送信者起動マルチキャストにおける輻輳制御方法の提案

Proposal for Congestion Control Method on Sender Initiated Multicast

村本 衛一, 米田 孝弘, 鈴木 史章, 鈴木 良宏, 中村 敦司
Eiichi Muramoto, Takahiro Yoneda, Fumiaki Suzuki, Yoshihiro Suzuki, Atsushi Nakamura
松下電器産業株式会社
Matsushita Electric Industrial Co., Ltd.

概　要

送信者起動マルチキャストに対して、受信者を送信速度毎に異なるクラスに分類し、TFRC[4]に準じた輻輳制御を行う方式を提案する。この輻輳制御方式を用いれば、TCP のフローとの帯域の公平性を満たしながら、致命的な振動状態に陥らずに収束する輻輳回避が実現される。また、同一ストリームの受信者間の公平性に関しても、従来の受信者起動マルチキャストにおいて提案された方式よりも優れた特性を持つ方式となっている。

1. はじめに

我々は、インターネットでのマルチキャストを用いたグループ通信の実現をめざし、様々な種類の情報を複数地点で共有できる方式の研究開発を行なっている。

インターネットでは、情報を配送する経路上のすべてのリンクにおいて、競合する他の TCP フローと公平性を保ちながら、輻輳を回避することが望まれる[1]。また、同一の情報配送のセッションにおいては、それぞれの受信者に、許容される最大限の送信速度で情報が配送されることが望ましい [2]。

本稿では、送信者起動のマルチキャスト(Sender Initiated Multicast)[1]を対象とし、これらの条件を満たしながら輻輳制御を実現する方式を提案する。

2. マルチキャストの輻輳制御への要求

複数の受信者とマルチキャストを用いてセッションを確立する通信を行う場合には、そのセッションの輻輳制御方式の特性や他の通信への影響が課題となる。ここでは、インターネットにおいてそのような通信を実現する場合に、輻輳制御方式に対して求められる項目を列挙する。

2.1 セッション間の公平性

インターネットに特定のフローを流す場合、競合する他のフローと公平に帯域を共有する必要がある[1]。この公平性は、他のセッションのフローとの公平性を指すことから、本稿では“セッション間の公平性”と呼ぶ。

現在のインターネット上のトラフィックの大半は TCP であることから、セッション間の公平性を満たすためには、TCP のフローとの帯域の公平性について考えれば充分である[1][4][6]。

マルチキャストでは、送信者から受信者に至る論理的な配送木を構築し、その配送木にしたがってパケットを配送する。セッションが存在する期間中、その配送木のリンクで発生する輻輳を検知し、輻輳の発生しているすべてのリンクにおいて、リンクを共有する TCP のフローとのセッション間の公平性を保ちながら、輻輳を回避する必要がある。

2.2 輻輳回避アルゴリズムの振動回避

同時に動作する輻輳回避のアルゴリズムが相

[1]ここでは、少数の受信者にデータを配送する第三層の技術 [3] を指す。第四層のトランスポートプロトコルでは、ACK ベースのプロトコルを送信者起動と表現することがあるが、これとは異なるものである。

互に影響することで、互いの制御動作が致命的な振動状態に陥ることがある。この振動は、同じアルゴリズム自身との相互作用による振動と他のアルゴリズムとの相互作用による振動に分けられる。我々の提案する輻輳回避のアルゴリズムに関しては、いずれの振動に対しても収束することを保証する必要がある。

2.3 セッション内の公平性

マルチキャスト特有の公平性に関する問題として、“セッション内の公平性(Intra session fairness)”に関する問題があげられる[2][7]。この問題は、ある受信者にとっては、送信者と受信者の間の帯域が空いているにも関わらず、他の遅い受信者の速度に全体の送信速度が制限されるため、空いている帯域を有効に利用できないという問題である。

我々の提案する輻輳制御方式は、このセッション内の公平性に関しても考慮し、様々な帯域を利用できる受信者が共存できるものとする必要がある。

3. 輻輳制御方式 SICC

本章では、送信者起動のマルチキャストに対して、これまでに説明した要件を満たす輻輳制御方式 SICC(Sender Initiated Congestion Control)を提案する。

