

## 自律分散型ネットワークQoS制御方式に関する研究

|          |   |
|----------|---|
| 著者       | 林 秀樹  |
| 発行年      | 2004-03   |
| その他のタイトル | A Study of an A Network QoS Control Sheme Based on Autonomous Distributed Control |
| 学位授与番号   | 26402甲第31号  |
| URL      | <a href="http://hdl.handle.net/10173/207">http://hdl.handle.net/10173/207</a>     |

平成 16 年 3 月修了  
博士 ( 工学 ) 学位論文

( 和文題目 ) 自律分散型ネットワーク QoS 制御方式  
に関する研究

( 英文題目 ) A Study on A Network QoS Control Scheme Based  
on Autonomous Distributed Control

公開論文審査及び最終試験

平成 16 年 2 月 12 日

高知工科大学大学院 工学研究科 基盤工学専攻 ( 基盤工学コース )

林 秀樹

Hideki Hayashi



# 要 旨

## 自律分散型ネットワーク QoS 制御方式 に関する研究

林 秀樹

本論文は、ネットワーク QoS 制御の自律分散型実現に関する一連の研究成果を、5 章に分けてまとめたものである。その研究の主眼点は、ネットワークの拡大と伝送帯域の大容量化に伴い多様化していく、サービスやアプリケーションの品質 (QoS) 保証のための QoS 制御の要件を明確にすることと、今後の発展すべき方向性の提示にある。

第 1 章序論では、本研究の目的ならびにその工学上の意義、および QoS 制御やフロー制御の関連分野での研究の現状について述べ、本研究で得られた新しい成果について概説する。

第 2 章では、将来のネットワークにおけるサービスやアプリケーションの多様化に伴い、既存のキューイング制御方式では対応できない問題を示し、この問題を解消するため、自律分散型ネットワーク QoS 制御方式に基づいた、自己タイミング型パイプラインを応用する新しいキューイング機構の特長を明らかにし、その基本的な構成を述べる。さらに本機構が状態遷移方程式を用いて解析可能であることを示し、原理的に優先制御機構向きの構成であり、機能分散により従来のキューイング方式には見られない遅延制御や廃棄制御を包含した優れたキューイング特性を持つことを明らかにする。

第 3 章では、ルータやスイッチ (ノード) 技術の進展に伴い、将来のネットワークにおける新旧ノード間の性能格差の拡大と、それに伴う、ネットワーク性能向上を十分にできない問題について述べ、これを解消するために、従来方式の延長にはない、本方式による新しい自律分散型フロー制御方式が必要であることを明らかにしている。さらに、このような自律

分散型制御アーキテクチャを適用したフロー制御方式が，自律的に分散先の各リンク上でのパケット廃棄を抑制し，中継先の処理能力に応じて過負荷を与えないフロー分散により，キューイング遅延を抑制でき，この結果，処理性能，柔軟性，および信頼性の観点から，原理的に非常に優れた方式であることを明らかにする．

第4章では，本研究で明らかにした自律分散型制御アーキテクチャ実現法の評価手法について述べ，シミュレーションによる評価結果を提示し，本研究によるアーキテクチャ実現法の有効性，および，優れた性能を発揮できることを明らかにする．

第5章では，本論文の各章で得られた成果を総括し，ノードからネットワークの制御まで，一貫した自律分散型制御に関して，今後の課題ならびに将来展望について述べる．

キーワード 自律分散制御, QoS, フロー制御, キューイング

# Abstract

## A Study on A Network QoS Control Scheme Based on Autonomous Distributed Control

Hideki Hayashi

This doctoral dissertation, organized into five chapters, describes a study on a network QoS control scheme based on autonomous distributed control. This study has been conducted to give suggestions on the future high-speed network QoS control because the significance of the flexible QoS control will increase to realize various network services and applications under the future large-scaled and broadband networks.

In the introduction chapter, the objectives of this study and its technological significances are addressed through a survey of the current status of relevant studies as well as related fields of QoS controls and flow controls. The key point is to introduce the autonomous distributed control in any layers of the network QoS control. Then, two network QoS control schemes proposed in the dissertation are overviewed in terms of the autonomous distributed control.

In the next chapter, a self-timed priority-based queueing scheme is firstly proposed as an autonomous distributed QoS control scheme at the inner node level of the network. The proposed scheme differentiates Internet packet flows per class without introducing any centralized control functions. Several variants of the scheme are also discussed to improve its queueing characteristics, i.e., queueing delay and packet loss probability. Furthermore, an analytical model based on the state-transition probability equations is

formulated to evaluate the scheme.

Secondly, an autonomous distributed flow control scheme is proposed, in the third chapter, to realize the inter-network level QoS control of the network. The proposed scheme distributes packet flows to multiple links of relay nodes to avoid traffic congestions as well as to reduce queueing delay. Especially, the scheme will efficiently work at a heterogeneous network which is mixed both old low-performance routers or switches (nodes) and new high-performance ones equipped with dramatically advanced technology. Therefore, the scheme gives network designers flexibility even when they expand the existing network.

In the fourth chapter, both proposed schemes are evaluated through simulation experiments under various practical traffic conditions and network configurations. The simulation results show that both schemes perform efficiently in comparison with several related schemes proposed previously.

The final chapter summarizes the proposed schemes and their remarkable features and then discusses future studies of this doctoral work and perspectives of it from the view point of potentiality of the autonomous distributed control.

***key words***     autonomous distributed control, QoS, flow control, queueing

## 関連発表論文

### ( 1 ) 論文誌等に発表した論文

- [1] 林 秀樹, 岩田 誠, 寺田 浩詔: “クラス別 QoS 制御向き自己同期型優先キューの性能評価,” FIT2003 情報科学技術フォーラム情報技術レターズ, vol.2, pp.313–314, Sep. 2003.
- [2] 林 秀樹, 岩田 誠, 島村 和典: “異種混合ネットワークにおける自律型フロー分散制御方式,” 情報処理学会論文誌, vol.45, no.2, pp.426–437, Feb. 2004.
- [3] 林 秀樹, 岩田 誠, 寺田 浩詔, 島村 和典: “自己タイミング型パイプラインによる優先キューイング制御方式,” 電子情報通信学会論文誌 B(採録, 掲載予定).

### ( 2 ) 国際会議等に発表した論文

- [1] Daichi Morikawa, Makoto Iwata, Hideki Hayashi, Hiroaki Terada: “Super-pipelined IP-Address Lookups in a Data-Driven Network,” *Proceedings of the IASTED International Conference, Parallel and Distributed Computing and Systems*, pp.431–436, Anaheim, Aug. 2001.
- [2] Hideki Hayashi, Makoto Iwata, Toshihiko Hosomi, and Hiroaki Terada: “A Self-Timed Pipeline Implementation of Class-Based QoS Control,” *International Workshop on Embedded Systems at IEEE High Performance Computing Conference 2001*, pp.6–10, Hyderabad, Dec. 2001.
- [3] Hideki Hayashi, Makoto Iwata, and Hiroaki Terada: “An Autonomous Class-Based QoS Control Utilizing a Self-Timed Folded Pipeline,” *The 4th International Conference on Advanced Communication Technology*, pp.469–474, Phoenix Park, Feb. 2002.
- [4] Makoto Iwata, Michihiro Ogura, Yuko Ohishi, Hideki Hayashi, and Hiroaki



Terada: "100MPackets/s Fully Self-Timed Priority Queue: FQ," *2004 IEEE International Solid-State Circuits Conference*, Session 8.1, San Francisco, Feb. 2004.

( 3 ) 学会等に発表した論文

- [1] 林 秀樹, 岩田 誠, 孔 令杰, 米田 進, 寺田 浩詔: "超高帯域光ネットワークにおける分散型マルチプロトコルルータ," 情報処理学会研究報告 2001-DSM-22, pp.1-6, July 2001.
- [2] 細美 俊彦, 林 秀樹, 岩田 誠, 寺田 浩詔: "自己同期パイプラインによる Diffserv 制御機構," 平成 13 年度電気関係学会四国支部連合大会講演論文集, pp.199, Sep. 2001.
- [3] 細美 俊彦, 岩田 誠, 林 秀樹, 寺田 浩詔: "自己同期パイプラインによる SLA 制御の実現法," 2002 年電子情報通信学会総合大会," B-7-112, March 2002.
- [4] 林 秀樹, 島村 和典: "自律抑制型フォロワー制御によるネットワーク性能の改善," 2003 年電子情報通信学会総合大会, B-7-69, March 2003.
- [5] 朝日山 輝久, 林 秀樹, 岩田 誠, 寺田 浩詔: "折り返し型自律 QoS 制御方式におけるジッタ抑制法の一考察," 2003 年電子情報通信学会ソサエティ大会, B-6-78, Sep. 2003.
- [6] 林 秀樹, 岩田 誠, 寺田 浩詔, 島村 和典: "折り返し型自律優先キューイング機構における順序補償方式の一考察," 2004 年情報処理学会第 66 回全国大会, 4J-4, March 2004.

# 目次

|       |                                |    |
|-------|--------------------------------|----|
| 第1章   | 序論                             | 1  |
| 1.1   | 研究の背景と目的                       | 1  |
| 1.2   | キューイング制御における課題                 | 3  |
| 1.3   | フロー制御における課題                    | 5  |
| 1.4   | 研究の内容                          | 8  |
| 第2章   | 自律分散型制御アーキテクチャに基づく優先キューイング制御方式 | 11 |
| 2.1   | まえがき                           | 11 |
| 2.2   | 自律分散型優先キューイング制御方式              | 13 |
| 2.2.1 | 基本構成                           | 13 |
| 2.2.2 | 優先度に基づく廃棄制御                    | 16 |
| 2.2.3 | 優先度に基づく遅延制御                    | 18 |
| 2.3   | マルコフ連鎖による解析モデル                 | 20 |
| 2.4   | 順序補償方式                         | 22 |
| 2.4.1 | 順序逆転現象                         | 23 |
| 2.4.2 | 自律分散型アーキテクチャにおける順序補償方式         | 23 |
|       | 優先クラス別メモリとバイパスフラグ              | 23 |
|       | バイパス動作                         | 25 |
|       | パケット廃棄時の制御                     | 28 |
|       | 優先クラス別メモリのクリア                  | 30 |
| 2.5   | まとめ                            | 32 |
| 第3章   | 自律分散型制御アーキテクチャに基づくフロー分散制御方式    | 35 |
| 3.1   | まえがき                           | 35 |

|             |                                      |           |
|-------------|--------------------------------------|-----------|
| 3.2         | 複数経路フロー分散・集約方式 . . . . .             | 37        |
| 3.2.1       | 要件 . . . . .                         | 37        |
| 3.2.2       | フロー分散方式 . . . . .                    | 38        |
| 3.2.3       | フロー集約方式 . . . . .                    | 40        |
| 3.3         | 個別リンクのフロー制御方式 . . . . .              | 43        |
| 3.4         | まとめ . . . . .                        | 46        |
| <b>第4章</b>  | <b>評価</b>                            | <b>49</b> |
| 4.1         | まえがき . . . . .                       | 49        |
| 4.2         | 自律分散型制御に基づく優先キューイング制御方式の評価 . . . . . | 49        |
| 4.2.1       | 評価方法 . . . . .                       | 50        |
| 4.2.2       | 特性評価 . . . . .                       | 52        |
| 4.2.3       | 既存方式との比較評価 . . . . .                 | 59        |
| 4.2.4       | 順序補償効果の比較評価 . . . . .                | 63        |
| 4.3         | 自律分散型制御に基づくフロー分散制御方式の評価 . . . . .    | 65        |
| 4.3.1       | 特性評価 . . . . .                       | 67        |
| 4.3.2       | 典型的なネットワークへの適用例の評価 . . . . .         | 71        |
| 4.3.3       | 例外的なケースの評価 . . . . .                 | 80        |
| 4.4         | まとめ . . . . .                        | 82        |
| <b>第5章</b>  | <b>結論</b>                            | <b>85</b> |
|             | <b>謝辞</b>                            | <b>91</b> |
|             | <b>参考文献</b>                          | <b>93</b> |
| <b>付録 A</b> | <b>折り返し型 STP のマルコフ連鎖モデル</b>          | <b>99</b> |
| A.1         | パケットの存在確率 $ON$ . . . . .             | 99        |

|       |                                |     |
|-------|--------------------------------|-----|
| A.1.1 | 往路側ステージの $ON_{i,0}$ . . . . .  | 99  |
| A.1.2 | 復路側ステージの $ON_{i,1}$ . . . . .  | 101 |
| A.2   | パケットのバイパス確率 $p$ . . . . .      | 102 |
| A.3   | パケットの前方段移動確率 $r$ . . . . .     | 102 |
| A.3.1 | 往路側ステージの $r_{i,i+1}$ . . . . . | 102 |
| A.3.2 | 復路側ステージの $r_{i,i-1}$ . . . . . | 103 |
| A.4   | パケットの出力確率 $\mu$ . . . . .      | 104 |



# 目次

|      |                               |    |
|------|-------------------------------|----|
| 2.1  | STP における自律動作                  | 13 |
| 2.2  | 自律分散型優先キューイング制御方式の基本構成        | 14 |
| 2.3  | バイパス経路でのハンドシェイク手順             | 16 |
| 2.4  | 自律分散型優先キューイング制御方式の廃棄制御        | 17 |
| 2.5  | 自律分散型優先キューイング制御方式の遅延制御        | 19 |
| 2.6  | 自己タイミング型キューイング方式の状態遷移         | 21 |
| 2.7  | パケットの順序逆転現象                   | 24 |
| 2.8  | 優先クラス別メモリとバイパスフラグ             | 26 |
| 2.9  | バイパスフラグに基づくバイパス動作             | 27 |
| 2.10 | パケット廃棄時の制御                    | 29 |
| 2.11 | 優先クラス別メモリクリア用のリセットパケット        | 31 |
|      |                               |    |
| 3.1  | トラヒック負荷分散を必要とするネットワーク構成例      | 38 |
| 3.2  | 高速送信側での自律型フロー分散方式             | 39 |
| 3.3  | パケット順序補正が可能な環状型受信バッファ機構       | 41 |
| 3.4  | 個別リンクのフロー制御信号シーケンス            | 45 |
|      |                               |    |
| 4.1  | 離散事象シミュレーションモデルにおけるパケット移動の一例  | 51 |
| 4.2  | 理論解析とシミュレーション結果の比較            | 53 |
| 4.3  | WFQ と自律分散型優先キューイング方式の遅延特性の比較  | 61 |
| 4.4  | WRED と自律分散型優先キューイング方式の廃棄特性の比較 | 62 |
| 4.5  | 遅延特性における順序補償の効果               | 64 |
| 4.6  | 基本評価モデル                       | 66 |

|      |   |    |
|------|---|----|
| 4.7  | 同種ノードで構成された中継経路の最小転送間隔 $1/LP$ による最大ネットワークレイテンシーと最大スループット到達時間の特性 . . . . . | 68 |
| 4.8  | 異種ノードで構成された 6 種類の複数経路による評価モデル . . . . .                                   | 69 |
| 4.9  | 異種ノードで構成された中継経路へのフロー分散順序による最大ネットワークレイテンシーと最大スループット到達時間の特性 . . . . .       | 70 |
| 4.10 | 同種ノードで構成された中継経路へフロー分散したネットワークレイテンシー特性 . . . . .                           | 72 |
| 4.11 | 異種ノードで構成された複数経路による評価モデル . . . . .   | 74 |
| 4.12 | 異種ノードで構成された中継経路へのフロー分散順序パターンの違いによるネットワークレイテンシー特性 . . . . .                | 75 |
| 4.13 | 異種ノードで構成された中継経路へのフロー分散順序パターン A の遅延変動とパケットロスに対するネットワークレイテンシー特性 . . . . .   | 76 |
| 4.14 | 応用評価モデル . . . . .   | 78 |
| 4.15 | 送信側高速ノードが 2 つの場合のネットワークレイテンシー特性 (flow 0) . . . . .                        | 79 |
| 4.16 | 送信側高速ノードが 2 つの場合のネットワークレイテンシー特性 (flow 1) . . . . .                        | 80 |
| 4.17 | 例外ケースの評価モデル . . . . .   | 81 |
| 4.18 | フロー到着時間比較 . . . . .   | 82 |

# 表目次

|                                   |    |
|-----------------------------------|----|
| 4.1 優先キューイング制御方式の総合評価結果 . . . . . | 57 |
|-----------------------------------|----|





# 第 1 章

## 序 論

### 1.1 研究の背景と目的

昨今の光通信技術の飛躍的な進展を受けて、波長多重や光増幅技術など、伝送路(リンク)技術の進展には著しいものが見られ、これら技術進展を背景とした、伝送帯域の広帯域大容量化が進んでいる。これに伴い、ネットワークを介して様々な情報の疎通を行うことが期待されており、実際に、ストリーミング配信、VoIP( Voice over IP(Internet Protocol) ) [1]、E-learning、電子商取引など、多様なサービスやアプリケーションが試みられ、導入が進められている。このようなサービスの多様化は今後一層進んでいくと考えられ、特にインターネットにはあらゆるサービスを包含していく動向が見受けられる。

しかしながら、ネットワークの広帯域大容量化は、ビットレートの飛躍的向上と共に、ネットワーク内に存在する種々の遅延を相対的に顕在化させることになる。このため、これらの遅延の累積が、ネットワークに導入された個々のサービスやアプリケーションの性能や特性に無視できない影響を及ぼしていく。さらに、インターネットのように拡張が続き、不均質なネットワークでは、これらの遅延は変動し、その変動は統計的事象として評価ができたとしても、正確な予測が困難である。このような遅延の影響を、エンドホスト間のみで補正する場合、非常に大きなバッファと許容遅延時間を設ける必要があり、サービスやアプリケーションの要求する品質(QoS: Quality of Service)を満足することが難しい。すなわち、このようなネットワークでは、ネットワークを構成するルータやスイッチ(ノード)間において、処理遅延の影響を限定的に留め、さらにノード内においても処理遅延の影響の累積を解消することにより、エンドホスト間の QoS 保証を満たすネットワーク QoS 制御が必

要である。したがって、各ノードの処理遅延の影響を局所化し、トラフィックの変動やネットワークの拡張に柔軟な自律分散型制御システムの実現が重要な課題となる。

筆者は既に、ノードアーキテクチャにおいて、パケットの転送や多重処理に徹底した自律分散型制御システムを採り入れた、分散型マルチプロトコルルータを提唱している [2]。このルータは、従来方式のような集中制御に伴う、プロセッシングスケジューリングを不要とする動的データ駆動アーキテクチャ [3] の活用により、多様なプロトコルの処理を並列にシームレスに行える。このため、従来方式のようなスケジューリングオーバーヘッドに起因する、プロセッシングリソースの活用できない時間を原理的に持たない。また、それらプロトコル処理のために、徹底して活用された自己タイミング型パイプラインアーキテクチャによる機能分散化の恩恵を受け、個々の処理に伴う遅延が局所的に解消され、過負荷なトラフィックによる輻輳をある程度吸収できる、エラスティックな特性を持つ。これらの特性により、既存方式と比較すると、特にマルチプロトコル処理において、優れた性能を示すことが確認された。

近年のサービスやアプリケーションの多様化を受けて、ノードアーキテクチャの中でも優先制御処理機構は重要視されつつある。また、それに伴い、この処理にかかる負荷と機能の複雑さに起因する、大きな遅延が問題視されている。また、ネットワークを疎通する個々のフローにおいても、ネットワークの不均質さから被る遅延の増加や変動が、サービスやアプリケーションの安定供給の上で問題となりつつある。本論文では、まず、このノード内の優先制御処理機構に目を向け、ここに自律分散型制御アーキテクチャを導入することによって、ノード処理の遅延の低減化と性能改善を目指す。次に、ノードが連結されて構成されたネットワークを導通するフローの制御法に目を向け、各ノードの処理遅延の累積、または、隘路<sup>あいろ</sup>の経路が要因となる、ホスト間の遅延の増加を未然に防ぐ適切なフロー制御を、自律分散型制御アーキテクチャに基づいて構築し、フローの安定化とネットワークリソースの有効活用を目指す。

このノード内の優先制御に伴う遅延と、ノード間のフロー制御に伴う遅延の大きな要因は、実は共通したものである。これらの要因は、各ノードの処理待ちに伴うキューイング遅

延であることが明確になりつつある．近年のノードの処理能力の向上に伴い，処理速度（スループット）の向上は著しい．しかし，既存の処理方式では，入力されたパケットを入力順にキューイングメモリに格納し，処理を待たせるため，優先制御のように複数のフローを処理する制御では，他のフローのキュー出力可否によって，自フローのキュー出力に影響を受ける．また，既存のノード間のフロー制御方式では，処理能力の低いノードのキューイングメモリにパケットが蓄積され，フローの遅延は，このノードのキューイング遅延に支配される．すなわち，ノード内およびノード間の遅延低減化はキューイング遅延を抑制し，かつ累積させないことが課題解決において肝要である．そこで，本研究では，大きなキューイング遅延が見込まれる，各ノードにおける優先制御，およびノード間のフロー制御に着目し，ノードからネットワークに至る自律分散的制御手法の確立によって，自律分散型ネットワーク QoS 制御方式の実現を目指している．すなわち，

- 1) ノードのキューメモリ内部でのキューイングを必要最小限に抑え，低遅延かつ機能的に優先制御を行い，入出力の遅延変動を吸収できる自律分散型制御アーキテクチャ，ならびに，
- 2) ノードの処理能力が不均質であることを前提とし，フローの出力先ノードに過度な負荷を与えないようにフローを送出し，フロー送出ノードに余剰能力があれば，フローを分散して送出するフロー分散制御が行える，自律分散型制御アーキテクチャ，

を柔軟かつ拡張性の高い，自己タイミング型制御方式を基礎として，確立することを目指している．

## 1.2 キューイング制御における課題

一般にキューイング制御とは，入力されたパケットを宛先ノードに送出するためにメモリに蓄積し，そのパケットに処理される順序を与え，その処理が行われるまで待たせる制御である．したがって，キューイング制御では，必然的に処理待ちのための遅延を生じる．また，処理待ちのパケットが増え，メモリに空きが無くなり，蓄積できなくなったパケットは

廃棄されることになる。このような廃棄方式を Drop Tail と呼ぶ。入力された順にパケットを処理して出力する，最も簡単なキューイング制御を FIFO ( First In First Out ) と呼び，このキューイングは実装が簡単であることから広く用いられている。しかし，パケットの入力順のみに依存するキューイング制御であるため，QoS 制御で必要な，特定のパケットの優先的な取り扱いができないという問題点を持つ。また，廃棄方式が Drop Tail であるため，バースト的なトラフィックにキューイングメモリを占有されやすく，ノードを制御するパケットまで廃棄され，サービス不能に陥る危険を併せ持つ。したがって，QoS 制御には不向きなキューイング制御であると言える。

一方，パケットに優先度を与え，この処理待ちの遅延を優先度に応じて差別化できるキューイングを優先キューイングと呼ぶ。単一のキューを想定した優先キューイングには，Drop Tail の欠点を改善した RED ( Random Early Detection ) [4] を応用し，キューイングの閾値を優先度別に割り当てることによって，キューイングの差別化を行う，FRED ( Fair Random Early Drop ) [5]，RIO ( Random Early Drop gateway with In/Out bit ) [6]，WRED ( Weighted RED ) [7] などがあげられる。RED では，キューイング前に平均キュー長に応じて確率的にパケットを廃棄し，FIFO メモリにキューイングする。これにより，バースト的なフローによるキューイングメモリの寡占や，自フローの過剰なキューイングによる他のフローのパケット廃棄への影響を抑制している。例えば，WRED は，優先度別に，この廃棄確率の閾値を与え，優先度に応じたキューイングの差別化を可能にしている。このように，平均キュー長を制御し，事前に輻輳を予防し，QoS の保証を目指したキューイング制御を一般に AQM ( Active Queue Management ) [8] と呼ぶ。

複数のキューを想定した優先キューイングには，優先度別にキューを与え，それぞれのキューからの出力の順序を制御する，PQ ( Priority Queueing ) や FQ ( Fair Queueing ) [9, 10, 11, 12, 13, 14, 15] などがあげられる。PQ は，優先度の高いパケットがキューイングメモリ内にある限り，優先的に出力するキューイング制御であり，FQ は，優先度に応じた重みにより，出力帯域の割り当てを応分に差別化するキューイング制御である。このように，複数のキューからの出力の順序を制御することによって，優先度に応じた差別化を行う

キューイング制御を一般にスケジューリング制御と呼ぶ。

AQM とスケジューリング制御は、組み合わせて用いることが可能であり、近年、拡張性の高いネットワーク QoS 制御方式であるとして注目される DiffServ ( Differentiated Services ) [16] における優先制御の実装手段として、用いられることが多い。しかしながら、これらの方式を組み合わせた優先キューイングでは、優先度に基づいた廃棄率の差別化やキューイング遅延の差別化を行うことはできても、キューイング自体の遅延の抑制には十分な効果を与えることができない。AQM は、FIFO メモリにキューイングを行うため、たとえば、キューイング時の確率的な廃棄により、輻輳が予防できたとしても、平均キュー長が長くなることによるキューイング遅延の増加を避けることができない。特に、複数の優先度を持つパケットを単一のキューにキューイングする WRED などの場合、いずれのパケットのキューイング遅延も、全てのパケットの遅延に影響を与え、全体のキューイング遅延を増加させる。スケジューリング制御では、複数の優先度のパケット間のキューイング遅延の差別化は行えても、キュー長の制御や廃棄制御には関与しないため、全体のキューイング遅延の抑制には寄与できない。

したがって、このキューイング遅延を可能な限り低減できる、新しいキューイングアーキテクチャが求められる。本論文では、自己タイミング型制御方式を基礎とした、自律分散型制御アーキテクチャを目指す研究の一環として、この課題を解決する新しいキューイングアーキテクチャを示す。

### 1.3 フロー制御における課題

インターネットのように、多様な処理能力のノードで構成される不均質なネットワークでは、送出したパケットが受信側ホストに転送される過程で、無駄に廃棄されることのないように、また、ネットワークに過負荷を与えないように、送信側でパケットの流量 ( フロー ) を調整する必要がある。このフローの調整機能をフロー制御と呼ぶ。インターネットで標準的プロトコルとして用いられる TCP/IP ( Transmission Control Protocol / IP ) のフロー

制御では、広告ウィンドウと呼ばれる TCP コネクションの受信側ホストから通知される許容受信量の情報を基礎として、送信側ホストのフローの送出を調整する。このフロー制御方式はウィンドウ制御方式とも呼ばれる。

ウィンドウ制御方式は、ホスト間のフロー制御方式であるため、経路を構成し、ホストには直接接続されないノードの処理能力を知ることができない。このため、送信側ホストは、受信側ホストから返される ACK パケット ( Acknowledge packet ) に記録された送信パケットの受領情報を元に、送信パケットの送達確認や廃棄検出を行い、廃棄パケットの再送とフローレートの試行錯誤的な調整を行う [17, 18, 19, 20, 21]。このフロー制御方式は、原理的にどのような不均質なネットワークでも動作できるという利点がある。しかしながら、近年のノードの処理能力の進展は目覚しく、著しく処理能力の高いノード ( 高速ノード ) が導入され、ネットワークの不均質格差は拡大しつつある。このようなネットワークにおいて、ウィンドウ制御方式によるフロー制御を用いた場合、パケットの廃棄、再送、ウィンドウサイズの縮小を繰り返し、フローレートの安定に時間がかかり、特に高速ノードの処理能力が十分に発揮できないという問題が顕在化してきている。また、近年ストリーミング等のアプリケーションの通信プロトコルとして、よく用いられる UDP/IP ( User Datagram Protocol / IP ) の場合、フロー制御が行われないため、パケットが大量に処理能力の低いノード ( 低速ノード ) に流入し、最悪の場合、低速ノードがフリーズする可能性もあり、ネットワークの信頼性を大きく損ねてしまう。

相対的に処理能力が顕著に劣るノードは、悪意を持ってノードに負荷を与える DDOS ( Distributed Denial of Service ) 攻撃等の対象にもなりやすく、ネットワークに脆弱さを内在することとなり、安定運用の面でも問題がある。仮にこれら不均質なノードを経由して適切にフロー制御が行えたとしても、高速ノードの処理能力を活かし、かつ低速ノードが隘路にならないようにフローを送出するためには、トラフィックを負荷分散することが必須となる。

この課題に対して、フローを複数経路に均等に負荷分散して送出し、経路あたりの負荷を下げる方法が提案されている [22, 23]。例えば、文献 [22] では、複数経路にフローを分散す

る場合の課題として、負荷分散の均質化とパケットの順序補正を挙げている。前者の課題に対しては、FQ アルゴリズムを逆転応用させ、可変長パケットに対応できる負荷分散手法を提案し、後者の課題では、パケットにシーケンス番号が付加しなくてもほぼ順序補正を行えるように、ラウンドマーカーパケットをデータパケットと共に周期的に受信側へ送出し、マーカーパケットを補正的に用いてラウンドロビンのようにフローを集約することを提案している。つまり、均等な負荷分散を目指したパケット単位、あるいは、パケットを細分化したセル単位のフロー分散を行うと、フロー分散および集約のためのスケジューリングや、パケット順序補正のために高速な専用ハードウェアが必要となることが伺える。

このような努力を払っても、この手法では均等な負荷分散が行えないことが懸念される。なぜなら、高速ネットワーク内の中継ノードの処理能力は、パケット長よりもむしろパケット数に依存するため、FQ のようなサービスレートの公平さを目指したスケジューリング制御を用いてフロー分散を行っても、パケット長によって、分散送出された経路ごとの処理負荷が偏ってしまうからである。すなわち、長いパケットが経由する経路中のノードの処理負荷が小さく、短いパケットが経由する経路中のノードの処理負荷が大きくなる。この処理負荷の違いのため、短いパケットの経由する経路の遅延が大きくなるリスクを生じる。

また、各経路の処理遅延が変動した場合、ラウンドロビンのようにフローを集約すると、各経路からのパケットの読み出し間隔が揺らぎ、これがフロー全体の遅延の増加につながる。すなわち、均等な負荷分散を目指してフロー分散を行っても、見合う効果を得られず、非効率的であると言わざるを得ない。簡易な実装を目指して、フロー単位で分散する経路を選択する方式 [24] も提案されているが、フローの総量が大きいと、やはり負荷が偏る場合があり、適切な負荷分散が難しい。

フロー分散時に必要とされている均等な負荷分散の本来の目的は、ネットワークのある部分（ノードやリンク）に負荷が偏り、その部分を経由するフローの隘路となることを避けるために、各リンク、あるいは各ノードに負荷を均等に割り当て、ネットワーク全体のリソースを有効に活用しようとする考えに基づいている。その均等な負荷分散のアルゴリズムのために、フローを分散または集約するノード、あるいは各経路中のノードが過負荷になるよう



では目的から大きく外れることになる。それに，インターネットのような不均質なネットワークでは，遅延変動は不可避であると考えざるを得ない。それでもなお，経路中で順序保証を行おうとすれば遅延の蓄積は必至となる。

したがって，均等な負荷分散には拘らずに，ノードに過負荷を与えないことを目指し，遅延変動を許容でき，ネットワークリソースをできるだけ活用できるフロー分散制御方式を追求することが本来のフロー分散の目的に適い，必要とされると考える。本論文では，前節同様の，自律分散制御アーキテクチャを目指す研究の一環として，この課題を解決する新しいフロー分散制御方式を示している。

## 1.4 研究の内容

本論文では，ネットワークの著しい進展に伴う，伝送帯域の大容量化とサービスやアプリケーションの多様化の結果，既存の制御方式やアーキテクチャでは解決できず，顕在化していく新たな遅延の問題を解決し，ネットワークリソースの有効活用と拡張性の高い QoS 制御を行える自律分散型ネットワーク QoS 制御方式を提示する。また，この方式の優位性を，定性的および定量的に論じ，この研究で得られた成果を明らかにする。

第 2 章は，1.2 節に述べたキューイング制御の課題を受けて，将来のネットワークにおける既存のキューイング制御方式の抱える問題を挙げ，この問題を解消する自律分散制御アーキテクチャに基づく新しいキューイングアーキテクチャを提示する。局所制御性に優れた自己タイミング型パイプラインを応用するこのキューイングアーキテクチャが，マルコフ連鎖による確率モデルに基づき解析可能であることを示し，原理的に優先制御機構向きの構成であることを明らかにする。また，このキューイングアーキテクチャが，柔軟性と拡張性に優れ，種々の制御を機能分散することにより容易に適用でき，従来 of キューイング方式には見られない遅延制御や廃棄制御を包含した，優れたキューイング特性を持つことを明らかにする。

第 3 章は，1.3 節に述べたフロー制御の課題を受けて，ノード技術の進展に伴い，将来の

ネットワークにおける新旧ノード間の性能格差が原因となって生じるネットワークの不均質差の拡大と，それに伴って，ネットワーク性能の向上と，ノードの処理能力の活用が十分にできない問題について述べ，これを解消するために，従来方式の延長上にはない，新しいフロー制御方式が必要であることを明らかにしている．さらに，自律分散型制御アーキテクチャを適用したフロー分散制御方式を提示し，この方式が，自律的に分散先の各中継経路上でのパケット廃棄を抑制でき，中継先のノードの処理能力に応じてフローを分散させることにより，キューイング遅延を抑制できることを示す．これらのことから，処理性能，柔軟性，および信頼性の観点から，原理的に非常に優れた方式であることを明らかにする．

第4章では，本研究で明らかにした自律分散型ネットワーク QoS 制御方式の評価手法について述べ，2章および3章で具体的に示した，本方式に基づくキューイング制御方式とフロー分散制御方式の，シミュレーションによる評価結果を提示する．これらの結果から，本研究による QoS 制御アーキテクチャの有効性を示すとともに，この方式が優れた性能を発揮できることを明らかにする．



## 第 2 章

# 自律分散型制御アーキテクチャに基づく優先キューイング制御方式

### 2.1 まえがき

インターネットは今や，社会や生活の基盤として広く浸透し，一般の利用者が使用可能な通信速度も飛躍的に広帯域化しつつある．これに伴い，多種多様な通信サービスやアプリケーションが登場し，導入され，利用されている．これらのサービスやアプリケーションは，同一のネットワークで提供され，かつ，それぞれに多様な通信環境の条件（遅延，帯域，廃棄率など）を要求する．これらの多様な通信品質（QoS）をそれぞれに適切に保証するため，多くの QoS 制御技術が提案されてきた [25, 26, 27, 28]．これらの一部は商用のルータやスイッチ（ノード）内に既に実装され，QoS 保証の必要なサービスやアプリケーションのために用いられつつある [29]．中でも，DiffServ[16] は，IntServ[30] に比べて，ネットワークのスケラビリティを維持したまま，簡易に QoS 制御を実現する技術として各所で研究開発が進められている [31, 32]．

筆者は，この DiffServ を想定した QoS 制御の一実現法として，高機能 LSI モジュールの実現に適した自己タイミング型パイプライン STP（Self-Timed Pipeline）[33] を応用した，優先キューイング制御方式について提案してきた [34, 35, 36, 37, 38, 39]．本章では，この自律分散型制御アーキテクチャに基づく新しい優先キューイング制御方式について述べる．

優先度に基づく制御には一般に，キューイング時の廃棄制御を基礎にしたアクティブキュー管理方式 AQM（Active Queue Management）[8] と，優先度を持つクラス（あるい

は、フロー)毎に設けられた複数のキューからの読み出し制御(スケジューリング)を目的としたフェアキューイング方式 FQ( Fair Queueing ) [9, 10, 11, 12, 13, 14, 15] が組み合わせて用いられる。そして、両者の特性が相補的に働き、廃棄率、遅延、帯域などが優先度に応じて差別化される。

