IP Fragment 情報を利用したルータにおける効率的なパケット破棄手法

田村陽介[†] 戸辺義人[†] 徳田英幸[†,††]

IP 層において上位アプリケーションのフレームを考慮する Application Level Framing は長年の課題である．ルータが IP ヘッダ以外の情報をもとにパケットを制御することはスケーラビリティの点で不利である．また同時にルータのパケット中継の作業を複雑にしてしまう恐れがある．本論文で我々は送信ホストは IP Fragment を積極的に活用することにより，ルータに対してアプリケーションデータの境界線を知らせるべきであると主張する．我々が提案する Transport framE Discard（TED）はルータがパケットを破棄する際に，この IP Fragment の情報を用いてパケットを破棄する順序を動的に変更させる．TED を利用することでルータはアプリケーションデータの単位でパケットを破棄することが可能となる．我々は TED を FIFO，WFQ，RED，Random Drop FIFO に組み合わせて実装し，評価した．その結果，ネットワーク内における全体のパケット欠損率を変化させずに，無駄なパケットの中継を回避し，上位アプリケーションに対して高い通信性能を実現することが実証された．

TED: IP Fragment is Best Information for Application Level Framing

Yosuke Tamura,[†] Yoshito Tobe[†] and Hideyuki Tokuda[†,††]

Application Level Framing at the IP layer has been a long time issue. Using information other than IP header to control the packet can cause a disadvantage in scalability. Also, it may complicate the packet forwarding process of the router. In this paper, we claim that the source hosts aggressively use the IP fragment to inform the application data boundary to routers. Our proposed scheme, TED (Transport framE Discard), dynamically changes the order of discarding packets by using this information. By doing so, it becomes possible to discard packets in units that the application specifies. We evaluated the performance by combining TED with FIFO, WFQ, RED, and Random Drop FIFO. As a result, TED improves the performance of applications without changing the total packet discard ratio in the network.

1. はじめに

インターネット環境では，ネットワーク資源は複数のフローに共有されている．アプリケーションで利用する巨大なデータをそのままネットワークに配信すると全体的に性能が悪化する．そのため，Internet Protocol（IP）[1)]はアプリケーションデータをネットワークに最適な大きさのパケットに分割して送信する．アプリケーションのデータが増加するほど，より多くの IP パケットに分割されることになる．

IP はパケットを受信側に確実に届けることを保証していない．ベストエフォート型の配信を基本としたインターネットでは，ネットワークが過負荷状態になるとルータは許容量を超えたパケットを破棄する．アプリケーションデータを構成するパケットが 1 個でも欠損して受信側に到着すると，アプリケーションデータは完全に復元することができないために，届いたすべてのパケットが無駄になる．

我々は，送信ホストが ADU（Application Data Unit）を IP Fragment してネットワークに送信することを推奨する．IP Fragment の情報によりルータがアプリケーションデータの境目を IP ヘッダの情報のみを利用して判断することが可能となる．本論文で提案する TED（Transport framE Discard）は，ルータに実装し，アプリケーションデータの境目を IP 層で判断することにより無駄なパケットを減少させ，アプリケーションに対して高い性能を提供することが可能となる．

本論文は次のように構成されている．2 章で現在の

† 慶應義塾大学政策・メディア研究科
Graduate School of Media and Governance, Keio University
†† 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University

インターネット上でルータのパケットの取扱いに関する現状を述べる．特に IP Fragment されたパケットが欠損すると上位層に対してどのような影響を及ぼすか実験した結果を示す．その結果をもとに，3 章において TED を提案し，その設計と実装に関して述べる．4 章では TED を利用して実環境で実験評価した結果を述べる．5 章で TED の性能に関して議論する．6 章で関連研究を述べ，最後に 7 章でまとめと今後の課題を述べる．

