

## 複数フローに対する拡散型フロー制御方式

会田 雅樹<sup>†</sup> 高野 知佐<sup>††</sup> 村中かほり<sup>††</sup>

<sup>†</sup> NTT 情報流通プラットフォーム研究所

〒180-8585 東京都武蔵野市緑町 3-9-11

<sup>††</sup> NTT アドバンステクノロジー トラヒックエンジニアリング事業部

〒180-0006 東京都武蔵野市中町 2-4-15 ハクエイビル 4F

E-mail: <sup>†</sup>aida.masaki@lab.ntt.co.jp, <sup>††</sup>{chisa,kaori}@mitaka.ntt-at.co.jp

あらまし 我々はこれまで、高速ネットワークに要求される厳しい制御遅延の要求を満たすためのフロー制御方式として、拡散型フロー制御技術を提案してきた。この技術は、個々のノードが自分の知りうる局所情報に基づいて、予め定められたルールにより高速で自律動作し、その動作がネットワーク全体として望ましい性能を実現するフロー制御技術である。これまでの研究では、単一のフローに着目した技術を扱ってきたが、本稿では複数フローに対する拡散型フロー制御技術を提案する。この技術により、輻輳の回復時間を短縮し、複数フロー間での公平性を実現することができる。

キーワード フロー制御, 拡散, 自律分散制御, 公平性

## Diffusion-Type Flow Control Scheme for Multiple Flows

Masaki AIDA<sup>†</sup>, Chisa TAKANO<sup>††</sup>, and Kaori MURANAKA<sup>††</sup>

<sup>†</sup> NTT Information Sharing Platform Laboratories,

NTT Corporation, Musashino-shi, Tokyo 180-8585 Japan

<sup>††</sup> Traffic Engineering Division, NTT Advanced Technology Corp.,

Hakuei Bldg. 4F, 2-4-15, Nakcho, Musashino-shi, Tokyo 180-0006 Japan

E-mail: <sup>†</sup>aida.masaki@lab.ntt.co.jp, <sup>††</sup>{chisa,kaori}@mitaka.ntt-at.co.jp

**Abstract** We have proposed the diffusion-type flow control mechanism as a solution of severely time-sensitive flow control required for high-speed networks. In this mechanism, each node in a network manages its local traffic flow only on the basis of the local information directly available to it, by using of predetermined rules. In addition, the implementation of decision-making at each node can lead to optimum performance for the whole network. Our previous studies concentrate on the flow control for a single flow. In this article, we propose the diffusion-type flow control mechanism for multiple flows. The proposed scheme enables us to quickly recover from congestion and to achieve fairness among flows.

**Key words** flow control, diffusion, autonomous decentralized control, fairness

### 1. はじめに

近年のインターネットの急速な普及・需要の拡大により、将来をにらんだ高速なネットワークの構築がすすめられている。ネットワークの高速化によりノードでのパケット転送処理速度が高速化すると、エンド-エンドの遅延はノードでの処理遅延よりもリンク伝搬遅延が支配的になる。このことにより、ネットワーク制御に関して以下の問題点が顕在化すると予想される。

- 超高速で動作する制御機構の必要性  
ネットワーク状態が短時間で大きく変化するため、非常に短い制御遅延で動作するネットワーク制御技術が必要になる。
- リアルタイムな情報把握の難しさ

リンク伝播遅延(光速度により決まる)により、ネットワークの広範囲な状態情報を短時間で収集することができない。

このことから、高速ネットワークに適用可能な新しい制御の枠組みを作っていく必要があると考えられる [1]。

フロー制御の課題に対する多くの研究は、数理計画問題として定式化されていて [2]~[6]、ネットワークの状態情報を収集することを前提にしている。しかし、高速ネットワークでは広範囲な状態情報をリアルタイムに収集することができない。また、数理計画問題の求解には十分な時間が必要なので、高速ネットワークに要求される非常に短い制御遅延を満足することが難しい。このように、ネットワーク全体の情報を一ヶ所に収集し、管理することを前提とした集中制御の枠組みは、超高速ネット

ワークでのフロー制御として適切でない．このため，自律分散制御に基づくフロー制御機構が必要とされる．

現在，自律分散型のフロー制御として，例えば TCP に代表されるエンドホストによる制御が広く用いられ，その特性について活発に研究されている [5] ~ [7]．エンドホストによるフロー制御では，制御遅延の時間スケールはラウンドトリップタイム (RTT) 程度となる．低速なネットワークでは，RTT 程度の制御遅延による影響を無視することが可能だが，ネットワークが高速化すると RTT 程度の制御遅延がネットワーク性能に大きな影響を与える可能性がある．これは，ネットワークの高速化に対して RTT 自身は不変であるが，ノードの動作速度の時間スケールを時間の基準として見た RTT は大きくなるからである．RTT よりも短い時間スケールで俊敏に動作する制御を実現するためには，ノードが局所情報に基づいて自律動作するタイプの制御機構が必要になる．