FQ は、優先度の高いフローに対して、多くの帯域を割り当てるため、結果として遅延も低減させる効果がある。しかし、この制御の前段で行うキューイングによって生じる遅延(キューイング遅延)の低減には寄与しない。この結果、フロー数が多いと最大遅延が大きくなることが知られている [40]。一般に、このキューイング遅延の低減には、FQ と組み合わせて用いられる AQM の廃棄制御が効果を与える。

既存の AQM、例えば、RED[4] や、これを応用した RIO[6]、WRED[7] 等では、キューイング前に平均キュー長に応じて確率的にパケットを廃棄した後に、単純な FIFO メモリにキューイングしている。このため、輻輳時に FIFO メモリ内のキュー長が長くなると、全てのクラスに属するパケットの平均キューイング遅延が増大してしまう。このため、特に、リアルタイム性が要求されるフローが、低廃棄率を要求するフローの影響を受け易くなり、結果的に遅延の増大を招くという問題がある。

さらに、FQ では、低遅延のフローには廃棄率を低くしか設定できない。また、AQM では逆に、低廃棄率のフローには遅延時間を長くしか設定できない。これらの制約は両方式の特性に起因するものであり、柔軟な優先度の設定が不可能である。

これに対して本章では、STP を応用した自律分散型優先キューイング制御方式が、個々のデータの一時記憶と優先制御機能を局所的に分散実装できることに着目する。すなわち、分散的な実装によって、キューメモリ内部でのキューイングを必要最小限に抑えられ、低遅延かつ機能的に各種の優先制御を実現できることを述べる。次節では、直線状の STP を折り返した可変長キューイング機構を構成することによって、パケット廃棄制御と優先出力制御を低遅延で差別化できる優先キューイング制御方式について詳述する。そして、この方式のマルコフ連鎖モデルを 2.3 節で定式化し、性能特性を解析可能であることを提示する。このモデルの解析結果、およびこのモデルと等価なシミュレータを用いた性能特性評価結果に

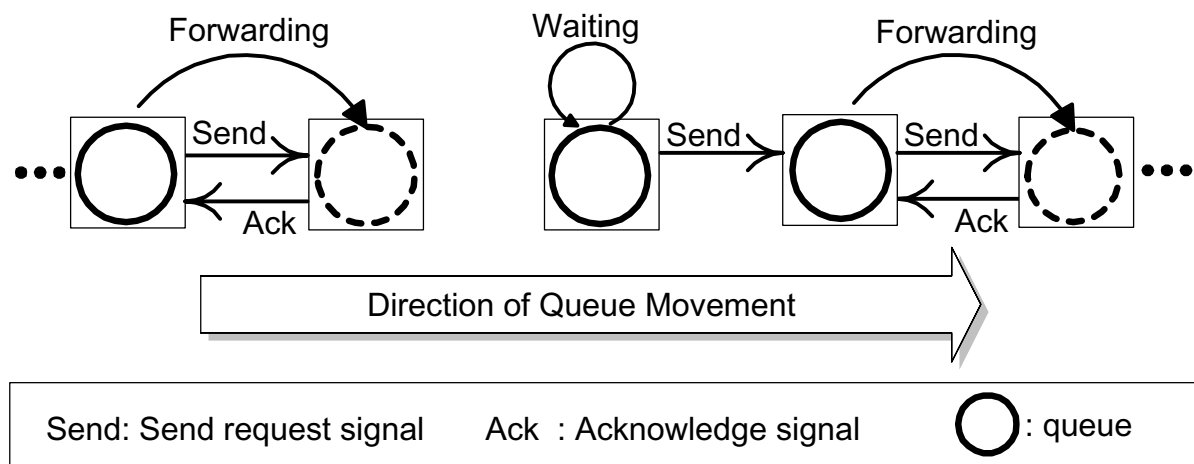


図 2.1 STP における自律動作

については 4 章で述べる。また，2.4 節では，自律動作に伴うパケットの順序逆転現象とその補償方式について述べ，この順序補償方式により，特性が改善できることを示す。

## 2.2 自律分散型優先キューイング制御方式

本節では，自己タイミング型パイプライン STP による優先キューイング制御方式の基本的な構成と原理的特性を述べた後，優先度に応じた廃棄制御機能および遅延制御機能をキューイング機構内部で自律分散的に実現する手法について考察する。

### 2.2.1 基本構成

直線状の STP 内では各パケットは図 2.1 ように自律的に移動する。すなわち，あるパケットがあるパイプライン段（ステージ）に到着すると，そのステージから，到着したパケットの移動方向のすぐ前方のステージに，送出要求信号（Send）を送る。その前方のステージが他のパケットを保持していなければ，Send を送出したステージに承認信号（Ack）を返す。パケットを保持したステージが Ack を受信すれば，パケットは Ack を返した前方のステージに移動する。このように，隣接するステージ間でのみ Send 信号/Ack 信号（ハンドシェイク信号）を授受して，パケットの保持または移動を行う。

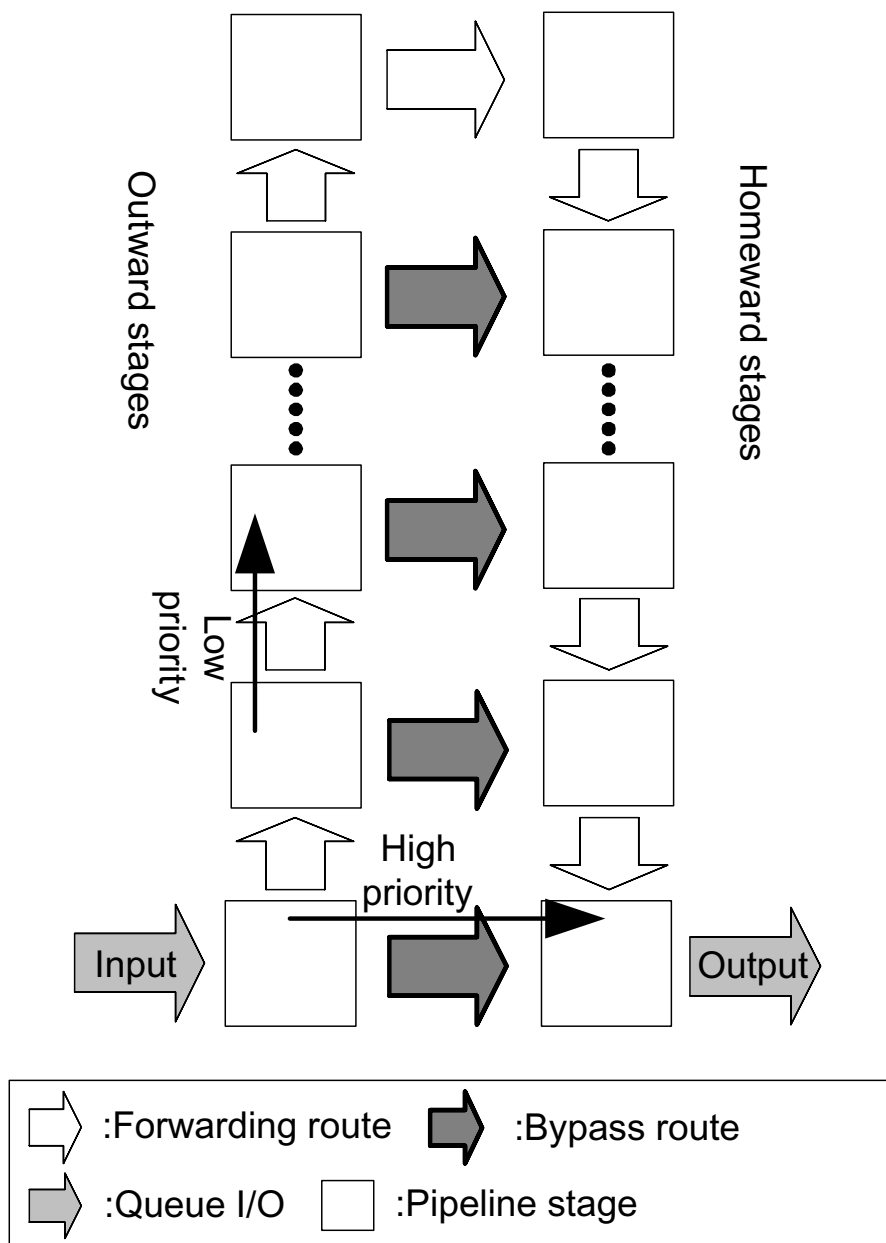


図 2.2 自律分散型優先キューイング制御方式の基本構成

したがって，STP には，局所的な信号伝達のみによって動作時にだけ電力を消費する省電力特性や，パケット流量の変動に対する自律緩衝能力（エラスティック性）等，クロック同期回路では実現不可能な高集積回路実現に適した特性があり，データ駆動型チップマルチプロセッサの実用化にも活用されている [33] 。

これらの特性をさらに応用したものが図 2.2 の STP による優先キューイング制御方式の

基本構成である．この構成は STP を半分に折り，折り返し点までの往路/復路の各段にバイパス経路を設けた可変長キュー構造を採る．このキューイング制御機構では，自律的な優先制御を実現するために，各パケットに自身の優先度情報<sup>\*1</sup>を付与している．往路パイプライン内に存在するパケットは原則として，まずバイパス経路への移動を試みる．復路側ステージにおいて，バイパス先のステージが塞がっている場合，またはバイパス先のステージへの移動が優先度の高いパケットと競合する場合は，往路パイプライン内の前方ステージへ移動する．

このような動作によって，キュー出口が輻輳していない場合は，入力パケットを最短遅延時間でキューから送出し，輻輳時には，パケットを徐々に往路パイプライン内にキューイングするエラスティックな仕組みが実現できる．さらに本基本構成では，バイパス先のステージへの移動を試みた時，復路側ステージに競合するパケットが存在する場合でも，往路側のパケットの優先度が高ければ，バイパス先に割り込み，先行できる．これによって，優先度が高いパケットはより短いキューイング遅延時間でキューから出力できることになる．

各ステージでパケット相互の優先度を比較してバイパス経路を通過すべきか否かを判断するための，ハンドシェイク手順を以下に示す．ただし，往路側ステージのパケットを  $P_o$  (優先度  $k_o$ )，復路側ステージのパケットを  $P_h$  (優先度  $k_h$ ) とする．

- 1) 移動先が競合する  $P_h$  が存在しない場合： $P_o$  はバイパス経路に移動する．
- 2) 移動先が競合する  $P_h$  が存在し，かつ  $k_o > k_h$  の場合 (図 2.3(a))：移動先ステージでは，双方のパケットから受信した送出要求信号 (Send) に対して， $P_o$  側に対して承認信号 (Ack) を返す．Ack を受信した  $P_o$  はバイパス経路に移動する．この時， $P_o$  は往路側パイプラインの前方段からも Ack を受信しているが，バイパス経路への移動を優先する．また， $P_h$  はこの間待機する．
- 3) 移動先の競合する  $P_h$  が存在し，かつ  $k_o \leq k_h$  の場合 (図 2.3(b))： $P_h$  に対して Ack を返し，先入れ先出しキューイングを行う．この時， $P_o$  は往路側パイプラインの前方段から Ack を受信すれば，当該前方段へ移動する．

---

<sup>\*1</sup> DiffServ の場合，IP ヘッダ中の DSCP ( DiffServ Code Point ) が用いられる．



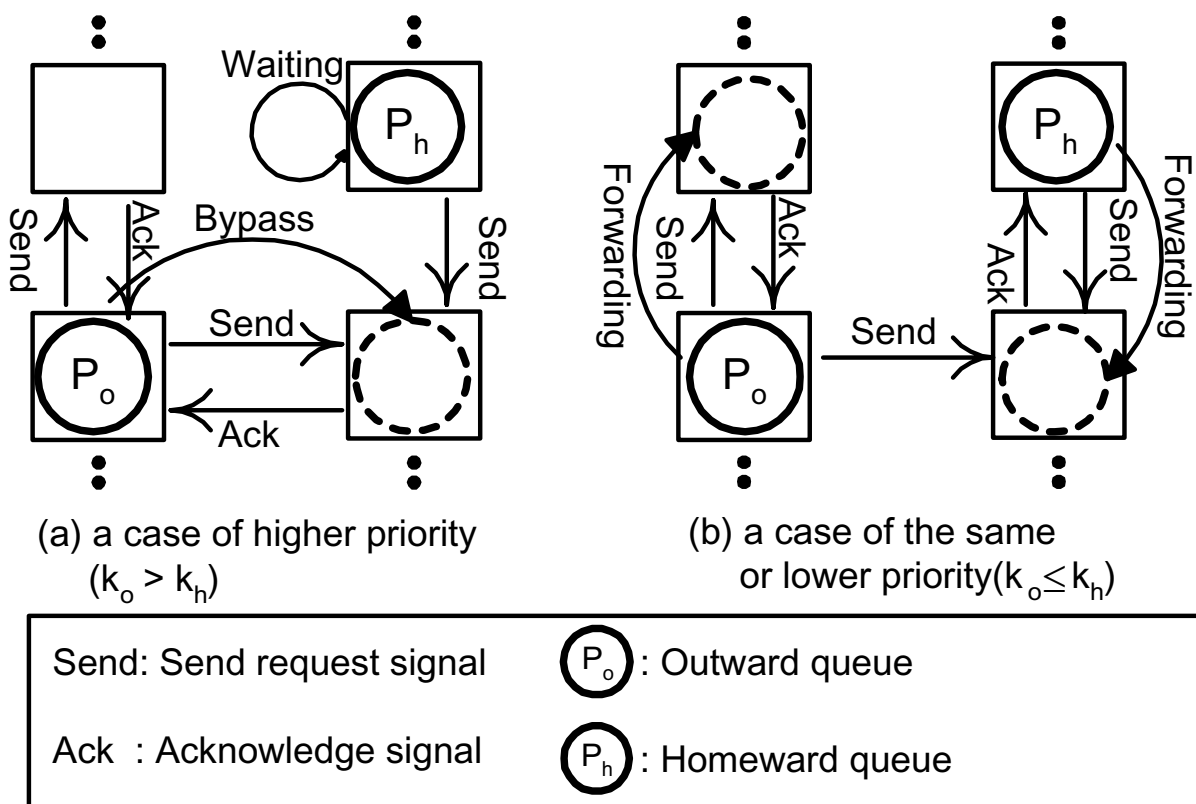


図 2.3 バイパス経路でのハンドシェイク手順

この自律分散型優先キューイング制御方式では，このような局所的な振る舞いの結果として，優先度の高いパケットが優先度の低いパケットを追い越せるため，単純な FIFO メモリを用いた従来の AQM に比べて，輻輳時のキューイング遅延を低減できる．

### 2.2.2 優先度に基づく廃棄制御

一般のキューイング制御では，大量のパケットをキューイングする時に，未然にバッファ溢れを防ぐための廃棄制御方式が導入されている．RIO や WRED は，さらに廃棄対象のパケットを差別化するために，廃棄するパケットの廃棄優先度に応じて，予め設計された廃棄確率や廃棄閾値を基にしてパケット廃棄の有無を決定している．自律分散型優先キューイング制御方式においても，廃棄率の差別化のために，次の二つの廃棄制御を導入する(図 2.4)．

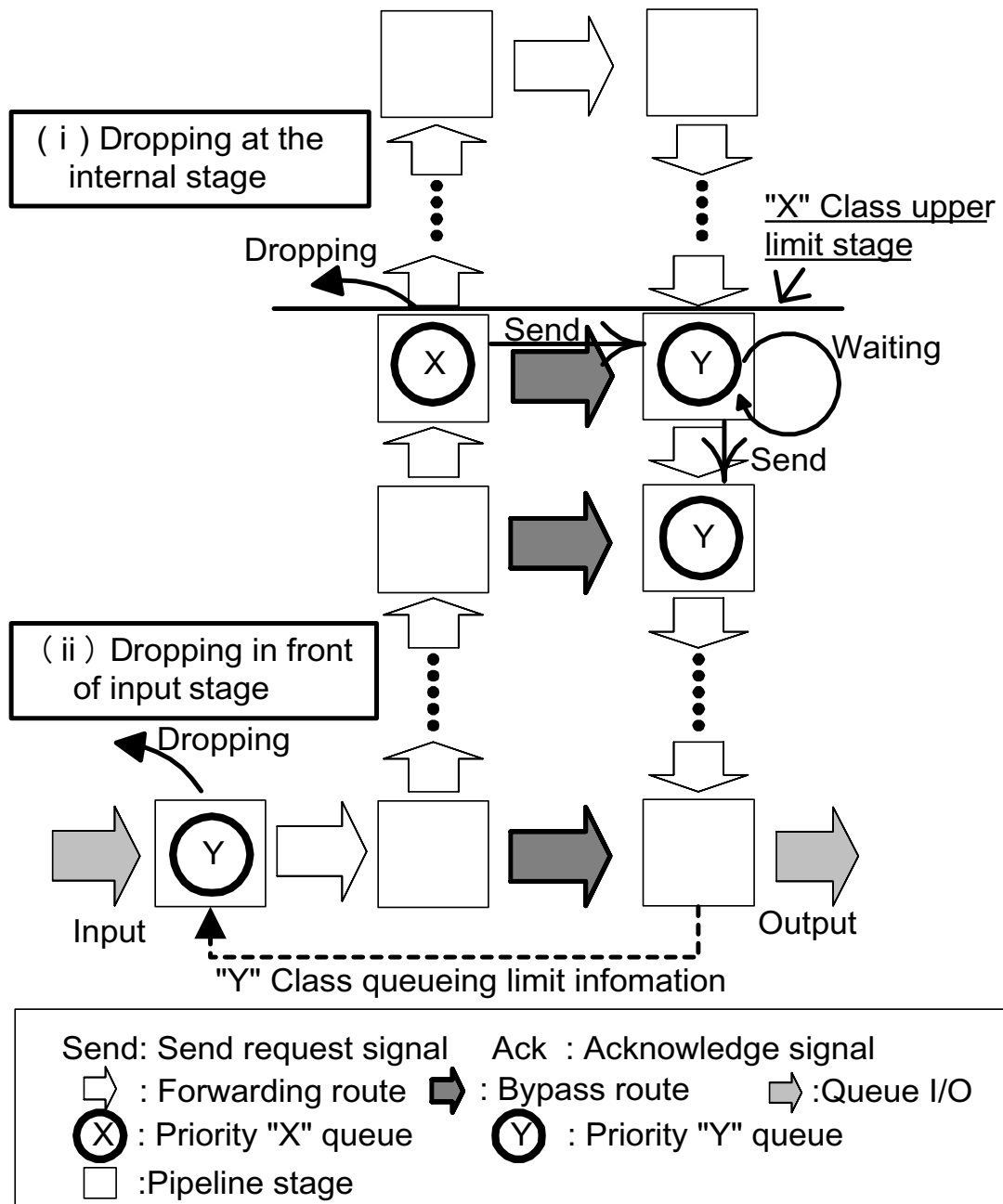


図 2.4 自律分散型優先キューイング制御方式の廃棄制御

- (i) 内部廃棄制御：キューイング機構内でパケットを廃棄する方式であり，優先度別にキューイング上限のステージを予め設定しておき，この上限に到達したパケットを全て廃棄する．この方式は，廃棄率とキューイング遅延時間を共に差別化できる効果があると考えられる．
- (ii) 事前廃棄制御：キューイング機構の入口で，廃棄優先度に応じてパケットを廃棄する．この時の廃棄率は，キューイング機構内の該当クラスのパケット数に基づき決定される．この方式は，キューイング機構が過負荷になることを未然に防ぎ，キューイング機構内の優先制御動作向上に効果があると考えられる．

### 2.2.3 優先度に基づく遅延制御

自律分散型優先キューイング制御方式では，クラス間の遅延特性をより明確に差別化するために，各パケットの優先度に応じてバイパス可能なステージ（以下，BS と呼ぶ）を制限する機能を，往路側の各パイプライン段に付加し，BS の範囲を複数段階に差別化できる．その基本構成を図 2.5(a) に示す．この構成は，BS の範囲を 3 段階に差別化して設定することによって，3 種類のクラスに属するパケットの平均キューイング遅延を明確に差別化することを意図している．ただし，BS の制限によって，キューイング機構内で選択できるバイパス経路が減少し，全体の遅延が増加するおそれがある．そのため，ここで次の 2 種類の手法を提案する．

- (iii) クラス別入力段分離法：図 2.5(b)

入力ステージを優先度によって分離し，入力後のパケットは基本構成の場合と同様に振舞う．この手法では，優先度に依存した入力ステージから上段のステージが，その優先度を持つクラス（優先クラス）の BS となるため，設定された BS まで到達する遅延を解消でき，かつ，遅延偏差の縮小化が期待できる．

- (iv) 優先クラスリマーク法：図 2.5(c)

低負荷時，すなわち，キュー内の総パケット数が閾値より少ない時のみ，入力パケット

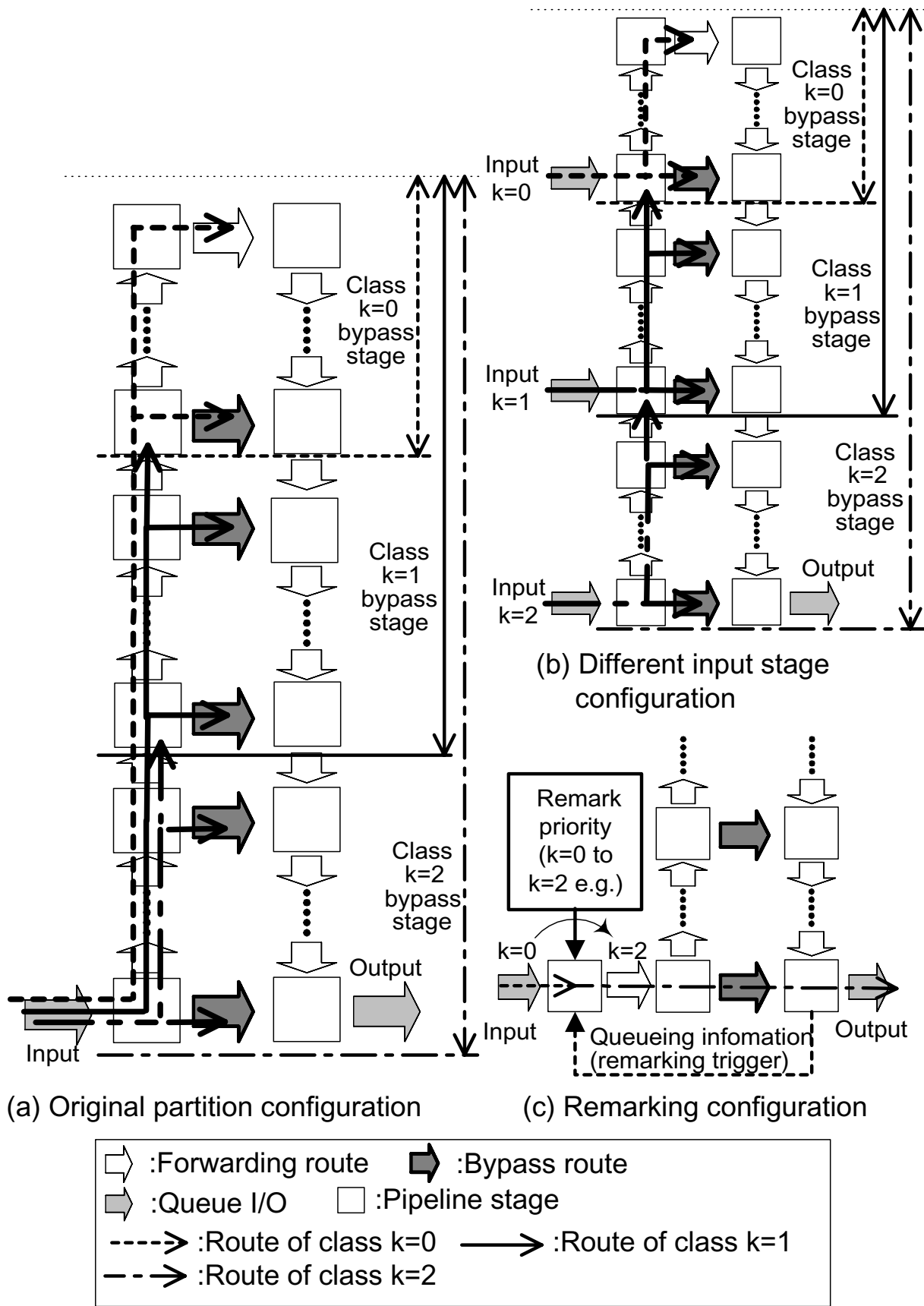


図 2.5 自律分散型優先キューイング制御方式の遅延制御

の優先度を見かけ上高く（リマーク）し，より短いキューイング経路が可能な BS 範囲が適用される．ただし，往路側の各ステージでは，バイパス先が競合する場合には元の優先度に基づくハンドシェイク手順により，バイパスまたは前方段への移動を決定する．この手法では低負荷時にはできる限り最短経路でキューイングするため，遅延時間の平均ならびに偏差の低減が期待できる．

以上のように，本優先キューイング制御方式は，乱数による確率的計算等を用いない，ごく簡単な優先転送/廃棄制御機能を STP の各ステージに付加することによって，廃棄率と遅延時間を共に低遅延で差別化することを意図した方式である．このため，DiffServ の AF クラス [41] のように相対的優先制御が要求される場合にでも，本節で述べた制御手法の組み合わせで差別化できることが期待できる．

## 2.3 マルコフ連鎖による解析モデル

本節では，自律分散型優先キューイング制御方式の性能特性解析を目的としてマルコフ連鎖によるモデルを定式化する．本方式における折り返し型 STP の各ステージで非同期的に発生する事象（動作）は，起こりうる状態の確率の推移によってモデル化が可能である [39]．具体的には，2.2.1 節に述べた Send 信号および Ack 信号の伝搬時間を 1 : 1 とし，パケットの移動時間が 1 単位時間で行われると仮定すると，図 2.6 に示すように各ステージでのパケットの存在確率が離散時間で推移するモデルによって表せる．

ここで，折り返し型パイプラインの入出力ステージを基準として，キューイングの深さ  $i$  のステージを，往路側が第  $[i, 0]$  段，復路側は第  $[i, 1]$  段とする． $k (\geq 0)$  はパケットの優先度を示し，値が大きいほど低遅延が要求される． $t$  は経過時間， $\lambda_i(k, t)$  は第  $[i, 0]$  段の入力パケットの到着確率， $\mu(k, t)$  は出力パケットの出力確率を示す．パケットが第  $[i, 0]$  段から，第  $[i, 1]$  段へバイパスする確率は  $p_i(k, t)$ ，前方段へ移動する確率は  $r_{i,i+1}(k, t)$  である．また，2.2.2 項 (i) の内部廃棄制御を適用し，上限の第  $[i, 0]$  段にてパケットは  $D_i(k, t)$  の確率で廃棄される．このように，パケットの廃棄確率を設定したステージを廃棄ステージと呼

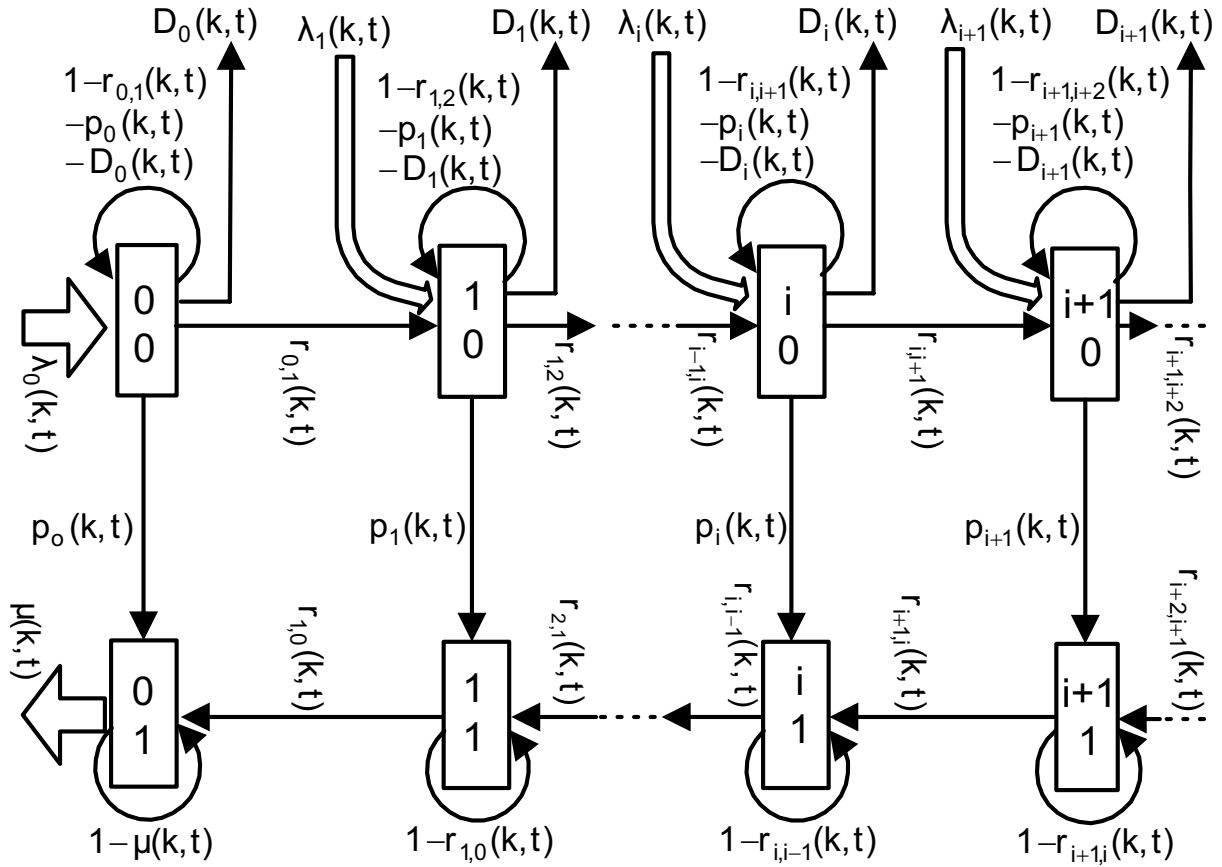


図 2.6 自己タイミング型キューイング方式の状態遷移

ぶ．復路側の第  $[i, 1]$  段においても，同様に前方段に移動する確率は  $r_{i,i-1}(k, t)$  である．

さらに， $k$  クラスの packets が時刻  $t$  に第  $[i, 0]$  段に存在する確率を  $ON_{i,0}(k, t)$ ，キューイングされている packet 数を  $q(k, t)$ ，2.2.2 項 (ii) の事前廃棄制御による廃棄確率を  $D_A(k, q(k, t))$  とする．これらを用いて各確率を定式化すると，第  $[i, 0]$  段での，packet の存在確率  $ON_{i,0}(k, t)$  の関係式は次のようになる．

$$\begin{aligned}
 ON_{i,0}(k, t) &= \lambda_i(k, t) \times \{1 - D_A(k, q(k, t))\} \times \{1 - r_{i-1,i}(k, t) \times ON_{i-1,0}(k, t-1)\} \\
 &\quad + r_{i-1,i}(k, t) \times ON_{i-1,0}(k, t-1) \\
 &\quad + \{1 - r_{i,i+1}(k, t) - p_i(k, t) - D_i(k, t)\} \times ON_{i,0}(k, t-1) \quad (2.1)
 \end{aligned}$$

ここで，

$$D_i(k, t) = \begin{cases} 1 - p_i(k, t) & ([i, 0] \text{ 段が } k \text{ クラスの廃棄ステージの時}) \\ 0 & (\text{それ以外の時}) \end{cases}$$

また,  $p_i(k, t)$  は, ハンドシェイク手順の動作から次式で表される.  $N$  は  $k$  の最大値である.

$$p_i(k, t) = \left\{ 1 - \sum_{l=k}^N ON_{i+1,1}(l, t-1) \right\} \times ON_{i,0}(k, t-1) \\ \times \left[ \sum_{k=0}^N r_{i,i-1}(k, t-1) + \left\{ 1 - \sum_{k=0}^N ON_{i,1}(k, t-1) \right\} \right] \quad (2.2)$$

ここで, キューからの出力が  $B(t)$  の確率でブロックされるとすると  $\mu(k, t)$  は次式で示される.

$$\mu(k, t) = (1 - B(t)) \times ON_{0,1}(k, t-1) \quad (2.3)$$

ただし, 式 (2.1) ~ (2.2) は, 入出力ステージと最上段ステージの場合を除く, 中間ステージを対象とした式である\*2. 全ての式は  $t \geq 1$  で成り立ち,  $t = 0$  時の各確率は全て 0 である.

$\lambda_i(k, t)$  を与えて, これらの式を  $\mu(k, t)$  について解くことにより, 各クラスのパケットの遅延や廃棄率を求める. この式を代数的に解くことは非常に困難なため, 代入計算を繰り返すプログラムを用いてこの式を解析した. 解析の過程で各パケットの移動経過を記録すれば, 遅延時間を算出可能である. また, 平均した  $\lambda_i(k, t)$  と  $\mu(k, t)$  の比から廃棄率を求められる. 解析結果については, 特性評価や既存方式との比較評価と共に, 4 章において述べる.

## 2.4 順序補償方式

2.2, 2.3 節において, 自己タイミングパイプライン STP の折り返し構成を用いて, 特別なスケジューリング機構を用いなくとも, 効果的に遅延や廃棄の差別化を行うことができる自律分散型優先キューイング制御方式を提示した. この優先キューイング制御方式は, 各パイプライン段(ステージ)において必要な制御を局所的に集約できるため, 制御機能の単純化と構成の多様性において優れた特性を持つ. しかし, 一方で自由度の高い自律動作のた

\*2 入出力ステージから最上段のステージまで含めた, 折り返し型 STP 全体を表す一般化した状態の確率については, 付録 A に示す.