2. インターネットの現状と問題

従来のインターネットは OSI7 階層モデルに従い階層化された通信サービスを実現してきた．この階層化されたモデルはプロトコルの設計において非常にすぐれたモデルであり，インターネット普及へ大いに貢献している．しかし階層化されたシステムはその順応性や拡張性に優れている反面，各層で独立した処理をしているため，全体的な性能を考えた際，必ずしも効率的であるとはいえない．

2.1 ルータの仕組みとパケット欠損の分類

ルータは入力インタフェースから受け取ったパケットを宛先アドレス情報をもとに適した出力インタフェースから送出する役目を持つ．ルータは突発的に流れ込むフローに備えて，一時的に出力インタフェース部分にパケットを格納しておくためのキューを保持している．キュー構造，パケットスケジューリング，パケット破棄方法を工夫することによってルータはフローに対して QoS 制御を実現することが可能となる．これらの仕組みは総称してキューイングと呼ばれる．

ネットワーク上でのパケット欠損のほとんどは，パケットがルータで中継される際に起こる．ハードウェアやソフトウェアの障害を除くとパケット欠損の要因は主に次のように分類することができる．

- 資源不足によるパケット破棄
  既存のルータの大部分が利用している最も簡単なキュー構造は FIFO（First In First Out）キューである．パケットは到着した順に送出される．FIFO キューのパケット破棄はキュー長が制限値を超えた場合に起こる．FQ（Fair Queueing）[2]や WFQ（Weighted Fair Queueing）においてもキュー長が制限値を超えた場合にパケットは破棄される．
- 故意的なパケット破棄
  資源不足によるパケット破棄が起こってからでは，ルータは複雑な処理を実現することが難しい．RED（Random Early Detection）[3]は平均キュー長に対応した確率でパケットを破棄することによって能動的にキューを管理する仕組みである．RED はキューが破綻する前に送信ホストに輻輳を知らせることができる．そのため，TCP の同期現象を防ぎ帯域利用率が向上する．またディファレンシエーテッドサービスの枠組みにおいてエッジルータでのラベリングをもとにコアルータで各フローに対して公平な帯域を割り当てる CSFQ（Core-Stateless Fair Queueing）[4]が提案されている．CSFQ ではキュー長に関係なく，各パケットにラベリングされた通信速度からパケットごとに破棄率を計算している．計算された破棄率で，そのパケットは破棄される．

2.2 IP Fragment の現状と問題

データリンクを流れるデータの最大長は MTU（Maximum Transmission Unit）として定義されている．送信側の IP 層では MTU を超えたデータがトランスポート層から渡されると，そのデータを MTU を超えないように複数のパケットに分割する．受信側では，この複数のパケットがすべて揃うともとのデータに復元することができる．ネットワーク内でパケットを処理するルータは基本的に，どのパケットが分割されているか，いないかなどといったことには感知しない．そのため，パケットの落ち方によって同数のパケットが欠損しても上位層が受け取れるデータの数が異なる可能性がある．

図 1 は IP 層で分割されたパケットが 3 個欠損した際にアプリケーションにとって最悪の場合と，最良の場合を図示している．3 個の TDU（トランスポート層から IP 層に渡されるデータ）がネットワーク層で各々 3 個のネットワーク層のパケットに分割されている．

受信側では，3 個すべてのパケットを受信しなければ 1 個の TDU を復元することができない．最悪の場合とは 9 個のパケットのうち欠損した 3 個のパケットが，それぞれ別の TDU の一部であった場合である．この場合，受信側では TDU を 1 個も復元することができない．結果的に 6 個のパケットがネットワーク上

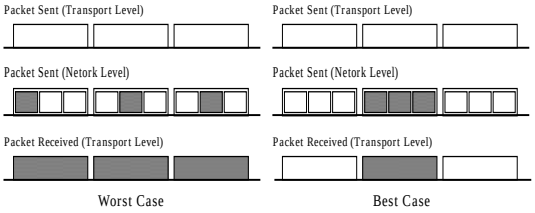

図 1 ネットワーク層で 3 個のパケットが欠損するときの最悪の場合と最良の場合
Fig. 1 The worst and best case when three packets drop in the network layer.