SICC の設計にあたっては、次のような項目を考慮した。

- 動画や、画面、音声などのストリーム情報を対象とする
- 複数のクラスを設け、受信者をクラスに分類し、クラス毎にデータ配送を行う
- 受信者毎に、TFRC に準拠したクラス評価を行う
- 送信者起動マルチキャストを用いることでクラスの更新動作を高速に実現する
- 送信者起動マルチキャストが対象とする受信者の数を上限とし、簡素な設計とする

3.1 クラスの構成

SICC では、送信速度の上限 B (bps)を設定し、それに対して 1/2 ずつ減速した複数のクラスを用意してデータを配送する[1]。受信者は、このクラスのいずれかへと分類される。図１に 3 つのクラスでの配送の例を示す。

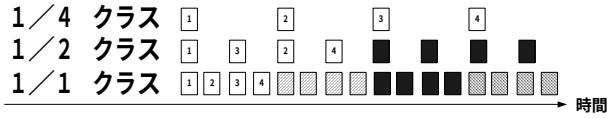

図 1 クラス毎のパケット配送

ひとつのクラス C_i において、情報を送信する速度 Bi (bps)は、送信者が扱うクラスの数を n とすると、以下のようになる。

$$B_i = \frac{B}{2^i}$$

ここで、i は $0 \le i \le n-1$ の整数

また、送信開始時にのみ使用される特別なクラスとして C_s を用意する。これについては、3.7 で説明する。

3.2 各クラスにおけるデータ配送

SICC の対象とするデータは、動画転送やコンピュータ画面の転送のように、アプリケーションのデータ単位（ADU: Application Data Unit）で、コマ落しが可能な情報である。SICC は、このようなデータを時間制約付の高信頼配送 (semi-reliable and time bounded delivery)[16]で伝送する方式である。

それぞれのクラスでは、送信速度と ADU のサイズから、ADU の配送に要する時間の上限が制約として定められる。受信者は、この時間の範囲内で再送が可能である場合、NAK を発行する。この時間の範囲で再送が不可能と判断した場合、それまでに受信した ADU を破棄する。

[1] TCP が AIMD(Additive Increase Multiplicative Decrease)の挙動を示すことから、送信速度の抑制を実現するため 1/2 ずつ減速するクラスを採用した。

3.3 TFRC によるクラス評価

インターネット上で、他の TCP フローとの帯域の公平性を保ちながら UDP ユニキャストで通信を行なうための送信レート制御方式として、TFRC[4]がある。

SICC では、受信者が報告するラウンドトリップタイム(RTT)と損失率に基づき、TCP フローとの帯域の公平性を保ちながら送信の可能な最大送信速度 X (bps)を推定し[5]、この Xを用いて受信者のクラス分けを行う。X の推定には、TFRC で定義される TCP reno の平均スループット推定式(式１)を用いる。

$$X = \frac{8s}{R\sqrt{2bp/3} + T\_RTO((3\sqrt{3bp/8}) * p * (1 + 32p^2)} \quad \cdots \text{式}1$$

式１の変数の意味を以下に示す。

R	ラウンドトリップタイム(秒)
p	損失率
s	パケットサイズ(byte)
b	TCP の肯定応答単位(1)
T_RTO	TCP のタイムアウト値(4* R)

SICC は、この Xを用いて次に示すクラス評価式（式 2）を満たすクラス Ci に受信者を割り当てる。

$$B_i > X \geq B_{i+1} \quad \cdots \text{式}2$$

以上に述べたクラス分けにより、あるクラスに属する受信者に対しては、TFRC で許容される送信レート以内で配送が行なわれることが保証される。

3.4 動的な送信レート制御

受信者との間の通信状況は、時々刻々と変化する。それに伴い X も変化するため、受信者のクラス分けも動的に行う必要がある。以下に、受信者と送信者のそれぞれについて、どのように動作するのかを説明する。