め、場合によっては、同一優先度のパケットがキューイング中に順序が入れ替わるリスクを有する。本節では、このリスクを解消する順序補償方式について以下に記す。

### 2.4.1 順序逆転現象

図 2.7 は、本優先キューイング機構へ入力されたパケットが、キューイング中に順序を入れ替えられるケースを示している。ここでは、A、B、C の 3 クラス（優先度： $A > B > C$ ）の中で、中優先度の B クラスに注目する。先に入力された B2 パケットが、バイパスできずにキューイングされ、上段のステージに移動し、後から入力された同一クラスの B3 パケットが、バイパス先の競合する C クラスのパケットとの優先度比較の結果、先にバイパスする様子を示している。この結果、B2 パケットと B3 パケットの出力順序が入れ替わり、B クラスのフローに順序逆転を生じる。アプリケーションによっては、このような順序逆転をエンドホストで適切に補正できず、パケットの再送や廃棄のために性能が劣化することもある。次項にこの順序逆転現象を補償する方式について提示する。

### 2.4.2 自律分散型アーキテクチャにおける順序補償方式

自律分散型制御方式では、集中制御方式のようにシステム全体の状態を把握する中央制御部を持たない。このため、各機能部は、直近の情報より判断し、動作する必要がある。本項では、パケットの入力時に優先クラス別にシーケンス番号を付与し、この番号をステージへ記録、またはステージから参照することにより、パケットの動作を決定し、システム全体として順序補償される、自律分散型の順序補償方式 [44] を提示する。

#### 優先クラス別メモリとバイパスフラグ

本キューイング機構に入力されたパケットは、パケットの入力順序を記録するため、優先クラス別にシーケンス番号が与えられる。復路側ステージでは、ある優先度のパケットが到着すると、ステージに設置された優先クラス別メモリ上の、当該優先度のメモリエリアに



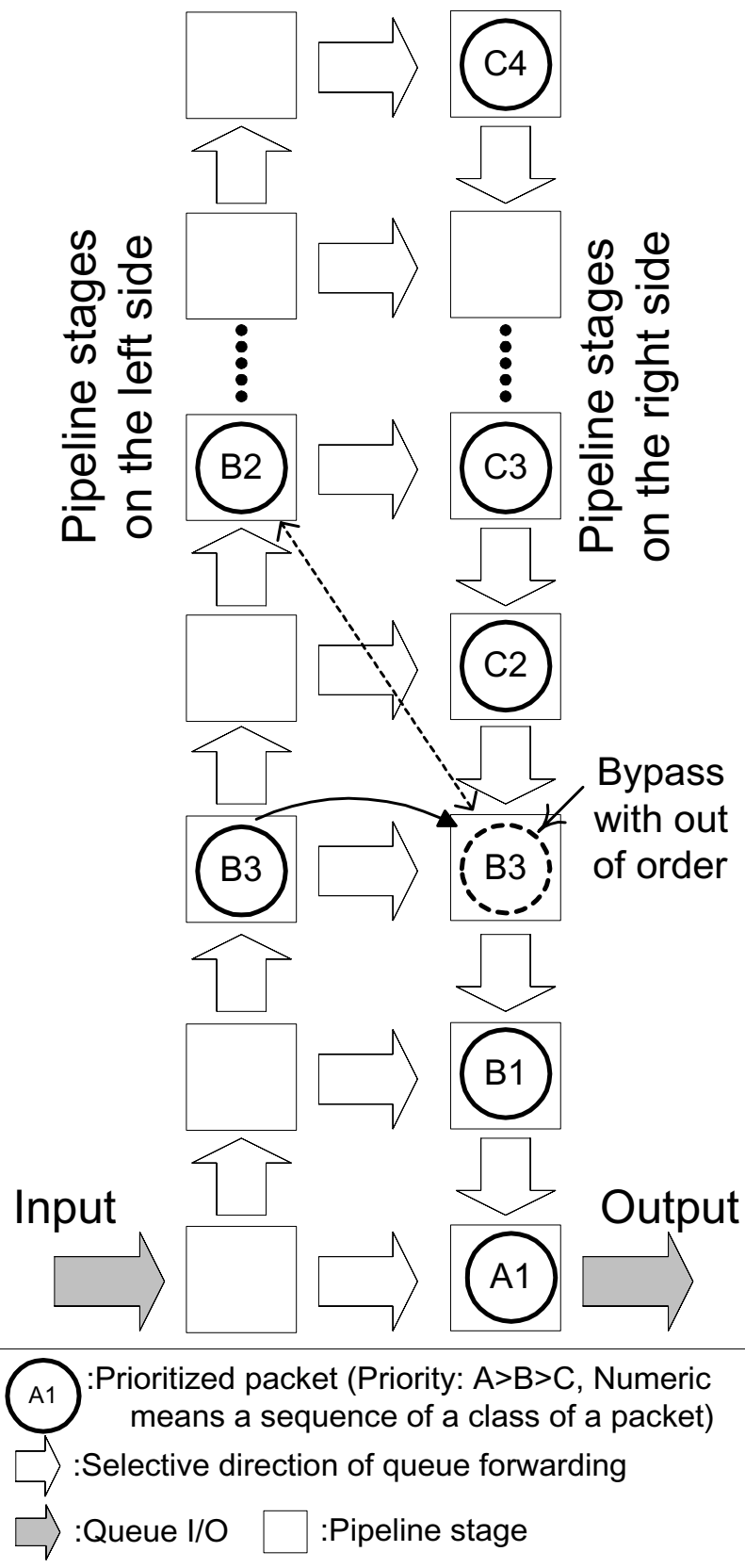


図 2.7 パケットの順序逆転現象

シーケンス番号を記録していく。一方，往路側ステージの packets は，自身のシーケンス番号を，復路側ステージの優先クラス別メモリ上に記録された当該優先度のシーケンス番号と比較し，連番であるか（差が 1 であるか）を判断し，連番であれば順序補償可と判断し，バイパスフラグを ON（有効）にする。

図 2.8 に，この優先クラス別メモリとバイパスフラグの設定の様子を示している。

- (a) B クラスの優先クラス別メモリと B3 packets のシーケンス番号の差が 1 でないことから，バイパスフラグは OFF（無効）とする。packets は，バイパスされずに前方段にキューイングされる。
- (b) 往路側ステージに到着した B2 packets にその優先クラス別メモリが 2 に更新され，下方段から移動してきた B3 packets のシーケンス番号と，優先クラス別メモリのシーケンス番号の差が 1 となる。よって，バイパスフラグが有効となる。

バイパスフラグが有効となった B3 packets は，これ以後，シーケンス番号の比較が不要となり，図 2.3 の優先制御の判断のみによって，バイパスかキューイングの動作を行う。

### バイパス動作

バイパスフラグの操作は，バイパスフラグが有効となるまで，優先して行われる。バイパスフラグが有効となるまで，packets は優先度にかかわらずバイパスできない。バイパスフラグが有効となった packets は，バイパス先のステージが空いており，移動が競合しないか，あるいは，競合しても競合 packets の優先度が自身より低ければバイパスする。図 2.9 は，バイパスフラグの有効・無効の違いによる packets のバイパス動作の様子を示している。

- (a) バイパスフラグが有効な場合の動作
  - (a-1) バイパス先への移動が競合する Cy packets より優先度が高いため，バイパスする。
  - (a-2) バイパスと同時に優先クラス別メモリの値を自身のシーケンス番号で更新する。

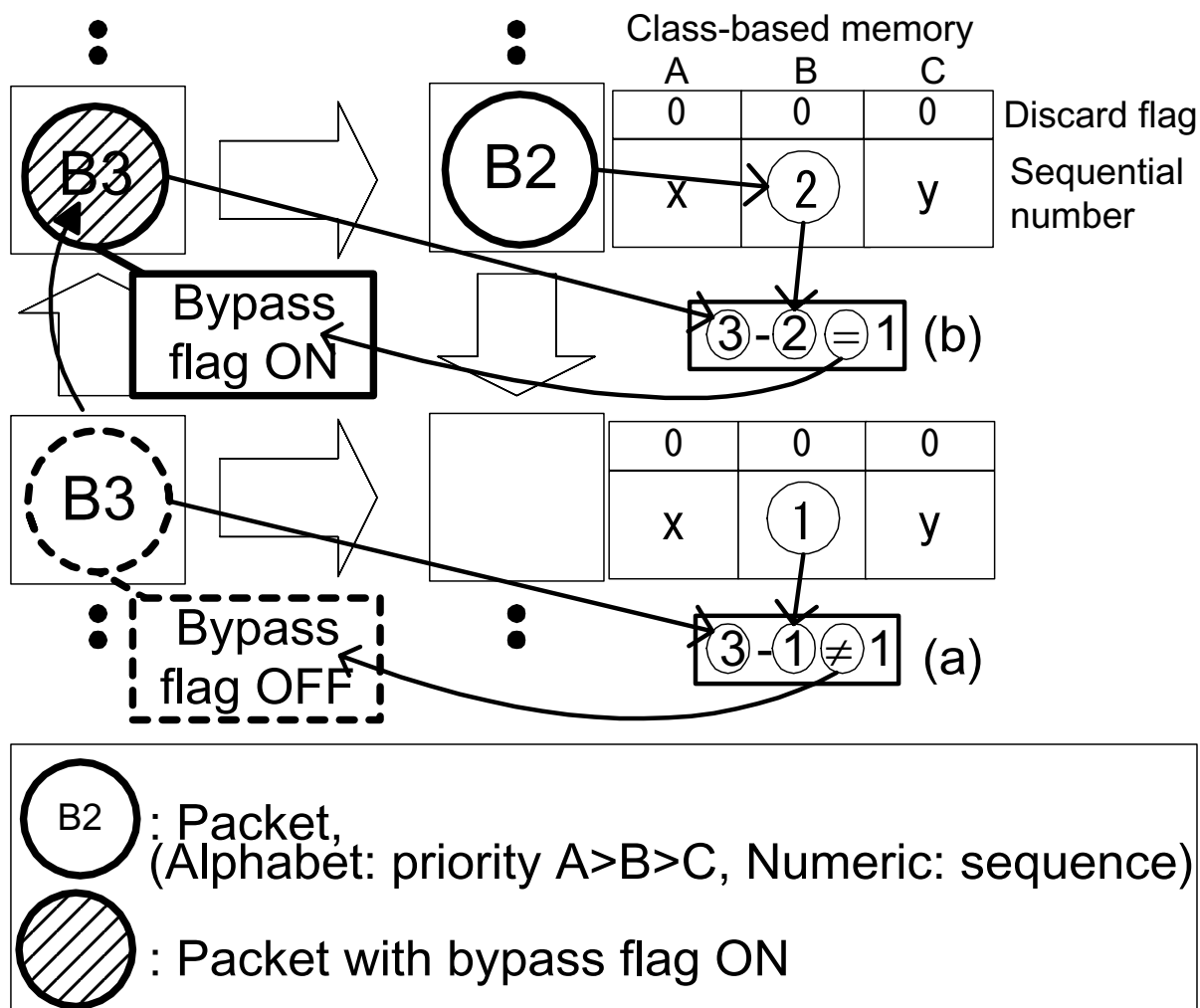
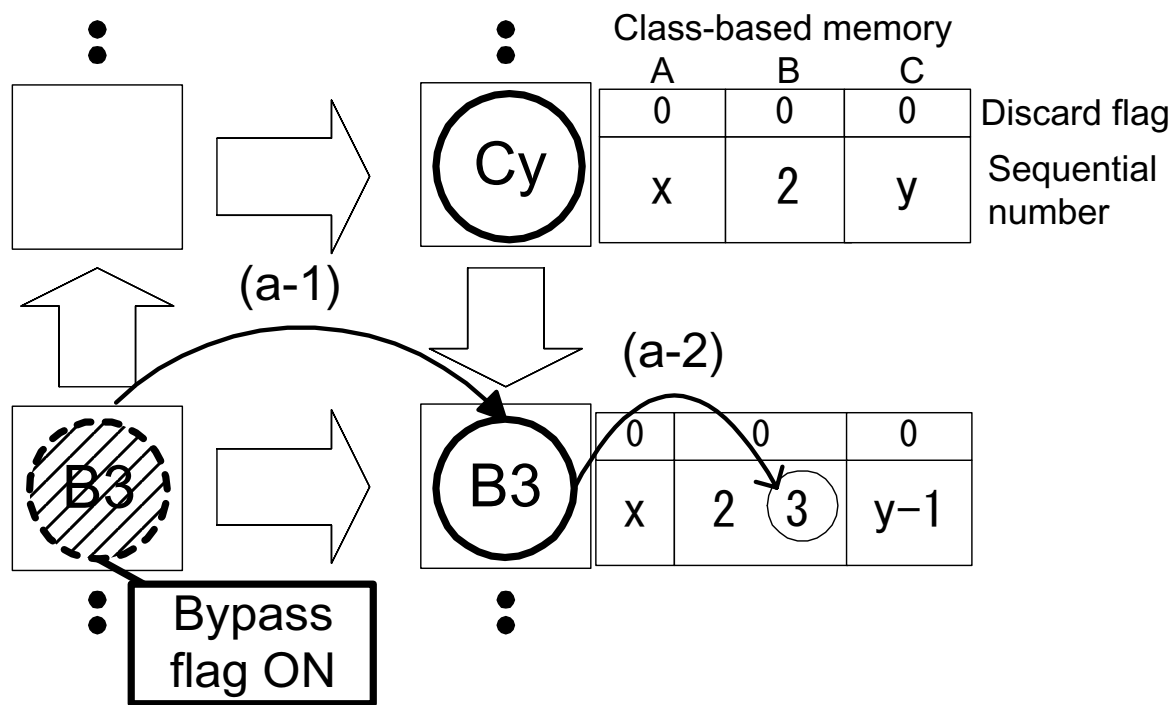


図 2.8 優先クラス別メモリとバイパスフラグ

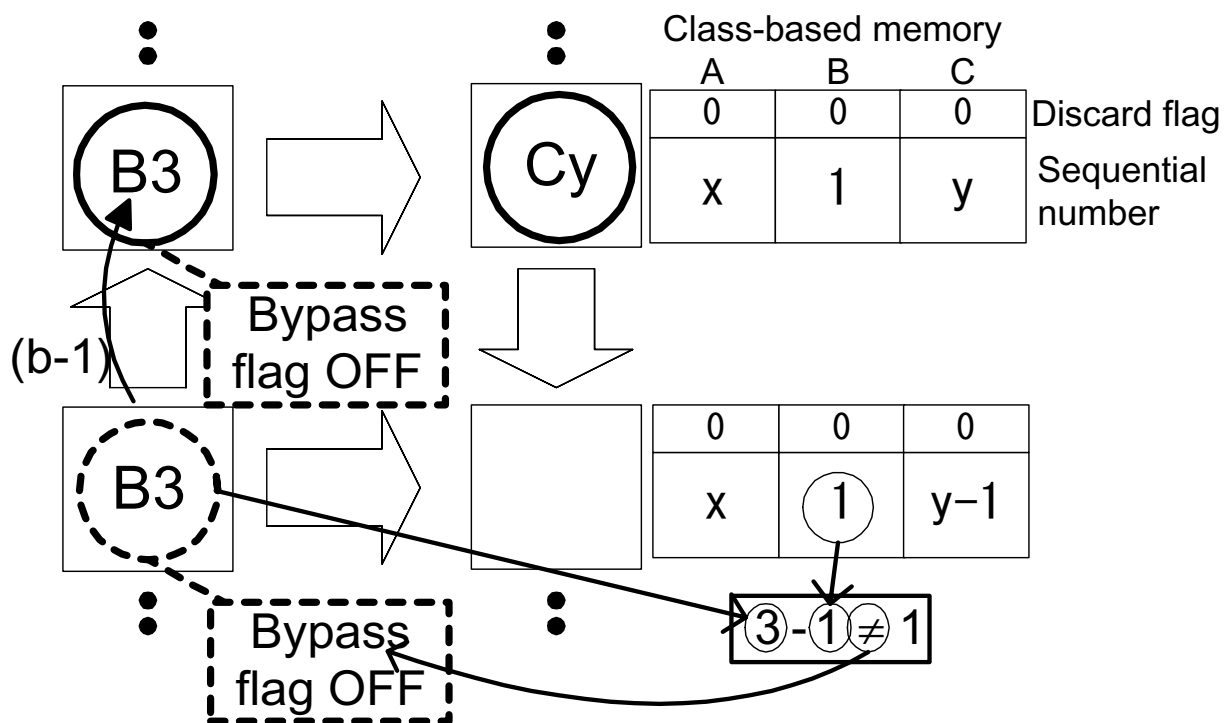
(b) バイパスフラグが無効な場合の動作

(b-1) バイパス先への移動が競合する  $C_y$  パケットより優先度が高くても，前方段にキューイングされる．この動作は，競合パケットが無くとも同様である．

このように，バイパスフラグを活用することにより，順序補償を考慮したバイパス動作を簡易に制御できる．



(a) A case of bypass flag ON



(b) A case of bypass flag OFF

図 2.9 バイパスフラグに基づくバイパス動作

## パケット廃棄時の制御

本キューイング機構の廃棄ステージにてパケットが廃棄された場合、パケットのシーケンス番号に欠番が生じる。順序補償制御をする際、これを放置するとバイパスフラグの設定に不具合を招く。このため、パケットが廃棄された場合、速やかに廃棄パケットのシーケンス番号を記録する必要がある。さらに、廃棄パケットと同じ優先度で、すぐ次のシーケンス番号を持つパケットは順序逆転しないため、速やかにバイパス可能とすることが要求される。このため、廃棄パケットのシーケンス番号をできる限り早く下段に伝え、バイパスフラグの有効化を進める必要がある。本章の順序補償制御では、この情報を、下段に移動する他のクラスのパケットにピギーバックさせて伝える方法を採用。図 2.10 にこの廃棄時の優先クラス別メモリの操作制御を示す。

### (a) 廃棄ステージでパケットが廃棄された場合

(a-1) 廃棄されたパケットのシーケンス番号を、すぐ右側往路ステージの優先クラス別メモリの当該優先クラスに記録する。

(a-2) さらに、当該優先クラスの廃棄フラグを有効(1)にする。

### (b) 廃棄フラグが有効となっている往路側ステージで、あるパケットが前方ステージに移動する場合

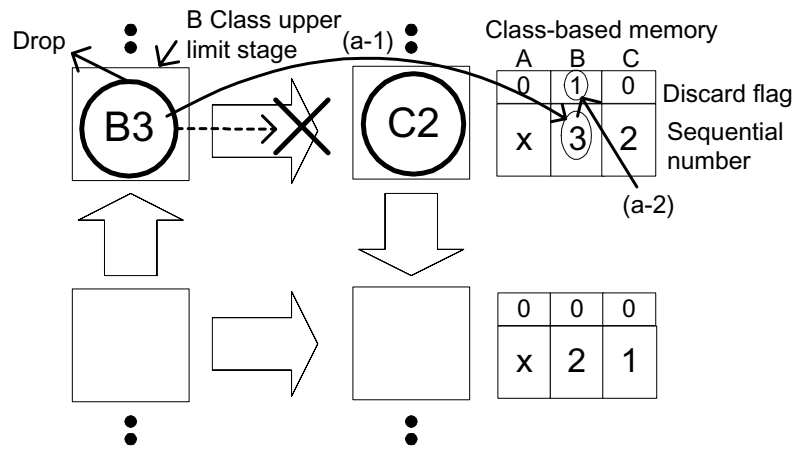
(b-1) 前方ステージの優先クラス別メモリの中のそのパケットの優先クラスに自身のシーケンス番号を記録する。

(b-2) 廃棄フラグが有効となっていた優先クラスの番号もその前方段に記録する。

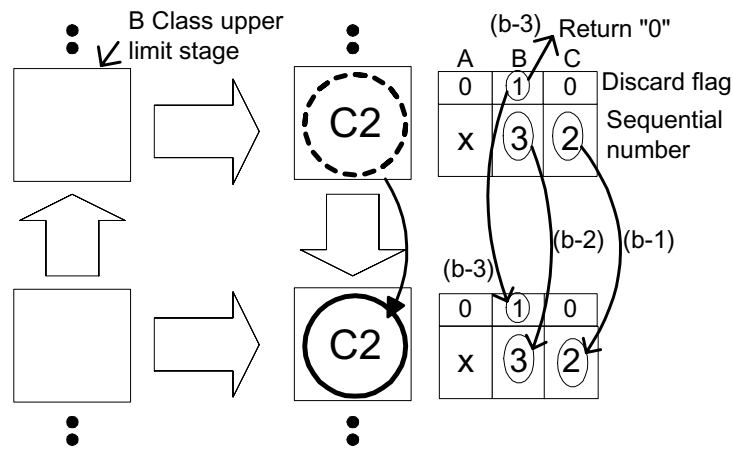
(b-3) かつ、廃棄フラグを有効にする。この際、移動元のステージの廃棄フラグは無効(0)状態に戻す。

### (c) 廃棄フラグが無効である往路側ステージで、あるパケットが前方ステージに移動する場合

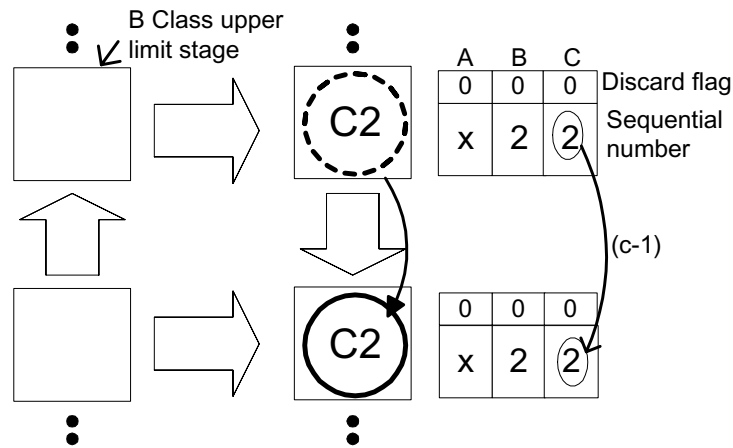
(c-1) 前方ステージの優先クラス別メモリの中のそのパケットの優先クラスだけに自身のシーケンス番号を記録する。



(a) A case of dropping a packet



(b) A case of discard flag ON



(c) A case of discard flag OFF

図 2.10 パケット廃棄時の制御

この一連の操作により，廃棄されたパケットの情報が速やかに下段に伝わり，廃棄パケットに続いて入力されるパケットが無駄にキューイングされることを防ぐことができる．

### 優先クラス別メモリのクリア

本キューイング機構の入力時に，パケットに与えられた優先クラス別のシーケンス番号は，パケットのフレームやキューイング機構のステージのメモリの大きさが限られていることから，有限値の循環番号となる．このため，番号が循環した時，パケットのシーケンス番号や優先クラス別メモリ上に記録した番号は，循環前と循環後のものが混在することとなり，情報を誤認識して，バイパスフラグの設定に不具合を生じることがある．二巡以前の番号が残らないことが確実であれば，巡回前の番号を誤認識しないように制御することが可能である．しかし，シーケンス番号の循環周期やトラフィックの変動状況によっては，キューイング機構の上段部に数巡前に記録された番号が残るケースも十分考えられる．このため，このような状況が起こらないように，シーケンス番号が循環した時に，キューイング機構内のステージに記録された番号を順次クリアしていくことが必要となる．

図 2.11 に，ステージの記録されたシーケンス番号をクリアするリセットパケットの動作について示す．

- (a) シーケンス番号の循環したパケット ( B0 ) がバイパスし，優先クラス別メモリの該当優先クラスを自身のシーケンス番号で更新する．
- (b) シーケンス番号の循環したパケットがバイパスした時，同一の優先クラスとシーケンス番号を持つリセットパケットを生成し，これを往路側の前方段へ送る．
- (c) リセットパケットが到着したステージでは，復路側ステージのパケット存在の有無に拘わらず，優先クラス別メモリの該当優先クラスを，このシーケンス番号で更新 ( リセット ) する．
- (d) リセットパケットは最上段まで順次移動し，(c) の操作を繰り返す．
- (e) 最上段に到着したりセットパケットは，(c) の操作を行うとともに，廃棄される．

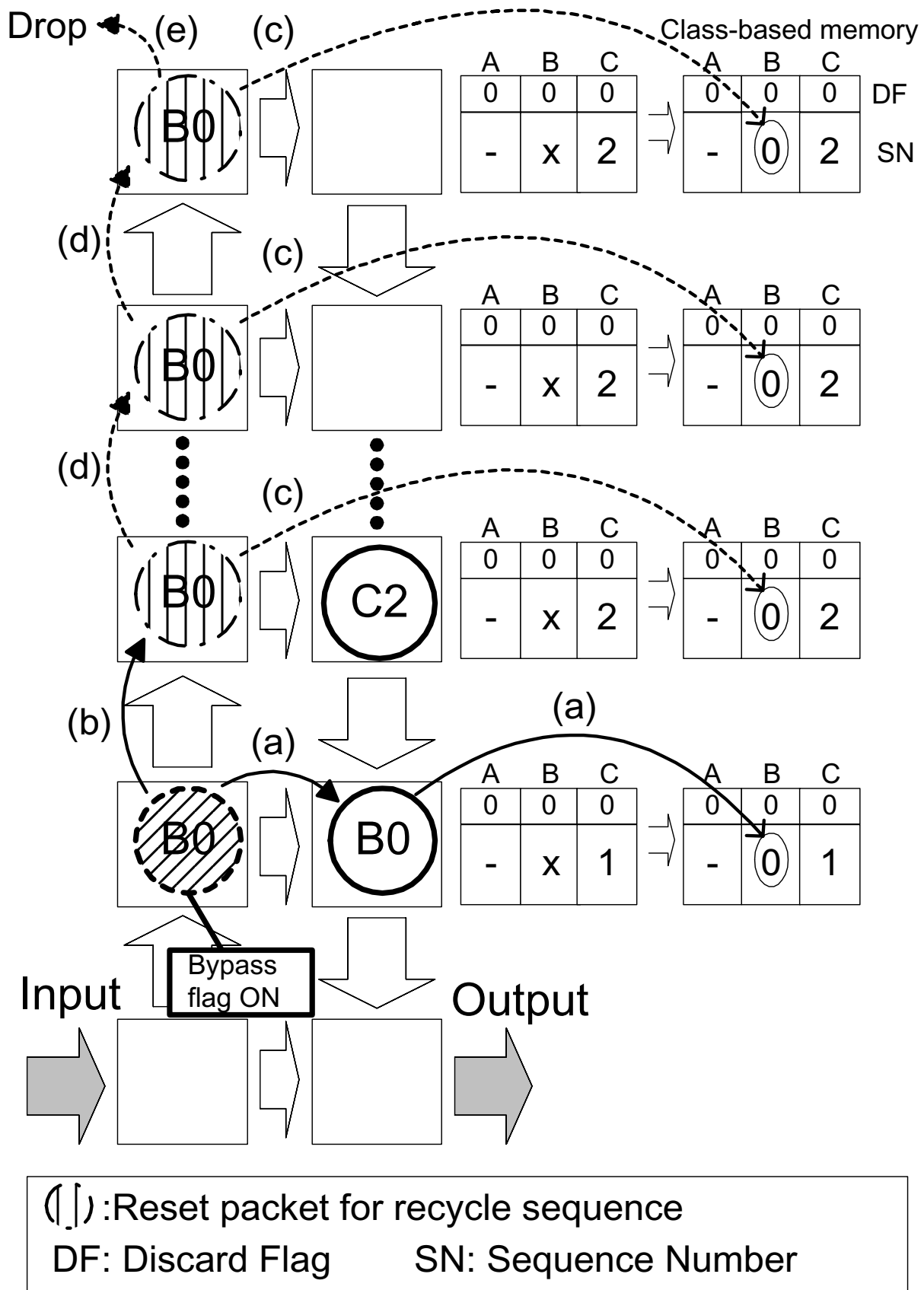


図 2.11 優先クラス別メモリクリア用のリセットパケット



シーケンス番号の循環したパケットが廃棄された場合も，リセットパケットが生成され，同様にシーケンス番号のリセットが行われる．

これらの一連の処理によって，パケットのシーケンス番号が循環した場合に，循環前のシーケンス番号の記録が残されることなく，バイパスフラグ設定の判断を正常に行うことができる．

## 2.5 まとめ

本章では，高機能 LSI モジュールの実現に適した，自律分散型制御アーキテクチャに基づく自己タイミング型パイプライン機構 STP を応用して，DiffServ 向きの優先キューイング制御方式が実現できることを明らかにした．また，この優先キューイング制御方式が，パケットの優先度に応じて，廃棄率およびキューイング遅延時間を同時に差別化することができることを示し，さらに，制御機能が分散化したアーキテクチャを活用することにより，従来のキューイング制御方式には不可能な遅延制御や廃棄制御をキューイング中に行え，これらの制御の組み合わせにより多種多様なキューイング特性を表現できる方式であることを示した．また，この優先キューイング制御方式を，マルコフ連鎖モデルに基づく解析可能な数学モデルで表せることを示し，これらの特性を原理的に明らかにできることを示した．この数学モデルを活用することにより，本優先キューイング制御方式の発展や，STP を応用する制御メカニズムの解析に役立つことができると考える．

STP は，その制御方式の局所性の恩恵を受け，従来の同期型パイプラインに比べ，大規模かつ多様な構成のパイプライン形成を可能にしている．このため，STP を利用したこの自律分散型優先キューイング方式は，理論的には無限に大きなキューイング構造を採ることができる．しかも，パケットの自律動作に基づいた優先キューイング制御方式であるため，複雑な優先度に基づくスケジュール操作を行うことなく，無数の優先クラスを同時に取り扱うことが可能である．既存のキューイング制御方式のように，キューイング構造が大規模化し，取り扱う優先クラスやフロー数の増加に伴い，全ての優先クラスの平均遅延が全般的に

増加する問題についても，優先度に応じてパケットがバイパスして出力可能な方式であるため，根本的に解消されている．局所的な自律動作のために，キューイング中に同一優先度のパケットの順序が入れ替わるリスクについても，提案する順序補償方式によって解決できることを示した．

これらのことから，このキューイング制御方式は，ネットワークの高速大容量化に伴い，顕在化するキューイング遅延に伴う問題を解消し，多様化の進むサービスそれぞれに対して，適切に QoS を提供できる有用な技術であると考えられる．

関連研究では，自律分散型優先キューイング制御方式の実現可能性の検証のために， $0.18\mu\text{m}$  の 6 層配線 CMOS 技術による LSI チップを試作している [42, 43]．この LSI チップは基本構成のみではあるが，約 100 M パケット/秒程度のキューイング速度を達成しており，この優先キューイング制御方式が 10 G bit/s クラスのリンク速度に対応できる見通しを得ている．

本章では，自律分散型優先キューイング制御方式の基礎的な平均キューイング特性を高めるための，廃棄制御法，遅延制御法，順序補償方式を示した．今後の課題として，TCP の各種フロー制御との親和性の評価についても考案中であり，本方式の，より広範な適用性を明らかにしたい．



## 第 3 章

# 自律分散型制御アーキテクチャに 基づくフロー分散制御方式

### 3.1 まえがき

近年，波長多重・光増幅技術の発展によって，信号の誤りや損失，遅延が極めて少ない高品質な高速光伝送が可能になり，あらゆる形式の情報を自由に疎通できる情報通信ネットワークの実現が可能になりつつある [45]．

しかし，昨今，このような高速光リンクを収容可能な高速ルータやスイッチ（ノード）を新規にネットワークに導入しても，その性能を十分に発揮できない問題が顕在化している．これは，新設ノードと既設ノードの性能差のため，パケットロスや処理遅延の変動が生じ，これらの既設ノードがネットワークの隘路になるからである．具体的には，TCP/IP の場合，既設ノードの能力を超える広告ウィンドウによってフローが制御されると，送信側でのパケット廃棄の検出を通してのみ，経路上のボトルネックを間接的に検出することになる．このため，新旧ノード間の性能差が大きい場合，パケット廃棄検出，再送，ウィンドウサイズの縮小を何度も繰り返し，これがフローの遅延や揺らぎを増加させる原因となる．また，フロー制御を行わない UDP/IP の場合，パケットが大量に低速ノードに流入し，最悪の場合，低速ノードがフリーズする可能性もあり，ネットワークの信頼性を大きく損ねてしまう．

このため，通常，トラヒックの増加傾向やその分布傾向を予測して，ネットワーク上に隘路が生じないように，複数の新規高速ノードが設備される．しかし，今日そして今後将来のインターネットについて考えた場合，インターネット接続事業者内では計画的にインフラ整

備を行えても、このネットワークの性格上、複数の事業者を含む統合的なネットワークに対して、計画的に新設ノードが設備されるとは考えにくい。また、トラフィックの分布状況は日々変動するため、これらの変動に応じて設備拡張を緻密に計画することも困難である。このため、今日そして今後将来を考えても、インターネットでは、ノード間の性能差を直接的に解消することが原理的に不可能であり、ネットワーク内の不均質格差は解消できないと考えざるを得ない。よって、このような性能差によるパケットロスや遅延変動を可能な限り低減するフロー制御方式が必要になる。

さらに、仮に新旧ノード間で適切にフロー制御ができたとしても、既設ノードがネットワーク内で隘路にならないように、トラフィックを負荷分散することが必須になる。フローを複数経路に負荷分散して送出し、経路当りの負荷を下げる方式も多数提案されている[22, 23]。しかし、パケット単位あるいはパケットを細分化したセル単位で負荷分散する方式はスケジューリングやパケット順序補正を要するため、高速な専用ハードウェアが必要になり非効率的である。簡易な実装を目的として、フロー単位で分散する経路を選択する方式[24]も提案されているが、フローの総量が大きいと特定のノードが高負荷になる場合があり、細粒度での負荷分散が難しい。

そこで本章では、より柔軟で信頼性の高いネットワークを構築するために、新旧ノード間の性能差に伴うパケットロスや遅延変動を低減できるフロー制御を行うと同時に、フローをパケット単位で複数経路に自律的に負荷分散することによって、ネットワーク内のスループットを向上を目的とした、自律分散型制御アーキテクチャに基づくフロー分散制御方式について述べる。

## 3.2 複数経路フロー分散・集約方式

### 3.2.1 要件

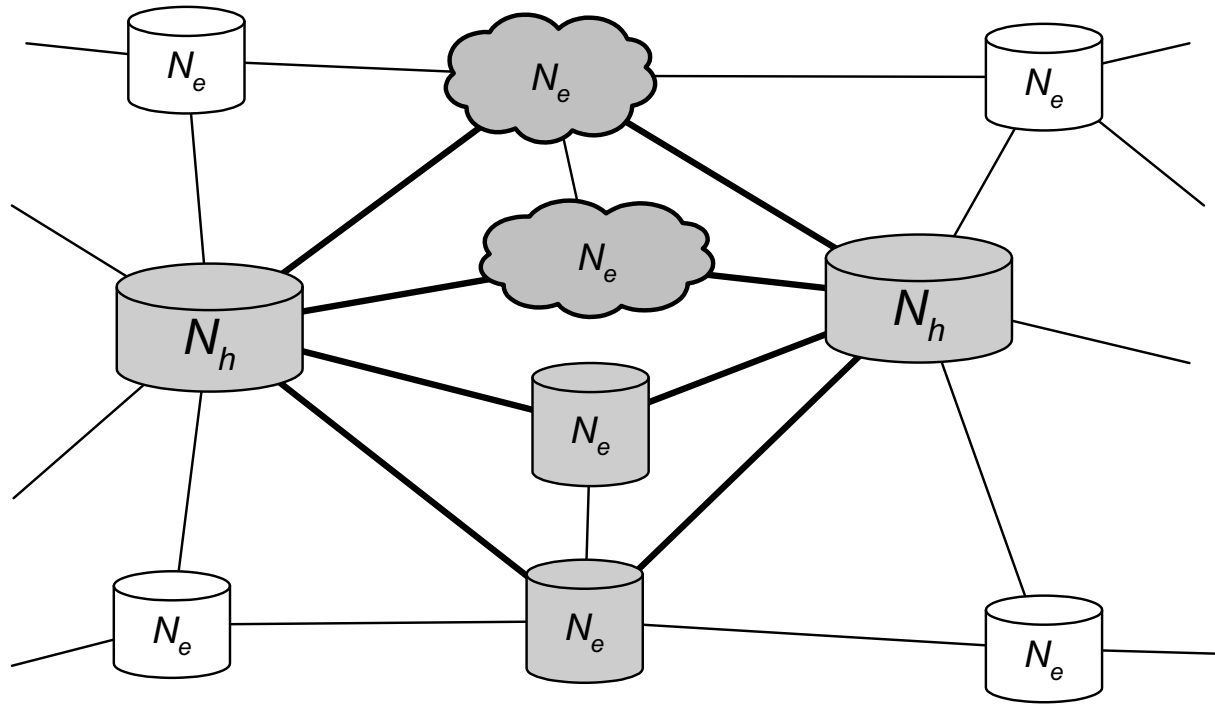
高速ノード<sup>\*1</sup>を新規に導入した場合に，ボトルネックが生じるネットワーク構成の典型として，図 3.1 に示すような，高速ノード  $N_h$  間にそれよりも劣る性能の複数の既設中継ノード（ないしは既設中継ネットワーク） $N_e$  が接続される例を考える．高速ノード間に単一経路しか設定されない場合には，TCP/UDP パケットいずれの場合でも，前述したように，パケット廃棄や処理遅延の変動が増大し，高速ノードの転送性能を活用できない．たとえ，OSPF[46] 等の経路選択アルゴリズムや IPv6 の経路制御ヘッダを用いて，複数経路のルーティングが選択されても，現状のノード装置には動的に変動するトラフィックに対して効果的に負荷分散する機能が実装されていないため，ネットワーク全体のスループット向上は困難である．

このようなネットワークでは，経路上の中継ノード  $N_e$  のパケット転送能力，経路の伝送遅延は一般に異なっており，かつ，動的に変動する．このような複数の経路上にフローを効果的に負荷分散するために，以下の点に着目している．