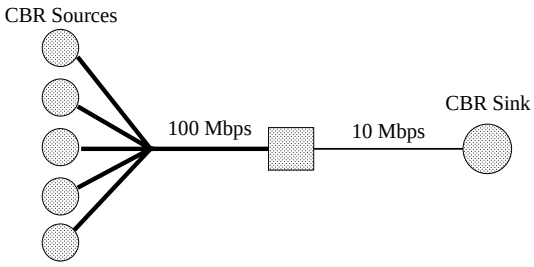

図 2 実験環境
Fig. 2 The network topology.

の資源を無駄に利用し受信側に届いたことになる．最良の場合は，3 個のパケットが同一の TDU 内で欠損した場合である．この場合は 1 個の TDU が欠損するだけで，残りの 2 個の TDU は復元できる．

2.3 パケット欠損の特徴の解析

図 2 の実験環境でパケット欠損率と TDU 欠損率の関係を調べる実験をした．100 Mbps のファーストイーサネットにつながれた送信ホストから 1 台のルータを経由して 10 Mbps のイーサネットにつながれた受信ホストに 5 本の CBR（Constant Bit Rate）のフローを同時に送信する．CBR はトランスポートプロトコルとして UDP を利用した．実験マシンは，すべて同一の性能（FreeBSD-3.4R，Celeron 500 MHz，RAM 128 MB）である．中間ルータは ALTQ（Alternate Queueing）[5] のプラットフォームを利用し FIFO，WFQ，RED，Random Drop FIFO を動作させて実験をした．

Random drop FIFO は一定の確率でパケットを破棄する FIFO である．実験では，この確率は 1 パケットの大きさの TDU を FIFO を利用して送信したときの欠損率に設定した．WFQ のトークンの大きさは 512 byte に設定した．ルータのキューの制限値はパケット 30 個分に設定した．RED の 2 個の閾値はパケット 10 個分と 25 個分に設定した．

各リンクの MTU は 1500 bytes である．TDU の大きさをパケット 1 個分（1472 bytes），2 個分（2952 bytes），3 個分（4432 bytes），4 個分（5912 bytes），5 個分（7392 bytes）に設定してデータを測定した（分割された最初のパケットにのみ UDP の 8 byte のヘッダが付加する）．

また，パケット破棄の特徴を調べるために，欠損したパケットの間隔を測定した．この実験では 5 個のパケットで構成される TDU を 20 万個送信した．グラフの X 軸は指数軸であるため，0 の表示ができない．そのため，0 のときは Y 軸上に表示している．

2.3.1 負荷状態の軽いネットワーク

近年のフローの多くは TCP に代表されるように終端ホストで輻輳制御を行う機構を備えている．UDP を利用した通信でも RTP（Real-Time Transfer Protocl）[6]/RTCP（RTP Control Protocol）のようにアプリケーションの性能を向上させるために欠損率を考慮して通信量を調整する独自のプロトコルを利用しているものが存在する．

このようなフローだけを中継しているルータは極度の過負荷状態になることはほとんどない．しかし，欠損を指標として通信制御を行っているものがほとんどであるためパケットの欠損が完全になくなるというわけではない．

ここでは通信量が調整されているフローを想定して実験する．10 Mbps のイーサネットに対して 2 Mbps の CBR を 5 本送信した．イーサネットは仕様的には 10 Mbps のフローが利用することができるが，実際には物理層の処理のオーバヘッドなどが入るために，上位層で 10 Mbps を転送すると数パーセントのパケット欠損率となる．

図 3 は TDU の欠損率を示している．1 個分のパケットの TDU ではどのキューイング方式を利用しても 4.5%程度の欠損であった．Random Drop FIFO で破棄する確率は 4.5%に設定した．5 個分のパケットの TDU を送信すると 25%近くの TDU が欠損した．