我々はこれまで，高速ネットワーク環境でも適切に動作するフロー制御方式として，拡散型フロー制御方式を提案し，その動作原理を述べると共に，ネットワーク性能の安定性，ネットワーク状態への適応性を評価してきた [1], [8]．この制御は，局所的な情報に基づいてノードが非常に高速で自律動作することにより，間接的にネットワーク全体の状態を望ましい方向に誘導するタイプの新しい制御機構である．この制御は，あるフローに沿った経路上のノード内パケット数を平滑化することで，輻輳ノードでのバッファ溢れによるパケット損失を予防する効果を狙っている．しかしながら，これまでの研究はある特定のフローのみに着目した制御であり，経路に沿った一次元的なパケット数の平滑化であるため，ネットワーク全体の「面」としてのパケット平滑化には結びついていなかった．

本稿では複数のフローが混在する状況を考え，その枠組みでノード内パケット数平滑化の再考察及び分類を行なう．これにより目的を明確にした上で，単一フロー内だけでなく複数フロー間でノード内パケット数が平滑化する拡散型フロー制御を提案する．更に，提案した制御方式の効果をシミュレーションにより確認する．

## 2. 単一フローに対するフロー制御方式の概要

単一フローに対する我々のフロー制御方式は，ある経路に沿った特定フローのノード内パケット数を平滑化する主要部分 (拡散型フロー制御方式) と，ネットワーク入口でのトラヒック流入量を適正化するシェーピング機能とからなる．シェーピング機能については，ネットワーク内のパケット数を適正化することが目的であるので，TCP のウィンドウ制御等の他の制御に置き換えることが可能である．

### 2.1 拡散型フロー制御方式

拡散型フロー制御はノードが局所情報に基づいて自律動作することで，ネットワーク全体としてノード内パケット数分布を均一化し，ネットワーク状態を秩序ある方向に導くことを目的としている．この制御の狙いと原理については [9], [10] で報告している．ここでは，拡散型フロー制御の具体的な動作について簡単にまとめる．

あるフローに沿って一次元的に連結したノード  $i$  ( $i = 1, 2, \dots, N$ ) を考える (図 1)．ノード  $i$  は，時刻  $t$  における下流ノード  $i+1$  へのパケット転送レート  $J_i(\alpha, t)$  を自分が知りうる情報のみに基づいて決定し，パケット転送を行なう．また，上流ノード  $i-1$  に向けてノード  $i$  の情報  $F_i(t)$  をフィードバックする．パケット転送レート  $J_i(\alpha, t)$  の決定は，

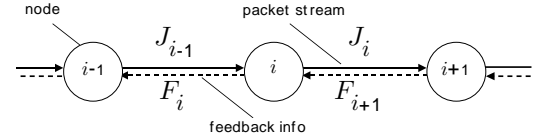


図 1 ノードの動作モデル．

下流ノード  $i+1$  から通知される情報  $F_{i+1}(t-d_i)$  の到着毎に実行する．ここで  $d_i$  はノード  $i$  と  $i+1$  間の伝搬遅延であり， $F_{i+1}(t-d_i)$  は現在時刻  $t$  より  $d_i$  だけ過去に下流ノード  $i+1$  で生成された情報であることを意味する．上流ノード  $i-1$  に向けたフィードバック情報  $F_i(t)$  の通知は，ノード  $i-1$  と  $i$  間の伝搬遅延  $d_{i-1}$  に比例した時間間隔で行なう．パケット転送レート  $J_i(\alpha, t)$  は

$$J_i(\alpha, t) = \max(0, \min(L_i(t), \tilde{J}_i(\alpha, t))) \quad (1)$$

$$\tilde{J}_i(\alpha, t) = \alpha r_i(t-d_i) - D_i(n_{i+1}(t-d_i) - n_i(t)) \quad (2)$$