- 既存の負荷分散方式 [22, 23, 24] のような各  $N_e$  への負荷の均一化ではなく，各  $N_e$  に少しでも過度な負荷を与えないようにして，各  $N_e$  の余剰処理能力を適応的に有効活用してフローを分配する．ただし，分配する経路に高速リンクで接続する  $N_h$  を含む場合，負荷分散せず当該  $N_h$  のみに中継する．
- パケット単位またはパケットを細分化したセル単位で細粒度の負荷分散制御を行い，総合的なスループットを向上する．
- 複数の高速ノード  $N_h$  が同一の中継ノード  $N_e$  を負荷分散経路として共有する場合で

---

<sup>\*1</sup> ノードにおけるパケット転送処理は通常の場合パケットヘッダのみが処理対象となる．このためノードの性能は，単位時間当たり転送可能なパケット数 PPS ( packets per second ) で規定される．一般的には，最小サイズの IP パケットの転送能力が，そのノードが収容可能な伝送リンク速度の総和以下の場合も多く見受けられる．



$N_h$  : Higher-performance node  
 $N_e$  : Existing node or subnetwork

図 3.1 トラヒック負荷分散を必要とするネットワーク構成例

も、適応的に負荷分散が可能なこと。

- エンドホスト間でパケットの順序逆転を完全に解消するのは一般に困難な問題とされている [47] ため、集約側で順序補正を行う。ただし、複数経路に分散されたパケットを集約側（受信側）高速ノードで受信した時点でパケットの順序がなるべく正しくなる方式を導入して、受信側でのパケット並び替えに伴う遅延時間を短縮する。
- 負荷分散・集約処理に要する遅延時間（中継経路での転送遅延を除外した時間）を経路に依存せずに均一にし、パケット順序逆転を低減する。

### 3.2.2 フロー分散方式

本章で述べるフロー分散制御方式は、図 3.2 に示すように、負荷分散先の選択に要する処理の並列化のために、当該経路に送出可能かを適応的に判断する制御部 AC (Adaptive flow

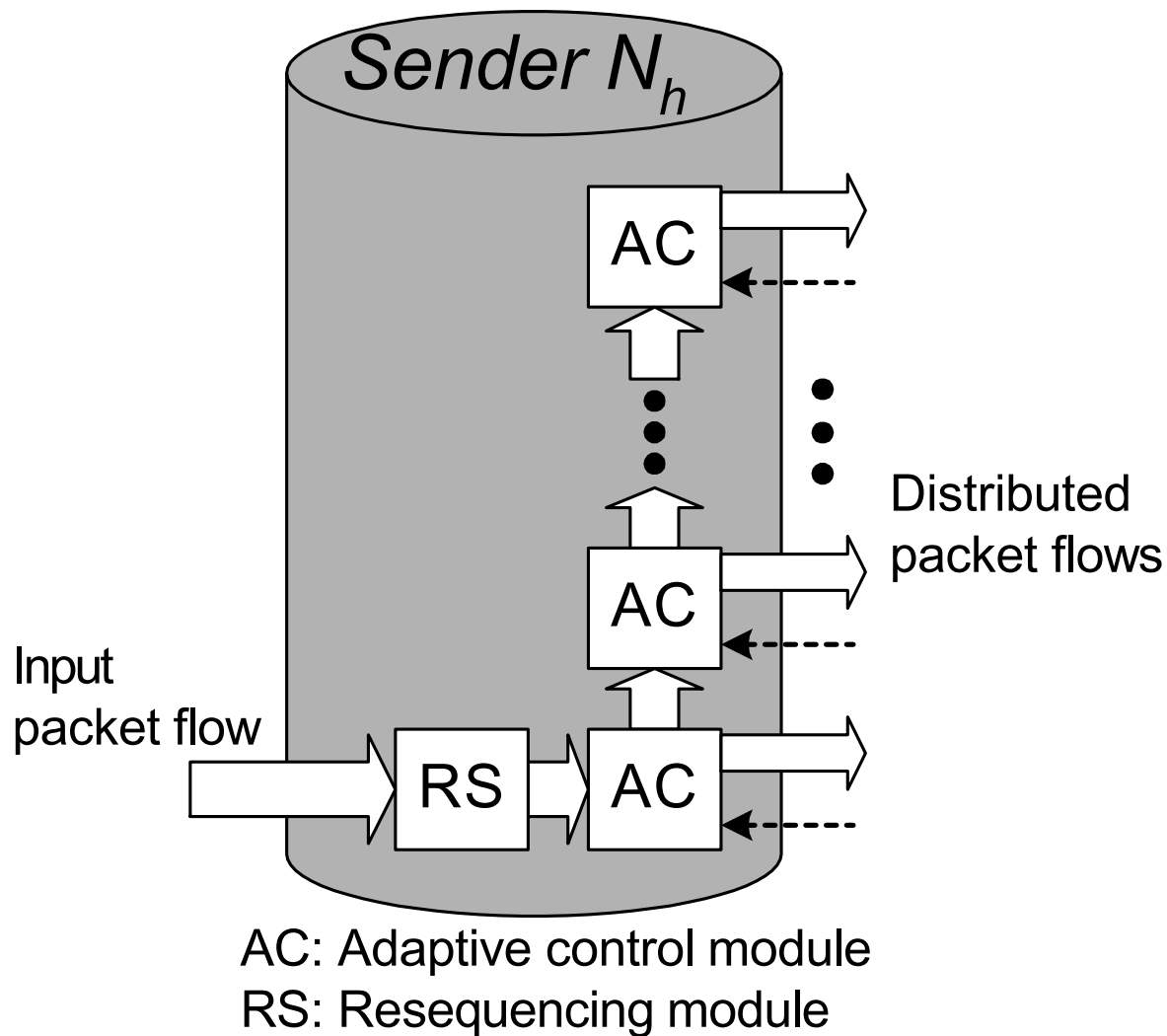


図 3.2 高速送信側での自律型フロー分散方式

control module)を各経路毎に用意し、以下のようにパイプライン並列に実現が可能である。

- (1) 送信番号付与部 RS：高速ノードから送出されるパケットはまず、受信側での順序保証のために、送信  $N_h$  ID および送信番号を付与される。送信番号の記録は、IPv4 パケットではヘッダオプションの未使用部（予約クラス）の利用、また IPv6 パケットでは拡張ヘッダの宛先オプションの活用が考えられる。
- (2) 適応的フロー制御部 AC：送信  $N_h$  ID と番号が付与されたパケットは最初に、最下部の AC 部に入力され、当該ポートに送出可能かどうかを接続先  $N_e$  の負荷状況に基づいて判断される。個別リンクのフロー制御については次節に詳述する。



- (2-1) 送出可能と判断した場合，当該ポートよりパケットが送出される．
- (2-2) 当該ポートへは送出すべきでない（接続先  $N_e$  が過負荷である）と判断した場合，パケットは隣接する上部の AC 部へ送られ，再度そのポートへ送出可能かどうか判断される．同時に，後続のパケットは並行して，上記（2-1）の操作対象となる．
- (2-3) 各パケットは，送出可能なポートに到達するまで上部の AC 部へ送られる．

この一連の動作の結果，低速ノードの転送能力に制限されることなく，高速ノードの送出フローは並列に分散されて転送される．すなわち，本方式によって，低速ノードの転送能力に依存せず，代替経路がある限りは高速にフローを転送できる．

### 3.2.3 フロー集約方式

複数の経路を経由して到着したパケットを元のフローに集約する際，到着間隔の揺らぎや到着順序逆転のリスクがある．これらの問題は不均質なネットワーク上で，しかも自律的にパケット単位で負荷分散する場合，可能性を零にはできない．このリスクを可能な限り少なくするには，負荷分散・集約処理に要する遅延時間  $T_{da}$ （中継経路での転送遅延を除外した時間）が経路に依存せず等しいことが望ましい．これは，中継経路の転送遅延時間が一般には既知ではなく，経路ごとに  $T_{da}$  を差別化しても一般的な性能向上には寄与しないためである．

送信側高速ノードでは基本的に下部のポートから順にパケットを送出していくため，受信側ノードにおいても下部のポートから順にパケットを受信して整列する方式を採用し， $T_{da}$  を均一にしている．しかし，必ずしも下部のポートから送出順にパケットを受信できるとは限らない．このため，受信バッファを環状型に接続して，複数経路から到着した受信パケットを集約すると同時に，順序逆転したパケットのみ再び巡回して再整列する，図 3.3 に示す環状型受信バッファ機構を導入した．

この受信バッファの出力部に順序補正用の検査機構を設け，順序カウンタに従って次に送出すべきパケットが判定する．順序逆転し，先に出力部に到着したパケットは再度受信バッ

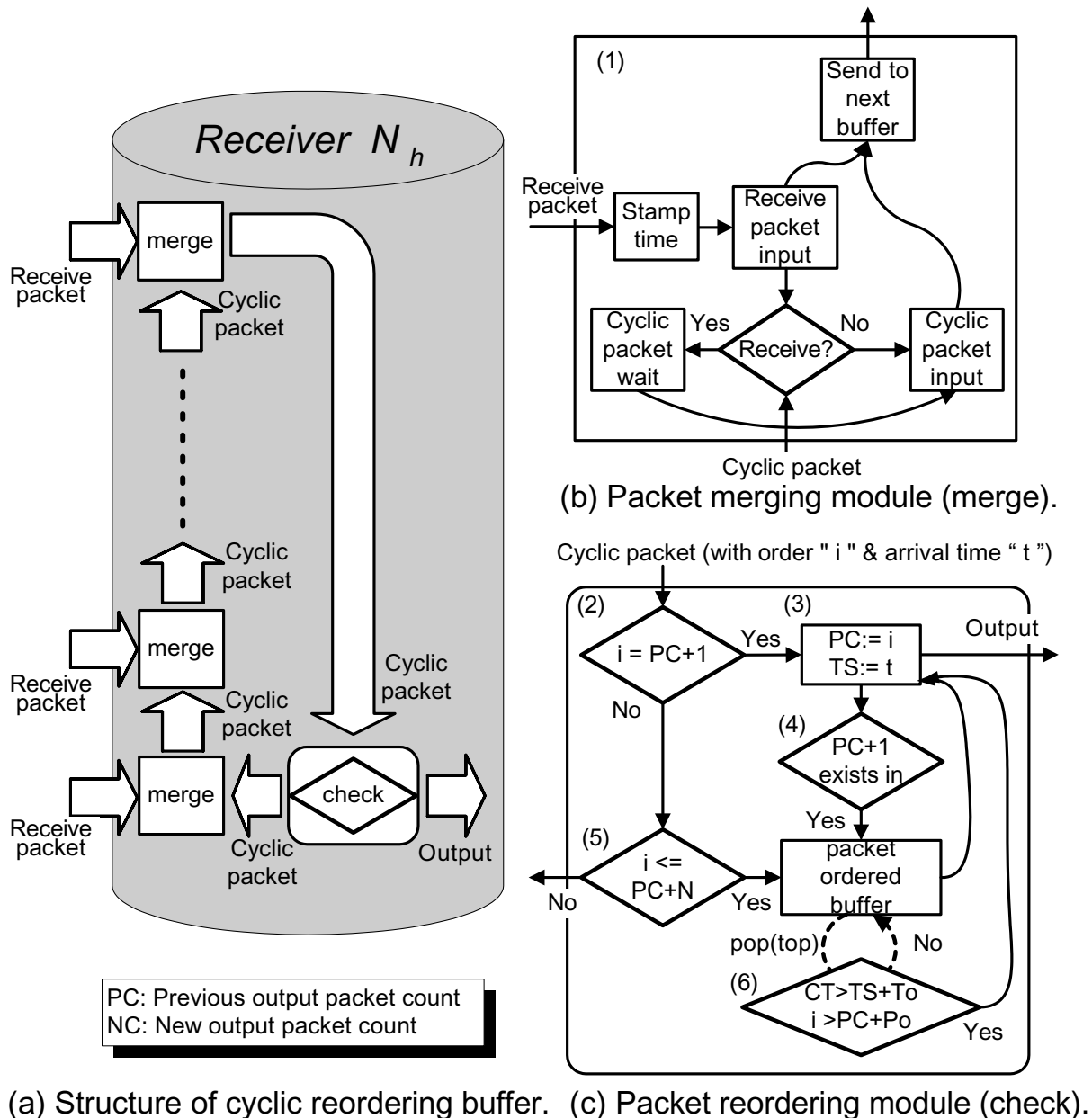


図 3.3 パケット順序補正が可能な環状型受信バッファ機構

ファを巡回する。ただし，出力すべきパケット順が僅差のものは，わずかなタイミングで順序逆転したことも考えられ，引き続き出力すべきパケットが到着する可能性が高い。このようなパケットを再度巡回させると，フローをこの巡回制御の一巡分遅延させることになる。したがって，出力すべきパケットの順序に近い順序逆転パケットは巡回させず，出力部のバッファにおいて待機させ，無用な遅延を避けている。この小容量の出力待ち合わせ用バッ

ファを組み合わせれば，効率的なパケットの整列とこれに伴う遅延時間の短縮が同時に実現される．

この環状型受信バッファ機構の動作を図 3.3(b)(c) のフローチャートに従って以下に説明する．

- (1) 環状型受信バッファ機構に接続する複数経路から負荷分散されたパケットが到着する．この時，順序補正待ちのため巡回するパケットと到着パケットが同一合流部 (merge) に同時合流する場合，到着パケットを優先させ，巡回パケットを待たせる．また，各パケットには受信時に時刻印  $t$  を付与する．
- (2) パケットが巡回して検査部 (check) に到着すると，その送信  $N_h$  ID と送信番号  $i$  を，同一の送信  $N_h$  ID から受信したパケットの中で直前に出力されたパケットの送信番号  $PC$  と比較し， $PC$  より 1 だけ大きい場合，すなわち連番であればそのパケットを出力可能と判断する．以降の手順では，送信  $N_h$  ID 毎に各変数を管理する．
- (3) パケット出力時に  $PC$  の値を  $i$  に更新する．さらに，最新出力パケットの受信時刻  $TS$  を  $t$  に更新する． $TS$  は手順 (6) のタイムアウト処理で利用する．
- (4) さらに検査部はバッファ内に保留していたパケットの送信番号を検索し，新しい  $PC$  より 1 だけ大きい番号を持つパケットがバッファ内に存在するか検査する．もし存在すれば，引き続いてそのパケットも出力するために，手順 (3)(4) を繰り返し，該当するパケットが存在する限りバッファを検索する．
- (5) 検査部に到着したパケットの送信番号  $i$  が  $PC$  と比較して 2 以上大きい場合，バッファに保留する範囲 (バッファレンジ:  $N$ ) の僅差の順序逆転であるかどうかを判断し，該当するものはバッファに保留し，そうでないものは再度環状バッファを巡回させる．
- (6) 分散経路上でのパケットロス及び受信  $N_h$  を迂回するパケットに対応するためにタイムアウト処理を行う．すなわち，パケットバッファ中で最小の送信番号を持つパケットを周期的に参照し， $CT > TS + T_o$ ，かつ， $i < PC + P_o$  であれば，タイムアウトと判定し，そのパケットおよび後続の連番パケットを出力するために，手順 (3)(4) を繰り返す．

す．ここで， $CT$  は現在時刻， $To$  はタイムアウト時間， $Po$  はタイムアウト時に強制的に出力すべきパケットの送信番号の差である．

以上のフロー分散・集約方式によって，遅延の大きな経路から小さな経路へとスケジューリングすることで，スケジューリング遅延（空いている AC 部に到着するまでの遅延）と経路間の差を相殺することが可能になる．同時に，遅延の小さな経路を通過したパケットが先に受信側に到着する可能性が高くなり，順序補正に要する遅延時間も短縮できる．したがって，多様な性能のノードが混在するネットワークにおけるパケット単位の負荷分散方式の課題である，パケットの到着間隔の揺らぎや順序逆転の問題が軽減される．

さらに，フロー分散機構ならびに集約機構共にパイプライン並列に実現可能な方式を採用したため，各 AC 部，合流部，検査部は接続リンク速度で動作すれば十分である．従来の負荷分散方式 [22, 23] では，送信側では全ての経路の負荷状況を集約してスケジューリングし，さらに，受信側でも全ての経路からパケットを受けて順序補正する必要があったため，経路数に比例した負荷状況集約速度，順序補正速度が要求される．これに対して，本方式の要求性能は経路数に依存しないため，比較的低速なハードウェアで容易に実現でき，さらに，経路数の増加にも容易に対応できるスケーラブルな方式であると言える．

### 3.3 個別リンクのフロー制御方式

前節に述べたフロー分散・集約方式を効果的に実現するには，送信側の AC 部で実現すべき個別リンクのフロー制御を，負荷分散を前提とした方式にする必要がある．隣接ノードの情報をフィードバックしてリンク毎のフローを制御する方式として，昨今，AQM (Active Queue Management) が注目されている [48]．AQM は，Drop-tail 方式や，この方式を発展させた廃棄方式を活用して，隣接する送信先の輻輳状況の程度をフィードバックしてフローを制御する方式である．フィードバック情報として，残存キュー量，到着レート，接続経路の容量等を用い，これらを基に様々な閾値関数を計算する方式が提案されている．しかし，これらはいずれも，単一リンクないしはそれを含む経路のみを対象として，スループッ

ト向上，遅延時間短縮，およびパケット廃棄率低減を目指しており，複数経路による負荷分散を前提にはしていない．

負荷分散を前提にした場合，隣接する送信先の入力キューにパケットを滞留させるよりも，分散先が存在する限りは，そのパケットを他の経路に分散するほうが総合的な性能が向上する．キューに滞留するとそのノードの処理時間と滞留したパケット数に乗じた値に比例してフローの遅延時間が増加するためである．したがって，本章で述べるフロー分散制御方式で採用する，個別リンクのフロー制御方式では，低速の中継ノードにパケットをできる限りキューイングしないように，送出レートを制御する方式を導入した．

このフロー制御方式は，中継ノードからのリプライ情報（リプライ間隔やキューの情報）によって，送信側の高速ノードのフロー送出を自律的に抑制する．これによって，送出側の高速ノードのバーストフローを中継ノードの転送能力のレベルに速やかに抑える．このリプライパケットには，ICMP パケット [49] の Information Reply ( type 16:未使用 ) を利用するなど，既存のプロトコルの活用が想定可能である．

本フロー制御方式では，リプライ情報から求められるキュー量の増減から，接続先ノードの転送能力を推定して，これにパケットの送出間隔 SI を調整している．しかし，リプライパケットを受信できない，初期のパケット送出の場合には，転送能力が未知であるため，中継ノードのキューが溢れる恐れがある．この問題を解消するために，送信ノードと中継ノード間で送受する信号シーケンスを図 3.4 のように規定している．

初期制御：まず最初に，リンク先の中継ノードの転送能力を以下のように推定して，SI の初期値を設定する．

- (1) 送出側の高速ノードは続けて  $M$  個 ( $M \geq 2$ ) のパケットを同じポートへ送出し，2 つ以上のリプライパケットを受ける．
- (2) 各リプライパケットの到着間隔  $RI$  を中継ノードの転送能力を反映したフィードバック情報であると仮定し，該当ポートへのパケット送出間隔  $SI$  の初期値として設定する．  
SI の初期値を設定するまで，リプライパケットを待つ．

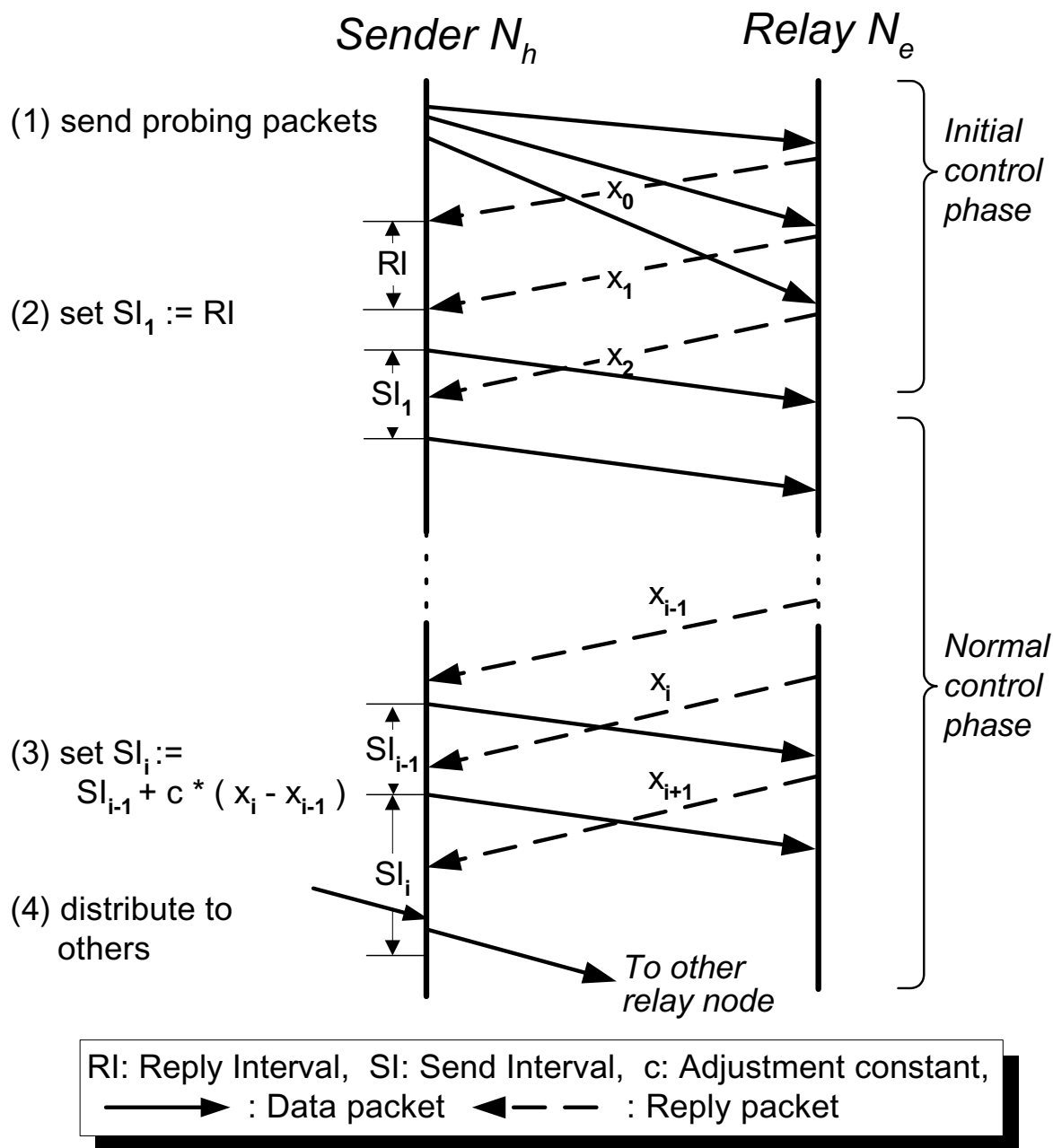


図 3.4 個別リンクのフロー制御信号シーケンス

通常制御 :

- (3) 中継ノードからのリプライパケットが保持するキュー情報(バッファされているパケット数  $x_i$ )をもとに SI を更新する . すなわち , 中継ノードのバッファ内のパケット数の増減に応じて SI も , 次式を用いて増減させる .

$$SI_i := SI_{i-1} + c * (x_i - x_{i-1})$$

ただし,  $c$  は高速ノードの転送性能に合わせた重み係数である.

- (4)  $SI$  が初期化されれば, 以前にパケットを送出した時間から現在までの時間経過  $PI$  を監視する. 次の送出パケットは,  $PI \geq SI$  であれば送出される. そうでなければ, 別の負荷分散先がある限り, 次の  $AC$  部へ転送される.

結果として,  $SI$  は中継ノードの転送能力に漸近し, 自律的に中継ノードの転送能力に応じたフローレートでパケットの送出を続けることが可能となる.

この最初に2つ以上のパケットを送出する方法は, Packet-pair による帯域推定的方式 [50] を参考にした. Packet-pair 方式は, TCP/IP のコネクションエンドからの Ack パケットの受信間隔情報から, 事前にコネクションリンクのボトルネック帯域を推定し, 過度のパケット送出を抑制する方式である. 本章のフロー分散制御方式では, この手法を応用し, 初期制御時にリプライパケットの到着間隔から, 中継ノードの転送能力を推定している.

### 3.4 まとめ

本章では, ネットワークの拡充時の柔軟性やスケーラビリティを提供するために, 高速ノードのトラフィックフローをそれと比較して低速な既設ノードで受信する場合でも, パケットロスや処理遅延を低減でき, 同時に, 複数経路に高速トラフィックを負荷分散してネットワーク内の多様なノードを活用できる自律分散型制御アーキテクチャに基づくフロー分散制御方式について述べた.

ネットワークの拡大や発展時には, 新規ノードの導入や部分的な設備更改を行うと, どうしても残された既存ノードが, 結果としてボトルネックとなりがちであった. このフロー分散制御方式を活用することによって, このような問題も, 緩和されることが期待できる.

ウィンドウ制御方式を始めとする従来のフロー制御方式では, 経路中のノードのトラフィックの処理性能や処理状況を知らずともデータを疎通できるように, エンドホスト間で試行錯誤的にフローレートの調整やパケットの再送を行う制御方式を採用している. イン

ターネットのような不均質なネットワークにおいて、とにかくデータを疎通させるために、これらの方式は至極もっともな有用な手法であったと考える。しかし、近年のインターネットの発展とともに、不均質格差が拡大し、さらにトラフィックの増加に対して設備増強が間に合わない状況においては、従来のフロー制御方式では、ネットワークリソースを活用することが十分にできず、廃棄の有無をプローブとするフローレートの調整機能は、むしろ再送処理にともなうリソースの消費が懸念される。

この自律分散型のフロー分散制御方式は、低い性能のノードを活用しつつ、高速ノードの送出する高速フローを導通できるため、設備更改のどのような段階においても、ネットワーク内のボトルネックを緩和し、ネットワーク全体の性能を引き上げることが可能となる。インターネットのように、計画的な設備更改が難しく、常に発展途上のネットワークにおいては、そのネットワーク性能をできるだけ活用するために、このフロー分散制御方式は大いに役立つと考える。さらにこの方式の特長である自律分散制御機構では、接続形態に合わせて分散スケジューリングを設計することが不要となる。このことは、ネットワークの進展や変化に伴って、行う必要のあったネットワークの設計や設備更改の計画にかかる労力を緩和する効果を生むと期待する。

本章で述べたフロー分散制御方式を適用すれば、送出ノードは接続された中継ノードの性能を自律的に知ることにより、適切にフローの分散化を図ることが可能である。ネットワークの拡大や発展が継続し、接続形態が固定できない現在および将来のネットワークにおいて、この自律分散型制御アーキテクチャに基づくフロー分散制御方式は、非常に有効かつ意義のある技術であると考えられる。





# 第 4 章

## 評 価

### 4.1 まえがき

本章では，本論文で述べる自律分散型ネットワーク QoS 制御方式における，具体的適用例として示した，ノードレベルの QoS 制御方式である自律分散型アーキテクチャに基づく優先キューイング制御方式(2章)と，ネットワークレベルの QoS 制御方式である同アーキテクチャに基づくフロー分散制御方式(3章)の評価手法について述べ，それらの評価モデルをそれぞれ示し，これらの評価モデルに基づきシミュレーション評価を行う．

また，得られた評価結果から，これらの方式の特性を考察し，実用上の優位性について，既存方式との比較や実環境条件に則した評価を行った結果から，定量的に述べる．4.2 節では，2章で述べた自律分散型の優先キューイング制御方式の評価について述べ，4.3 節では，3章で述べた自律分散型のフロー分散制御方式の評価について述べる．そして，4.4 節で，これら評価から得られた結果を総括し，方式全般の特性を考察する．

### 4.2 自律分散型制御に基づく優先キューイング制御方式の評価

本節では，STP による自律分散型優先キューイング制御方式の特性評価のために，2.2 節で提案した廃棄制御手法 (i)(ii) と遅延制御手法 (iii)(iv) を組み合わせた 9 種類の構成の特性評価結果を示し，それぞれの特質を議論する．さらに，最も良好な特性を示す構成と，既存方式とをそれぞれ遅延特性/廃棄特性における差別化について比較し，本方式の優位性を

述べる．最後に，2.4 項で提案した，キューイング時の順序補償方式の導入によって，平均遅延特性を比較し，この方式の導入効果について述べる．

#### 4.2.1 評価方法

本評価では，多様な構成を許容できる自律分散型優先キューイング方式に対して，その動作を柔軟に模擬して特性評価を行うために，STP 間のハンドシェイク信号の授受を基本イベントとする離散事象シミュレータを作成した．

まず，本シミュレータによる評価の信頼性を確認するために，2.3 節に述べたマルコフ連鎖モデルによる解析結果との比較を行った．比較条件を合わせるために，本シミュレータの Send 信号および Ack 信号の伝搬時間，ならびにパケットの移動時間は 2.3 節と同様とする．

図 4.1 に，本シミュレータのパケット移動操作の一例を示す．各ステージ上のパケットは，前方段やバイパス先が空いていれば移動し，空いていなければ待つ．ただし，バイパス先への移動と復路側での移動が競合する場合は，各パケットの優先度に基づき動作が決定される．ここで， $k_{ij}$  は  $k_i$  クラスの  $j$  番目のパケットを意味し， $i$  の値が大きいほど優先度が高く，優先的に出力される．

- ( 1 )  $k_{02}$  は，バイパス先への移動が競合する  $k_{01}$  より優先度が高くないため，前方段に移動する．また， $k_{11}$  は， $k_{12}$  より優先度が低くないため，前方段に移動する．  
( $t = n \rightarrow t = n + 1$ )
- ( 2 )  $k_{21}$  は，バイパス先への移動を優先させ，競合する  $k_{01}$  より優先度が高いため，バイパスする．( $t = n + 1 \rightarrow t = n + 2$ )
- ( 3 )  $k_{11}$  パケットが移動しないため，一部詰まったパイプライン内においても， $k_{12}$  パケットは前方段が空いているため可動である．したがって， $k_{12}$  パケットに続くパケットは  $t = n + 3$  以降可動となる．

本シミュレータは，( 3 ) のように輻輳時でも，空きステージがあればパケットが移動し，ある程度処理が継続できるエラスティック性を包含したシミュレーションが可能である．

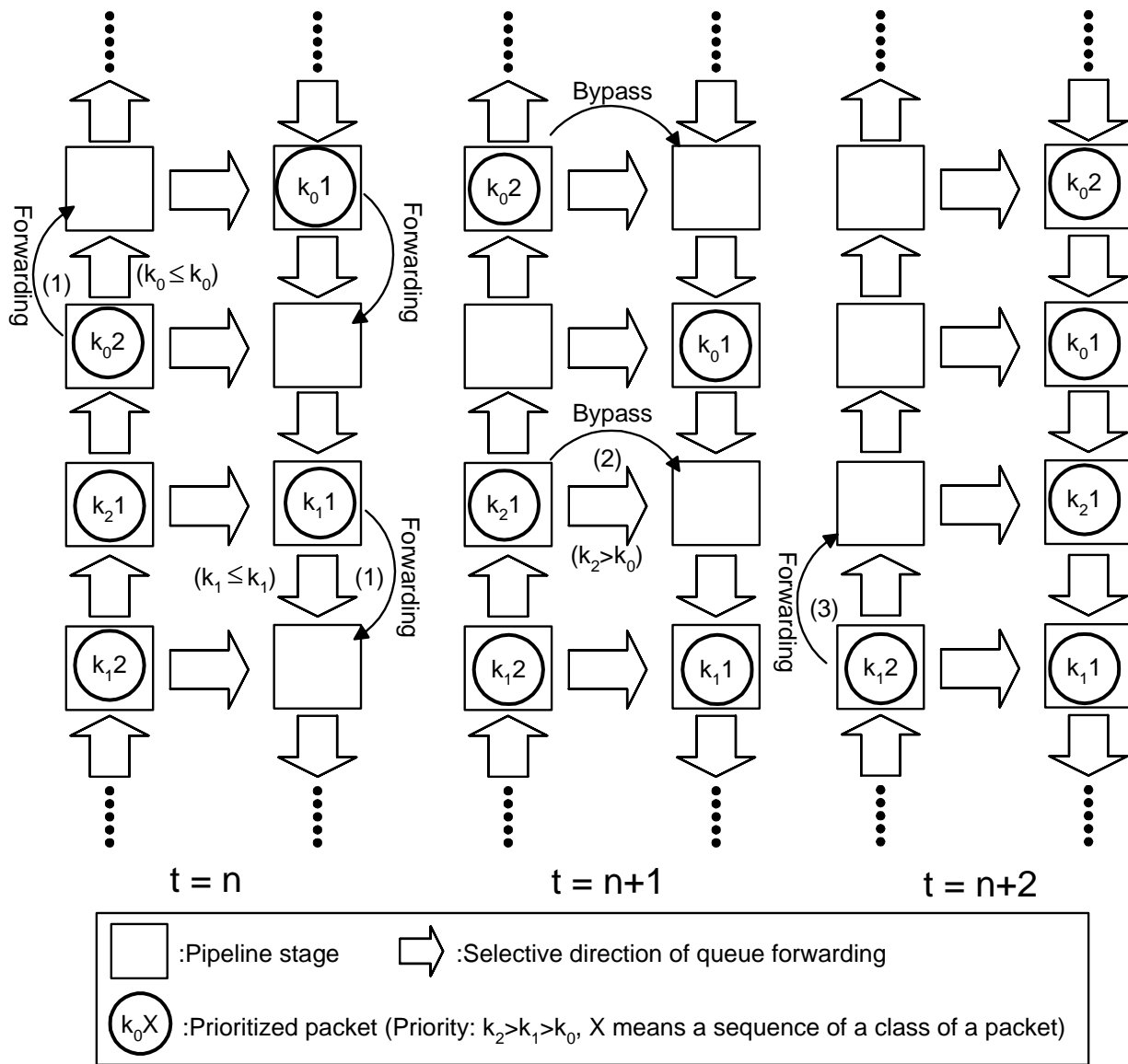


図 4.1 離散事象シミュレーションモデルにおけるパケット移動の一例

自律分散型優先キューイング方式は，キューの成長/縮退過程，および，負荷条件によって，特性が異なると予想される．このため，初期状態からキューが成長し，最大キュー容量に達するまで入力を受け，その後，キューが消滅するまでの平均遅延と廃棄率特性を観測するものとした．さらに，輻輳のため，出力ポートの競合や出力バッファ溢れを生じ，瞬間的にキュー出力できない確率を，出力帯域の閉塞率  $B(t)$  (出力閉塞率) と呼び，これを 0.1 ~ 0.9 まで変化させ，相対的な優先制御が要求される DiffServ の AF クラス [41] を想定して，

優先度が3クラス( $k = 0, 1, 2$ )のパケットを交互に順に入力して、クラス毎に特性を計測した。ここで、出力閉塞率が0である場合、入力帯域と出力帯域は等しいものとする。

