

動的な重み付けによる WFQ の提案と評価

藤川 達也 †

木村 成伴 ‡

海老原 義彦 ‡

† 筑波大学 第三学群情報学類

‡ 筑波大学大学院 システム情報工学研究科

1 はじめに

DiffServ では、トラフィックを複数のクラスに分類し、パケットをクラスごとに異なるキューに格納する。そして、各キューから優先度に応じてパケットを送り出すことで、クラス間の相対的な QoS を保証する。パケットの分類方法や送出方法は、採用するパケットスケジューラによって異なる。代表的なスケジューラとして、WFQ (Weighted Fair Queuing) などがある。

2 Weighted Fair Queuing

WFQ では、パケットの分類をトラフィックフロー単位で行う。具体的には、送信元と宛先の IP アドレス、ポート番号、プロトコルの種類、IP Precedence 値をもとに、送信元から宛先までのトラフィックフローを動的に検出して、フローごとに個別のキューを作成して格納する。ここで、IP Precedence 値とは、IP ヘッダに格納されている各パケットの優先度を表す値である。格納するフィールドとして、IPv4 の場合は ToS (Type of Service) フィールドが、IPv6 の場合は Traffic Class フィールドが使用される。この IP Precedence 値の取り得る範囲は、CISCO 社製品 [2] の場合は 1 から 8 である。各トラフィックフローに割り当てられる帯域は、主にこの優先度 (= Weight) によって決定され、以下の式 (1) により近似値を計算することができる。

$$BW(i) = \frac{\text{LinkBW} \times \text{Weight}(i)}{\sum_i \text{Weight}(i)} \quad (1)$$

ここで、 $BW(i)$ は i 番目のフローに割り当てられる帯域、 LinkBW はリンクの帯域、 $\text{Weight}(i)$ は i 番目のフローの優先度である。

キューに格納されたパケットは、分類の際に式 (2) により求められた終了時間 $F(i, t)$ が小さいものから順に送り出される。

$$F(i, t) = S(i, t) + \frac{L_{in}}{BW(i)} \quad (2)$$

$$S(i, t) = \begin{cases} F(i, t - \tau) & (i \in B(t)) \\ \max\{F(i, t - \tau), V(t)\} & (i \notin B(t)) \end{cases} \quad (3)$$

A Proposal and Evaluation of WFQ with Dynamic Weighting

†Tatsuya Fujikawa ‡Shigetomo Kimura ‡Yoshihiko Ebihara

†College of Information Science, Third Cluster of Colleges, University of Tsukuba

‡Graduate School of System and Information Engineering, University of Tsukuba

$$V(t + \tau) = \max\{V(t) + \tau, \min_{i \in B(t)}\{S(i, t)\}\} \quad (4)$$

$$\tau = \frac{L_{out}}{\text{LinkBW}} \quad (5)$$

ここで、 i はフロー番号、 t は時刻である。また、 $S(i, t)$ は式 (3) で求められる開始時間、 $V(t)$ は式 (4) で求められる仮想時間、 $B(t)$ は送信待ちのパケットがキューの中に存在するフローの集合、 L_{in} は受け取ったパケットのサイズ、 L_{out} は送信するパケットのサイズである。また、WFQ では、パケットの破棄方法として Tail Drop 方式を使用する。この方式では、キューの使用率が閾値を越えると、それ以降に届いたパケットを全て破棄する。

なお、本論文では、上記の WFQ のパケットスケジューリング方法に少し変更を加えた WF2Q (Worst-case Fair WFQ)[3] を対象とする。WFQ は、全てのパケットの中から F が最小のものを選択するが、WF2Q では $S(i, t) \leq V(t)$ を満たすフローの中から F が最小のパケットを選択する。

例えば、図 2.1 のように同じ長さのパケットが中継ルータに届いたとする。ここで、 $\text{LinkBW}=10\text{pps}$ (packet per second)、 $\text{Weight}(0)=5$ 、 $\text{Weight}(1)=4$ 、 $\text{Weight}(2)=1$ とする。このとき、WF2Q のスケジューリングに従うと、パケットの送信順序は図 2.2 のようになる。