と決定する．ここで， $n_i(t)$  は時刻  $t$  でノード  $i$  に存在するパケット数， $r_i(t-d_i)$ ， $n_{i+1}(t-d_i)$  は下流ノード  $i+1$  からのフィードバックにより (リンク伝播遅延  $d_i$  を伴って) 通知される情報で，それぞれ下流からの指示レート，下流ノード内のパケット数を表し， $L_i(t)$  はノード  $i$  から  $i+1$  に向かうリンクの利用可能帯域である．また  $D_i$  は

$$D_i = \frac{D}{d_i} \propto (d_i)^{-1} \quad (3)$$

のように決める．ここで  $\alpha (\geq 1)$  と  $D (> 0)$  は定数で，各々フロー強度係数，拡散係数と呼ぶ．一方，ノード  $i$  が生成するフィードバック情報  $F_i(t)$  は

$$F_i(t) = (r_{i-1}(t), n_i(t)) \quad (4)$$

からなり，これを上流ノード  $i-1$  に通知する．ここで，

$$r_{i-1}(t) = J_i(1, t) \quad (5)$$

とする．

上記に加えてネットワークの境界部分についての規則として以下が必要である [11]．

ネットワーク外のノード又はエンドホストは拡散型フロー制御方式をサポートしていないことを前提とし，ネットワーク入り口での入力パケットレートは，フィードバック情報  $r_0(t-d_0)$  で指定されたレートにシェーピングされるものとする．これは，ネットワーク外のノード  $i=0$  では式 (2) によるレート  $\tilde{J}_0(\alpha, t)$  の計算ができないためである．ノード  $i=1$  が通知する情報  $r_0(t)$  は，ネットワーク外のノードまたはエンドホストのバッファに存在するパケット数を  $n_0 = 0$  と仮定し，式 (2) によってレート  $\tilde{J}_0(\alpha, t)$  を計算する．これにより，ネットワーク外からの入力パケットレートとネットワーク内部の拡散型フロー制御によるレートとの親和性を高めることができる．つまり，ノード  $i=1$  で計算する情報  $r_0(t)$  については，式 (5) の代わりに

$$r_0(t) := \alpha J_1(1, t) - D_0 n_1(t) (= \tilde{J}_0(\alpha, t + d_0)) \quad (6)$$

を用いて決定する． $r_0(t)$  はノード  $i=1$  で取得可能な量のみによって計算することができる．

## 2.2 ネットワーク入口でのパケットレート制限法

拡散型フロー制御は局所情報に基づくノードの自律動作を用いた制御なので、ネットワーク内の状態を高速に安定化させる効果には優れるが、ネットワーク内に外部から流入する余分なパケットを防ぐ効果は少ない。そこで、必要に応じてネットワーク入口でパケットレートを制限する機能を併用し、ネットワーク入口でのパケット流入量を適切に制限することが必要である。

これを実現するためには、ネットワーク入口部分でフローに沿った経路上のネットワーク状態を把握し、その状態に合わせてネットワーク入口でのパケットシェーピングを施すことが有効である [12]。

ノード  $i$  が生成するフィードバック情報  $F_i(t)$  にリンクの最大利用可能帯域に関する情報  $\ell_i(t)$  を加え、

$$F_i(t) = (r_{i-1}(t), n_i(t), \ell_i(t)) \quad (7)$$

を上流ノード  $i-1$  に通知する。最大利用可能帯域に関する情報  $\ell_i(t)$  は、

$$\ell_i(t) = \min(L_i(t), \ell_{i+1}(t - d_i)) \quad (8)$$

のように生成する。つまり、 $\ell_i(t)$  の計算は、下流のリンクのうち一番小さい（とノード  $i$  が認識する）帯域の値を用いる。

ネットワーク内のノード ( $i = 1, 2, \dots, N$ ) での転送レート  $J_i(\alpha, t)$  の決定には、新しく追加された情報  $\ell_i(t)$  は使用せず、ネットワーク入口でのパケットレートの制限にのみ利用し、ネットワーク入口のパケットシェーピングのレート  $J_0(\alpha, t)$  を以下のように決める。

$$\begin{aligned} J_0(\alpha, t) &= \max(0, \min(\ell_0(t), r_0(t - d_0))) \\ &= \max(0, \min(\ell_0(t), \tilde{J}_0(\alpha, t))) \end{aligned} \quad (9)$$

ここで、 $\ell_0(t)$  の計算は、ノード  $i = 1$  から通知された  $\ell_1(t - d_0)$  とネットワークのアクセス帯域  $L_0$  (伝搬遅延は  $d_0$ ) から決めることができるので、実装では、式 (9) をネットワーク外のノードが計算することは必ずしも必要では無い。ネットワーク外のノードが拡散型フロー制御のレート決定規則をサポートしていなければ、ノード  $i = 1$  がシェーピングレートとして  $\min(\ell_1(t), r_0(t))$  をネットワーク外のノードに通知すればよい。但し、ネットワーク外のノードは、下流から指定されたレートでシェーピングすることができる機能を持つ必要がある。