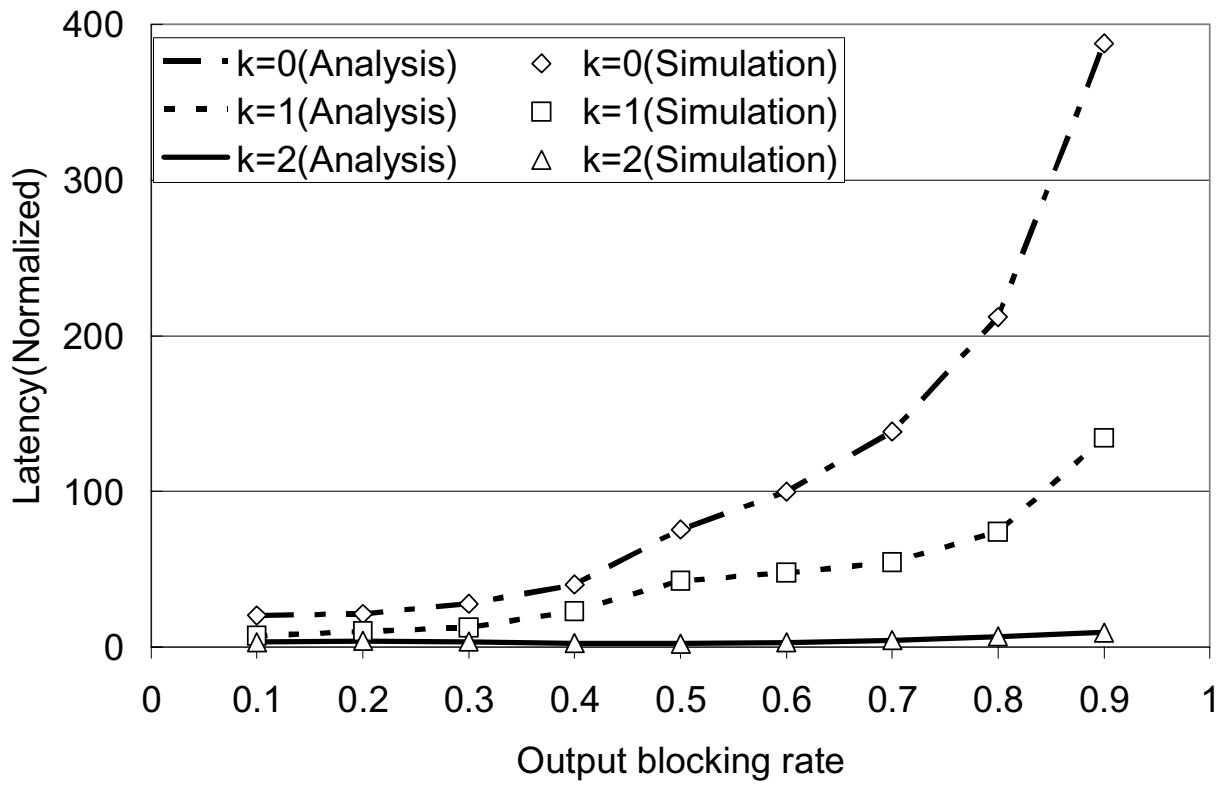
図4.2は、往路/復路パイプライン段が共に100段( $i = 0 \sim 99$ )で、事前廃棄率  $D_A(k, q(k, t)) = 0$ 、入力ステージが  $[0, 0]$  段( $k = 0, 1, 2$ )、廃棄ステージが、 $[9, 0]$  段( $k = 2$ )、 $[49, 0]$  段( $k = 1$ )、 $[99, 0]$  段( $k = 0$ )の場合のキューイング遅延特性と廃棄率特性の比較結果を示している。ただし、入力パケットは、最小サイズのIPv4パケットを用い、投入間隔を、ハンドシェイク手順が連続して行われる十分短い間隔とした。そして、キューイング遅延時間は入力パケットの投入間隔の最小値を1として正規化している。他のいくつかの条件でもシミュレーション評価と解析結果は一致しており、作成したシミュレータの信頼度は充分であると判断した。

よって、以降の特性評価では本シミュレータを用いた。比較評価においても本シミュレータに既存方式を実装し、自律分散型優先キューイング制御方式の遅延特性/廃棄特性を比較した。順序補償方式の評価の場合も、2.4節で示すアルゴリズムを本シミュレータに実装することにより、適用効果を比較評価している。

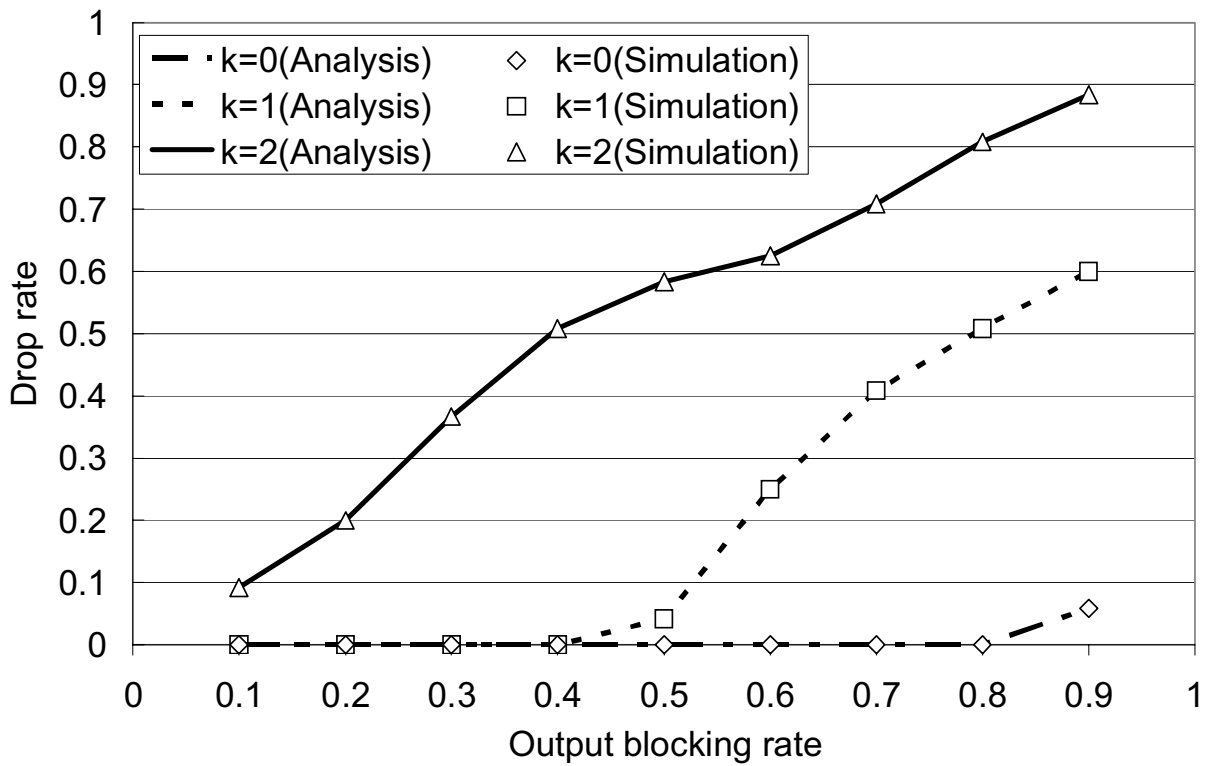
#### 4.2.2 特性評価

QoS評価項目を、遅延特性(平均遅延)と廃棄特性(廃棄率)、遅延分散特性(遅延分散)、帯域差別化特性(帯域差)の4つとし、これらの特性に対する目標値をバランス良く実現することを指標として、2.2節に述べた各制御手法を組み合わせた9種類の構成の特性を評価した。なお、これらの特性はパケットの順序保障を前提とし、順序逆転したパケットは全て廃棄するものとし、各構成におけるパケットの順序逆転率についても評価対象とした。評価条件の、パイプライン段数、クラス、出力閉塞率の変化範囲については図4.2と同様とし、入力パケットは、入力間隔を同様の条件とし、キューイング機構が過負荷状態に達し、一定時間経過するまで投入するものとした。また、この入力パケットの各クラスの構成比を均一とし、かつ同一クラスのパケット間隔をランダムにした。

この評価条件下で、各目標値を次のように設定した。



(a) Latency characteristics



(b) Drop rate characteristics

図 4.2 理論解析とシミュレーション結果の比較

- 平均遅延：400 以下<sup>\*1</sup>，および差別化可
- 廃棄率：差別化可
- 遅延分散：各クラスの分散値が 20000 以下<sup>\*2</sup>
- 帯域差：廃棄率の低いクラスほど占有帯域が高いこと，および中廃棄率のクラスはほぼ占有帯域を変化させずに維持できること
- 順序逆転：順序逆転したパケットの廃棄が，廃棄率の差別化に影響しないこと

これらの目標のうち，差別化可は必須の目標とする．

まず，各制御手法におけるパラメータの影響ならびに適切な値を調べるために，シミュレータを用いて事前評価した結果について考察する．

#### (1) 内部廃棄制御：廃棄ステージ

キューイング機構内の全てのパイプラインステージを有効に活用するためには， $k = 0$  クラスの廃棄ステージは  $[99, 0]$  段が望ましい．その他のクラスの廃棄ステージは，できるだけ下部に設けた方が，順序逆転が軽減される．これは，クラスごとのパケットの動きが抑制され，同一クラスのパケットの順序が入れ替わりにくくなるためと思われる．ただし，廃棄ステージの位置がキューの入出力口に近すぎると，深くキューイングできないため，輻輳時に廃棄率が上昇する．遅延制御を組み合わせる場合は，クラス毎のバイパスステージ BS が設けられ，廃棄ステージがバイパスできる上限ステージとなる．このため，各 BS の範囲が重複しないように区分して設定でき，廃棄ステージをそれほど下部に設けなくても，クラスごとの遅延特性が顕著になり，明確に差別化ができる．これは，区分化により，同一クラスに属するパケットの動作を均質化できるためである．このため，遅延制御を組み合わせると廃棄ステージの位置を調整すると，廃棄率と順序逆転の低減により効果がある．

<sup>\*1</sup> 4.2.1 項の条件で正規化した値．後述の比較評価のため，WFQ 方式を試行した結果，最も遅延のクラスでも 400 以上（出力閉塞率 0.9 の時）であることが分かっている．

<sup>\*2</sup> 正規化した遅延に基づく値．上述と同様に試行の結果，最も遅延分散の小さなクラスで 40000 以上（出力閉塞率 0.9 の時）となり，この半分の値を目標とした．

具体的には，最も遅延の大きい  $k = 0$  クラスの BS と 2 番目に遅延の大きい  $k = 1$  クラスの廃棄ステージを隣接させずに，間を空けると順序逆転および廃棄率の低減に効果があることが分かった．この理由を以下に考察する．

$k = 0$  クラスのパケットは，廃棄されにくいいため，パイプラインステージの上部に滞留する傾向にある．このため， $k = 1$  クラスのパケットは，パイプライン上部でバイパスしにくくなり，廃棄ステージで廃棄されやすくなる．もし，パイプライン上部で  $k = 1$  クラスのパケットが  $k = 0$  クラスのパケットをバイパスすると，復路ステージの前方にも， $k = 0$  クラスのパケットが存在する可能性が高い．この際，先にバイパスしたステージより下部のステージで別の  $k = 1$  クラスのパケットが， $k = 0$  クラスのパケットをバイパスし， $k = 1$  クラスに順序逆転を生じる可能性がある．したがって， $k = 1$  クラスのパケットを上部のパイプラインステージに移動しないように，廃棄ステージを低めに設定すると，このような状況が起こりにくくなり，廃棄率や順序逆転現象が改善される．

以上より， $k = 1$  クラスの廃棄ステージを， $[44, 0]$  段  $\sim$   $[49, 0]$  段， $k = 2$  クラスでは  $[4, 0]$  段  $\sim$   $[9, 0]$  段程度に設定することが望ましいことが分かった．

## (2) 事前廃棄制御：クラス別キューイング閾値

廃棄率が高いクラスほど小さくすると，廃棄率の差別化に有効であるが，遅延の差別化にはあまり寄与しない．ただし，各クラスの閾値の合計を小さくすると，遅延も全体的に減少する．全般に合計値は 60  $\sim$  70 程度が望ましいことが分かった．この理由は，キューイング機構への過剰なパケット流入を抑制すると，キューイング機構内でパケットが円滑に移動できるためと考えられる．

## (3) クラス別入力段分離法：入力ステージ

事前廃棄制御のみを適用する場合，入力ステージのステージ間隔が大きい方が廃棄特性が改善されることが分かった．しかし，全般的に特性は悪い．この理由は，入力ステージへの入力パケットと，キューイングされ前方段に送られるパケットの競合によると考えられる．内部廃棄制御を適用する場合は， $k = 1$  クラスの BS が広くとれるように入



カステージを置くことが望ましく、廃棄ステージの前方段を入力ステージにするとより特性が良くなる。この理由は、内部廃棄により、前述の入力ステージでのパケットの競合を解消でき、 $k = 1$  クラスが、廃棄されにくい  $k = 0$  クラスの滞留の影響を回避できるためであると考えられる。クラス別の入力ステージは、廃棄ステージが設定される場合、その前方段とし、 $[0, 0]$  段 ( $k = 2$ )、 $[5, 0]$  段 ~  $[20, 0]$  段 ( $k = 1$ )、 $[50, 0]$  段 ~  $[80, 0]$  段 ( $k = 0$ ) が望ましいことが分かった。

#### (4) 優先クラスリマーク法：バイパス下限段とリマーク用閾値

クラス別のバイパス下限段は、内部廃棄の場合、既に廃棄ステージの説明で述べたように、低めのステージに設定し、最も遅延の大きいクラスについては間を空けて設定する方が特性が良い。しかし、事前廃棄の場合は、最も遅延の大きいクラスのバイパス下限段を、さらに下部側に設定し、他のクラスのそれを調整した方が遅延分散を抑制できる。この理由は、キューイング機構内でパケットが廃棄されないため、早くバイパスして外部に出力される方が、全体の遅延や遅延分散を抑制できるためと考えられる。

また、リマーク用のクラス別キューイング閾値は、パケットの挙動に大きな影響を与える。この値が大きすぎると順序逆転を多発させる傾向がある。リマーク用閾値は、制御手法の組み合わせにもよるが、概ね 20 パケット程度に設定すると、順序逆転を避けながら、低負荷時の最短経路の活用を行えることが分かった。

以上の各制御手法のパラメータに関する考察に基づいて、適切なパラメータを用いた 9 種類の優先キューイング制御方式の特性評価を行った。前述の QoS 評価項目の目標値を基準とし、基準を満たし最も特性の良い場合を (3 点)、それ以外でいずれかの基準を満たす場合を (2 点)、基準を満足できない場合を × (1 点) として三段階評価を行い、これらを合計したものを総合評価とした。この結果を表 4.1 に示す。

表 4.1 の評価結果より、事前廃棄制御のみの場合を除けば平均遅延が安定していることが分かる。これは、提案方式が、低遅延で遅延の差別化を行えるキューイング特性を有していることを裏付けている。事前廃棄のみ適用した場合の平均遅延は、差別化が困難であり、ク

表 4.1 優先キューイング制御方式の総合評価結果

| 廃棄制御              | 遅延制御 | 平均遅延<br>(*1) | 遅延分散<br>(*1) | 廃棄率 | 帯域差 | 順序逆転 | 総合評価<br>(*2) |
|-------------------|------|--------------|--------------|-----|-----|------|--------------|
| 内部廃棄              | なし   | (792)        | (15018)      |     |     | ×    | 9            |
|                   | 入力段  | (754)        | (11668)      |     | ×   | ×    | 8            |
|                   | リマーク | (648)        | (6815)       | ×   | ×   |      | 9            |
| 事前廃棄              | なし   | ×            | ×            |     |     | ×    | 7            |
|                   | 入力段  | ×            | (17922)      |     |     | ×    | 8            |
|                   | リマーク | ×            | (12764)      |     |     | ×    | 8            |
| 内部廃棄<br>と<br>事前廃棄 | なし   | (411)        | (5064)       |     |     | ×    | 9            |
|                   | 入力段  | (380)        | (3499)       | ×   | ×   | ×    | 7            |
|                   | リマーク | (370)        | (3415)       |     |     |      | 13           |

(\*1) 括弧内の数値: 4.2.1 項の条件で正規化した各評価結果の参考値 .

不可: 差別化不可

(\*2) : 3点, : 2点, ×: 1点とした合計値

ラス間の差もほとんど無いことが分かった．このため，事前廃棄だけではキューイング遅延に関する差別化が難しいと考えられる．これは遅延分散の特性が悪い要因でもある．ただし，事前廃棄制御は内部廃棄制御と組み合わせると，副次的な効果を生むことが評価結果から分かる．

遅延制御別に遅延分散を見ると，優先クラスリマーク法の場合の特性が最もよい．これは，クラスごとに BS が設定されるため，クラスごとのパケットの動作が均質化するためと考えられる．クラス別入力段分離法においても BS は定義されている．しかし，元々同一の入力口から入力されたパケットが，各 BS に振り分けられ，その後復路側ステージで合流する優先クラスリマーク法とは異なり，クラス別に入力段を分離して入力されるこの手法の場合，復路側ステージの合流時に競合を起こしやすくなる．これが遅延分散を大きくする原因と考える．

次に，組み合わせた廃棄制御別に遅延制御の効果を見ると，内部廃棄や事前廃棄を単独で遅延制御に組み合わせても，総合的な QoS を向上できないことが分かる．

内部廃棄制御は，クラス毎に異なる廃棄ステージを指定するため，簡易に廃棄率とキューイングの深さを差別化でき，同時に基本的な遅延の差別化も行える．しかし，これに遅延制御を組み合わせるとキューイング機構内のパケット分布に偏りが生じ，キューイング過程でクラス毎に廃棄する内部廃棄制御の利点を活かさない状況が生じる．この影響は，クラス別入力段分離法を組み合わせた場合の廃棄特性に顕著に現れる．この手法でいずれのクラスの BS も均等にした場合，パケットが入力してからほぼ同じタイミングで廃棄ステージに到達することになり，内部廃棄制御では廃棄率の差別化が難しい．したがって，不均等に BS を設定するほうが廃棄特性が良くなる．優先クラスリマーク法においても，バイパス範囲が指定されるため，遅延制御を行わない場合に比べ，高輻輳時に廃棄率の低いクラスのパケット廃棄率が上昇し，差別化できない場合が起こる．

一方，事前廃棄制御は，キューイング機構内への過剰なパケット流入を防ぎ，本提案方式の特長であるキューイング機構内でのパケットの自律可動性を補償する．このため，高輻輳時でもキューイング機構内で優先度に応じた差別化を行えることが期待される．ただし，優

先度別に流入パケット数を制御できても，キューイングの深さの差別化はできない．特にクラス別入力段分離法では，入力したパケットとキューイングしたパケットが競合する場合があります，遅延や廃棄率の特性に悪い影響を与え，遅延変動を伴う．したがって，事前廃棄制御だけでは，遅延分散の抑制や遅延の差別化が難しい．

以上の議論より，内部廃棄や事前廃棄だけでは QoS 特性を全般的に満足することができない．より厳密に差別化するには，遅延制御と共に，内部廃棄制御と事前廃棄制御を組み合わせる構成が望ましいと言える．この結論は，表 4.1 の総合特性評価の結果にも現れており，優先クラスリマーク法による遅延制御に加えて内部廃棄制御と事前廃棄制御を組み合わせ適用した構成が総合的に最も優れている．この構成では，クラスによるバイパス範囲の指定が，遅延の揺らぎを抑制し，リマーキング処理による最短経路の活用が平均遅延も抑制する．また，バイパス範囲が指定され，パケットの可動性が制限される欠点を事前廃棄制御によって補えるため，遅延制御と廃棄制御を相互に効果的に活用できる構成であると考えられる．

### 4.2.3 既存方式との比較評価

特性評価の中で最も優先制御に優れた本方式の遅延特性を，帯域割り当ての重みにより遅延の差別化を容易に行える WFQ[10] と比較した．ただし，WFQ の各キューには，キューイング遅延を低減させるため，RED が組み合わされているものとする．廃棄特性の比較対象としては，WFQ は本来廃棄制御方式でなく，各キューの Drop Tail に依存した固定的な廃棄特性を示し，妥当でない．このため，RED の廃棄パラメータをクラス毎に異なるように設定して，優先度に応じた廃棄率の差別化を行える WRED を用いた．

この際，本方式における内部廃棄制御の廃棄ステージは  $[9, 0]$  段 ( $k = 2$ )， $[49, 0]$  段 ( $k = 1$ )， $[99, 0]$  段 ( $k = 0$ ) であり，事前廃棄制御のクラス別キューイング閾値は  $(q(0, t), q(1, t), q(2, t)) = (45, 15, 5)$  である．また，優先クラスリマーク法における，バイパス下限段は  $(k = 0, k = 1, k = 2) = (80, 10, 0)$  であり，リマーク用閾値は  $(k = 0 \rightarrow 1, k = 1 \rightarrow 2, k = 0 \rightarrow 2) = (20, 0, 0)$  である．

比較条件を合わせるため，WRED のキューバッファサイズは，自律分散型優先キュー

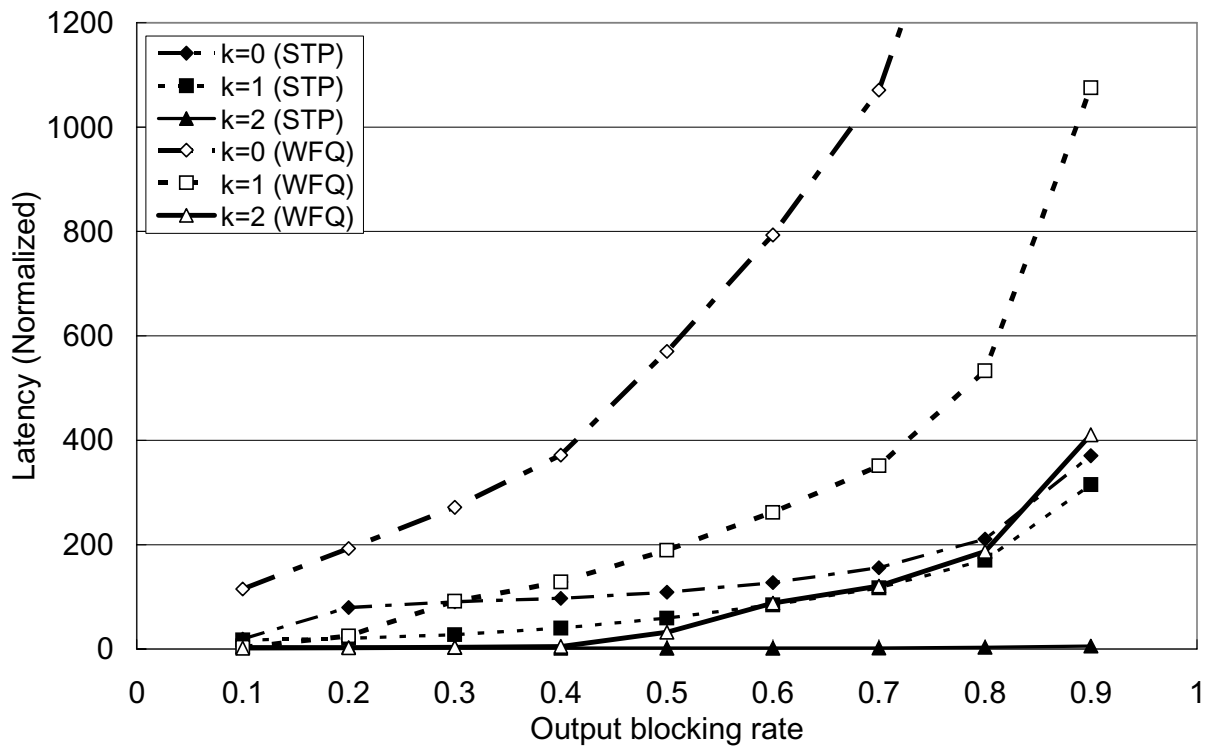
イング方式のパイプライン段数に合わせた 200 パケットとし，WFQ の各キューバッファサイズは，200 パケットを 3 クラスで均等に分割した 66 パケットとしている．また，WRED の  $k = 0$  相当クラスの  $max_{th}$  を，本方式のクラス別キューイング閾値の合計値とし，WFQ に組み合わせた RED の  $max_{th}$  は，この値の  $1/3$  とした．その他の RED パラメータについては，文献 [4, 6] を参考にして， $w_q = 0.002$  とし， $max_{th} = 3min_{th}$  を目安に試行し，最も特性の良かったものを適用した．具体的には，WRED の  $k = 0$  に相当するクラスでは， $(min_{th}, max_{th}, max_p) = (25, 65, 0.02)$ ， $k = 1$  相当クラスでは， $(min_{th}, max_{th}, max_p) = (15, 25, 0.1)$ ， $k = 2$  相当クラスでは， $(min_{th}, max_{th}, max_p) = (5, 15, 0.3)$  とした．WFQ に組み合わせた RED では， $(min_{th}, max_{th}, max_p) = (7, 22, 0.02)$  とした．また WFQ の帯域割り当ての比が，各クラス遅延にほぼ反比例することから，本方式の各クラスの BS 経路の平均長を参考にして， $k = 0, k = 1, k = 2$  相当クラスの帯域割り当て比を，それぞれ  $1 : 3 : 9$  とした．入力パケットのクラス構成比は，パケットの到着確率に合わせ，その他の評価条件は前節同様である．

図 4.3 に各優先クラスの遅延特性に関する比較結果を示し，図 4.4 には廃棄特性に関する比較結果を示す．

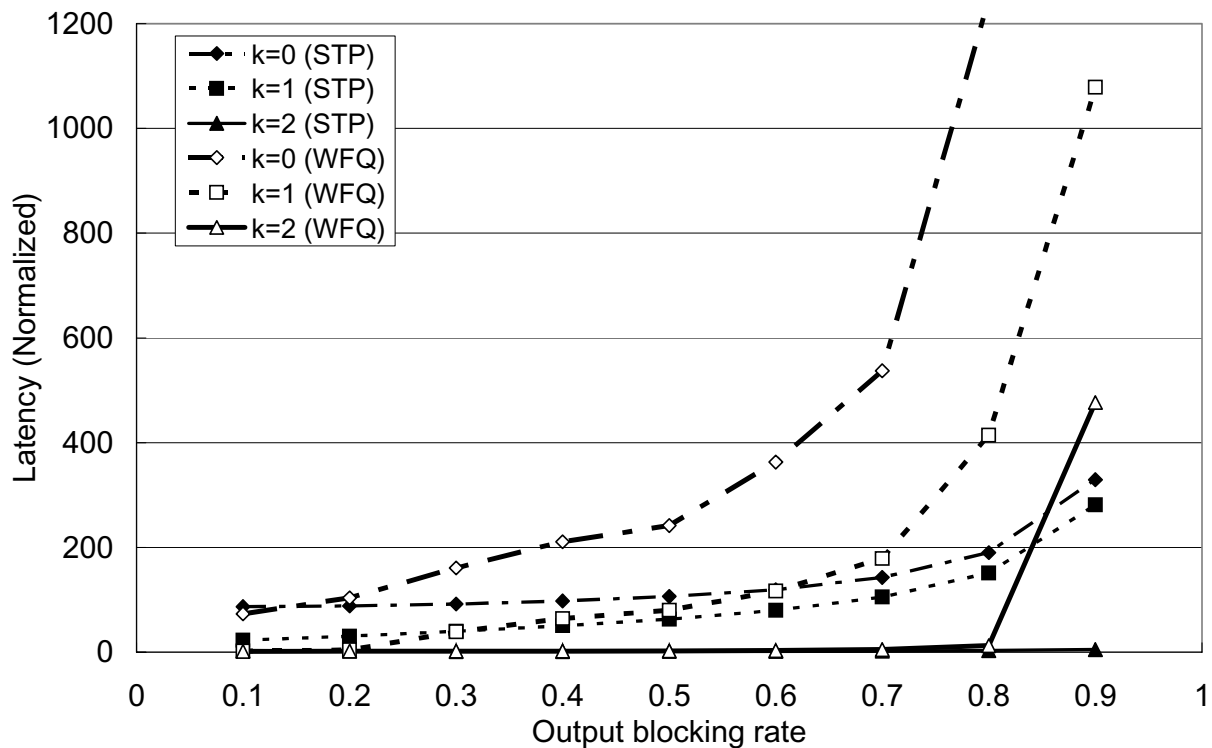
図 4.3 から，広範な輻輳状況において，本方式のキューイング遅延は小さく，かつ遅延の差別化が行われていることが分かる．このうち図 4.3(b) は，優先出力されるパケットが少なく，WFQ 全体の平均遅延が小さくなる場合の結果である．この比較でも，本方式は全般にキューイング遅延が小さいことが示されている．このことから，本方式のバイパスによる遅延抑制効果は，入力負荷バランスや出力閉塞率にかかわらず有効であると言える．また，遅延の差別化によって低遅延を要求するアプリケーションの性能改善が期待される．

本方式の廃棄特性に関しては，図 4.4 から，低廃棄率のクラスの廃棄率特性を維持しながら，高廃棄率のクラスの，出力側の閉塞率に伴う廃棄率の上昇が抑制できていることが分かる．さらに図 4.4(b) より，高廃棄率のクラスのパケットが少なく，このクラスのパケットが廃棄されやすい状況でも，同様に廃棄率の上昇を抑制できることが分かる．

図 4.3(a)，図 4.4(a) の評価では，本方式の投入フローのうち最大 7% のパケットの順序逆

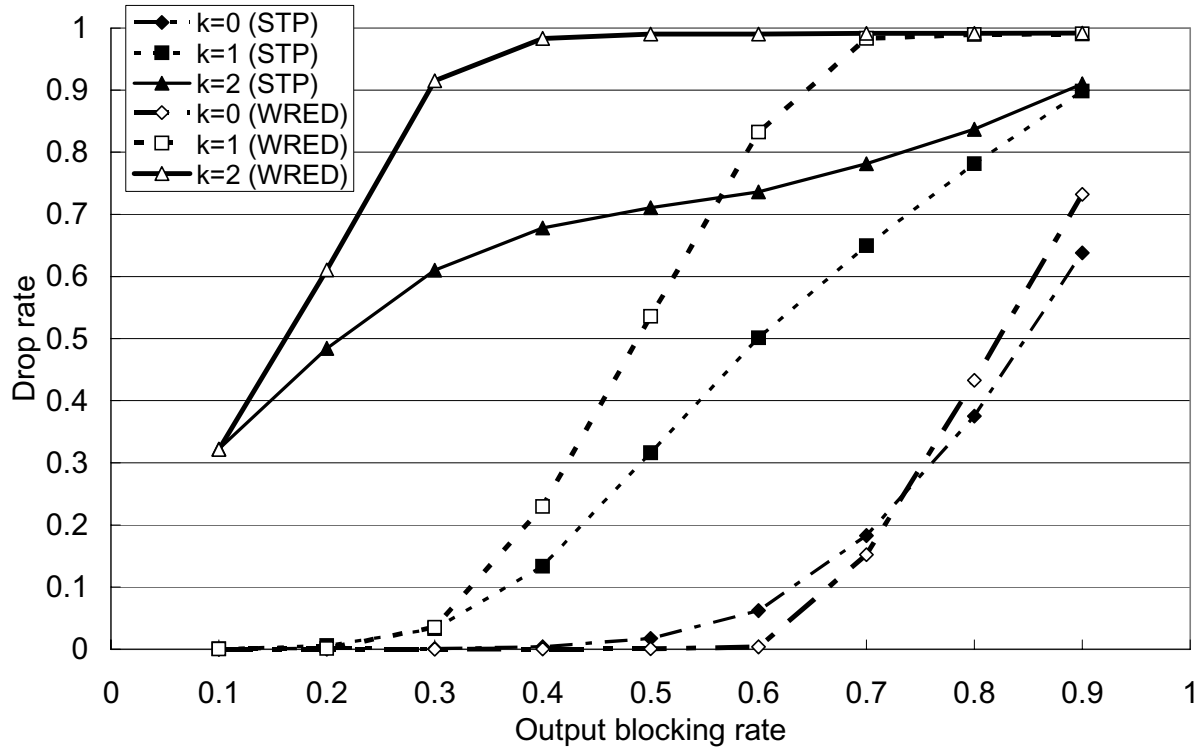


(a) A case of  $\lambda_0(0,t) : \lambda_0(1,t) : \lambda_0(2,t) = 1:1:1$

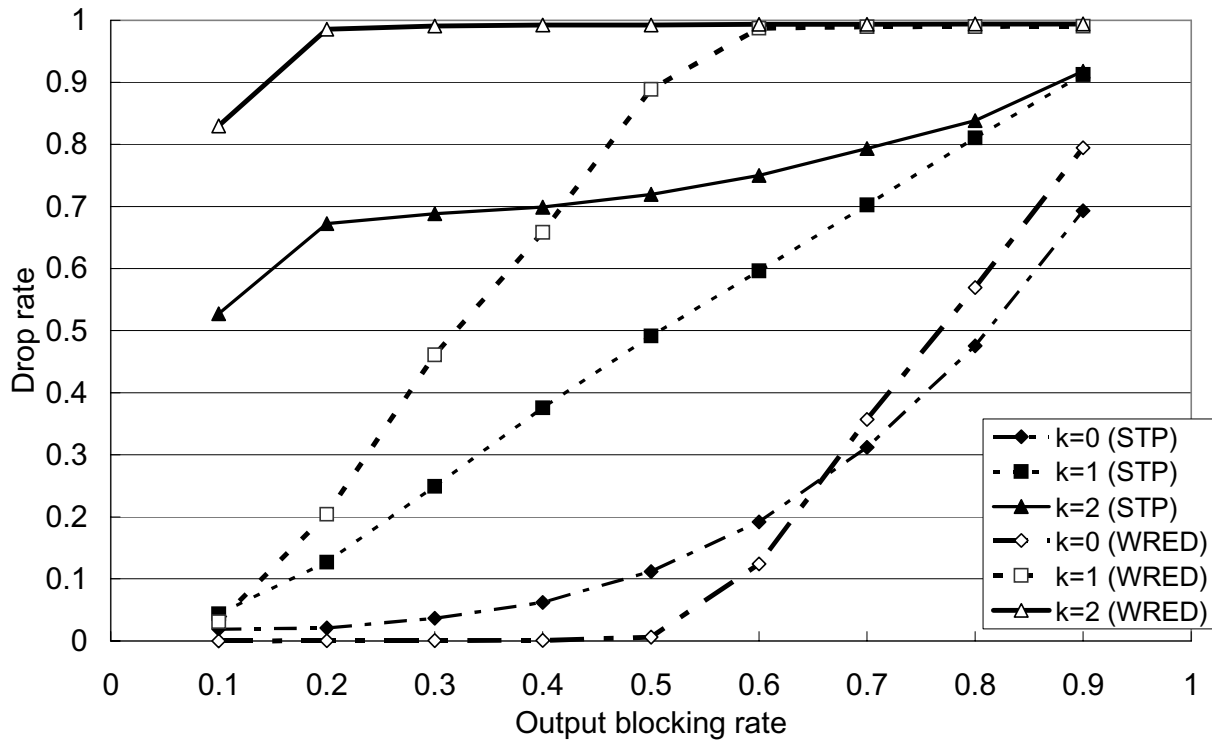


(b) A case of  $\lambda_0(0,t) : \lambda_0(1,t) : \lambda_0(2,t) = 4:4:1$

図 4.3 WFQ と自律分散型優先キューイング方式の遅延特性の比較



(a) A case of  $\lambda_0(0,t) : \lambda_0(1,t) : \lambda_0(2,t) = 1 : 1 : 1$



(b) A case of  $\lambda_0(0,t) : \lambda_0(1,t) : \lambda_0(2,t) = 4 : 4 : 1$

図 4.4 WRED と自律分散型優先キューイング方式の廃棄特性の比較

転が生じ，これらの廃棄を加味した特性を示し，比較している．図 4.3(b)，図 4.4(b) では，最大 0.5% 順序逆転パケットを生じ，これを廃棄した特性で比較評価している．順序逆転パケットを廃棄しても，本方式の特性には十分に優位性がある．

以上の評価結果を通して，自律分散型優先キューイング方式が既存の WFQ 方式に比較して，キューイングに伴う遅延を抑制し，既存の WRED 方式に比較して，出力側が輻輳しても廃棄を抑制できることを確認した．また，廃棄制御や遅延制御を組み合わせることによって，本来の低遅延特性に加えて，多様な差別化特性を示せることも明らかになった．