5 個分のパケットの TDU の場合，図 1 で示した考え方に基づき 4〜6%が最良の場合であると 25〜30%が最悪の場合となる（eg. 我々の別の実験ではデータグラムの大きさを変化させると欠損率が 4%から 6%程度変化するという結果が得られている）．

図 4 は，欠損したパケットの間隔を示したグラフである．RED と Random Drop FIFO は近似したグラフになっている．FIFO と WFQ はある周期的な間隔で欠損が起っていることが分かる．この周期はキューの制限値によって変化する．負荷状態の軽いネットワークではパケット欠損が突発的に起こらずにランダムに起こるため，最悪の場合になることが多い．

2.3.2 負荷状態の重いネットワーク

インターネットを流れるフローのすべてが TCP のように徐々にネットワークへ送信するデータ量を増加させていくわけではない．UDP を利用したフローの一部はあらかじめ設定された送信量でパケットを送信する．このようなフローがルータを経由する場合，一時的にルータは許容量を大幅に超えたパケットを受け取ることとなる．

ここではルータが極度の過負荷状態になったときに，どのようにパケットの欠損が起こるか調べた．10 Mbps のリンクにつながれたルータに対して 5 Mbps の CBR

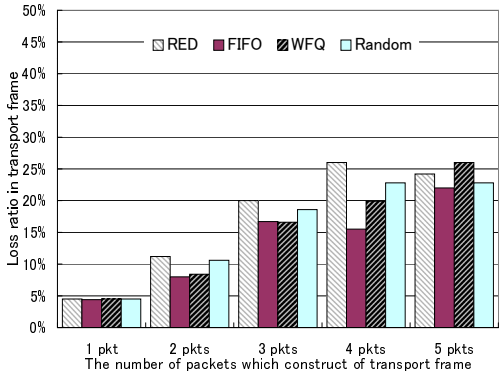

図 3 2 Mbps の CBR を 5 本流したときの TDU の欠損率
Fig. 3 The TDU loss ratio in five 2 Mbps-CBR flows.

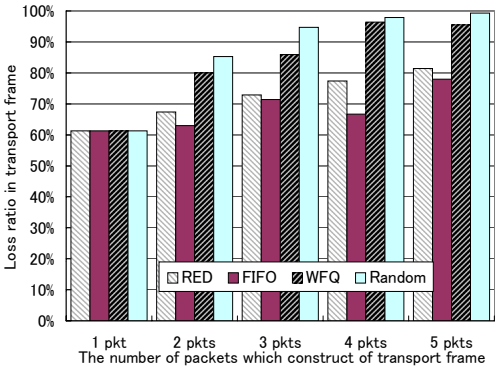

図 5 2 Mbps の CBR を 5 本流したときの TDU の欠損率
Fig. 5 The TDU loss ratio in five 5 Mbps-CBR flows.

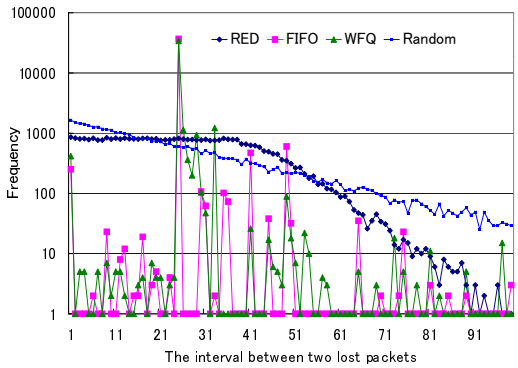

図 4 欠損したパケットの間隔
Fig. 4 The interval between two lost packets.

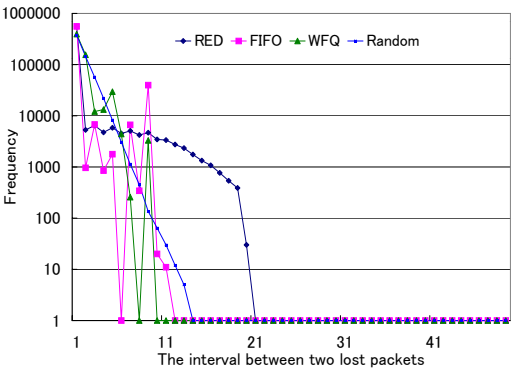

図 6 欠損したパケットの間隔
Fig. 6 The interval between two lost packets.

