

優先順位を利用した輻輳制御に関する考察

趙 晩熙† 木村 成伴‡ 海老原 義彦‡

†筑波大学大学院 工学研究科 ‡筑波大学 電子・情報工学系

概要

ATM で代表される高速ネットワークはその伝送速度が速くなるにつれ、伝搬遅延の占める割合が相対的に大きくなりつつある。したがって、送信されたパケットに対しての ACK を受信側から受け取ってフローを制御するような既存のウィンドウフロー制御は、高速ネットワークのフロー制御としては適切でない。そこで、ネットワーク内部に流れ込むデータの入力を規制することでフロー制御を行うリーキーバケット法がその主流となっている。また、最近のネットワークは違う属性を持つ様々なデータに対してもサービスを提供することをその目標としている。本論文では、それぞれの要求品質を満たしながら輻輳を制御する方法に関して考察を行い、優先順位を持つフロー制御を提案する。

A Study of Congestion Control Scheme Considering Priority

Manhee Jo, Shigetomo Kimura and Yoshihiko Ebihara
University of Tsukuba

Abstract

Widely used window based flow control mechanisms generally rely on end-to-end exchange of control messages such as ACKs in order to regulate traffic flow. But, in high speed networks, as propagation delays across networks typically dominate, such mechanisms are not suitable. An algorithm named leaky bucket scheme controls the traffic flow by regulating data input rate. Meanwhile, it is becoming important to support applications with diverse performance objectives. In this paper, we propose a new scheme to match these objectives and to control the congestion of network.

1 序論

データネットワークにおいて、ユーザーの要求が多種多様になり、ハードウェア技術の目覚ましい発展にともなってその規模や環境等が新たな局面を向かえている。B-ISDN は、そのような色々なユーザーの要求に対し、異なるサービスを統合して提供できるようにした通信規格であり、ATM 等の高速ネットワーク技術をその基礎としている。

高速ネットワークはその伝送速度が速くなるにつれ、通信にともなう総遅延の内、伝搬遅延 (propagation delay) の占める割合が相対的に大きくなる。既存のネットワークでのフロー制御¹によく用いられたウィンドウフロー制御 [ML80, F.H92, DR92] は、送信側から送られたパケットに対しての

¹[ITU93]では、トラフィック制御と輻輳制御を区別して記述しているが、本論文ではこれらの語彙をフロー制御と同じ意味として取り扱う。

ACKを受信側から受け取ることでフロー制御を行うので、伝搬遅延の割合が大きい高速ネットワークではあまり望ましくない方法である。このため、既存の方式とは違った形のフロー制御方法が必要となってきた。

また、これまでは属性の異なったデータに対して同一の扱いを行っていたので、リアルタイムデータなど、特定の要求目的を持つデータの処理に充分に対応仕切れなかった。最近のネットワークは違う属性を持つ様々なデータに対してもサービスを提供することをその目標としている。このため、データへの依存性(data dependency)、又は、サービスの品質(Quality of Service)を考慮したフロー制御が求められるようになった。本論文では、既存のフロー制御について比較、検討し、データの優先順位をフロー制御に取り入れた新しいアルゴリズムを提案する。

2章では研究の背景や基本となった様々な技術について述べ、3章では、優先順位付きアルゴリズムを提案する。4章で今後解決しなければならない問題点について述べる。

2 研究の背景

高速ネットワークのフロー制御はその方法が簡単であればあるほど良いと言われている[GP88]。簡単で実現しやすいということで、スライディングウィンドウフロー制御(Sliding Window Protocol; 以下、ウィンドウ制御と呼ぶ)方法がネットワークのフロー制御によく用いられる[ML80][F.H92][DR92]。この方法は、エラーの制御までも簡単に行えるため広く使われている技術である。しかし、相手からの受信確認(ACK; acknowledge)を利用してフローの制御を行うので、ネットワークの総遅延中で伝搬遅延の割合が相対的に大きい高速ネットワークではあまり好ましくない技術である。これに対して、ネットワークの入口でパケットの入力を規制(input rate regulation)するリーキーバケット(leaky bucket)という方法が注目をあびている[J.S86]。

以下では、この論文の背景になったいくつかのリーキーバケットスキームの方法について述べる。

2.1 リーキーバケットスキーム

輻輳を制御する方法の1つとしてリーキーバケットスキーム(Leaky Bucket Scheme)というアルゴリズムがある[J.S86]。これは1986年にTurnerが考案した方法で、その概要を図1にあらわす。