一方，実装面では，既存の WFQ を ASIC により高速実現する場合，スケジューリング可能な最大クラス数  $N_c$  に依存して，仮想時刻の計算部，および，出力部のソータのハードウェア規模が増大する．本方式の場合， $N_c$  はハードウェア規模の増加の主要因にはならない．しかし，WFQ のキューが単純なオンチップ RAM モジュールで実装可能であるのに対し，本方式の場合，総パイプライン段数がキューの総容量となるため，キューの容量分  $N_q$  だけパケット転送制御回路や優先度比較回路が必要になる．一般には， $N_c \ll N_q$  であるため，例えば， $N_c$  が 16 クラスで  $N_q$  が数十 k バイト以上になると WFQ に比べて本方式のほうがハードウェア費用を要する．しかしながら，今後の集積化技術の進展を考慮すれば多少のコスト増は無視できるため，従来の FQ や AQM では不可能であった柔軟な差別化が低遅延で実現できる利点を持つ本方式に優位性があると考えられる．

#### 4.2.4 順序補償効果の比較評価

本項では，2.4 項に示した自律分散型優先キューイングにおける順序補償方式を適用し，この適用効果が最も高いと考えられる平均遅延特性において，未適用の場合との比較評価を行い，その効果を確認する．この評価においては，順序補償方式の適用効果のみを比較できるように，本キューイング方式において，最も単純な制御の組み合わせ（廃棄制御，遅延制御）＝（内部廃棄，なし）で行い，廃棄ステージは以下のように設定した．

- 内部廃棄制御の廃棄ステージ

[29, 0] 段 ( $k = 2$ )，[59, 0] 段 ( $k = 1$ )，[99, 0] 段 ( $k = 0$ )



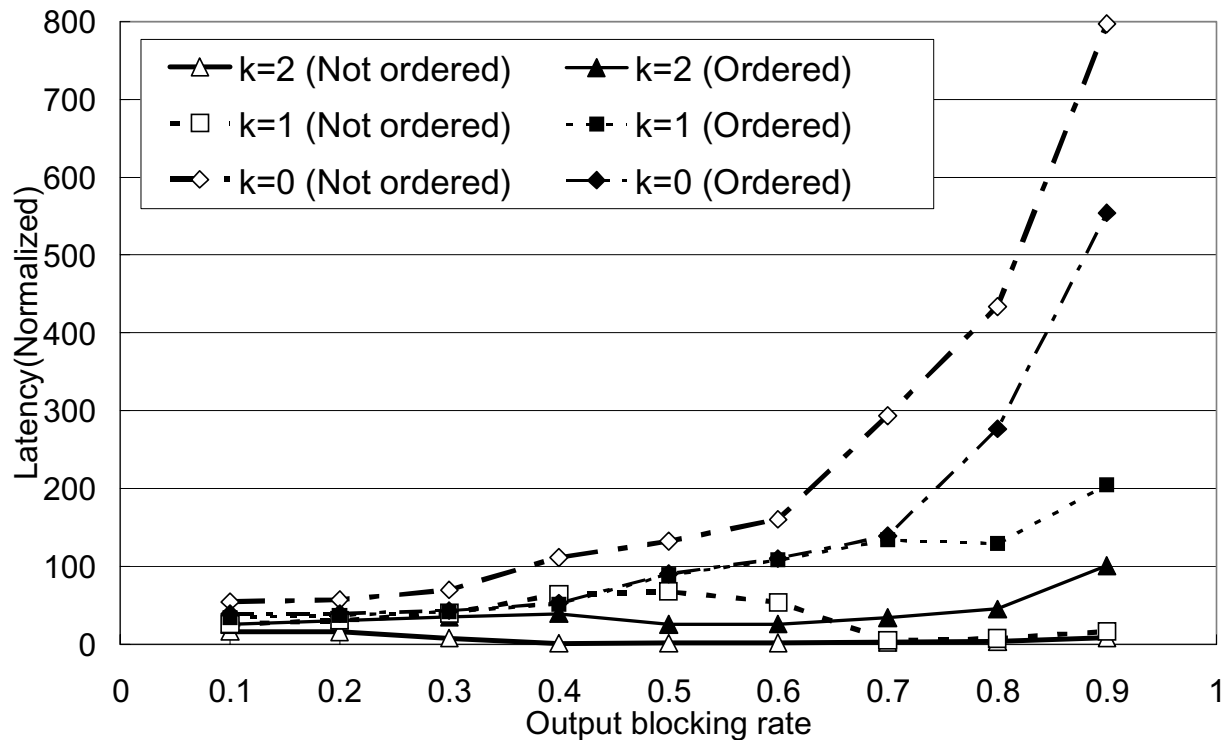


図 4.5 遅延特性における順序補償の効果

その他の評価条件は 4.2.2 項と同じである。

図 4.5 に、順序補償方式の適用/未適用の特性比較結果を示している。この結果から、順序補償によって、遅延特性に改善効果が見られることが分かる。特に、未適用時の回線輻輳率が高い場合を見ると、遅延の大きなクラスが一方的に遅延が増える傾向が改善され、高輻輳時にも適切に遅延特性が差別化されていることが分かる。さらに、中遅延のクラスにおける遅延特性においても、輻輳の度合いに応じて、穏やかに遅延が増加しており、未適用時の遅延変動が改善されることが示されている。

本キューイング方式は、自己タイミング型の制御動作を行うため、パケット個々の振る舞いは隣接するステージに存在するパケットの影響を受ける。STP の基本構成である、直線パイプラインのように、折り返し構成によるバイパス経路が存在しなければ、往路側ステージや復路側ステージにおいて、あるパケットの前後に位置するパケットが変わることがほぼなく、パケットの移動にはこれらのパケットの影響が連続的に続くことになる。この場合

は、動作が変動しても収束していくことが容易にわかる。しかしながら、バイパス経路の存在する本キューイング方式は、この前後のパケットの影響に加えて、バイパス先の競合するパケットからも影響を受ける。また、前後のパケットもバイパスや廃棄によって、変化していく。したがって、自律的動作の自由度が増し、個々のパケットの動作に注目すると直線パイプラインより変動が大きいと考えられる。

このような制御方式の回路に順序補償の制御を加えると、個々のパケットは、シーケンス番号によって、同一クラスの前後に位置していないパケットの影響も受ける。すなわち、キューイング機構内で、自身と同様な経路を通過する同一クラスのパケットからの影響が加わることになる。これは、前述の直線パイプラインにおける前後パケットからの連続的な影響と同様な効果となり、変動の収束と、動作の均質化に貢献すると考えられる。このため、遅延特性が改善し、遅延の低減効果が示されると考える。

以上の評価結果を通して、自律分散型優先キューイング制御方式が既存の AQM 方式と比較して、キューイングに伴う遅延を抑制し、出力側の輻輳時でも廃棄を抑制できるエラスティック性を持つことを確認した。また、廃棄制御や遅延制御を組み合わせることによって、本来の低遅延特性に加えて、多様な差別化特性を示せることも明らかになった。さらに、順序補償によって、広範な輻輳状態の遅延特性に大きな改善効果が期待でき、本キューイング制御方式が、多様な条件下のサービスやアプリケーションに対応できることが見込まれる。

### 4.3 自律分散型制御に基づくフロー分散制御方式の評価

本節では、3章で提案した自律分散型制御アーキテクチャに基づくフロー分散制御方式の有効性を、シミュレーションにより評価し、その結果について考察を行う。図 4.6 はこの自律分散型のフロー分散制御方式に関するシミュレーションの基本モデルを示している。この評価では、20Bytes の IPv4 パケットの連続フローを想定したトラフィックを用い、定常状態に遷移するまで投入する。単位時間は、高速ノード (Sender および Receiver) の転送

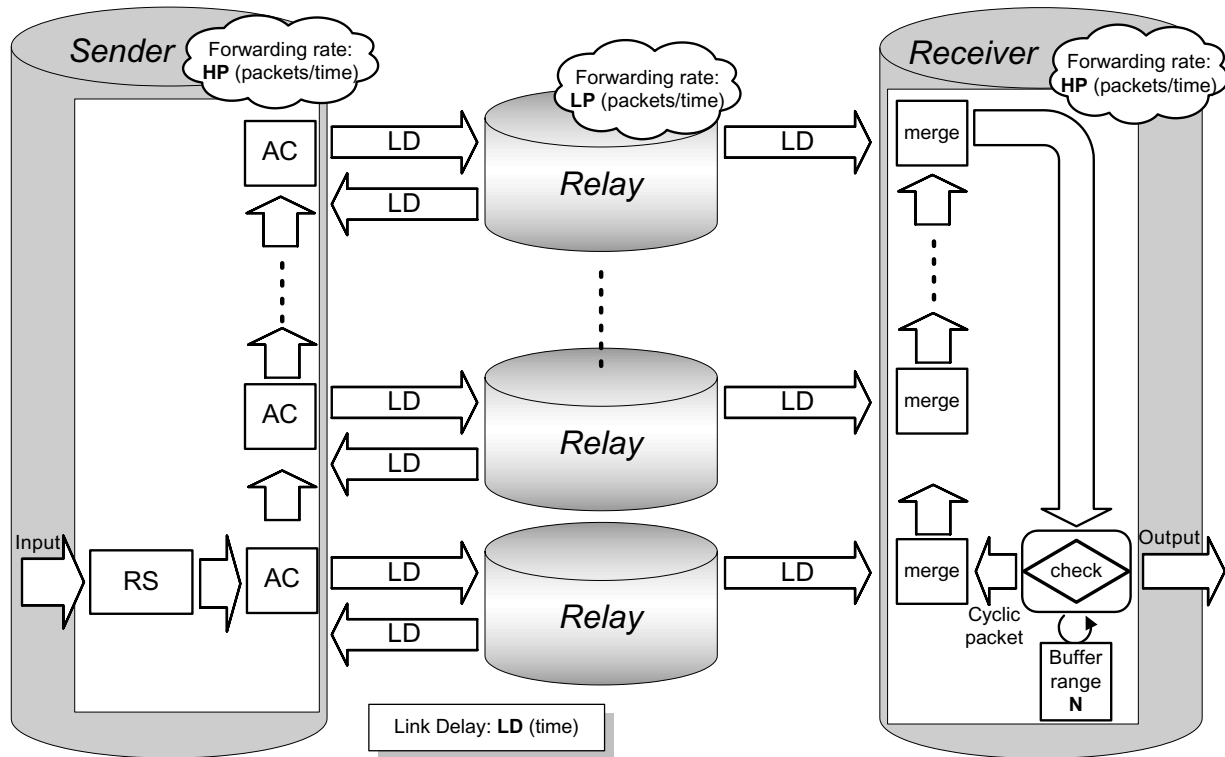


図 4.6 基本評価モデル

能力にスケールさせ、高速ノードがこのパケットを連続転送可能な時間間隔（最小転送間隔）とし、これを  $1(\text{time})^{*3}$  とする。高速ノードの AC 部間パケット移動時間と環状型受信バッファ機構内のモジュール間パケット移動時間もここで定義した  $1(\text{time})$  とする<sup>\*4</sup>。この単位時間を用いて高速ノードの転送能力は、 $HP = 1(\text{packet}/\text{time})$  と表される。また、中継ノード (Relay) の転送能力を  $LP(\text{packet}/\text{time})$  とすると、中継ノードの最小転送間隔は、 $1/LP(\text{time}/\text{packet})$  と表される。以降の評価では、この  $1/LP$  を用いて中継ノードの転送能力を表す。さらに高速ノードと中継ノード間のリンクに伴う遅延を  $LD(\text{time})$  としている。本評価では、フロー分散から、フロー集約までの方式評価を行うものとし、高速ノードに内在する遅延は対象外とする。

まず本フロー分散制御方式が複数経路上の中継ノードの転送能力に応じて、効果的にフ

<sup>\*3</sup> 高速ノードが 20Bytes のパケットを 10Gbits/s で連続投入できる場合、最小転送間隔は 16ns となる。

<sup>\*4</sup> パケット転送に関わるノード内部の一部の処理であるため  $1(\text{time})$  より短縮できると考えられるが、方式のオーバーヘッドの最悪条件として  $1(\text{time})$  を定義する。

ローの負荷分散が行えることを確認するために、複数経路上のノードのパケットの転送能力のみを対象とした特性評価を行う。その効果を確認した上で、現行のノード装置に定常的に加わる負荷による遅延、およびその遅延変動とパケット損失を想定し、実際のネットワークへの適用効果を評価する。

#### 4.3.1 特性評価

本スロー分散制御方式に対する特性評価として、高速ノードからの送出フローを複数種の中継ノードに分散させ、受信側の高速ノードで受信して出力されるまでの、ネットワークスループットとネットワークレイテンシーの評価を行った。ここで、ネットワークスループットとは、送出側の高速ノードから、受信側の高速ノードへのフローのスループットを指す。ネットワークレイテンシーとは同じ高速ノード間のフローのレイテンシーを指し、具体的にはフロー分散方式のRS部の入力から、フロー集約方式の環状型受信バッファ機構の出力までのレイテンシーを指す。この評価では、中継ノードの転送能力のみ対象とし、局内接続を想定した最小限のリンク遅延のみ付加している。

特性評価では、高速ノードのフローをもれなく分散して中継できる最小限の並列中継ノード数を用い、ネットワークスループットとネットワークレイテンシーを測定している。いずれの測定結果でもネットワークスループットは高速ノードの転送能力(最大スループット)に達することを確認した。よって評価結果は同種/異種ノードの並列構成パターン別の、ネットワークレイテンシー、および最大スループット到達時間に着目して特性評価を行っている。なお、以降の特性評価に用いたリンク遅延はすべて  $LD=3(\text{time})$  としている。

##### (1) 同種ノードで構成される複数経路を用いた評価

まず、全経路の中継ノードの転送能力が同一である複数経路に、フローを分散させた場合の評価を行った。図4.7は、 $1/LP$ がそれぞれ5, 10, 15, 20, 30(time/packet)の中継ノードを想定し、これらより同一のノードで並列構成する、5種の複数経路で評価した最大ネットワークレイテンシーと最大スループット到達時間の特性である。中継ノード

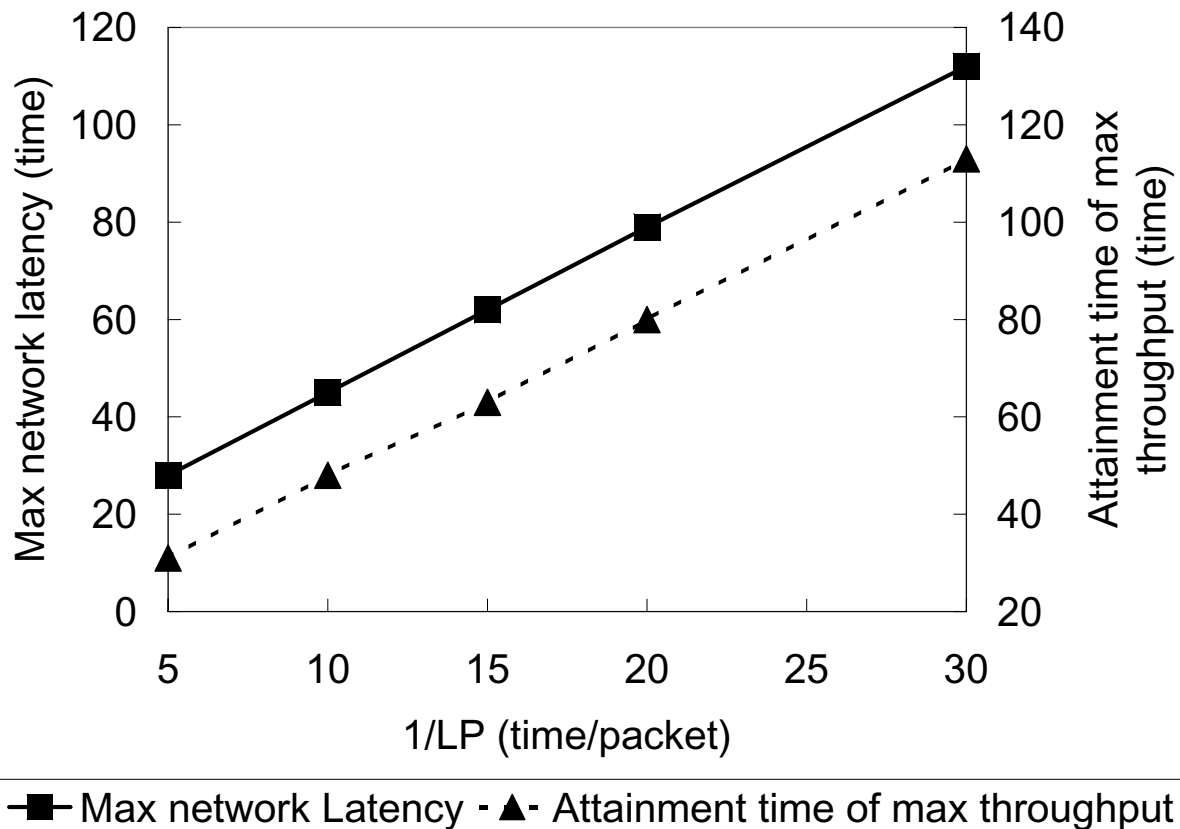


図 4.7 同種ノードで構成された中継経路の最小転送間隔  $1/LP$  による最大ネットワークレイテンシーと最大スループット到達時間の特性

ドの最小転送間隔に応じて，両特性とも線形増加を示し，この方式が分散構成した中継ノードの  $1/LP$  に応じたスケーラビリティを持つことを示している．

## (2) 異種ノードで構成される複数経路を用いた評価

次に，各経路の中継ノードの転送能力が異なる複数経路に，フローを分散させた場合のネットワークレイテンシーと最大スループット到達時間の特性を評価した． $1/LP$  が，5，10，15(time/packet) の3種の異種中継ノードを想定し，各経路にいずれかの中継ノードを用いた複数経路を考える．ただし，下部の AC 部に接続する経路から上部の AC 部に接続する経路の方向に，3種の中継ノードをローテーション順に，各経路へ割り当てる．ローテーションの方法は，順列  ${}_3P_3=6$  より6通り想定し，6種の複数経路を考える．この評価モデルを図 4.8 に示す．

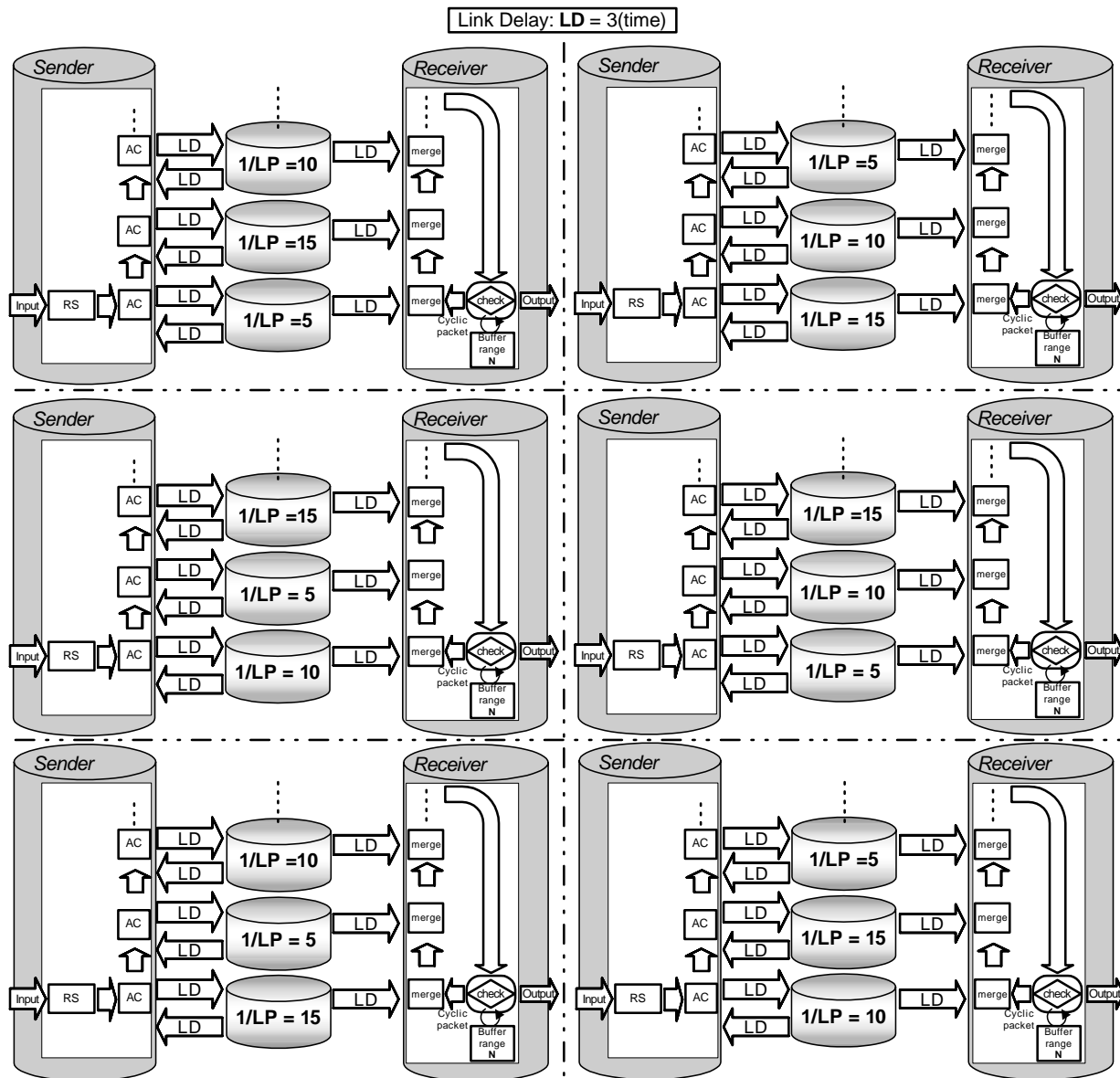


図 4.8 異種ノードで構成された 6 種類の複数経路による評価モデル

図 4.9 は、これら 6 種の複数経路で最大ネットワークレイテンシーと最大スループット到達時間の特性評価を行ったものである。この図 4.9 から、中継ノードの性能が低い経路順にフローが分散するパターン、すなわち  $1/LP$  が  $\dots 15 \ 10 \ 5 \ \dots$  の順の複数経路構成の方が、 $\dots 5 \ 10 \ 15 \ \dots$  の構成より、最大スループット到達時間が早く、ネットワークレイテンシーが小さい傾向が見られた。この理由は、前者の構成では下部 AC 部からの送出レートより、上部 AC 部からの送出レートが速い

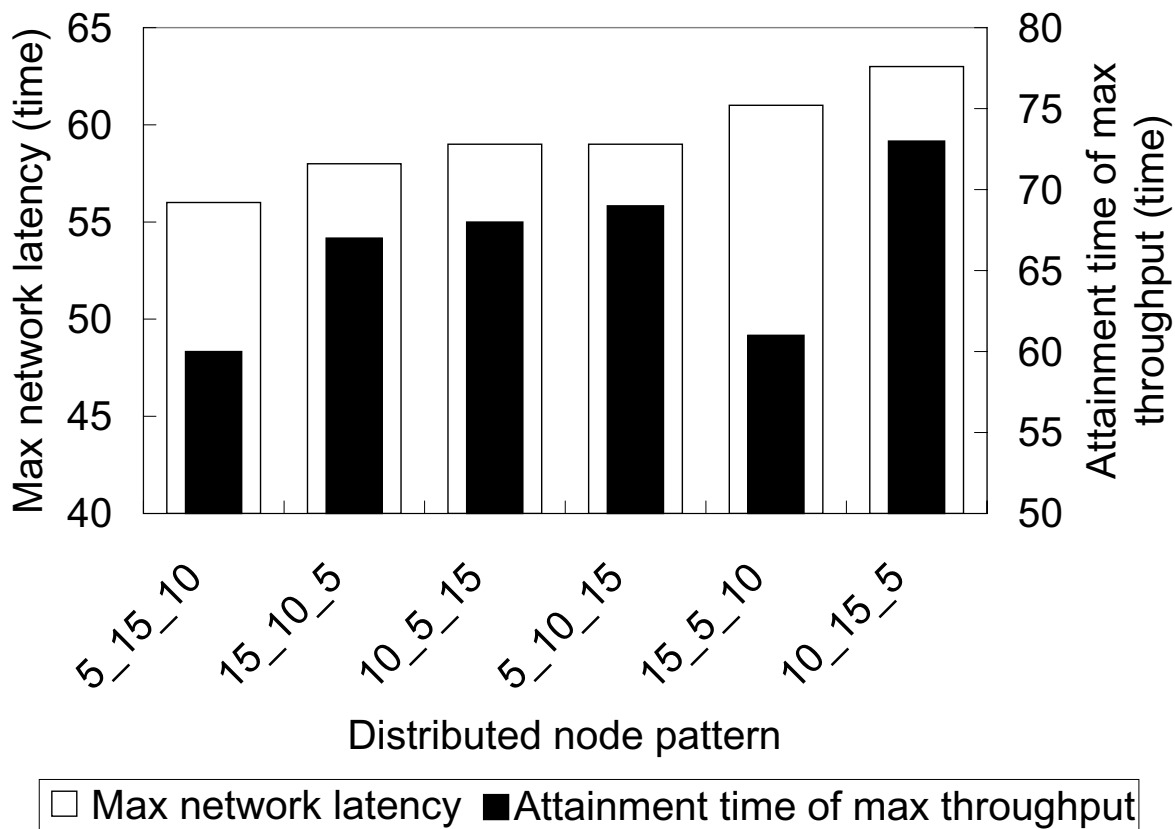


図 4.9 異種ノードで構成された中継経路へのフロー分散順序による最大ネットワークレイテンシーと最大スループット到達時間の特性

場合が多く、パケットが下部 AC 部から送出されず、上部 AC 部に転送された場合でも、その上部 AC 部への転送遅延が受信ノード到着時には解消されるためである。逆順になる後者の複数経路構成では、上部 AC 部に転送されたパケットは下部 AC 部から送出されるより遅いレートで送出される場合が多く、このため受信ノードへのパケットの到着間隔が広がる傾向にある。さらに・・・ 15 5 ……のような転送能力の逆転差の大きい経路間で分散送出した連続パケットは、順序逆転することが多く、この順序補正の待ち合わせ遅延の累積が、ネットワークレイテンシーを大きくし、最大スループットの到達時間を遅くすると考えられる。15 5 10 ……の複数経路構成の最大スループット到達時間特性は、上述の傾向と異なる。この理由は、15 5 の転送能力逆転差が最下部の経路にあるため、早期に順序逆転による遅延が確定し、その後、遅

延変動が少なくなるためと考えられる。

以上のことから，予め中継ノードの性能差が既知である場合，本フロー分散制御方式では，性能の低い順にフローを分散させる方が，フロー分散効率を高めることがわかる。

#### 4.3.2 典型的なネットワークへの適用例の評価

実際の典型的なネットワーク構成を想定して，本フロー分散制御方式の実用可能性についても予備的評価を行った。

この評価では，図 4.6 の基本評価モデルにおいて，単位時間を基準に，送出ノード - 中継ノード間，および中継ノード - 高速ノード間のリンク遅延をそれぞれ  $LD=250(\text{time})$  とした。また，転送能力の異なる中継ノードを 3 種類想定し，それぞれ  $1/LP$  を，3, 5, 15( $\text{time}/\text{packet}$ ) とし，さらにこの中継ノードにそれぞれ 500, 1000, 1500( $\text{time}$ ) の処理遅延(ノード遅延:  $ND$ )を加味した。これらのパラメータは現行の数種のギガビットルータ [51, 52, 53] を参考にして設定している。

また，本評価では実際のノードで生じる遅延変動とパケット損失を想定し，これらの評価条件も設定した。遅延変動とパケット損失は中継ノードで生じるものとし，遅延変動については， $1/LP$  を平均値とする指数分布に従って転送間隔が変動するものとした。さらにその転送間隔の処理タイミングに合わせて， $ND$  もそれぞれ 500, 1000, 1500( $\text{time}$ ) を平均値とする指数分布に従って変動するものとした。パケット損失はランダムに発生するものとし，損失率を 5% とした。

これらの評価条件のもとで，複数経路を 9 経路に固定し，上記 3 種のノードを用いた同種 / 異種ノードの並列構成によるネットワークレイテンシーを測定した。

##### (1) 遅延変動とパケット廃棄がない場合

###### (a) 同種ノードで構成される複数経路を用いた評価

図 4.10 は，想定した 3 種の中継ノードを用いて，同一の中継ノードで並列構成する 3 種の複数経路にフロー分散させた時の，ネットワークレイテンシー特性であ



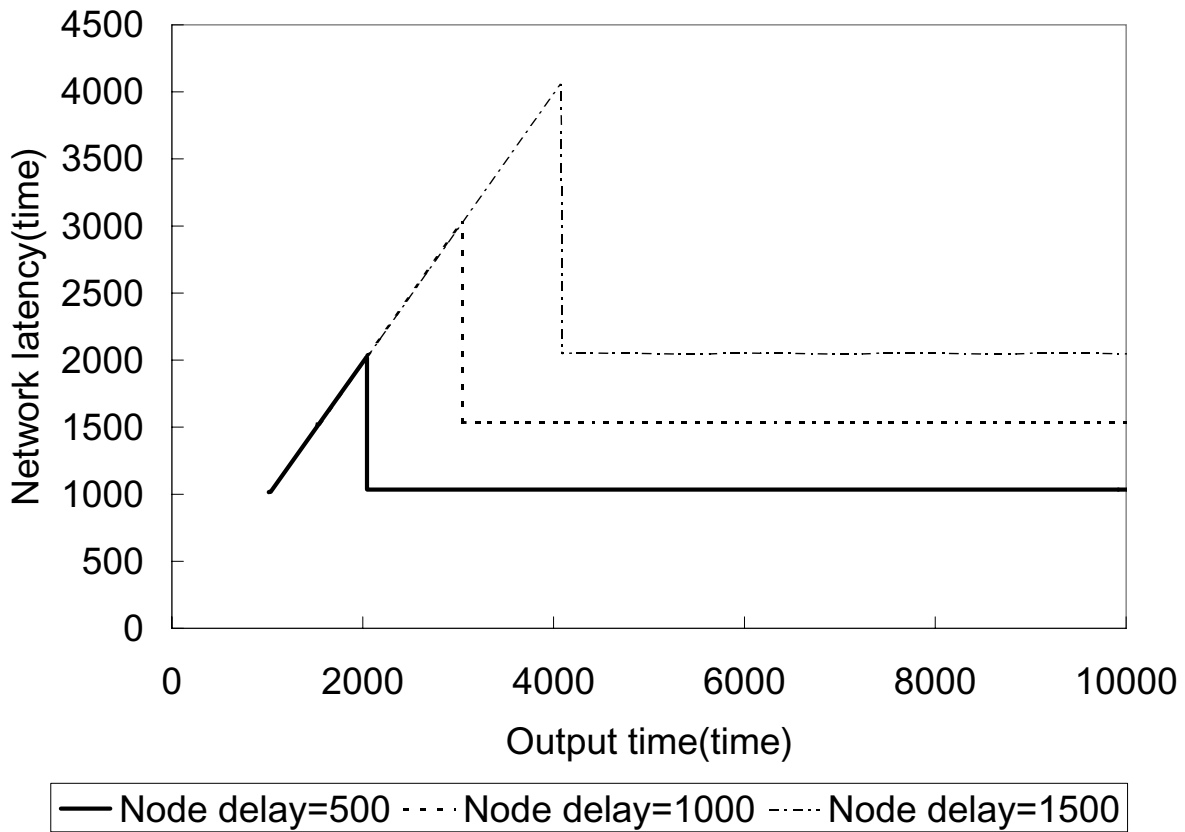


図 4.10 同種ノードで構成された中継経路へフロー分散したネットワークレイテンシー特性

る。いずれの複数経路構成の場合も，ノード遅延にリンク遅延合計 500(time) を加えた遅延，1000，1500，2000(time) にネットワークレイテンシーが収束している。この結果より，定常状態に遷移した，通常制御状態では本フロー分散制御方式の遅延の影響は見られず，効率よく分散されていることがわかる。初期の大きなレイテンシーは，初期制御時の SI 設定用のリプライパケット待ち時間のためである。例えば，ノード遅延が 500(time) の中継ノードへの初期制御の場合，リプライパケットが到着するまでの待ち時間（往復リンク遅延 + 中継ノードの遅延 = 1000(time)）に，経路上でのレイテンシー（リンク遅延合計 + 中継ノードの遅延 = 1000(time)）を加えた遅延が初期制御時の最大遅延となっているのがわかる。