## 3. ノード内パケット数平滑化の分類

ネットワークの輻輳を回避するためのノード内パケット数の平滑化について整理し、フロー制御による実現可能性を考察する。

ノード内パケット数の平滑化は以下のように分類することができる。

### (a) Serial Diffusion

あるフローの経路に沿ったノードに関して、そのフローに属するパケットのノード内パケット数を平滑化し、パケットロスを回避する方法。この平滑化は更に2種類に分類できる。

#### (a-1) Backward Serial Diffusion

ボトルネックに対してフローの上流方向にノード内パケット数を平滑化。

#### (a-2) Forward Serial Diffusion

ボトルネックに対してフローの下流方向にノード内パケット数を平滑化。

### (b) Parallel Diffusion

同一経路（の一部）を共有する複数のフローに関して、共通経路上にあるノード内パケット数のフロー間のばらつきを平滑化する。

これまでの単一フローに関する拡散型フロー制御は Serial Diffusion の実現を目指しており、特にノード  $i$  の利用可能帯域  $L_i$  がボトルネックとなる状況では、ノード内パケット数がフローの下流方向に平滑化することは無い (5.2 節参照)。つまり、Backward Serial Diffusion のみが実現している。より短時間でノード内パケット数を平滑化するには、Backward Serial Diffusion だけではなく Forward Serial Diffusion も同時に実現することが望ましい。これを実現するためには、特定のフローに関して利用可能帯域  $L_i$  を大きめに調整することが必要となる。

ノード内パケット数の平滑化を、フローの経路に沿った一次的な動きではなく、ネットワーク全体の面的な動きにするためには、Parallel Diffusion により複数フロー間での平滑化を実現することが必要である。Parallel Diffusion を実現するためには、ノード  $i$  を通過する複数のフローに対して利用可能帯域  $L_i$  を決定する際に、フロー毎に適当な重みをつけて、ノード内パケット数の多いフローに関して利用可能帯域  $L_i$  が大きめに調整されるようにしてやればよい。

このことから、複数フローの拡散型フロー制御において Backward Serial Diffusion だけでなく、Forward Serial Diffusion と Parallel Diffusion を併せて実現する為には、利用可能帯域  $L_i$  を適切に調整してやればよいことがわかる。

## 4. 複数フローに関する拡散型フロー制御方式

### 4.1 基本動作

本稿では、全てのフローを公平に扱い、アクティブな複数フローが等しく帯域を分け合うことを目的とする。フロー毎に異なる帯域を要求する場合への拡張は容易である。

ノード  $i$  と  $i+1$  の間のリンクを共有する  $M_i$  本のフローを  $j$  ( $j = 1, 2, \dots, M_i$ ) で識別することとし、フロー  $j$  に関する諸量を以下のように記述する。

$n_i^j(t)$ : 時刻  $t$  でノード  $i$  に存在するフロー  $j$  のパケット数。

$r_i^j(t - d_i)$ : 下流ノード  $i+1$  からのフィードバックにより (リンク伝播遅延  $d_i$  を伴って) 通知される指示レート。

$n_{i+1}^j(t - d_i)$ : 下流ノード  $i+1$  からのフィードバックにより (リンク伝播遅延  $d_i$  を伴って) 通知される下流ノード  $i+1$  に存在するフロー  $j$  のパケット数。

$L_i^j(t)$ : ノード  $i$  から  $i+1$  に向かうリンクのフロー  $j$  に関する利用可能帯域。

これらの量を用いて、フロー  $j$  に関する拡散型フロー制御の動作を考える。ノード  $i$  は、時刻  $t$  におけるフロー  $j$  の下流ノード  $i+1$  へのパケット転送レート  $J_i^j(\alpha, t)$  を自分が知りうる情報のみに基づいて決定し、パケット転送を行なう。また、上流ノード  $i-1$  に向けてノード  $i$  の情報  $F_i^j(t)$  をフィードバックする。パケット転送レート  $J_i^j(\alpha, t)$  の決定は、下流ノード  $i+1$  から通知される情報  $F_{i+1}^j(t - d_i)$  の到着毎に実行する。上流ノード  $i-1$  に向けたフィードバック情報  $F_i^j(t)$  の通知は、ノード  $i-1$  と  $i$  間の伝搬遅延  $d_{i-1}$  に比例した時間間隔で行なう。また、直前の連続する2つのフィードバック情報を生成する間に観測された、平均のアクティブフロー数 (ノード内に存在するパケットが属する異なるフロー数の平均値) を  $\bar{M}_i(t)$  とする。一般に  $\bar{M}_i(t) \leq M_i$  である。ノード  $i$  から  $i+1$  に向か