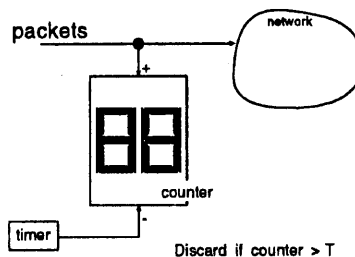


図 1: オリジナルリーキーバケット。

各ユーザーは、呼に対してカウンタとタイマーを持ち、パケットを送る度にカウンタの値を増加させる。増加した値があらかじめ決められた閾値(T)を超える場合パケットは廃棄される。カウンタの値はユーザーから与えられた時間間隔(ρ)をもとに定期的に減少する。Tと ρ はユーザーから受け取るパラメータである。

ネットワークの性能を大きく低下させる要因としてはデータのバースト性が挙げられる。ネットワークの規模が大きくなればなるほど、このバースト性データ(bursty data)がネットワークの性能に与える影響は大きくなる。このため、ネットワークの最適制御ができなくなる。このリーキーバケッ

トスキームは、データ伝送のバースト性を緩和させるために提案された方法である。この方法では呼の最大パケット伝送率は ρ に近くなる。これを単純化すると一般的に図 2 のようにあらわすことができる。 ρ はパケットがネットワークに送られる伝送率である。

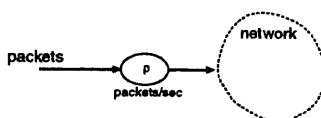


図 2: 単純リーキーバケット。

最初に考案されたリーキーバケットスキームはこのように入力バッファを持たせていない。ここに入力バッファを持たせ、いくつかの改良を行った例を次に述べる。

2.2 トークンバケットスキーム

リーキーバケットの変形として、トークンバケットスキーム (Token Bucket Scheme) というアルゴリズムがある [MI93]。それぞれの呼はパケットを送る前にトークンバケット (または、トークンプール) を調べる。トークンには予め割当量 (quota) が決まっており、トークンの割当量が 500 キロビットの時、1.5 メガビットのパケットに対しては 3 つのトークンが、2 メガビットのパケットに対しては 4 つのトークンが必要になる。トークンプールに十分なトークンがある時、パケットの大きさにあたる量のトークンをトークンプールから取り出し、ネットワークにパケットを伝送する。トークンプールのトークンが充分でない時はそのパケットは廃棄される。トークンプールへのトークンの生成は決められた時間間隔で行われる。その生成率を ρ とした時、パケットの最大伝送率は当然 ρ に近づくので、呼のパケット伝送率をトークンの生成率で制御することができる。この方法をもとにし、改良を加えたアルゴリズムがトークンバケットスキームであり、その基本動作の概要を図 3 に示す。

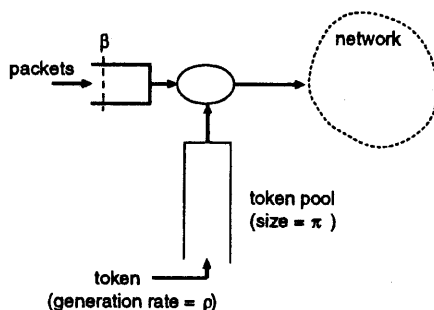


図 3: トークンバケット。

ここでは、捨てられるパケットの公平性を考慮して入力バッファを設け、そのバッファに閾値 (β) をおくことでネットワーク内部に入るパケット量を規制する。パケットがノードに到着すると、まず、入力バッファを確かめる。バッファ内のパケットの数が閾値を越えているとそのパケットは廃棄される。そうでない場合はバッファで順番を待つことになり、パケットが入力バッファの先頭にくると今度はトークンプールを調べる。十分なトークンがトークンプールに入っているとそのパケットはネットワークに流れるが、そうでない場合にはバッファの先頭で十分なトークンがたまのを待つ。

2.3 リーキーバケットとトークンバケットの混合

2.2節のトークンバケットスキームについて追加考察する。パケットが長い間到着しないとトークンがブールにたまり始める。トークンの数が π になった時に大きなパケットが到着するとそのノードは一瞬の内にその大きなパケットをネットワークに排出することになってしまう。これがトークンバケットスキームの欠点で、違った形でバースト性を持つ伝送を許すことになるので良くない。