を 5 本送信した．出力帯域は 10 Mbps であるため，15 Mbps（5 Mbps × 5 − 10 Mbps）のデータはルータで欠損することとなる．Random Drop FIFO では 61%の確率でパケットを破棄するように設定した．

図 5 に結果を示す．図 1 で示した考え方に基づき 61%の欠損が最良の場合だとすると最悪の場合は 100%であるが，それほど性能は劣化しなかった．それは突発的にパケットが破棄されるからである．特に RED や FIFO ではそれほど劣化はなかった．**図 6** は欠損の起ったパケットの間隔を示している．ほとんどの欠損パケットは連続的であることが分かる．

3. Transport framE Discard

本章では TED（Transport framE Discard）の設計と実装に関して記述する．TED では上位データフレームの境目を判断するために，IP ヘッダの `fragment offset` 領域を利用する．`fragment offset` が 0 のパケットは先頭パケットもしくは単独パケットだと判別できる．

TED は以下のような特徴がある．

- トランスポート層のフレームを構成するパケットが 1 個でも欠損した場合，ルータは，欠損したパケットと同一の `identification` 領域を持つパケットをすべて破棄する．
- TED を利用することで全体的なフローの破棄率は変更されない．
- TED はパケット破棄手法であるため，スケジューリング機構やキュー機構とは独立に実装することが可能である．
- IP ヘッダの `fragment offset` 領域と `identification` 領域を利用する．

3.1 TED で保持する変数と定数

TED では次のようなフローごとに保持する 2 個の変数と 1 個の定数を保持する．定数はキューの閾値である．以下にその説明をする．

- ted_credit_i
  ted_credit はフローごとに保持する変数である．TED では選択的にパケットを破棄する．そのた

め，破棄すべきパケットを破棄しなかったり，その逆もありうる．このような状態が一方的に続くとネットワーク資源を使い過ぎたり，余計に輻輳が起こったりする可能性がある．TED では ted_credit に破棄するべきパケットを破棄しなかった場合は 1 を加算し，その逆の場合は 1 を減算する．このことによりパケット破棄のバランスを保つ．

- ted_ident_i
  ted_ident はフローごとに保持する変数である．TED では同一の TDU を構成するパケットが 1 つでも喪失した場合，その TDU を構成するパケットをすべて破棄する．複数のフローが存在する場合，同一のフレームを構成するパケットが連続して到着するという保証はないため，TED では破棄したパケットの `identification` 領域の値を ted_ident に保持する．
- `TED_THRESH`
  TED_THRESH は定数である．この定数は資源不足によるパケット破棄が起こる FIFO などのキューイング方式にのみ利用される．TED では破棄すべきパケットを保持するときに，そのパケットに対してキューを空けておく必要がある．パケットがキューを制限値まで利用することをつねに許可しておくとパケットを保持する際にキューを確保できない可能性があるからである．

3.2 パケット破棄方法

図 7 と**図 8** は TED の主な動作を示す仮想コードである．

TED では次のような規則でパケット破棄を行う．

- キュー長にかかわらず，捨てられたパケットと同一の `identification` の領域を持つパケットは破棄する（図 7）．
- キュー長が TED_THRESH を超えた場合以下の規則でパケットを破棄する．
  - パケットが TDU を構成する先頭のパケットでかつ ted_credit が 0 よりも大きい場合．
  - キュー長がキュー制限値を超える場合．

図 9 は TED を利用したときのパケット破棄の例を図示している．TDU1，TDU2，TDU3 が各々3 つのパケットに分割され合計 9 つのパケットがネットワーク上に送信されると仮定する．