リンクの帯域を  $B_i$  とする．パケット転送レート  $J_i^j(\alpha, t)$  は

$$J_i^j(\alpha, t) = \max(0, \min(L_i^j(t), \tilde{J}_i^j(\alpha, t))) \quad (10)$$

$$\tilde{J}_i^j(\alpha, t) = \alpha r_i^j(t - d_i) - D_i(n_{i+1}^j(t - d_i) - n_i^j(t)) \quad (11)$$

によって与える．一方，ノード  $i$  が生成するフロー  $j$  に関するフィードバック情報  $F_i^j(t)$  は

$$F_i^j(t) = (r_{i-1}^j(t), n_i^j(t)) \quad (12)$$

からなり，これを上流ノード  $i-1$  に通知する．ここで，

$$r_{i-1}^j(t) = \max(0, \min(B_i/\bar{M}_i(t), \tilde{J}_i^j(1, t))) \quad (13)$$

とする．

ノード  $i$  がネットワークの入口 ( $i=1$ ) である場合，

$$r_0^j(t) := \alpha J_1^j(1, t) - D_0 n_1^j(t) (= \tilde{J}_0^j(\alpha, t + d_0)) \quad (14)$$

をネットワーク外のノード又はエンドホストに通知する．

ネットワーク入口でのシェーピングを施す場合は，以下の動作を加える必要がある．ノード  $i$  が生成するフロー  $j$  のフィードバック情報  $F_i^j(t)$  にリンクの最大利用可能帯域に関する情報  $\ell_i^j(t)$  を加え，

$$F_i^j(t) = (r_{i-1}^j(t), n_i^j(t), \ell_i^j(t)) \quad (15)$$

を上流ノード  $i-1$  に通知する．フロー  $j$  の最大利用可能帯域に関する情報  $\ell_i^j(t)$  は，

$$\ell_i^j(t) = \min(B_i/\bar{M}_i, \ell_{i+1}^j(t - d_i)) \quad (16)$$

のように生成する． $\ell_i^j(t)$  はネットワーク入口でのパケットレートの制限にのみ利用し，ネットワーク入口のパケットシェーピングのレート  $J_0^j(\alpha, t)$  を以下のように決める．

$$\begin{aligned} J_0^j(\alpha, t) &= \max(0, \min(\ell_0^j(t), r_0^j(t - d_0))) \\ &= \max(0, \min(\ell_0^j(t), \tilde{J}_0^j(\alpha, t))) \end{aligned} \quad (17)$$

#### 4.2 利用可能帯域の決定法

複数フローに関する拡散型フロー制御方式で，Backward Serial Diffusion だけでなく，Forward Serial Diffusion と Parallel Diffusion を併せて実現する為には，フロー間で利用可能帯域  $L_i^j$  を適切に調整してやる必要がある．

ノード  $i$  から  $i+1$  に向かうリンクの帯域を  $B_i$  とする． $L_i^j(t)$  ( $j=1, 2, \dots, M_i$ ) は

$$\sum_{j=1}^{M_i} L_i^j \leq B_i \quad (18)$$

となるように決めなければならない．

もし， $L_i^j(t)$  を  $B_i/M_i$  のように固定的に決めてしまうと，フロー間で帯域の干渉は起こらず，Forward Serial Diffusion と Parallel Diffusion が実現しなくなる．これは，他のフローとの干渉が無いため，単一フローの拡散型フロー制御の動作に帰するためである．

拡散型フロー制御では，理想的なパケット転送レートは  $\tilde{J}_i^j(\alpha, t)$  であり， $J_i^j(\alpha, t)$  は帯域の制限を考慮したものであるため，理想的な値からのずれが生じる．特に多くのフローが混在する環境では，

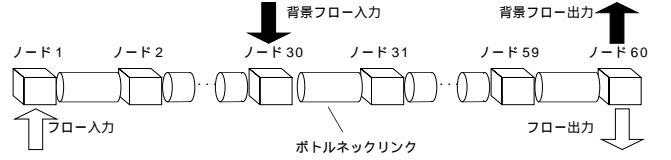


図2 ネットワークモデル.

$$\sum_{j=1}^{M_i} \tilde{J}_i^j(\alpha, t) > B_i \quad (19)$$

となる場合が多く現れ，各フローが理想的なパケット転送レート  $\tilde{J}_i^j(\alpha, t)$  を確保できる可能性は低い．このことは，ノード内パケット数が滑らかに平滑化することの妨げになる．各フローの理想的なパケット転送レート  $\tilde{J}_i^j(\alpha, t)$  を尊重しつつ，

$$\sum_{j=1}^{M_i} J_i^j(\alpha, t) \leq B_i \quad (20)$$

となるようにするために， $\tilde{J}_i^j(\alpha, t)$  による重みを付けて  $L_i^j$  を決定してみる．つまり，

$$L_i^j = B_i \frac{\tilde{J}_i^j(\alpha, t)}{\sum_{j=1}^{M_i} \tilde{J}_i^j(\alpha, t)} \quad (21)$$