上で述べた二つの方法を合わせたものが混合型 [C.P94] であり、その概要を図 4 に示す。

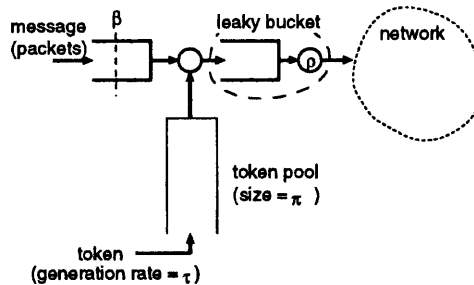


図 4: リーキーバケットとトークンバケットの混合。

これによりどのような場合でもパケットの伝送率をある一定水準に維持することができるようになる。

2.4 リーキーバケットスキームの改良型

ここではリーキーバケットスキームに更に改良を加え高速ネットワークに対応できるようにしたフロー制御について述べる。

ATM などの高速ネットワークでは、呼の設立時に一定の帯域をネットワークに要求する。この時ユーザーが要求した帯域が、ネットワークのサービス可能な範囲内にある時のみ、その呼は設立される。呼が確立した後、場合によっては約束した帯域を越える範囲のパケットを送ることもあり、BWM (Bandwidth Management) アルゴリズム [AD89b], [AD89a] ではこの協定を違反した部分に対してフラグを立てる。ネットワークの内部に送り込まれたパケットは中間ノードに転送されるが、この時そのノードの待ち行列の長さが予め決められた閾値を越えると、フラグが立っているパケットは廃棄される。

この BWM を改良した、一般化リーキーバケット方式 (Generalized Leaky Bucket Scheme) [KK90] について述べる。この方式は、グリーンとレッドの 2 種類のトークンを持つ。グリーントークンはネットワークの内部状況が輻輳状態でない時にもらうトークンで、レッドトークンは輻輳状態の時にもらうトークンである。ネットワークの内部状況を入力バッファの長さで推測し、それによって違う色のトークンを与えることで輻輳制御を行っている。

このスキームは以下のように動作する。まず、入力バッファの先頭に到着したパケットは、パケットの大きさにあたる量のグリーントークンがグリーントークンプールにあるかどうかを調べる。あった場合はグリーントークンをもって送信が始まるが、グリーントークンが足りない時は、その時の入力バッファの長さをチェックする。バッファの長さが予め決められたある値より小さい時、ネットワークはそれほど混んでいないと判断し、グリーントークンが充分たまるまで待つことができる。しかし、バッファの長さがその値を越えていると、たくさんのパケットがネットワークへの伝送を待っていると判断できるので、その場合は、レッドのトークンプールを調べる。レッドトークンが充分である時、レッドトークンをもらって送信を始めるが、トークンが足りない時はバッファの先頭に帰りもう一度

始めからやり直す。

レッドトークンで送信されたパケットは、ネットワーク内部の中間ノードを通る時、閾値を越えていたら廃棄されるので、グリーントークンで送られたパケットより低い優先順位で処理されることになる。

3 新しい方法の提案

ネットワークに送られるデータの属性は大きく、遅延制約 (delay-sensitive) データと損失制約 (loss-sensitive) データの2つに分けて考えることができる。遅延制約データはリアルタイム性のデータで、時間の制約があるデータである。これに対して、損失制約データは損失に対して制約があるデータで、完全性 (integrity) が保証されなければならない。

これまでに述べた方法は、ネットワークの輻輳制御が簡単に行えるという長所を持つが、データに対しての属性の相異を考慮していないため、ユーザの要求に充分に対応しきれない場合が発生する。そこで、本研究ではデータの属性を考慮し、その属性に基づいた優先順位をそれぞれのパケットに与えることで、データ属性に依存したフロー制御を可能にするアルゴリズムを提案する。これを、優先順位トークンプール (Priority Token Pool) 方式と呼ぶ。