3.4.1 受信者の動作

SICC では、送信されるすべてのパケットに送信時刻のタイムスタンプと通番が付加される。受信者は、受信したパケットに含まれる通番から、パケットの損失を検知する。損失を検知した受信者は、直ちに NAK パケットを送信者に対して送出する。この際、最後に受信したパケットに含まれている送信時刻のタイムスタンプとパケットの損失率を、この NAK パケットに埋め込み、送信者に報告する。埋め込まれるタイムスタンプは、送信者が RTT を算出する際に用いられる情報である。

また、送信者は定期的に報告要求を発行する。この報告要求を受信した場合にも、上記と同じ情報を報告する。

なお、損失率は TFRC に定義されている方法を用いて算出する。すなわち、RTT の時間内に含まれる、最近に起こった最大 8 つのパケット損失から、重み付け平均で損失率を算出する[1]。

3.4.2 送信者の動作

送信者は、受信者からの NAK に含まれる情報に基づき、受信者が属するクラスを再評価する。

NAK パケットが到着した時には、式１に基づいて、その受信者との間で許容される最大送信速度 X を算出する。受信者から報告要求に対する応答のみが連続して返答された時には、これを契機に、8s/RTT (bit/秒: s は byte 単位でのパケット長)だけ X を増加させる。このように算出した Xを式２に適用し、クラスを再評価する。

クラスの再評価の結果、現在よりも高速なクラスとなった場合には、この受信者を一つだけ高速なクラスに変更する。低速なクラスとなった場合には、直ちに式２で算出されたクラスへと変更する。X が最下位クラス C_{n-1} のクラスの送信速度も満たさなくなった場合には、その受信者に対するデータの配送は停止する。

SICC では、受信者が所属するクラスが変更された場合、受信者をそのクラスの配送リストから、別のクラスの配送リストへと移す。この動作により、実質的な配送木の更新が即座に実現される。

3.5 振動回避

クラスを更新する動作においては、Xが閾値の

[1] 受信者は RTT を算出できないため、3.7 で後述する GRTT を用いる。GRTT は送信者から配布されるものとする。

近辺で上下することにより、頻繁にクラスが変更される可能性がある。この頻繁な更新が、他の TCP や SICC のフローにおける制御動作と極端な振動状態に陥るべきではない[9]。このような状態を回避するために、受信者の高速なクラスへの変更をランダムに遅らせるものとする。

3.6 再送処理

送信者が NAK を受信した場合、前節で述べたクラス変更の判断を行なった後、その受信者が同じクラスに属しつづける場合にのみ、該当する NAK パケットを送信した受信者が属するクラスに対して再送パケットを送信する。

式 1 により輻輳状態を検知した場合、その受信者は低速なクラスに変更され、再送は行なわれない。このため、輻輳リンクへのオーバーシュートは発生しない。

3.7 スロースタートと初期クラス

SICC では、TFRC と同等のスロースタートを行なう。このスロースタートの初期状態においては、すべての受信者は、特別なクラス Cs に属している。受信者との RTT の最大値として推定される値を GRTT(秒)とすると、このクラスへの送信レートの初期値としては、$8s$/GRTT(bit/秒)を用いる。GRTT の時刻が経過するごとに、送信レートを $8s$/GRTT(bit/秒)ずつ増加させる。この過程において受信者から NAK パケットを受信した場合には、そのパケットに埋め込まれた情報にもとづき、式１で許容される最大送信速度を推定する。この値がスロースタートにおけるその時点の送信レートよりも小さい場合、式２にもとづいて受信者をクラスに割り当てる。

スロースタートは、送信レートが上限 B に達するか、あるいは C_s に属するすべての受信者のクラス分けが完了した時点で終了する。

GRTT は、受信者からの NAK パケットを受信した時点、および、報告要求に対する応答を受信した時点で、新たに算出された RTT にもとづき更新される。初期値としては、我々は文献[6]に示されている 500ms を採用する。