とする．これにより， $\tilde{J}_i^j(\alpha, t)$  が比較的大きいフローが大きなパケット転送レートを獲得でき，下流に多くのパケットを転送できると同時に，他のフローのパケット転送レートを低くする効果があり，Forward Serial Diffusion と Parallel Diffusion の実現が期待できる．

## 5. シミュレーション評価

本節では，複数フローが混在した場合の拡散型フロー制御の特性を調べるために，単一フローに関する拡散型フロー制御との比較をシミュレーション結果により示す．本稿での評価は，ネットワーク入口でのパケット流入量を調整するために，拡散型フロー制御と共に  $\ell_i(t)$  を用いたシェーピングを施している．また，拡散型フロー制御のパラメータは  $D=0.1$ ,  $\alpha=1$  とし，フィードバック情報  $F_i^j(t)$  の送信間隔は  $d_{i-1}$  とした．

### 5.1 Serial Diffusion の評価モデル

図2のような60ノードからなる部分ネットワークを考える．これは，あるフローに沿ってネットワークの一部を切り出したモデルである．ノード間のリンク伝播遅延は1(単位時間)とした．パケットは固定長で，リンクは単位時間あたり100パケットの転送速度を持つ．

シミュレーションシナリオは，時刻  $t=0$  でノード30から60に向けたフロー(単位時間あたり100パケットの速度)  $j=1$  が開始され，時刻  $t=1000$  でノード1から60に抜ける新しいフロー(単位時間あたり100パケットの速度)  $j=2$  が開始される．新しいフロー  $j=2$  のパケットが流入することにより，ノード30から31に向かうリンクがボトルネックとなり，予め定められている拡散型フロー制御の動作により各フローのパケット流量が制御される．このとき，単一フロー及び複数フローに関する拡散型フロー制御各々について，ネットワークの状態が時間と共にどのように変化するかを調べる．