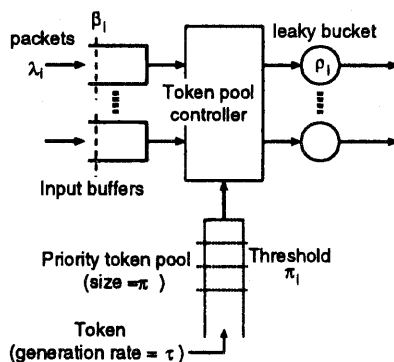


図 5: 優先順位トークンプール。

図 5 に、トークンプール制御部 (token pool controller) と、優先順位トークンプール (priority token pool) を設けた優先順位トークンプール制御法の概略を示す。データの属性、または要求するサービスの水準をパラメータとして、呼設立時に優先順位 p_i を定義し、その優先順位に合わせてトークンプールにそれぞれ違う閾値を持たせる。入力バッファにパケットが到着するとトークンプール制御部はパケットの優先順位とトークンの数を調べる。トークンの数がパケットの優先順位毎に設けた閾値以上であったらパケットを送る。閾値以下の場合、そのパケットが損失制約パケットならトークンが充分にたまるまで待たせるが、遅延制約のパケットなら廃棄する。トークンプール制御部によって充分なトークンを確認した呼は、リーキーバケットにパケットを送り、 ρ_i の伝送率でネットワークに伝送される。

このような方法で、データの属性を考慮したデータ依存のフロー制御が可能になり、リアルタイム処理などのサービスが統合して提供でき、B-ISDN などの応用に対応できると思われる。

4 まとめと今後の課題

これまでに考案されたいろいろな輻輳制御の方法について比較、検討し、リーキーバケットスキームを改良した優先順位トークンプール方式を提案した。現在、提案の方式をシミュレーションにより

評価中であるが、データの属性を考慮した優先順位を導入したことでよりインテリジェントな輻輳制御が行えるものと思われる。

遅延制約のデータの代表的なサービスであるリアルタイム処理は、ハードリアルタイム (hard real time) とソフトリアルタイム (soft real time) に分けることができる。ハードリアルタイムのデータは決められた時間が過ぎると、その時点でデータの価値が0になる制約をもち、ソフトリアルタイムな決められた時間が過ぎるとデータの価値が段階的に減少するという制約を持つ。優先順位にこの特徴を反映することが今後の課題である。もう一つの問題点として、このようにデータの特性を考えることでトークンプール制御部が複雑になるということである。どのようにして複雑なアルゴリズムを単純化するかも今後の研究課題として挙げられる。

参考文献

- [AD89a] D.T.Luan A.D.Eckberg and D.M.Lucantoni. "Bandwidth Management:A Congestion Control Strategy for Broadband Packet Networks – Characterizing the Throughput-Burstiness Filter". In *ITC Specialist Seminar*, pages 1769–1773, September 1989.
- [AD89b] D.T.Luan A.D.Eckberg, Jr. and D.M.Lucantoni. "Meeting the Challenge: Congestion and Flow Control Strategies for Broadband Information Transport". In *IEEE GLOBECOM '89*, pages 1769–1773, 1989.
- [C.P94] C.Partridge. "*Gigabit Networking*", pages 253–263. Addison-Wesley, 1994.
- [DR92] D.Bertsekas and R.Gallager. "*Data Networks*", pages 500–510. Prentice-Hall, 2nd edition, 1992.
- [F.H92] F.Halsall. "*Data Communications, Computer Networks and Open Systems*", pages 175–178. Addison-Wesley, 3rd edition, 1992.
- [GP88] R.G.H.Rogers G.M.Woodruff and P.S.Richards. "A Congestion Control Framework for High-speed Integrated Packetized Transport". In *IEEE GLOBECOM '88*, pages 0203–0207, November 1988.
- [ITU93] ITU-T. *ITU-T Recommendation I.371*, March 1993.
- [J.S86] J.S.Turner. "New Directions in Communications(or Which Way to the Information Age?)". *IEEE Communications Magazine*, 24(10):8–15, 1986.
- [KK90] I.Cidon K.Bala and K.Sohraby. "Congestion Control for High Speed Packet Switched Networks". In *IEEE INFOCOM '90*, pages 520–526, June 1990.
- [MI93] I.Cidon M.Sidi, W.-Z.Liu and I.Gopal. "Congestion Control Through Input Rate Regulation". *IEEE Transactions on Communications*, 41(3):471–477, 1993.
- [ML80] M.Gerla and L.Kleinrock. "Flow Control:A Comparative Survey". *IEEE Transactions on Communications*, COM-28(4):553–574, 1980.