4. 他の研究との比較

マルチキャストにより輻輳制御を実現する研究は、これまで、受信者起動のマルチキャストを前提として行われてきた。それに対して我々は、送信者起動のマルチキャストを用いてこそ、インターネットにおいて普及し得る輻輳制御が実現できるものと考えている。本章では、これまでになされた研究と SICC を対比することで、送信者起動のマルチキャストに対する輻輳制御が有望な技術であることを示す。

4.1 階層マルチキャスト

文献[8]は、受信者起動のマルチキャストにおいて、複数のマルチキャストグループを用いた輻輳制御方法を提案するものである。

この方式では、輻輳リンクの下流の受信者が一斉に離脱することで、輻輳回避を実現する。しかしながら、受信者起動のマルチキャストにおいて用いられる IGMP[13]や MLD[14]は問い合わせベースのプロトコルであり、実際に配送経路が更新されるまでの間に、ルータ間での配送木の枝狩りという、比較的長い時間を要する処理が必要となる。この処理による遅延は輻輳回避を遅らせる可能性があるため[9]、TCP フローとの帯域の公平性を保つことは困難である。

一方で送信者起動のマルチキャストにおいては、送信者がすべての受信者を把握しているため、受信者のクラス分けは、送信者のみが行えばよい。したがって、TCP フローにおける輻輳制御に対応可能な速度で受信者のクラス間の移動が実現される。

4.2 受信者起動の高信頼マルチキャスト

AER/NCA[10]は、受信者起動のマルチキャストによる NAK ベースの高信頼マルチキャストである。

AER/NCA は、通信状況の最も悪い受信者を代表者として選定し、その受信者にとって許容される最大送信速度でパケットを送信する。網の輻輳状態の変化に合わせて、代表者を選定することで、配送木のリンクで発生する輻輳を検知し、これを回避することができる。

しかし、この方式は、セッション全体の送信速度が最も遅い受信者に抑えられてしまうため、セッション内の公平性に問題がある。また、網

の輻輳状態によっては、代表者が転々と移り変わることにより、ひとつの TCP フローと比較しても、さらに悪い転送性能しか出せないという Drop to Zero 問題を内包することが指摘されている[9]。

一方、SICC では、セッションの期間の途中に、それぞれの通信状況に応じて受信者を即座にクラス分けし、クラス毎に独立した送信レートを適用できるため、セッション内の公平性を保ちながら、配送木のあらゆるリンクで発生する輻輳を回避できる。

4.3 サーバを用いた輻輳制御方式

文献[7]は、AER/NCA に改良を加えネットワーク上のサーバを用いることでセッション内の公平性に関する問題を改善する輻輳制御方式を提案するものである。この方式は、セッション内の公平性の問題を改善するため、ネットワーク内の配送木上のルータにサーバを配置し、このサーバが下流の受信者との間で独立した輻輳制御を行なうものである。

このようなサーバは、帯域の異なる通信を中継する必要があるが、そこでどのような処理を行えば良いかについては、通信の内容に依存することになる。このような上位層に依存する機能をネットワークの中継ノードであるルータに実装することは、汎用の目的で利用されるインターネットにおいては現実的ではない。

これに対して SICC では、クラス分けによりセッション内の公平性を解決し、各クラスに転送する内容は、始点ノードとなる送信者が決定することができる。転送する内容によって、間引きしたストリームを転送することもできれば、同じファイルをゆっくりと転送することも不可能ではない。また、送信者起動のマルチキャストに関しては、その実装の 1 つに XCAST6 がある。XCAST6 は、マルチキャストに対応していないルータのネットワークにおいても利用することが可能であることから、今すぐにでもインターネットで利用できる技術と言える。

4.4 アプリケーション層マルチキャスト

受信者起動のマルチキャストは、現在のインターネットにおいて、広域で使える状態にあるとは言い難い。その結果、複数の受信者に対して、同じ内容を個別に転送する実装を送信者のアプリケーションとして実現することが一般的に行われている。