### 5.2 Serial Diffusion の評価結果

単一フローに関する拡散型フロー制御を適用した場合の，フ

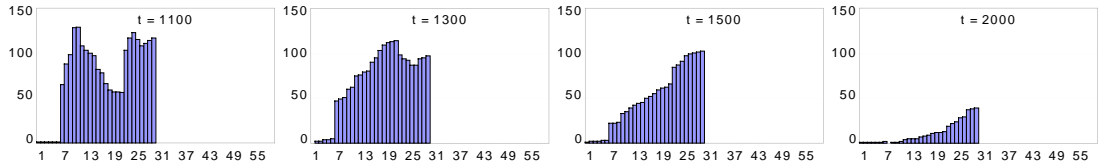


図 3 単一フローに関する拡散型フロー制御によるノード内パケット数時間変化．  
横軸はノード ID，縦軸はノード内パケット数を示す．

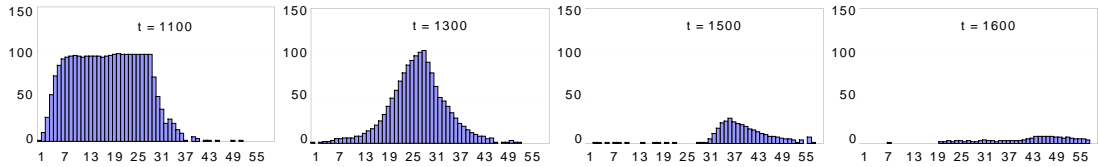


図 4 複数フローに関する拡散型フロー制御によるノード内パケット数時間変化．  
横軸はノード ID，縦軸はノード内パケット数を示す．

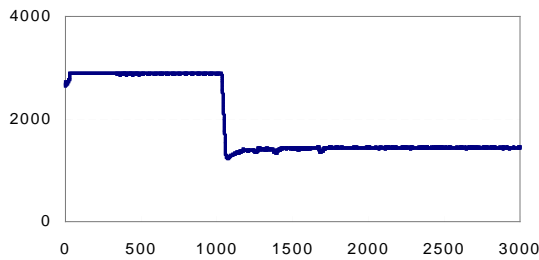


図 5 リンク上に存在する総パケット数の変化 (フロー  $j = 1$ )．  
横軸はシミュレーション時刻，縦軸はリンク上の  
総パケット数を示す．

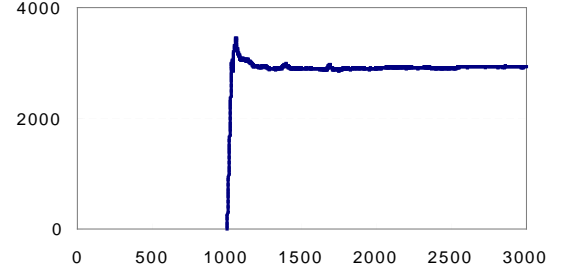


図 6 リンク上に存在する総パケット数の変化 (フロー  $j = 2$ )．  
横軸はシミュレーション時刻，縦軸はリンク上の  
総パケット数を示す．

ロー  $j = 2$  のノード内パケット数の振舞いを調べた結果が図 3 である．ボトルネック部分の利用可能帯域は， $B_{30} = 100$  [packets/unit time] を 2 本のフローが公平に分け合うとして，

$$L_{30}^1 = L_{30}^2 = B_{30}/2 \quad (22)$$

とした．ノード 30 では，パケットレート  $J_{30}^2(1, t)$  が  $B_{30}/2$  以下に抑えられるため，下流ノードでは輻輳が発生せず，ノード内パケット数が 0 になっている．ノード 30 で発生した輻輳は，フローの上流方向に拡散し，ボトルネックの一点にパケットの滞留が集中することを防いでいる．ここで，もし拡散型フロー制御が無ければ，パケットはノード 30 のみに滞留し，バッファ溢れによるパケット損失が発生する原因となってしまう．

複数フローに関する拡散型フロー制御を適用した場合の同様の結果が図 4 である．ボトルネック部分の利用可能帯域は式 (21) で決定され，状況に応じて  $B_{30}/2$  より大きくなることも小さくなることもありうる．この結果では，ノード 30 で発生した輻輳がフローの上流だけでなく下流方向へも拡散している．つまり，目的としていた通り Forward Serial Diffusion が実現していることが分かる．また，輻輳が収まるまでの時間も単一フローに関する拡散型フロー制御に比べて早い．

Forward Serial Diffusion の実現は，一時的にフロー  $j = 1$  の利用可能帯域を  $L_{30}^1 < B_{30}/2$  としたことによる．従って，このときのフロー  $j = 1$  への影響を調べる必要がある．フロー  $j = 1$  に関するノード内パケット数については，後述の Parallel Diffusion の評価で扱う．ここでは，複数フローに関する拡散型フロー制御が，ノード内パケット数の偏りを解消させつつ，同時にフロー  $j = 1, 2$  のパケットを公平に運んでいるかどうか

について調べる．各フローがどれだけのパケットを運んでいるかについては，ネットワークのリンク内に存在するパケット数をフロー毎に総計した値を用いる．この結果を示したのが図 5 及び 6 である．1 本のリンク上に存在できる最大パケット数は 100 個であり，フロー  $j = 1$  はフロー  $j = 2$  のほぼ半分のリンクを通過しているため，時刻  $t = 0$  から 1000 までは，3000 個が最大であり，フロー  $j = 2$  と競合する時刻  $t = 1000$  以降は 1500 個が最大となる．また， $t = 1000$  から入力するフロー  $j = 2$  は，最初からフロー  $j = 1$  と競合するので 3000 個が最大となる．どちらのフローについても，短時間で最大のパケット数に近くなり，ほぼ公平に帯域を分け合っていることが分かる．

### 5.3 Parallel Diffusion の評価モデルと結果

ここでは，複数フローに関する拡散型フロー制御により，複数フローに共通のノードに於いて，フロー毎のノード内パケット数がどのように変化するかを調べる．

ネットワークモデルは前述の評価と同じ次元ネットワークモデルを用いる．

時刻  $t = 0$  にノード 1 から 60 に抜けるフロー (単位時間あたり 100 パケットの速度)  $j = 1$  が開始される．また，時刻  $t = 1000$  にノード 30 から 60 に抜ける 2 本のフロー (単位時間あたり 20 パケットの速度)  $j = 2, 3$  が開始される．このとき，輻輳ノード 30 において，各フローのノード内パケット数の比較と，その時間変化を示したものが図 7 である．輻輳ノードからネットワークの入口が遠いフロー  $j = 1$  のノード内パケット数が一旦高くなるが，時間と共に平滑化する．もし，単