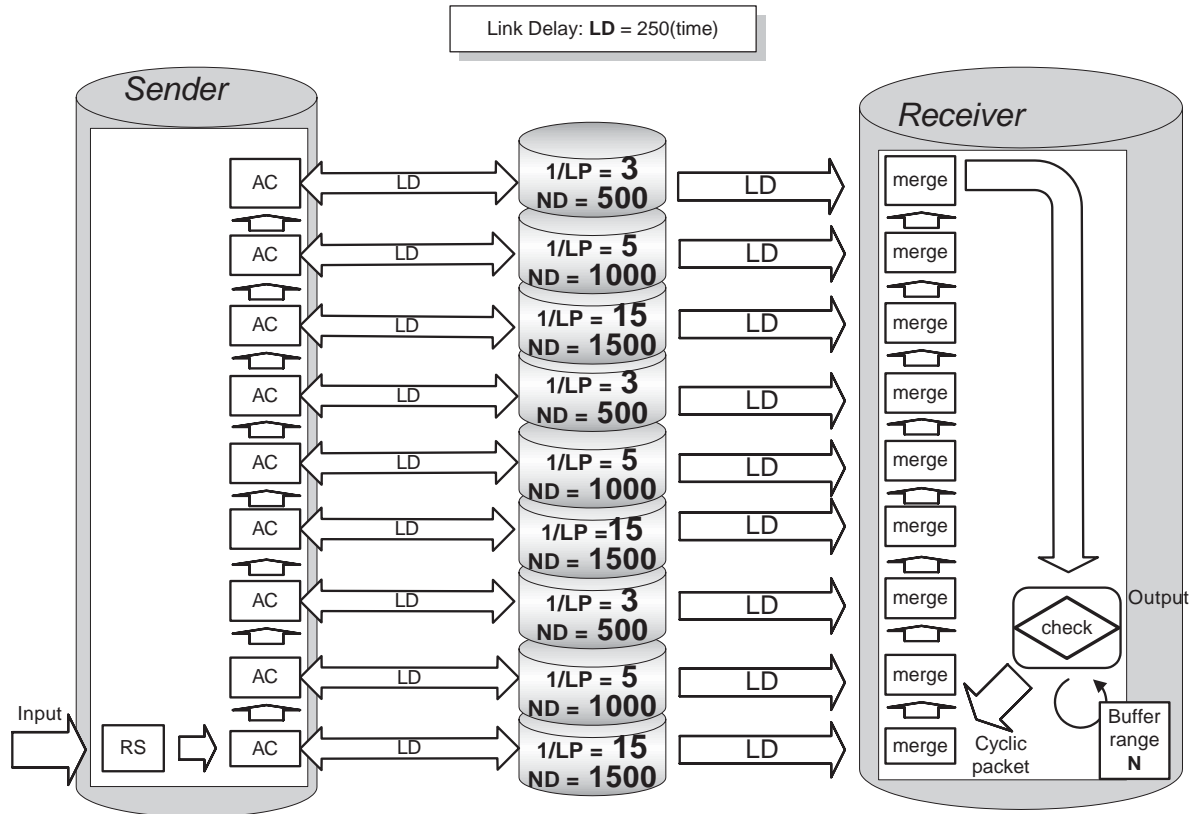
(b) 異種ノードで構成される複数経路を用いた評価

次に，図 4.11 の評価モデルに示すように，3 種の中継ノードをローテーションし

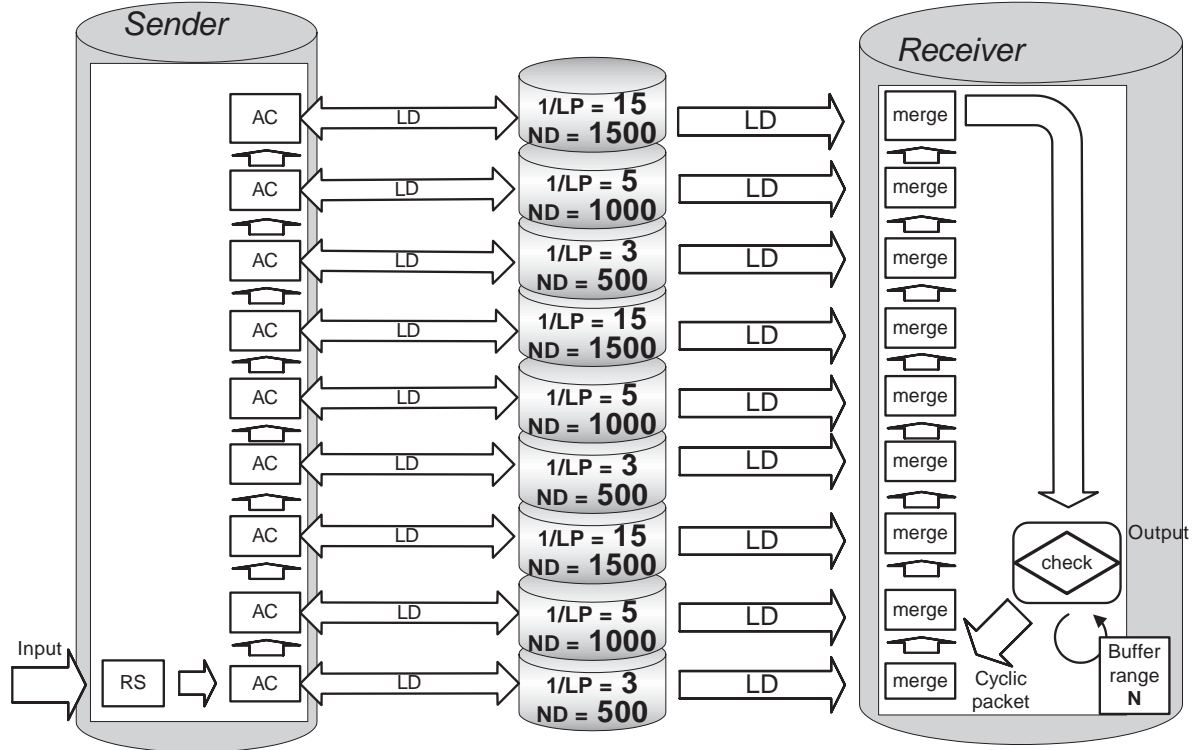
て各経路に割り当てた複数経路構成における特性評価を行う。図 4.12 は、この評価モデルを用いて評価を行って得られたネットワークレイテンシー特性である。特性評価で得られた結果を確認するために、この評価では中継ノードの性能の低い順にローテーションして割り当てた複数経路と、逆順に割り当てた複数経路の 2 種の構成を用いる。図 4.11(a) に示すパターン A は、性能の低いノード順（ノード遅延順で、1500 1000 500 1500 … の順序）に構成した複数経路であり、図 4.11(b) のパターン B はその逆順で構成した複数経路である。この結果は、前項の評価から見られた特性と同様、性能の低いノード順にフローを分散させた方が、ネットワークレイテンシーの特性がよいことを示している。

図 4.12(a) に示されるパターン A のネットワークレイテンシーでは、初期の特性に初期制御時のリプライパケット待ちの遅延の影響が見られる。図 4.10 と比較すると、この初期特性はノード遅延が 500(time) の場合の初期特性と一致しているのがわかる。初期制御時に、SI の初期値を設定するため、リプライパケットを待ち合わせる場合、送信番号の小さなパケットほど上部の AC 部で待ち合わせることになる。パターン A の経路構成の場合、上部 AC 部の方が性能の高い中継ノードに接続される傾向にある。したがって、上部 AC 部から先に SI が設定され、下部で待ち合わせていたパケットも続いて同じ AC 部から送出されていく。このため、順序逆転が起こりにくく、最小のノード遅延の影響しか出ないことになる。初期特性以降のネットワークレイテンシーは各経路の異なるノード遅延のため SI の設定時刻に差を生じ、早く SI が設定された経路からパケットが送出されるため、ネットワークレイテンシーが階段状に上昇している。

パターン B ( 図 4.12(b) ) の場合、最大ノード遅延 1500(time) とリンク遅延合計 500(time) に加えて、最大ノード遅延と最小ノード遅延の差（最大ノード遅延ギャップ）1000(time) が順序補正のための待ち時間として累積されるため、ほぼ 3000(time) にネットワークレイテンシーが収束する。初期の大きなレイテンシーは、図 4.10 のノード遅延が 1500(time) の場合の特性と一致し、複数経路内で最大

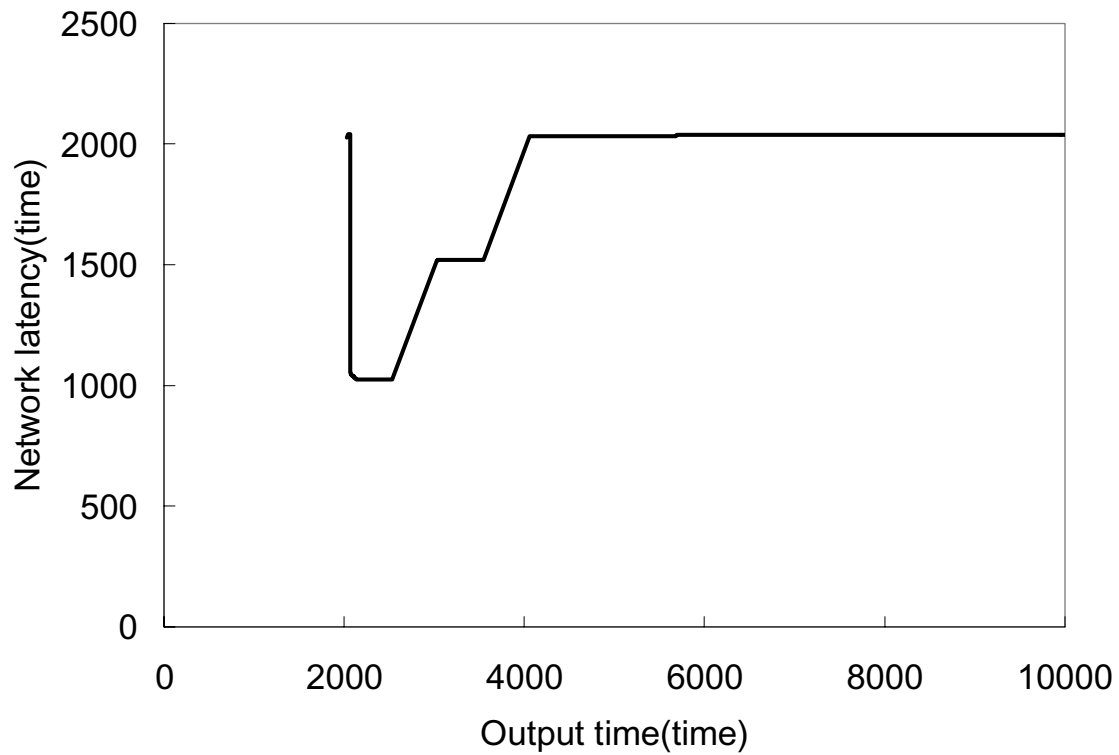


(a) Heterogeneous multiple pathways by pattern A

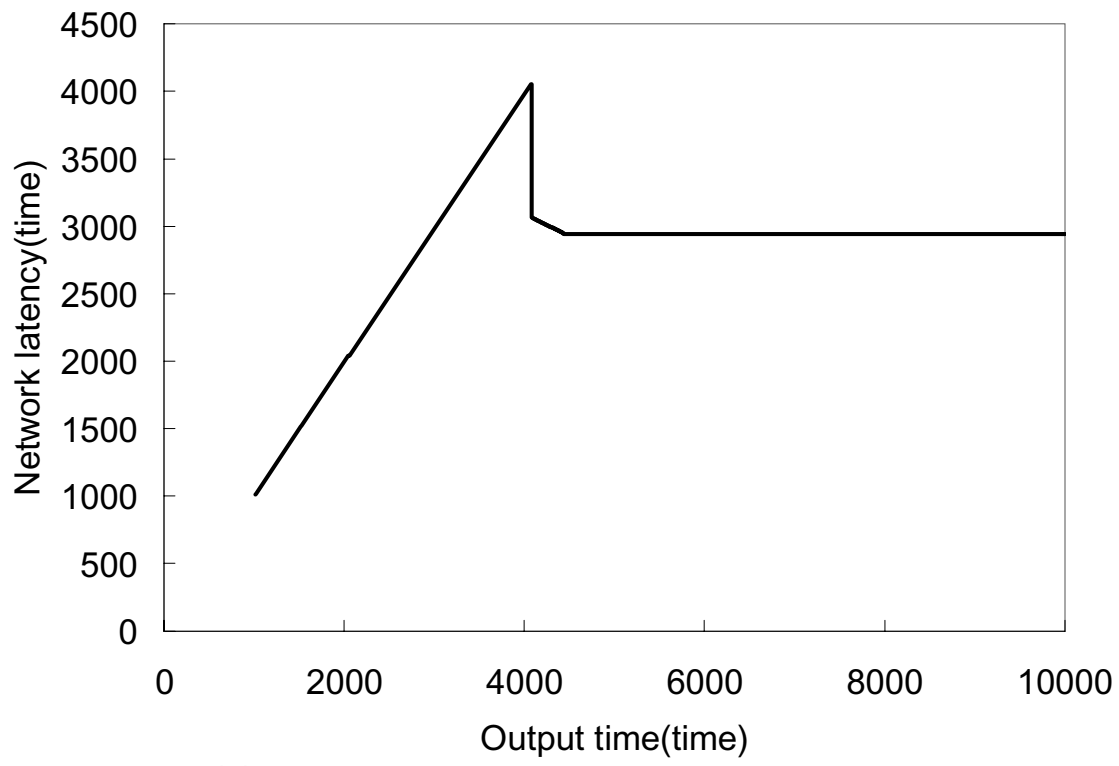


(b) Heterogeneous multiple pathways by pattern B

図 4.11 異種ノードで構成された複数経路による評価モデル



(a) Network latency characteristics by pattern A



(b) Network latency characteristics by pattern B

図 4.12 異種ノードで構成された中継経路へのフロー分散順序パターンの違いによるネットワークレイテンシー特性

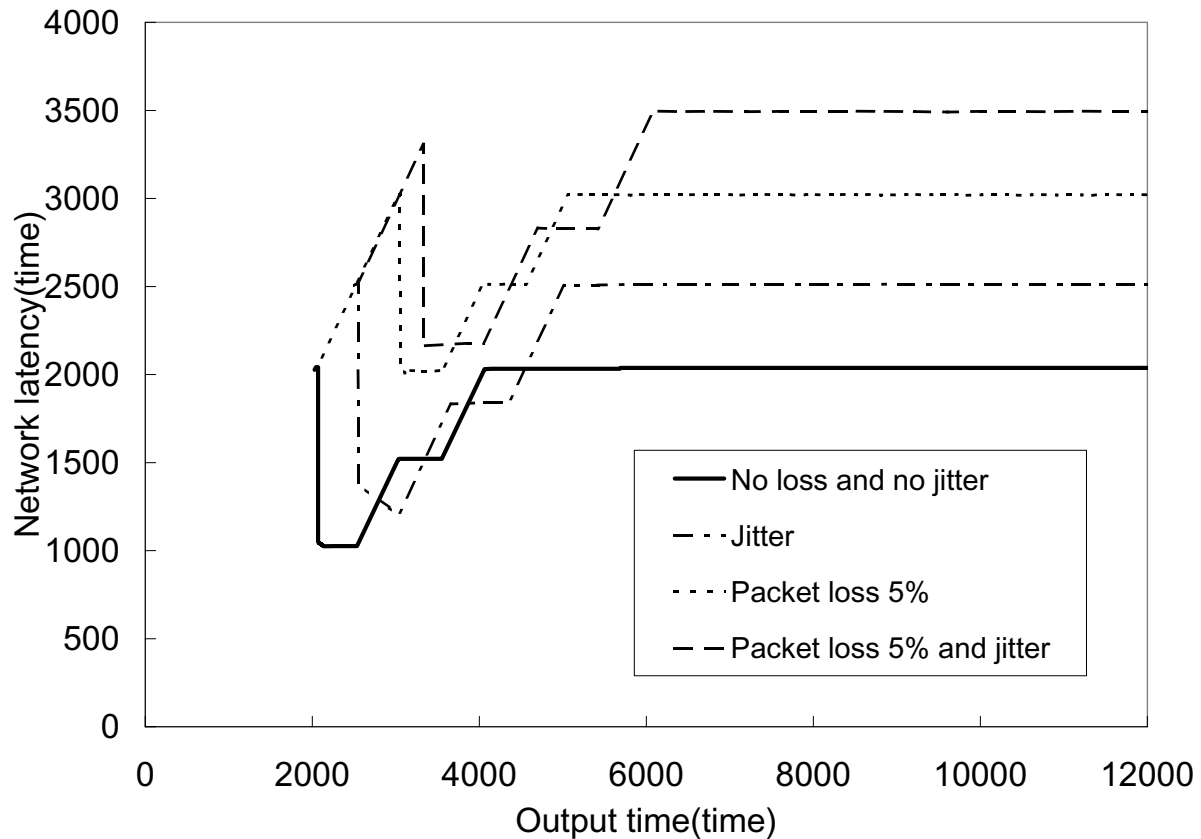


図 4.13 異種ノードで構成された中継経路へのフロー分散順序パターン A の遅延変動とパケットロスに対するネットワークレイテンシー特性

ノード遅延を持つ経路の初期制御時のリプライパケット待ち時間の影響であることがわかる。パターン B の場合，初期制御時，送信番号の小さいパケットが上部の性能の低い中継ノードに接続する AC 部で SI 設定待ちとなる傾向にある。このため送信番号の大きなパケットが先に送出され，順序補正の待ち合わせが必要となり，最大ノード遅延の影響が出る。本評価から，異種ノードから構成される複数経路にフロー分散する場合，最悪の場合でも，最大ノード遅延とノード間の最大遅延ギャップを基準にレイテンシーが想定可能であることが分かった。

## (2) 遅延変動とパケット廃棄のある場合

### (a) 単一フローによる評価

図 4.13 は，上述のパターン A に対して，遅延変動かパケットロス，およびその両

方の影響がある場合のネットワークレイテンシー特性である。遅延変動のみの場合、ほぼ 500(time) だけのネットワークレイテンシー増加で収まっている。遅延変動によるレイテンシーは、単一リンクのフローであれば、通常はその影響が積算されていくと考えられるが、本評価のように遅延変動の生じるノードの存在するリンクを並列して用いる場合、その影響は相殺され、遅延の増加が 500(time) 程度で収束することがわかった。パケットロスのみの場合はほぼ 1000(time) ネットワークレイテンシーが増加している。これは、この評価モデルの受信側高速ノード順序補正待ち合わせ時間を 1000(time) としているためであり、パケットロスの発生に伴った、パケット待ち合わせ時間がネットワークレイテンシーに積算されたと考えられる。遅延変動とパケットロスの両方が起こる場合は、これらの積算遅延分である、ほぼ 1500(time) だけネットワークレイテンシが増加することがわかる。いずれのケースも時間の経過とともにネットワークレイテンシーが一定となり、特性が安定することが分かった。

#### (b) 複数フローによる評価

図 4.14 は、本フロー分散制御方式を適用した複数の送信側高速ノードが、一部の中継ノードを共有し、共通の受信側高速ノードへトラフィックを分散させて送出する場合の評価モデルである。この評価でも前述の評価と同様に、中継ノードは 1/LP が 3, 5, 15(time/packet)、それぞれの処理遅延が 500, 1000, 1500(time) のものを 3 種類を想定し、受信側高速ノードに接続する経路は 9 経路とする。この 9 経路への中継ノードの割り当て方法は、前述のパターン A と同様とする。送信側高速ノード数は 2 とし、それぞれ複数経路を 6 つ持つ。そのうち送信側高速ノード間で共有する経路数は 3 つとした。また、この評価で用いた順序補正待ち合わせ時間は 3000(time) とした。

図 4.15 は、図 4.14 の評価モデル構成における flow0 に着目したネットワークレイテンシー特性である。単一の送信側高速ノードを適用した場合の特性である図 4.13 に比べて、ネットワークレイテンシーの増加が見られる。例えば、パケットロ

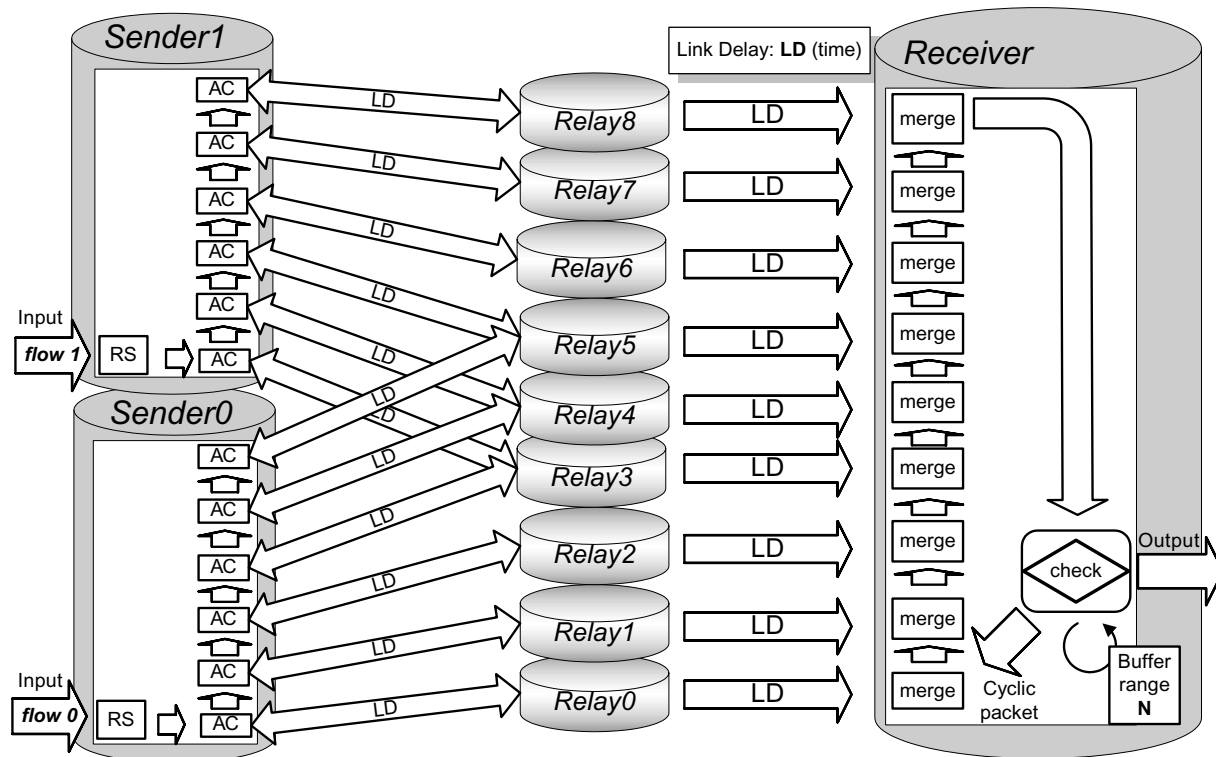


図 4.14 応用評価モデル

すなし，遅延変動なしの場合を図 4.15 と図 4.13 の特性から比較すると，図 4.15 のネットワークレイテンシーは，図 4.13 の場合の 2 倍近い値を示している．これは，両者の特性評価が同じ中継ノード構成を用いており，図 4.13 の評価時に単一のフローのみで用いていた帯域を図 4.15 の評価時には 2 つのフローで共有しているためである．すなわち，フローあたりの可用帯域がほぼ半分であるため，ネットワークレイテンシーが 2 倍になっている．遅延変動がある場合の特性も，使用できる帯域が狭いため，大きな遅延のあるリンクを自律的に避けることが難しくなり，この結果，図 4.13 の評価に比べてネットワークレイテンシーが増加している．パケットロスがある場合は，この評価で適用した順序補正待ち合わせ時間 3000(time) がネットワークレイテンシーに積算されていることが分かる．遅延変動とパケットロスのある場合は，これらの遅延積算分がネットワークレイテンシーに反映されている．以上のことは，図 4.16 の評価結果についても，同様なことが言える．ただし，図 4.15 と図 4.16 を比較すると，遅延変動がない場合に，若干図 4.16 の方がネッ

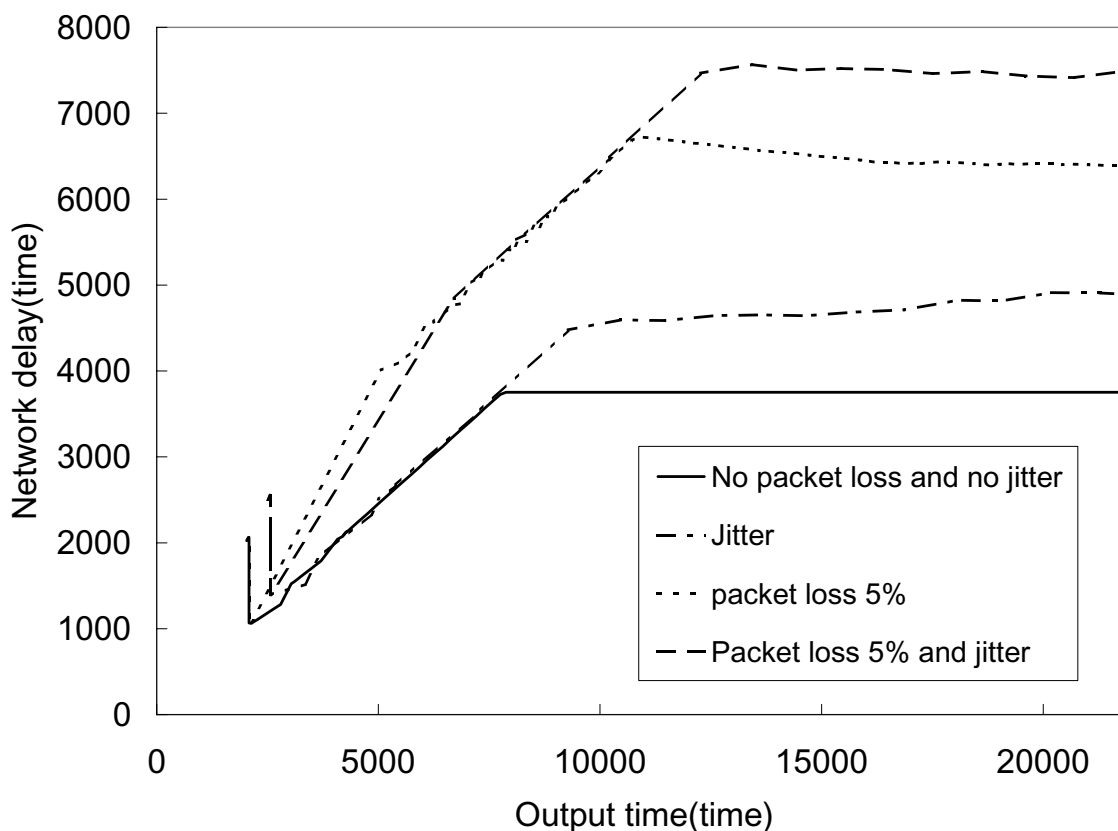


図 4.15 送信側高速ノードが 2 つの場合のネットワークレイテンシー特性 (flow 0)

トワークレイテンシーが少なく、最大値に達した時点で変動が見られる。この原因は、接続構成に起因すると考えられる。入力されたパケットはまず、送出側高速ノードの最下部の AC 部において、出力の可否を判定される。この評価構成では、flow1 が入力する送出側高速ノード (Sender1) の下部の AC 部は、別の送出側高速ノード (Sender0) の上部 AC 部とも接続される中継ノードに接続している。そのため、出力可否の判定には、Sender0 のトラフィックの影響を受ける。この結果、入力されたパケットが下部の AC 部で出力できるか、上位 AC 部に送られるかが、他のフローの影響を受け変動しやすいのでネットワークレイテンシーが若干不安定になると思われる。しかし、このような flow0 と flow1 の特性の差異は、遅延変動がある場合には見られない。つまり、遅延変動の影響は、このような接続構成に起因する特性よりも支配的であることが分かる。



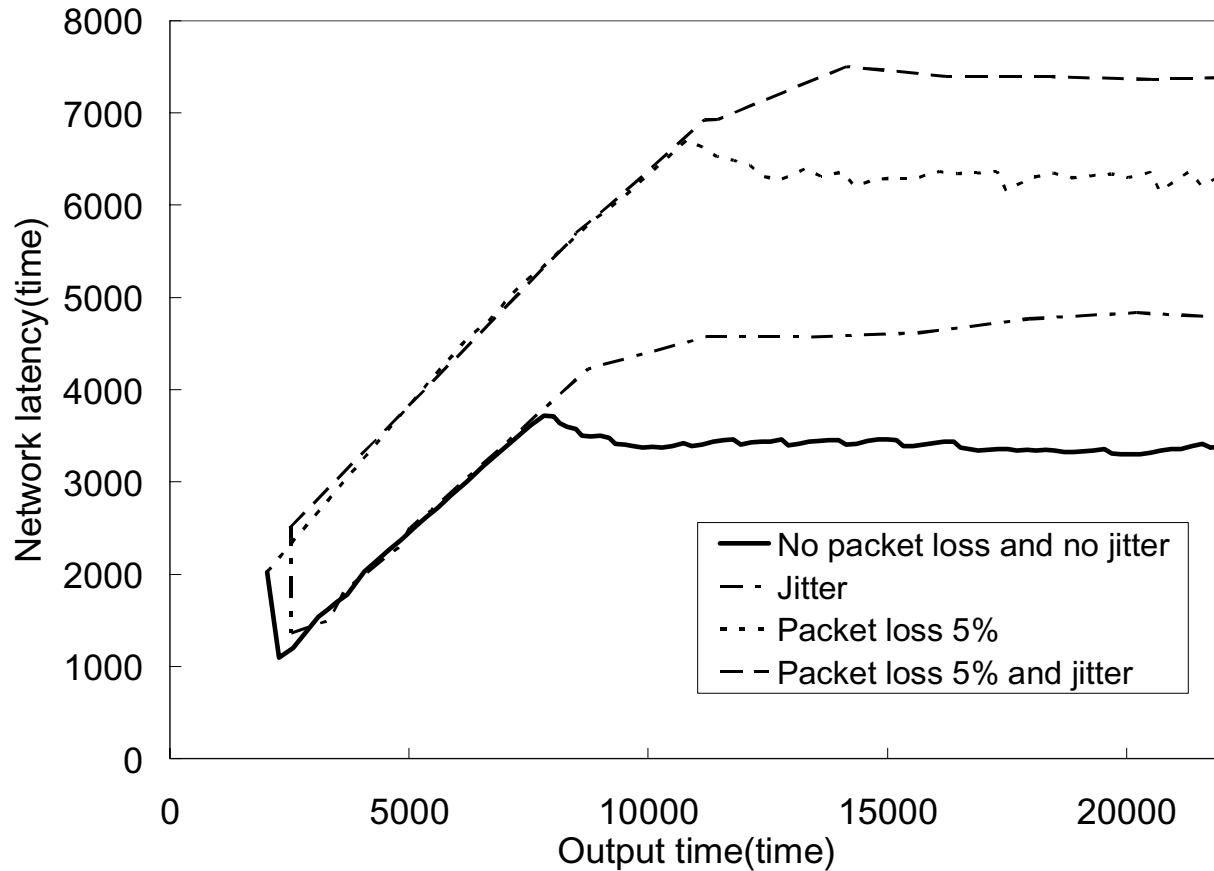


図 4.16 送信側高速ノードが 2 つの場合のネットワークレイテンシー特性 (flow 1)

これらのネットワークレイテンシー特性は、いずれのケースもほぼ一定値に安定し、その値は想定可能なものである。

### 4.3.3 例外的なケースの評価

次に、送出側高速ノードから複数の中継ノードに分散されたトラフィックが、中継ノードの設定誤り、または本方式の適用誤りによって、受信側高速ノードを経由せず、中継ノードの一部から直接宛先ノードに送られるトラフィックが発生する場合の性能評価を行う。

この場合、該当する中継ノードに到着したパケットは、受信側高速ノードに到達しないためパケットロスとみなされ、順序補正待ち合わせ時間を経過してから、続きの番号を持つパケットが受信ノードから送出される。この順序補正待ち合わせのための時間が、受信側高速ノードで待ち合わせしたパケットの遅延時間となる。本フロー分散制御方式を適用せず、当該

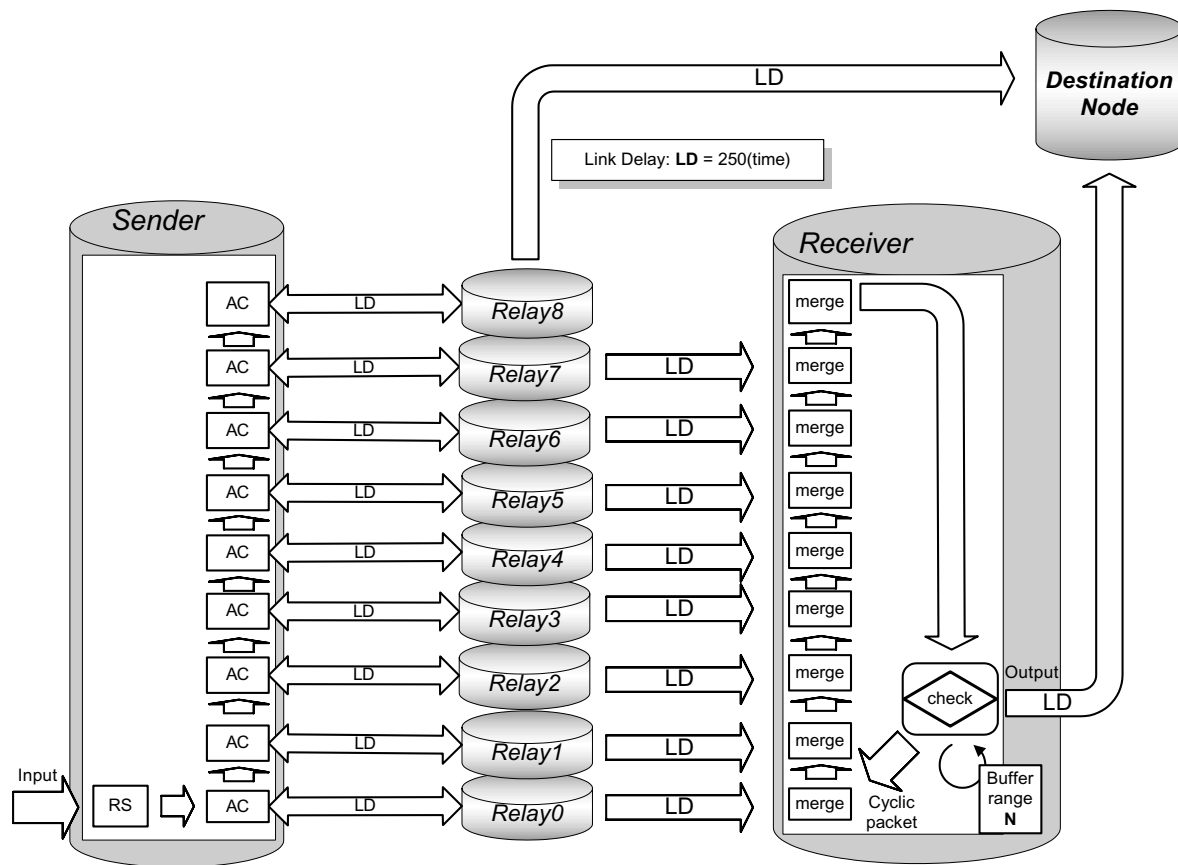


図 4.17 例外ケースの評価モデル

中継ノードのみを用いて、宛先ノードにトラフィックを送出する場合の性能と、本フロー分散制御方式を適用し、当該中継ノードのためパケットロスの誤認識が起こる状態で、宛先ノードへのトラフィックを分散・集約させた場合の性能を比較評価する。

この比較評価の構成図は、評価モデル図 4.17 のように示される。ここで、各中継ノードの転送間隔、および処理遅延は前節ののパターン A と同様とする。最上部の中継ノード（転送間隔 3(time)、処理遅延 500(time)）は、送信側高速ノードから受信したトラフィックを受信側高速ノードに送出せず、直接宛先ノードに送出するものとする。宛先ノードから当該中継ノードまで、また受信側高速ノードまでの伝送遅延はそれぞれ 250(time) とする。また、受信側高速ノードの順序補正待ち合わせ時間は 1000(time) とした。

この条件で総計 5000 個のパケットが宛先ノードに届くまでの時間を比較評価したもの

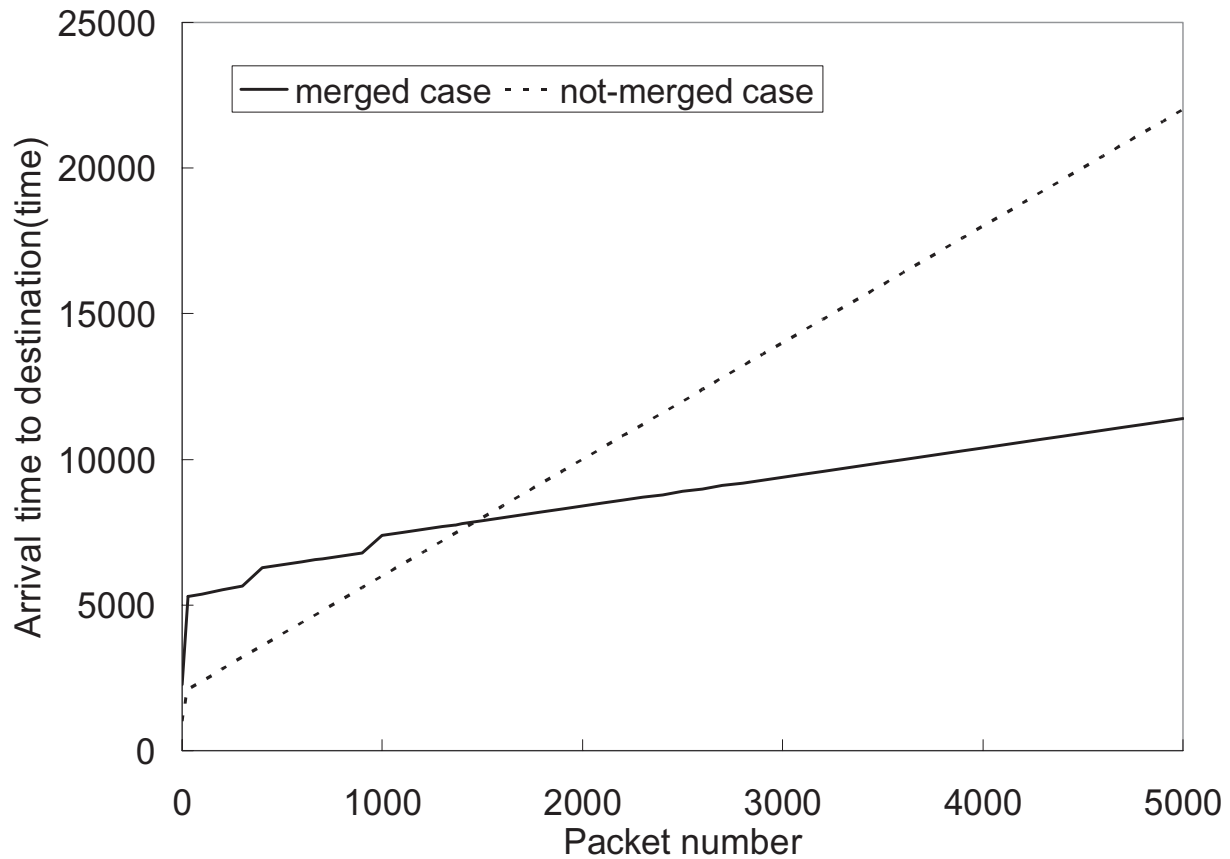


図 4.18 フロー到着時間比較

が、図 4.18 である。この結果より、本フロー分散制御方式の PACKET ロスの誤認識による順序補正待ち合わせ時間の遅延蓄積よりも、中継ノード単体で PACKET を送出していくことによる遅延蓄積の方が大きく、このような場合でも本方式の適用効果があることがわかる。