P2，P5，P9 が欠損する場合，従来のキューイング方式では TDU1，TDU2，TDU3 は受信側で結合することができずに，3 個の TUD は破棄されることとなる．一方，TED を利用すると P2 を破棄する際に P2 は先頭パケットではないため，破棄せずに ted_credit に 1 を加算する．P4 が到着したとき，P4 は先頭パケットで，かつ ted_credit が 1 であるため，P4 を破棄する．P4 が破棄されると P5，P6 は F2 を構成する同一の `identification` 領域を持つパケットである

```
if (P_id == ted_ident_i) {
        ted_credit_i − −;
        drop packet P;
}
```

図 7 仮想コード（その 1）．このコードは欠損したパケットと同一の **identification** 領域（P_{id}）を持つパケットを破棄するコードである．

Fig. 7 The pseudo-code (part 1). This code shows the function which discards the packet that have the same **identification** field as the discarded packet.

```
if (qlen > TED_THRESH){
    if (((P_off == 0) && (ted_credit_i > 0))
        || (qlen >= qlimit)) {
            ted_ident_i ← P_id;
            drop packet P;
    } else {
            ted_credit_i + +;
            do not drop packet P;
    }
```

図 8 仮想コード（その 2）．このコードはキュー長（qlen）が TED_THRESH を超えた場合の処理を示している．先頭パケット（P_{off}== 0）かつ ted_credit が 0 よりも大きい場合，または qlen がキュー制限値（qlimit）を超えた場合にパケットを破棄する．

Fig. 8 The pseudo-code (part 2). This code shows the funcation when the queue length (qlen) exceeds the value of TED_THRESH. The arrival packet is discarded if it is a first constituent packet (P_{off}== 0) and the ted_credit is bigger than or equal to 0, or qlen exceeds the queue limit (qlimit).

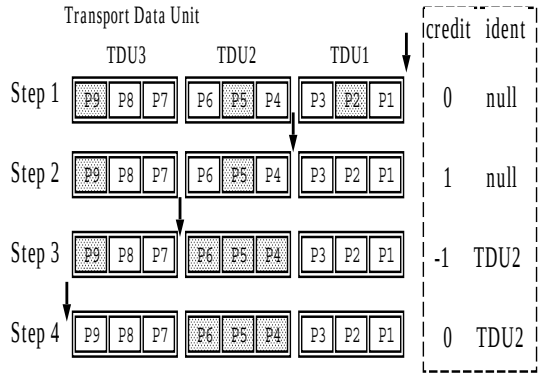

図 9 TED を利用したときのパケット破棄の例

Fig. 9 The details of the packet discard when using TED.

ため，破棄される．この時点で ted_credit は −1 になる．次に P9 が到着するが，P9 は先頭パケットではなく，また ted_credit も 0 よりも大きくないため，ted_credit に 1 が加算されて送信される．以上の処理で，TED は本来，一部のパケットが欠損したために結合できなかった F1，F3 を構成するパケットをすべて送信することが可能となる．

3.3 FIFO 以外のキューイング方式との組合せ方法

本章で述べた実装方法は基本的に FIFO を前提としている．TED はパケット破棄方法として独立しているため，既存のキューイング方式と組み合わせることが可能である．次の章で ALTQ[5] のプラットフォーム上で TED を FIFO，RED，WFQ，Random Drop FIFO と組み合わせて評価を行っているが，別のスケジューリング方式やキュー構造を持つキューイング方式と容易に組み合わせることも可能である．ここでは種類の異なるスケジューリング方式と TED を組み合わせる方法について述べる．