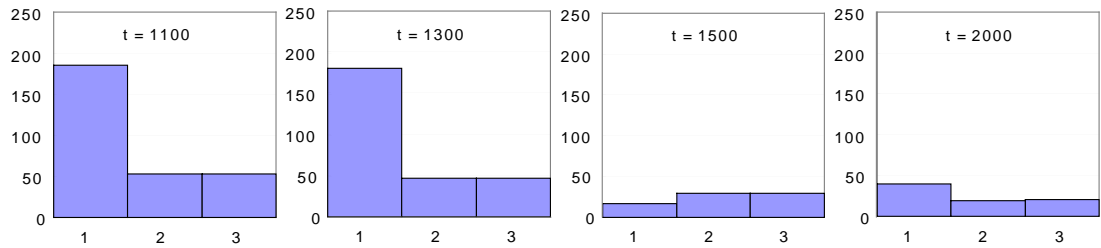


図7 ノード30におけるフロー毎のノード内パケット数の比較。  
横軸はフローID, 縦軸はノード内パケット数を示す。

一フローに関する拡散型フロー制御を用いて,

$$L_{30}^1 = L_{30}^2 = L_{30}^3 = B_{30}/3 \quad (23)$$

としたとすれば, フロー  $j = 2, 3$  についてはノード内パケット数が0になる. 図7では, フロー  $j = 2, 3$  に影響を与えることにより, フロー  $j = 1$  の輻輳を早期に回復させている. このときのフロー  $j = 2, 3$  のノード内パケット数は十分少なく, 影響は軽微であると考えられる.

## 6. おわりに

本稿では, 従来単一フローで考えてきた拡散型フロー制御を, 複数フローが混在する環境に適用する方法を考えた.

拡散型フロー制御は, ネットワーク内のパケットが特定箇所に過度に滞留することを防ぐことにより, パケット損失を防止する効果を狙っている. パケットの拡散方法には, フローに沿った Serial Diffusion と複数フロー間の Parallel Diffusion が考えられるが, 単一フローで考えてきた拡散型フロー制御では Serial Diffusion のうちの一部 (Backward Serial Diffusion) が実現するだけであった.

今回, 各フローの利用可能帯域を拡散型フロー制御と相性よく選ぶことで, Backward Serial Diffusion の他に, Forward Serial Diffusion と Parallel Diffusion を同時に実現することができ, 輻輳状態からの回復時間も短縮できることがわかった.

### 文献

- [1] C. Takano and M. Aida, "Stability and adaptability of autonomous decentralized flow control in high-speed networks," IEICE Transactions on Communications, vol. E86-B, no. 10, pp. 2882–2890, 2003.
- [2] Y. Bartal, J. Byers, and D. Raz, "Global optimization using local information with applications to flow control," Proc. the 38th Ann. IEEE Symp. on Foundations of Computer Science, Oct. 1997.
- [3] S. H. Low and D. E. Lapsley, "Optimization flow control-I: basic algorithm and convergence," IEEE/ACM Transactions on Networking, vol.7, no.6, pp.861–874, 1999.
- [4] K. Kar, S. Sarkar, and L. Tassiulas, "A simple rate control algorithm for maximizing total user utility," Proc. IEEE INFOCOM 2001, pp.133–141, 2001.
- [5] J. Mo and J. Walrand, "Fair end-to-end window based congestion control," IEEE/ACM Trans. Networking, vol.8, no.5, pp.556–567, Oct. 1999.
- [6] S. Kunniyur and R. Srikant, "A decentralized adaptive ECN marking algorithm," Proc. IEEE GLOBECOM '00, pp. 1719–1723, 2000.
- [7] R. Johari and D. Tan, "End-to-end congestion control for the Internet: Delays and stability," IEEE/ACM Trans. Networking, vol. 9, no. 6, pp. 818–832, Dec. 2001.
- [8] C. Takano, M. Aida and S. Kuribayashi, "Autonomous decentralized flow control in high-speed networks with inhomogeneous configurations," IEICE Transactions on Communications, vol. E87-B, no. 6, 2004.
- [9] M. Aida and C. Takano, "Principle of autonomous decentralized flow control and layered structure of network control with respect to time scales," in Supplement of the ISADS 2003 Conference Fast Abstracts, pp. 3-4, Pisa, Italy, Apr. 2003.
- [10] 会田 雅樹, 高野 知佐, "拡散型自律分散フロー制御の原理と狙い," 2003年信学ソ大, B-7-13, 2003.
- [11] 会田 雅樹, 高野 知佐, "拡散型フロー制御のネットワーク境界におけるレート決定法の検討," テレコミュニケーションマネジメント研究会, TM2003-52, Nov. 2003.
- [12] 高野 知佐, 会田 雅樹, "Open ネットワークに対する拡散型自律分散フロー制御適用法の検討," ネットワークシステム研究会, NS2003-136, Oct. 2003.