このような実装において、個々の受信者との通信に TFRC を用いれば、セッション間の公平性もセッション内の公平性も解決することが可能である。しかしながら、このような実装においては、すべての受信者に対して個別にユニキャストを用いて送信するため、ルータにおけるパケットの複製を利用することができない。その結果として、ネットワークにおいて大きな帯域を占拠することとなる。また、送信者において通信を複製する必要があるため、送信者の負荷も大きなものとなる。

これに対して SICC は、受信者の存在するクラスの数だけ送出すれば済むため、ネットワークに占める帯域も、送信者の負荷も比較的小さなものとなる。

5. まとめ

本稿では、送信者起動のマルチキャストにおいて、受信者を送信レートによりクラス分けし、それぞれのクラスに対して TFRC に準じた輻輳制御を実現することで、セッション間の公平性とセッション内の公平性の双方に優れた輻輳制御方式 SICC を提案した。

現在のところ、SICC は設計途上にあり、実際に利用できる環境にはない。また、受信者のクラス更新時の振動回避の挙動において、クラスによる送信レートの粒度が、TCP フローとの共存にどのような影響を与えるかに関しては、より詳細な解析が必要であると考えている。また、本稿では取り上げていないが、実際のインターネットへの普及を考えれば、RED[11]対応・非対応のルータが存在する網での挙動解析や、SICC の損失率計測法に対する ECN[12]の対応とその挙動解析も必要であると考えている。

今後、SICC を設計・実装し、これらの評価・検証を進めて行きたい。

謝辞

本研究の推進にあたっては、北陸先端科学技術大学院大学の篠田陽一教授、WIDE プロジェ

クト XCAST-WG のメンバにアドバイスをいただいた。また、本研究の一部は、TAO（通信・放送機構）のギガビットワーク利活用制度 JGN-P341019 の一環として実施されたものである。

参考文献

[1] S. Floyd,” Congestion Control Principles”,RFC 2914,September,2000

[2] S.Sarkar and L. Tassiulas,"Distributed algorithm for computation of fair rates in multirate multicast trees", Proc. IEEE INFOCOM 2000,pp52-61,March 2000.

[3] Small Group Multicast homepage, http://www.alcatel.com/xcast/

[4] M. Handley, S.Floyd, J.Padhye, J.Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification, RFC3448, IETF, January, 2003

[5] Padhye, J., Firoiu, V., Towsley, D. and J. Kurose, "Modeling TCP, Throughput: A Simple Model and its Empirical Validation", Proc. ACM SIGCOMM 1998.

[6] Joerg Widmer, Mark Handley,” TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification”, draft-ietf-rmt-bb-tfmcc-02.txt,IETF,July,2003

[7] 山口　誠、山本　幹、信頼性マルチキャストにおける intra-session fairness を考慮したふくそう制御方式、電子情報通信学会、論文誌 B、JN:S06222C　ISSN 1344-1756、J85?B NO. 10; PAGE. 1748-1756;

[8] L. Vicisano,L.Rizzo,J.Crowcroft," TCP-like Congestion Control for Layered Multicast Data Transfer",Proceedings of IEEE INFOCOMM,April,1998

[9] Brian Whetten, Jim Conlan, A Rate Based Congestion Control Scheme for Reliable Mulitcast", Technical White Pager, Global Cast Communications,Oct. 1998.

[10] Sneha K. Kasera, Supratik Bhattacharyya, Mark Keaton, Diane Kiwior, Jim Kurose, Don Towsley, Steve Zabele, Scalable Fair Reliable Multicast Using Active Services, IEEE/ACM Transactions on Networking

[11] S.Floyd, and V.Jacobson, Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413

[12] K. Ramakrishnan, S. Floyd, D. Black,The Addition of Explicit Congestion Notification (ECN) to IP,RFC3168,IETF, September 2001

[13] W. Fenner, Internet Group Management Protocol, Version 2, RFC2326, IETF, November 1997

[14] S. Deering, W. Fenner, B. Haberman, Multicast Listener Discovery (MLD) for IPv6, RFC2710, IETF, October 1999

[15] VNC homepage, http://www.realvnc.com/

[16] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, M. Luby, The Reliable Multicast Design Space for Bulk Data Transfer RFC2887 ,IETF, August 2000