以上の結果より、実ネットワークを想定した、大きな遅延時間差、遅延変動、また PACKET ロスを与えた評価の結果からも、自律分散型制御アーキテクチャに基づいた、本フロー分散制御方式の有用性と適用の優位性を示すことができたと考える。

#### 4.4 まとめ

本章では、自律分散型ネットワーク QoS 制御方式を、ノードレベルの QoS 制御方式である優先キューイング制御方式に適用した、自律分散型優先キューイング制御方式と、ネット

ワークレベルの QoS 制御方式であるフロー制御方式に適用した，自律分散型のフロー分散制御方式を評価モデルとして具体化し，この評価モデルに基づく評価において，その特性を明らかにした．また，実用に則した評価として，既存方式との比較や既存のネットワーク環境条件に基づく評価を行い，本方式の優位性を定性的あるいは定量的に示した．

本章の評価結果を要約すると次の通りとなる．

- 自律分散型制御に基づく優先キューイング制御方式

- － 基本構成における解析評価と本評価のために作成したシミュレータによる評価との比較

同一条件下の評価結果が一致し，シミュレータの信頼性が確認された．また，この優先キューイング制御方式が原理的に優先制御向きであることが示された．

- － 遅延制御と廃棄制御を機能分散させた構成における特性評価

基本構成に遅延制御や廃棄制御を様々な組み合わせた評価を行い，それぞれの特性とこれらの制御に必要な適切なパラメータ値を確認した．さらに，それらのパラメータ値を用いて，最も性能の良い特性を示した組み合わせ構成が，内部廃棄制御と事前廃棄制御を行い，優先クラスリマーク法による遅延制御を組み合わせたものであることが確認できた．

- － 既存方式との比較評価

既存の優先キューイング制御方式として，遅延特性は WFQ 方式，廃棄特性は WRED 方式を比較対象として，前述の評価の中で最も良い特性の得られた構成による本優先キューイング制御方式との比較評価を行い，本優先キューイング制御方式が，既存の優先キューイング制御方式に比較して，全般に低遅延であり，優先度別に所要遅延の差別化ができ，輻輳度に応じた緩やかな廃棄制御特性を示せることが確認できた．

- － 順序補償方式の適用効果に関する比較評価

本優先キューイング方式の最も基本的な内部廃棄のみ行う制御構成において，順序

補償方式を適用した場合の遅延特性の比較を行った結果，全般に遅延特性が改善され，遅延制御を行わない基本的な構成であっても，広範な輻輳状態において遅延差別化が行える特性が示された．

- 自律分散型制御に基づくフロー分散制御方式

- 特性評価

複数経路に用いたノードが同種の場合と異種の場合の特性評価を行った結果，同種ノードの性能に対して，ネットワークレイテンシーや最大スループットの達成時間のスケーラビリティがあり，ノードの性能に応じて最大遅延時間が線形的に見積もれることが確認された．また，異種ノードから構成される複数経路の場合，性能の低いノードから順にフローを分配する方が，ネットワークレイテンシーを低くできることが示された．

- ネットワークへの適用評価

実際のネットワークを想定した，ノードの処理遅延が大きな場合でも，特性評価で見られた特性が示され，有用な結果を示せることが確認できた．中継経路において遅延変動やパケットの廃棄がある場合，ネットワークレイテンシーの増加は見られるが，タイムアウト値などの設定値からこの遅延増加は想定可能なものであることが確認できた．

- 例外ケースの場合の評価

フロー分散や中継経路の設定誤りにより，本来のフロー集約ノードにパケットが届かなくても，適切なタイムアウト値を設定しておくことによって，フロー分散させずに，フローを伝達する場合と比較して，性能改善ができることが確認できた．

これらの評価結果から，ノードレベルの評価，そしてネットワークレベルの評価のいずれの場合も，従来方式では解決できない，ノードに内在する遅延にかかわる問題を解消または緩和できることが明らかにされた．

# 第 5 章

## 結 論

本論文では、飛躍的進展の続く今後将来のネットワークにおける、適切かつ有用な QoS 制御方式の在り方について述べ、既存の QoS 制御方式の適用時に顕在化しつつある問題を根本的に解決する、新しい QoS 制御方式を提示した。また、この QoS 制御方式に採用した自律分散型制御アーキテクチャが、今後将来のネットワークにおいてどのように有用であり、かつ、顕在化する問題をどのように解決できるかについて論じ、さらに、本 QoS 制御方式の具体的な適用例として、QoS 制御方式において最も重要な制御機構の役割を担う、キューイング制御とフロー制御における自律分散型アーキテクチャを提示し、定量的に評価によってこれらの特性を示し、評価結果を考察した。

本研究で得られた成果を以下にまとめる。

- ネットワークの進展に伴う問題定義とその解決法の提示

本研究では、まず第一に、ネットワークの大容量化やノード技術の進展の進む中、それらの技術の導入により、ネットワークの性能を向上させようと意図しても、期待した性能が得られず、また得られたとしても安定せず、あるいは、性能が劣化する場合もあるという問題に着目した。

この問題の根本的な原因は、ビットレートの飛躍的向上に伴い、相対的に顕在化しつつあるノードの処理遅延であり、この処理遅延の主要な要因であるキューイング遅延を可能な限り縮退させることが、本問題を解決する重要なポイントであることを示した。

本研究では、この問題の解決法として制御機能を分散させ、それら機能部の中で部分的に遅延を生じても局所的に解決でき、フローの伝達過程において遅延を蓄積しない、自

律分散型制御アーキテクチャの導入を提案した。

- 自律分散型ネットワーク QoS 制御方式の具体的構成例の提示

前述のキューイング遅延に大きな影響を受ける QoS 制御は、ノードレベルでは優先キューイング制御であり、ネットワークレベルの QoS 制御では、フロー制御が挙げられることを述べ、自律分散型ネットワーク QoS 制御方式の具体的効果を確認するために、自律分散型アーキテクチャを用いた、これらの制御方式の具体的構成例を示した。

- 自律分散型アーキテクチャに基づく優先キューイング制御方式

自律分散型アーキテクチャに基づく自己タイミング型パイプライン STP を応用することにより、キューイング遅延を低減できる優先キューイング制御方式が実現できることを示した。

- 自律分散型アーキテクチャに基づくフロー分散制御方式

自律分散型アーキテクチャを用いることにより、フローの送出先のキュー情報の変化に応じて、フローを送出するか、あるいは別経路へ分散させるかを、簡易な仕組みで判断する機能部を分散配置できる。この機能部の分散構成により、送出先の処理性能に応じた適切なフロー分散制御方式が実現できることを示した。

- 自律分散型ネットワーク QoS 制御方式の評価モデルの提示とその評価

前述の優先キューイング制御方式やフロー分散制御方式に対してその評価手法を提示し、この評価手法に基づく評価モデルを示した。また、この評価モデルに基づき、作成したシミュレータによるシミュレーション評価により、特性評価および実用に則した評価を行った。

- 自律分散型アーキテクチャに基づく優先キューイング制御方式

本優先キューイング方式が、STP の各ステージにパケットが存在する確率に基づいたマルコフ連鎖モデルで表すことができ、このモデルに基づく定式化により、解析可能であることを示した。さらに、STP の動作を模擬して作成した離散事象シミュレータとこの解析結果が一致することが確認され、シミュレーション評価の信頼性を明らかにし、この優先キューイング制御方式が原理的に優先制御向きである

ことが確認された。

本優先キューイング方式は、STP の各パイプラインステージに機能分散させて遅延制御や破棄制御を行うことにより、従来のキューイング制御方式には見られないユニークな優先キューイング制御を可能とし、これらの機能の組み合わせにより、多様な特性を示すことが確認された。

既存の優先キューイング制御方式として、遅延、および廃棄特性の差別化において、それぞれ代表的な方式である WFQ、および WRED 方式を対象とした比較評価から、本優先キューイング制御方式は、WFQ 方式と比較して低遅延でキューイングでき、かつ優先度に応じた遅延の差別化を行えることが確認され、また廃棄特性においても、WRED 方式と比較して、輻輳度に応じた緩やかな廃棄特性を持つことが確認された。

さらに、自律分散型アーキテクチャにも対応できる順序補償方式を提示し、この方式の適用によって、広範な輻輳時でも遅延特性が改善でき、より多様な状況に対応できる優れたキューイング方式が実現できることが見込まれた。

#### – 自律分散型アーキテクチャに基づくフロー分散制御方式

本フロー分散制御方式は、フロー分散にかかる AC 部やフロー集約にかかる merge 部が繰り返し配置される構成を採るため、評価モデル化が容易である。これらの機能部は自律動作をするため、構成や機能上のスケーラビリティが高い。同一の処理性能を持つノード（同種ノード）で複数経路を構成した評価モデルを提示し、この評価モデルにおいて、ノードの処理性能を変化させた時のネットワークレイテンシーの特性が線形的変化を示すことから、これらの特質が確認された。

本フロー分散制御方式は、各機能部が自律動作を行うため、経路上の処理遅延が大きく、かつ変動しても動的に対応が可能である。経路中でパケットが損失しても、適切なタイムアウト時間の設定により、フロー性能に大きな影響を与えることなく、フローの導通が行える。また、複数のフローが混在した場合でも、それぞれのフロー分散のためのスケジュール操作が不要である。実際のネットワーク環境を想



定した，処理遅延や遅延変動，パケット廃棄を発生させた評価モデルを構築し，この評価モデルにおいて，単一のフローの導通と複数フローの導通を評価した結果，安定した特性が得られたことから，これらの特質が確認された．

インターネットは，構築時の本来の目的に照らしてみても，また，その後の発展経過を振り返ってみても，自律分散的に制御されるべきネットワークであることが伺える．今まで十分な自律分散制御アーキテクチャを用いなくとも，本論文で提示したような問題が顕在化しなかった理由は，かつてはネットワークの可用帯域に対して，ネットワークノード性能に対するエンドホストの性能がかなり低く，エンドホストの利用できる接続回線の占める帯域も小さく，またネットワークを利用するサービスやアプリケーションも限られていたため，遅延に対して寛容であり，フローが隘路に遭うことも少なかったからであろう．

しかし，継続的な広帯域大容量化とノードやホストの性能向上が顕著であり，ネットワークで取り扱われるサービスやアプリケーションも飛躍的に増え続けていくであろう，今後将来のネットワークでは，その内部に性能格差にともなう不均質さを持つと，この不均質さを無視したフローの導通によって，たちまち隘路を生じかねない．しかしながら，拡大を続け構成の流動的なネットワークにおいて，これらの不均質さを考慮した緻密なリソース配分は不可能である．したがって，本論文で述べるようなフローの導通に対して自律分散的に制御の行える方式が必須となってくる．

自律分散制御を徹底したネットワークとはどのようなものか，ここで考えてみたい．ノード内の機能モジュールの自律分散化，そして，この機能モジュールを構成するプロセッサレベルの自律分散化を進め，プロセッサを構成するチップ稼動制御の自律分散化を行う．さらに，ここで制御されるパケットまでを細分化したビットレベルで自律分散制御する．このように自律分散制御を突き詰めて行くと，流体を扱うような制御に近づいていくと考えられる．すなわち，川が流れるように自然に情報トラフィックが疎通し，流れが堰きとめられると，流れることが可能な部分を探して自然に情報トラフィックが疎通していく，そのような自然の現象に近い動作をするネットワークへと近づいていくと想像できる．こうして，自律

分散化を極めたネットワークは人々にネットワークを意識させない，あらゆる形式の情報が疎通する，仮想的な自由空間として，その存在意義を変えていくのかもしれない．

今日のインターネットの飛躍的发展と，それに伴い生じつつある新しい遅延の問題は，ネットワーク制御において，本来の自律分散性の意味と，今後の目指すべき方向性を再考すべき転機であるとも考える．本論文で述べた自律分散型ネットワーク QoS 制御方式が，今後のインターネットの進展を支える礎の一端となれば幸いである．



# 謝 辞

本研究の全過程を通して、終始懇切なる御指導、御鞭撻、御示唆を賜った高知工科大学副学長寺田浩詔教授に衷心より感謝の意を表する。

本研究の遂行ならびに本論文の作成にあたり、終始御指導、御助言を頂き、常に温かい御激励を下された高知工科大学岩田誠教授に深謝の意を表する。

筆者の高知工科大学大学院基盤工学専攻の博士課程に在籍して以来、ご教示を賜り、出向先であった、通信・放送機構ギガビットネットワーク研究開発プロジェクトにおいても、新たな研究分野への手ほどきや御示唆をいただき、研究全般に渡る御助言や、温かい御激励を下された、同プロジェクトサブリーダーも兼任される高知工科大学島村和典教授に厚く御礼申し上げます。

筆者が高知工科大学大学院基盤工学専攻の博士課程に在籍して以来、情報工学一般及び各専門分野に関しご教示頂き、研究全般にも御示唆、御助言を賜った、高知工科大学坂本明雄教授、岡田守教授、清水明宏教授、篠森敬三教授、菊池豊助教授に厚く感謝する。

筆者が高知工科大学大学院基盤工学専攻の博士課程に在籍して以来、種々の面でひとかたならぬ御助言、御激励をいただいた、高知工科大学福本昌弘助教授、浜村昌則講師、妻鳥貴彦講師、大森洋一助手（現、九州大学）に厚く御礼申し上げます。

本研究の実施、ならびに、本論文の執筆の機会を与えてくださり、温かいご支援をくださった、日本テレコム株式会社専務執行役技術本部本部長兼 CTO 弓削哲也博士、同社常務執行役戦略企画本部副本部長中野豊氏、同社戦略企画本部情報通信研究所副所長米田進博士に深く感謝の意を表する。

本研究の実施に伴い、種々の技術的な御討論や有益な御助言を下された日本テレコム株式会社戦略企画本部企画部孔令杰氏に心から御礼申し上げます。

また、本研究の実施に伴い、ご協力いただいた通信・放送機構ギガビットネットワーク研究開発プロジェクトの関係諸氏、日本テレコム株式会社戦略企画本部の関係諸氏、高知工科

大学情報システム工学科岩田研究室および島村研究室の関係諸氏に深く感謝の意を表す次第である。

# 参考文献

- [1] B. Goode, “Voice Over Internet Protocol (VoIP),” *Proc. of the IEEE*, vol.90, no.9, pp.1495–1517, Sep. 2002.
- [2] 林, 岩田, 孔, 米田, 寺田, “超高帯域光ネットワークにおける分散型マルチプロトコルルータ,” 情報処理学会研究報告 2001-DSM-22, pp.1–6, July 2001.
- [3] Arvind and R. A. Iannucci, “A Critique of Multiprocessing von Neumann Style,” *Proc. of the 10th International Symposium on Computer Architecture*, pp. 426–436, June 1983.
- [4] S. Floyd, and V. Jacobson, “Random Early Detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol.1, no.4, pp.397–414, Aug. 1993.
- [5] D. Lin and R. Morris, “Dynamics of Random Early Detection,” *Proc. of ACM SIGCOMM '97*, 1997.
- [6] D. Clark and W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service,” *IEEE/ACM Transactions on Networking*, vol.6, no.4, pp.362–373, Aug. 1998.
- [7] <http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm>
- [8] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, “Recommendations on Queue Management and Congestion Avoidance in the Internet,” *RFC2309*, April 1998.
- [9] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulations of a Fair Queuing Algorithm,” *Proc. of ACM SIGCOMM '89*, 1989.
- [10] A.K. Parekh and R.G. Gallager, “A Generalized Processor Sharing Approach to

- Flow Control in Integrated Services Networks: The Single-Node Case,” *IEEE/ACM Transactions on Networking*, vol.1, No.3, pp.344–357, June 1993.
- [11] J.C.R. Bennett and H. Zhang, “WF<sup>2</sup>Q: Worst-case Fair Weighted Fair Queueing,” in *Proc. of IEEE INFOCOM '96*, 1996.
- [12] S.J. Golestani, “A Self-Clocked Fair Queueing Scheme for Broadband Applications,” in *Proc. of IEEE INFOCOM '94*, 1994.
- [13] M. Shreedhar and G. Varghese, “Efficient Fair Queueing using Deficit Round Robin,” *Proc. of ACM SIGCOMM '95*, 1995.
- [14] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, “Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip,” *IEEE Journal on Selected Areas in Communications.*, vol.9, no.8, pp.1265–1279, Oct. 1991.
- [15] S. Floyd and V. Jacobson, “Link-sharing and Resource Management Models for Packet Networks,” *IEEE/ACM Transactions on Networking*, vol.3, no.4, pp.365–386, Aug. 1995.
- [16] Y. Bernet, J. Binder, S. Blake, M. Carlson, B.E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss, “A Framework for Differentiated Services,” *draft-ietf-diffserv-framework-02.txt*, Feb. 1999.
- [17] W.R. Stevens, “TCP/IP illustrated vol.1,” *Addison-Wesley*, 1994.
- [18] G.R. Eright and W.R. Stevens, “TCP/IP illustrated vol.2,” *Addison-Wesley*, 1995.
- [19] W. Stevens, “TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms,” *RFC2001*, Jan. 1997.
- [20] Allman, M., Paxson, V. and Stevens, W.: TCP Congestion Control, *RFC2581*, April 1999.
- [21] V. Jacobson, “Congestion Avoidance and Control,” *Proc. of ACM SIGCOMM*, pp.314–329, 1988.
- [22] H. Adishesu, G. Parulkar, and G. Varghese, “A Reliable and Scalable Striping

- Protocol,” *Proc. of ACM SIGCOMM*, pp.131–141, 1996.
- [23] J. Wang and K. Nahrstedt, “Parallel IP Packet Forwarding for Tomorrow’s IP Routers,” *Proc. of IEEE Workshop on High Performance Switching and Routing (HPSR’01)*, 2001.
- [24] J. Y. Jo, Y. Kim, H. J. Chao, and F. Merat, “Internet Traffic Load Balancing using Dynamic Hashing with Flow Volume,” *Proc. of SPIE ITCOM*, 2002.
- [25] H. Lu and I. Faynberg, “An Architectural Framework for Support of Quality of Service in Packet Networks,” *IEEE Communication Magazine*, vol.41, no.6, pp.98–105, June 2003.
- [26] 阪田, “インターネットにおける QoS 制御,” *信学誌*, vol.85, no.10, pp.749–755, Oct. 2002.
- [27] X. Xiao and L.M. Ni, “Internet QoS: A Big Picture,” *IEEE Network*, vol.13, no.2, pp.8–18, March/April 1999.
- [28] 戸田, “ネットワーク QoS 技術,” オーム社, 2001.
- [29] S. Vegesna, “IP QoS 完全ガイド,” ソフトバンクパブリッシング, 2002.
- [30] R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: An Overview,” *RFC1633*, June 1994.
- [31] T. Aimoto and S. Miyake, “Overview of DiffServ Technology: ITS Mechanism and Implementation,” *IEICE Trans. Inf. & Syst.*, vol.E83-D, no.5, pp.957–964, May 2000.
- [32] B.E. Carpenter and K. Nichols, “Differentiated Services in the Internet,” *Proc. of the IEEE*, vol.90, no.9, pp.1479–1494. Sep. 2002.
- [33] H. Terada, S. Miyata, and M. Iwata, “DDMP’s: Self-Timed Super-Pipelined Data-Driven Multimedia Processors,” *Proc. of the IEEE*, vol.87, no.2, pp.282–296, Feb. 1999.
- [34] 細美, 林, 岩田, 寺田, “自己同期パイプラインによる Diffserv 制御機構,” 平成 13 年度



- 電気関係学会四国支部連合大会講演論文集, pp.199, Sep. 2001.
- [35] 細美, 岩田, 林, 寺田, “自己同期パイプラインによる SLA 制御の実現法,” 2002 年電子情報通信学会総合大会,” B-7-112, March 2002.
- [36] H. Hayashi, M. Iwata, T. Hosomi, and H. Terada, “A Self-Timed Pipeline Implementation of Class-Based QoS Control,” *International Workshop on Embedded Systems at IEEE High Performance Computing Conference 2001*, pp.6–10, Dec. 2001.
- [37] H. Hayashi, M. Iwata, and H. Terada, “An Autonomous Class-Based QoS Control Utilizing a Self-Timed Folded Pipeline,” *The 4th International Conference on Advanced Communication Technology*, pp.469–474, Feb. 2002.
- [38] 朝日山, 林, 岩田, 寺田, “折り返し型自律 QoS 制御方式におけるジッタ抑制法の一考察,” 2003 年電子情報通信学会ソサイエティ大会, B-6-78, Sep. 2003.
- [39] 林, 岩田, 寺田, “クラス別 QoS 制御向き事故同期型優先キューの性能評価,” FIT2003 情報科学技術フォーラム情報技術レターズ, Vol.2, pp.313–314. Sep. 2003.
- [40] 菅野, 日下, 松田, 木幡, 白鳥, “IP ネットワークにおけるパケットスケジューリング方式 IPS の提案と評価,” 情処学論, vol.40, no.1, pp.66–75, Jan. 1999.
- [41] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,” *RFC2597*, June 1999.
- [42] 小倉, 大石, 岩田, 寺田, “自己同期型折り返しパイプラインの一実現法,” 2003 年電子情報通信学会総合大会, A-1-31, March 2003.
- [43] M. Iwata, M. Ogura, Y. Ohishi, H. Hayashi, and H. Terada, “100MPackets/s Fully Self-Timed Priority Queue: FQ,” *2004ISSCC*, 8. 1. Feb. 2004.
- [44] 林, 岩田, 寺田, 島村, “折り返し型自律優先キューイング機構における順序補償方式の一考察,” 2004 年情報処理学会第 66 回全国大会, 4J-4, March 2004.
- [45] H. Terada, “Impact of Photonic Technology on the Future Communication,” *IEICE Trans. Comm.*, vol.E77-B, No.2, pp.96-99, Feb. 1994.

- [46] J. Moy, “OSPF Version 2,” *RFC2178*, July 1997.
- [47] J.C.R. Bennett, C. Partridge, and N. Shectman, “Packet Reordering is Not Pathological Network Behavior,” *IEEE/ACM Transactions on Networking*, Vol.7, No.6, pp.789–798, 1999.
- [48] B. Wydrowski and M. Zukerman, “QoS in Best-Effort Networks,” *IEEE Communication Magazine*, Vol.40 No.12, pp.44–49, Dec. 2002.
- [49] J. Postel, “Internet Control Message Protocol,” *RFC792*, Sep. 1981.
- [50] S. Keshav, “A Control-Theoretic Approach to Flow Control,” *Proc. of ACM SIGCOMM*, pp.3–15, 1991.
- [51] [http://www.juniper.co.jp/solutions/literature/test\\_report/lightreading03.html#01](http://www.juniper.co.jp/solutions/literature/test_report/lightreading03.html#01)
- [52] <http://www.hitachi.co.jp/Prod/comp/network/gr2000/>
- [53] <http://www.hitachi.co.jp/Prod/comp/network/switch/summit/>



# 付録 A

## 折り返し型 STP のマルコフ連鎖 モデル

自律分散型優先キューイング制御方式の折り返し型 STP の各ステージ上のパケットの動作は、ある時刻における各ステージでのパケットの存在確率の離散時間推移により表現できる。このパケットの存在確率は、一定の過去の状態によってのみ定まり、マルコフ過程として取り扱うことが可能である。提案方式の取り扱う優先クラス数が  $k_{tot}$  の有限値である場合、各ステージの状態は、いずれかのパケットが存在する  $k_{tot}$  通りの状態と、いずれのパケットも存在しない 1 通りの合計、 $k_{tot} + 1$  通りの整数個の状態空間がある。したがって、離散時間型有限マルコフ連鎖としてモデル化が行える。

以下に、最上段の往路側を  $[m, 0]$  段、復路側を  $[m, 1]$  段とする折り返し型 STP の、 $[i, 0]$  段及び  $[i, 1]$  段 ( $0 \leq i \leq m$ ) における、一般化した状態の確率を求める式を記す。特に説明のない限り、ここで用いる確率は 2.3 節で用いたものと同じである。

### A.1 パケットの存在確率 $ON$

#### A.1.1 往路側ステージの $ON_{i,0}$

$[i, 0]$  段にパケットの存在する確率  $ON_{i,0}(k, t)$  は、

(1) 新たに入力されるパケットの確率

$$\lambda_i(k, t) \times \{1 - D_A(k, q(k, t))\}$$

(2) 下段の packets が当該ステージへ移動する確率

$$r_{i-1,i}(k, t) \times ON_{i-1,0}(k, t-1)$$

(3) 当該ステージに過去に存在した packets が滞留する確率

$$\{1 - r_{i,i+1}(k, t) - p_i(k, t) - D_i(k, t)\} \times ON_{i,0}(k, t-1)$$

によって、決められる。ここで、 $D_A(k, q(k, t))$  は、2.2.2 節で述べた (ii) の事前廃棄制御による廃棄確率を表し、 $D_i(k, t)$  は 2.2.2 節で述べた (i) の内部廃棄制御による廃棄確率を表す。

また、下段の packets の当該ステージへの移動が無い場合のみ、入力 packets を許容すると、実際に入力される packets の確率は、

$$\lambda_i(k, t) \times \{1 - D_A(k, q(k, t))\} \times \{1 - r_{i-1,i}(k, t) \times ON_{i-1,0}(k, t-1)\}$$

と表せる。

最下段の  $[0, 0]$  段では、(2) は移動元が存在しないので、考慮の必要はない。また、最上段の  $[m, 0]$  段では、前方段が存在しないため、(3) における前方段への移動確率  $r_{i,i+1}(k, t)$  については考慮する必要がない。これは、廃棄ステージにおいても同様である。

したがって、これらの確率を 0 として、 $[i, 0]$  段に packets の存在する確率  $ON_{i,0}(k, t)$  の関係式を一般化すると次式のように表される。

$$\begin{aligned} ON_{i,0}(k, t) &= \lambda_i(k, t) \times \{1 - D_A(k, q(k, t))\} \\ &\quad \times \{1 - Q_i(k, t)\} + Q_i(k, t) \\ &\quad + \{1 - S_i(k, t) - D_i(k, t)\} \\ &\quad \times ON_{i,0}(k, t-1) \end{aligned} \tag{A.1}$$

ここで、 $Q_i(k, t)$ 、 $S_i(k, t)$  はステージに依存する確率関数である。これらは、次のとおり表される。

$$D_i(k, t) = \begin{cases} 1 - p_i(k, t) & ([i, 0] \text{ 段が } k \text{ クラスの廃棄ステージの時}) \\ 0 & (\text{それ以外の時}) \end{cases}$$

$$Q_i(k, t) = \begin{cases} r_{i-1,i}(k, t) \times ON_{i-1,0}(k, t-1) & (i \geq 1) \\ 0 & (i = 0) \end{cases}$$

$$S_i(k, t) = \begin{cases} p_i(k, t) & ([i, 0] \text{ 段が } k \text{ クラスの廃棄ステージ, または, } i = m \text{ の時}) \\ r_{i, i+1}(k, t) + p_i(k, t) & (\text{それ以外の時}) \end{cases}$$

### A.1.2 復路側ステージの $ON_{i,1}$

$[i, 1]$  段にパケットの存在する確率  $ON_{i,0}(k, t)$  は ,

(1) バイパスされてくるパケットの確率

$$p_i(k, t) \times ON_{i,0}(k, t - 1)$$

(2) 上段のパケットが当該ステージへ移動する確率

$$r_{i+1, i}(k, t) \times ON_{i+1, 1}(k, t - 1)$$

(3) 当該ステージに過去に存在したパケットが滞留する確率

$$\{1 - r_{i, i-1}(k, t)\} \times ON_{i, 1}(k, t - 1)$$

によって, 決められる .

ここで, 最上段の  $[m, 1]$  段では, 移動元ステージが存在しないため, (2) を考慮する必要はない . また, 最下段の  $[0, 1]$  段では, それより下段のステージがなく, 下段のステージに移動する確率の代わりに, キューから出力される確率を考慮する必要がある . 従って, (3) では,  $r_{i, i-1}(k, t)$  を  $\mu(k, t)$  に置き換える必要がある .

したがって,  $[i, 1]$  段での, パケットの存在確率  $ON_{i,1}(k, t)$  の関係式は次式のように一般化される .

$$ON_{i,1}(k, t) = p_i(k, t) \times ON_{i,0}(k, t - 1) + R_i(k, t) + \{1 - M_i(k, t)\} \times ON_{i,1}(k, t - 1) \quad (\text{A.2})$$

ここで,  $R_i(k, t)$ ,  $M_i(k, t)$  はステージに依存する確率関数であり, 次のとおりとなる .

$$R_i(k, t) = \begin{cases} r_{i+1, i}(k, t) \times ON_{i+1, 1}(k, t - 1) & (i \leq m - 1) \\ 0 & (i = m) \end{cases}$$

$$M_i(k, t) = \begin{cases} r_{i, i-1}(k, t) & (i \geq 1) \\ \mu(k, t) & (i = 0) \end{cases}$$

## A.2 パケットのバイパス確率 $p$

パケットが  $[i, 0]$  段から  $[i, 1]$  段へバイパスする確率  $p_i(k, t)$  は,

- (1) 復路側ステージに, バイパス先への移動が競合する, 自身の優先度以上のパケットが存在しない確率

$$1 - \sum_{l=k}^N ON_{i+1,1}(l, t-1)$$

- (2) バイパス先のステージに存在したパケットが既に移動したか, あるいは存在しない確率

$$\sum_{k=0}^N r_{i,i-1}(k, t-1) + \left\{ 1 - \sum_{k=0}^N ON_{i,1}(k, t-1) \right\}$$

によって決められる.

ここで,  $N$  は  $k$  の最大値である. また, 最下段の  $[0, 1]$  段では, (2) において, A.1.2 項 (3) と同様の理由から,  $r_{i,i-1}(k, t-1)$  を  $\mu(k, t-1)$  に置き換える必要がある.

パケットがバイパス可能な場合は まず 移動元ステージにパケットが存在し ( $ON_{i,0}(k, t-1)$ ), かつ上記 (1)(2) を同時に満たす必要がある. したがって, 確率  $p_i(k, t)$  を一般化して表すと, 次式のように表される.

$$p_i(k, t) = \left\{ 1 - \sum_{l=k}^N ON_{i+1,1}(l, t-1) \right\} \times ON_{i,0}(k, t-1) \times \left[ \sum_{k=0}^N M_i(k, t-1) + \left\{ 1 - \sum_{k=0}^N ON_{i,1}(k, t-1) \right\} \right] \quad (\text{A.3})$$

ここで,  $M_i(k, t)$  は前節と同様の確率関数である.

## A.3 パケットの前方段移動確率 $r$

### A.3.1 往路側ステージの $r_{i,i+1}$

パケットが, 往路側ステージの  $[i, 0]$  段から, 前方段の  $[i+1, 0]$  段へ移動する確率  $r_{i,i+1}(k, t)$  は,

- (1) 前方段のパケットがバイパスするか，さらに前方のステージへの移動するか，または，廃棄されたか，あるいは，もともと存在しない確率

$$\sum_{k=0}^N \{r_{i+1,i+2}(k, t-1) + p_{i+1}(k, t-1) + D_{i+1}(k, t-1)\} + \left\{1 - \sum_{k=0}^N ON_{i+1,0}(k, t-1)\right\}$$

- (2) 移動元のパケットがバイパスせず，廃棄されずに存在している確率

$$1 - p_i(k, t) - D_i(k, t)$$

によって決められる．

ここで， $[m-1, 0]$  段のステージにおいては，その前方段が最上段の  $[m, 0]$  段であるため，(1)における  $r_{i+1,i+2}(k, t-1)$  の考慮が不要である．これは，廃棄ステージの下段のステージにおいても同様である．

パケットが前方段に移動する場合は，まず，移動元ステージにパケットが存在し ( $ON_{i,0}(k, t-1)$ )，かつ上記 (1)(2) を同時に満たす必要がある．したがって，確率  $r_{i,i+1}(k, t)$  を一般化して表すと，次式のように表される．ただし， $0 \leq i \leq m-1$  とする．

$$\begin{aligned} r_{i,i+1}(k, t) &= ON_{i,0}(k, t-1) \\ &\times \left[ \sum_{k=0}^N \{S_{i+1}(k, t-1) + D_{i+1}(k, t-1)\} + \left\{1 - \sum_{k=0}^N ON_{i+1,0}(k, t-1)\right\} \right] \\ &\times \{1 - p_i(k, t) - D_i(k, t)\} \end{aligned} \quad (\text{A.4})$$

ここで， $S_i(k, t)$  は A.1 節と同様である．

### A.3.2 復路側ステージの $r_{i,i-1}$

パケットが復路側ステージの  $[i, 1]$  段から  $[i-1, 1]$  段へ移動する確率  $r_{i,i-1}(k, t)$  は，

- (1) 移動しようとするパケットより優先度の高いパケットのバイパスが無い確率

$$1 - \sum_{l=k+1}^N p_{i-1}(l, t)$$



(2) 移動先のステージの PACKET が , さらに前方段に移動したか , あるいはもともと存在しない確率

$$\sum_{k=0}^N r_{i-1,i-2}(k, t-1) + \left\{ 1 - \sum_{k=0}^N ON_{i-1,1}(k, t-1) \right\}$$

によって決められる .

ここで , 最下段の上段にあたる  $[1, 1]$  段では , (2) において , A.1.2 項 (3) と同様の理由から ,  $r_{i-1,i-2}(k, t-1)$  を  $\mu(k, t-1)$  に置き換える必要がある .

PACKET が前方段に移動する場合は , まず , 移動元ステージに PACKET が存在し ( $ON_{i,1}(k, t-1)$  ) , かつ上記 (1)(2) を同時に満たす必要がある . したがって , 確率  $r_{i,i-1}(k, t)$  を一般化して表すと , 次式のように表される . ただし ,  $1 \leq i \leq m$  とする .

$$\begin{aligned} r_{i,i-1}(k, t) = & ON_{i,1}(k, t-1) \times \left\{ 1 - \sum_{l=k+1}^N p_{i-1}(l, t) \right\} \\ & \times \left[ \sum_{k=0}^N M_{i-1}(k, t-1) + \left\{ 1 - \sum_{k=0}^N ON_{i-1,1}(k, t-1) \right\} \right] \end{aligned} \quad (\text{A.5})$$

ここで ,  $M_i(k, t)$  は A.1 節と同様である .

## A.4 PACKET の出力確率 $\mu$

PACKET が , パイプラインから出力される確率  $\mu(k, t)$  は次式で表される .

$$\mu(k, t) = (1 - B(t)) \times ON_{0,1}(k, t-1) \quad (\text{A.6})$$

ここで ,  $B(t)$  は 3 章で用いた出力がブロックされる確率と同様である .

以上の式 (A.1) ~ (A.6) は ,  $t \geq 1$  で成立し ,  $t = 0$  時の各確率は全て 0 である .