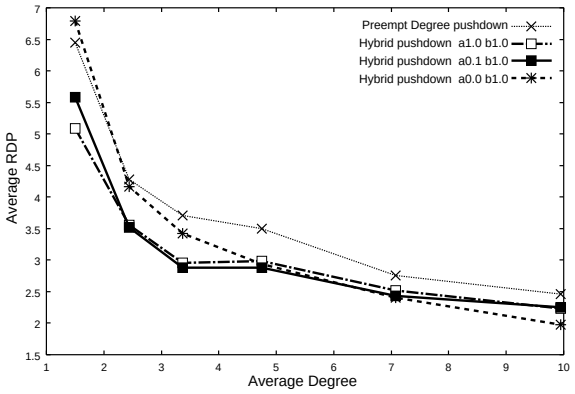

Fig. 8 Average RDP による Hybrid pushdown と Preempt Degree pushdown の比較

が悪くなった．

図 7 に Resource Usage の結果を示す．図 6 と同様に，Hop を重視するパラメータの Hybrid pushdown が最も低い Resource Usage となり，Preempt Degree pushdown が最も高い Resource Usage となった．Preempt Degree pushdown では，Hop を評価していないために親子ノード間の距離が遠くなり，その結果 Resource Usage が高い，つまりネットワーク資源の消費が大きくなった．

図 8 に Average RDP の結果を示す．RDP を下げ

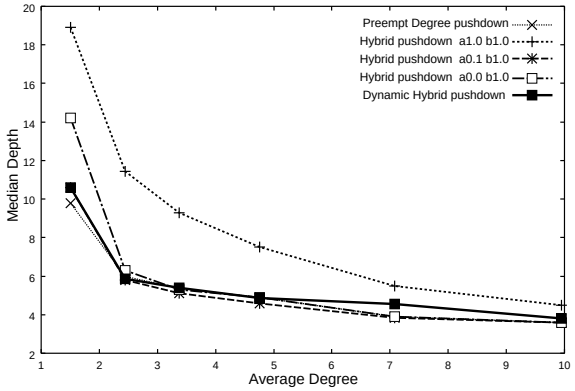

Fig. 9 Median Depth による Dynamic Hybrid pushdown と Preempt Degree pushdown の比較

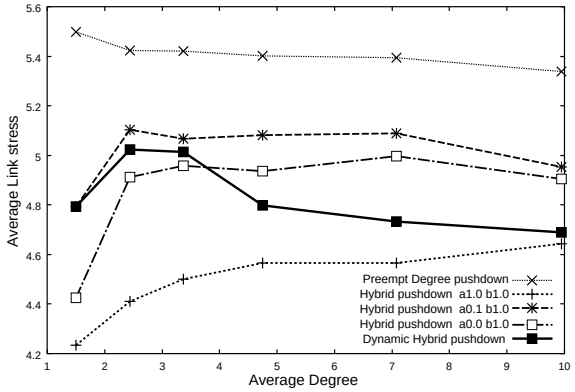

Fig. 10 Average Link stress による Dynamic Hybrid pushdown と Preempt Degree pushdown の比較

るためには木の高さと親子ノード間の距離の両方を小さくする必要がある．Preempt Degree pushdown は，図 5，6 から，木の高さは低いが親子ノード間の距離は遠いとことが分かる．その結果，RDP の値が高くなった．それに対し Hybrid pushdown では，Degree と Hop の両方を評価したことによって，木の高さと親子ノード間の距離の両方ともに小さくなり，RDP の値が低くなった．

5.2 Dynamic Hybrid pushdown の評価と考察

図 9 に Median Depth の結果を示す．Dynamic Hybrid pushdown によってパラメータを動的に変更しても，Preempt Degree pushdown と同じ程度に低い高さの木を構築できた．これは，パラメータの動的な変更によって，平均 Degree が小さい場合には Degree を重視した評価によって木の高さが低くなり，平均 Degree が大きい場合には，Degree が大きいために Hop を重視した評価をしても木の高さはあまり高くならなかったためである．

図 10 に Average Link stress の結果を示す．Dynamic Hybrid pushdown によってパラメータを動的に変更したため，平均 Degree が大きくなるほど，