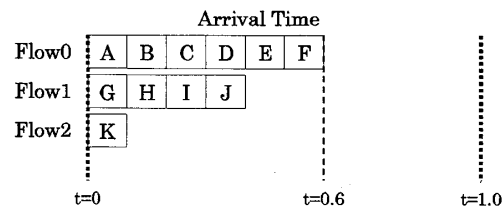


図 2.1: パケットの到着時間

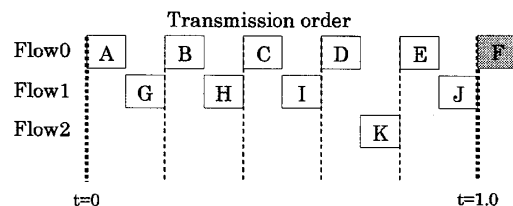


図 2.2: パケットの送信順序

3 提案方式

既存の WF2Q は、各トラフィックフローの優先度から、割り当てる帯域を固定的に決定する。そのため、高いビットレートを要求する動画のストリーミング配信などに広帯域を保証できない恐れがある。図 2.2 の例の Weight から各フローに割り当てられる帯域を計算すると、 $BW(0)=5\text{pps}$, $BW(1)=4\text{pps}$, $BW(2)=1\text{pps}$ となる。そのため、フロー 1 とフロー 2 に属するパケットは 1 秒間で全て送信されるが、フロー 0 に属するパケットの内 1 つ (図 2.2 のパケット F) は残ってしまう。これを繰り返すと、フロー 0 のキューが溢れてしまい、Tail Drop によりパケットが破棄されてしまう。そこで、提案方式では、式 (1) の $Weight(i)$ を式 (6) に変更することで、Tail Drop によるパケットの破棄が行われた回数に応じた柔軟な帯域分配を可能にする。

$$Weight(i, t) = \max\{Weight_{init}(i) + ndrop(i, t), \alpha(i, t) \times Weight(i, t - \Delta t)\} \quad (6)$$

ここで、 $Weight_{init}(i)$ はフロー i の重みの初期値、 $ndrop(i, t)$ は時刻 $t - \Delta t$ から t の間に破棄されたフロー i のパケット数に $(Weight_{init}(i, t) - 1)/7$ を掛けた値で、 $ndrop(i, t) = 0$ のときは $\alpha(i, t) = 0.95$ 、それ以外の場合は $\alpha(i, t) = 1.0$ とする。

4 測定実験

提案方式の有効性を確認するため、The Network Simulator version 2 を使用して、測定実験を行った。ネットワークトポロジとして図 4.1 を使用し、ルータ R0 で従来の WF2Q と提案方式 ($\Delta t = 0.5$ 秒) を動作させた。フロー 0 は動画伝送を想定した 10Mbps でパケットサイズが 1000byte の UDP/CBR、フロー 1 は音声通信を想定した 128kbps でパケットサイズが 200byte の UDP/CBR、フロー 2 はセグメントサイズが 1040byte の TCP/FTP である。図 4.2 にフロー 0 の平均パケット損失率を、図 4.3 にフロー 0 とフロー 2 のスループットを示す。なお、シミュレーション時間は 60 秒であり、測定結果として 10 回の試行の平均値と信頼レベル 95% の信頼区間を示した。図 4.2 より、破棄されたパケットの数に応じて重みを変化させることで、フロー 0 の平均パケット損失率がほぼ 0% になることが分かった。これにより、従来の方式と比較して、フロー 0 により高い QoS を保証することができたとと言える。その反面、図 4.3 より、フロー 2 に割り当てられる帯域が 25%~75% 狭くなってしまった。しかし、フロー 2 は、TCP によるファイル転送であり、リアルタイム性がない通

信であるので、転送にかかる時間がある程度増加しても、問題がないと考えられる。

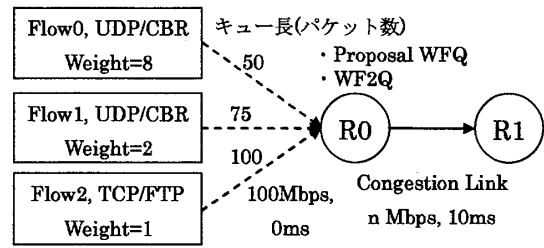


図 4.1: ネットワークトポロジ

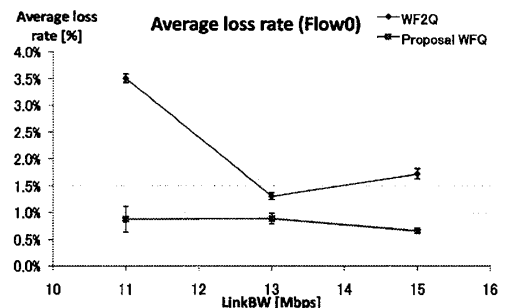


図 4.2: フロー 0 の平均パケット損失率

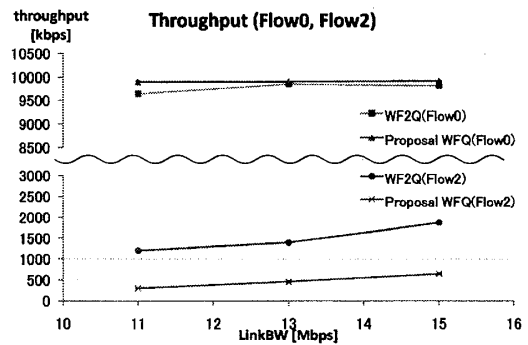


図 4.3: フロー 2 のスループット

参考文献

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," Request for Comments: 2475, 1998.
- [2] K. Solie and L. Lynch, "CCIE Practical Studies," Cisco Press, 2003.
- [3] J.C.R. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queueing," Proceedings of the IEEE INFOCOM'96, pp. 120-128, 1996.