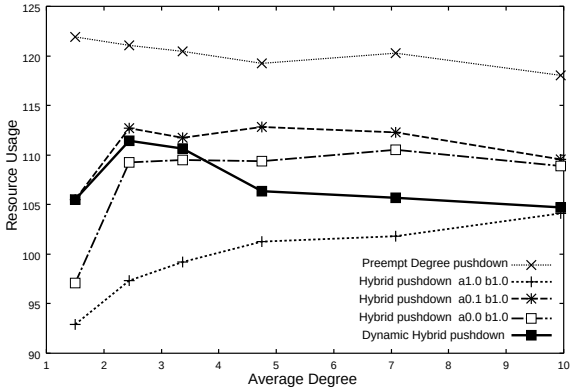

Fig. 11 Resource Usage による Dynamic Hybrid pushdown と Preempt Degree pushdown の比較

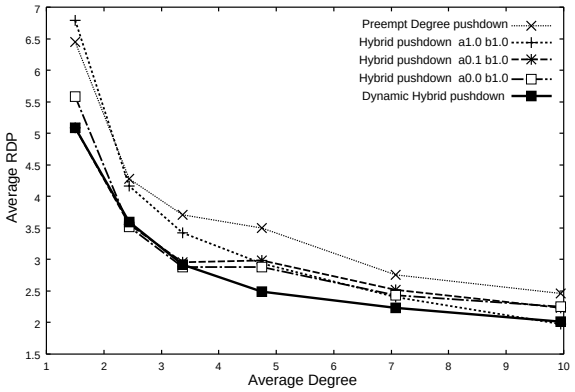

Fig. 12 Average RDP による Dynamic Hybrid pushdown と Preempt Degree pushdown の比較

Link stress が低くなり，ネットワークの利用効率が良くなった．

図 11 に Resource Usage の結果を示す．図 10 と同様に，Dynamic Hybrid pushdown によってパラメータを動的に変更したため，平均 Degree が大きくなるにつれて，Resource Usage が下がり，ネットワーク資源の消費が少なくなった．

図 12 に Average RDP の結果を示す．Dynamic Hybrid pushdown によってパラメータの動的な変更を行なうことで，全ての pushdown の中で最も低い RDP となった．これは，パラメータの変更によって，木の高さを低く抑えたまま，親子ノード間の距離を小さくすることができたためである．

6 まとめ

Scribe における従来の pushdown である Preempt Degree pushdown には，ノードのみを評価しており，リンクを評価していないという問題点があった．本研究では，pushdown 時にノードの評価に加え，リンクの評価も行なうことで，ネットワーク資源の消費を少なくすることができると考え，評価値を用いてノードとリンクの両方を評価する Hybrid pushdown を提案した．Hybrid pushdown では，Preempt Degree pushdwon と同じ程度の木の高さに抑えながらネットワーク資源の消費を少なくする事ができた．また，Hybrid pushdown におけるパラメータを周囲の状況から動的に変更させる Dynamic Hybrid pushdown を提案した．この Dynamic Hybrid pushdown によって，さらにネットワーク資源の消費を少なくする事が可能となった．

参考文献

[1] Y. Chu, S. G. Rao, S. Seshan, and H. Zhang, *A Case for End System Multicast*, Proceedings of ACM Sigmetrics, June, 2000.

[2] Y. Chawathe, *Scattercast: An Adaptable Broadcast Distribution Framework*, Multimedia Systems, Vol.9:104-118, July, 2003.

[3] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, *ALMI: An Application Level Multicast Infrastructure*, Proceedings of the 3rd Usenix Symposium on Internet Technologies & Systems (USITS 2001), March, 2001.

[4] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr. *Overcast: Reliable Multicasting with an Overlay Network*, Proceedings of the Fourth Symposium on Oerating System Design and Implementation, Oct, 2000.

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, *Internet Indirection Infrastructure*, Proceedings of ACM SIGCOMM, August, 2002.

[6] S. Ratnasamy, M. Handley, S. Shenker, and R. Karp, *Application-level Multicast using Content-Addressable Networks*, International Workshop on Networked Group Communication, 2001.

[7] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, *SplitStream: High-Bandwidth Multicast in Cooperative Environments*, Proceedings of SOSP, 2003.

[8] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, *SCRIBE: A large-scale and decentralized application-level multicast infrastructure*, IEEE Journal on Selected Areas in Communications Vol.20 No.8, Oct, 2002.

[9] A. Rowstron1, and P. Druschel, *Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), 2001.

[10] A. R. Bharambe, S. G. Rao, V. N. Padmanabhan, S. Seshan, and H. Zhang, *The Impact of Heterogeneous Bandwidth Constraints on DHT-Based Multicast Protocols*, Proceedings of 4th International Workshop on P2P Systems (IPTPS) 2005, Feb, 2005.

[11] *ns-2*, http://www.isi.edu/nsnam/ns/

[12] *GT-ITM:Modeling Topology of Large Internetworks*, http://www.cc.gatech.edu/projects/gtitm/