- RED
  RED は平均キュー長に対応した確率でパケットを破棄することによって，能動的にキューを管理する仕組みである．パケットの破棄率の計算には，平均キュー長と 2 個の閾値（最小閾値，最大閾値）が利用される．RED ではキューの制限値をキュー長が超える前にパケットを破棄するため，TED_THRESH を定義する必要はない．図 7 のコードをパケットの入力部に追加し，パケットを計算された確率で破棄する部分に図 8 のコードを追加する必要がある（図 8 の最初の行の `if` の判定文は必要ない）．
- WFQ
  WFQ は内部に複数のキューを持つ構造となっている．ラウンドロビンで各キューからパケットを送出する．各ラウンドでキューに対してトークンを与え，トークンの量がパケットの大きさを超えた場合，先頭パケットを送信する．ALTQ の WFQ の実装ではキューの制限値をキュー長が超えた場合，新規に到着したパケットを破棄の対象とせずにキューの先頭パケットを破棄する．したがってキューにすでに格納されているパケットが無駄なパケットとなる可能性がある．図 7 のコードをパケットの出力部に追加する必要がある．
- Random Drop FIFO
  確率をもとに到着したパケットを破棄するキューイング方式の一例として，一定の確率で到着パケットを破棄する Random Drop FIFO を TED と組み合わせた．Random Drop FIFO ではキュー長とは無関係にパケットを破棄する方式のため，TED_THRESH を定義する必要がない．図 7 のコードをパケットの入力部に追加し，パケットを一定の確率で破棄する部分に図 8 のコードを追加する必要がある（図 8 の最初行の `if` の判定文は必要ない）．

4. 評　　価

本章では ALTQ を利用して実環境 TED の評価を行った結果を述べる．FIFO，WFQ，RED，Random Drop に TED を組み合わせて 2.3 節と同一の実験環境（図 2）で実験を行った．各キューイング方式の設定は 2.3 節で述べたとおりである．キューの制限値を 30 パケットとした．TED の設定において，TED_THRESH を 25 パケットとした．

4.1 負荷状態の軽いネットワーク

2 Mbps の CBR を 5 本同時に転送し，TED を実装したルータを途中に設置した．**図 10** のグラフは TDU の大きさを変更させたときの TDU 欠損率を各キューイング方式で比較した結果である．TED を実装しない場合の実験結果の図 3 と比較すると，TED を利用すると，TDU の欠損率を大幅に減少させていることが分かる．特に FIFO，RED の実験では TDU の大きさを増加させても TDU 欠損率は 1 パケット分の TDU のときとほとんど変わらないという結果が出た．

前章で述べたとおり ALTQ における WFQ の実装は入力パケットを破棄の対象とせずに，キューの中の先頭パケットを破棄する．そのため，無駄なパケットも一時的にキューを占有することとなる．その結果，入力パケットを破棄の対象としている他のキューイング方式と比べてキューの利用効率が悪くなり性能が劣化する．

図 11 は 5 個のパケットで構成された TDU を 10 Mbps の CBR で転送したときの，欠損パケットの間隔を示した図である．図 4 と比較すると明らかに欠損の間隔の特徴が変化している．実験を行ったすべてのキューイング方式において連続した欠損が多くなった．また一定間隔で欠損の間隔が増加している部分がある．これは連続的な欠損が一定間隔で起こっていることを示している．

4.2 負荷状態の重いネットワーク

TED を利用したときにルータが極度の過負荷状態になったときのパケット欠損の特徴を調べた．**図 12** のグラフは 5 Mbps の CBR を 5 本同時に転送したときの結果を示している．中間ルータにおけるフローの

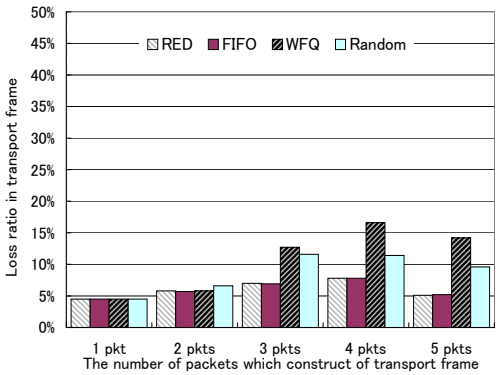

図 10 2 Mbps の CBR を 5 本流したときの TDU の欠損率
Fig. 10 The TDU loss ratio in five 2 Mbps-CBR flows.

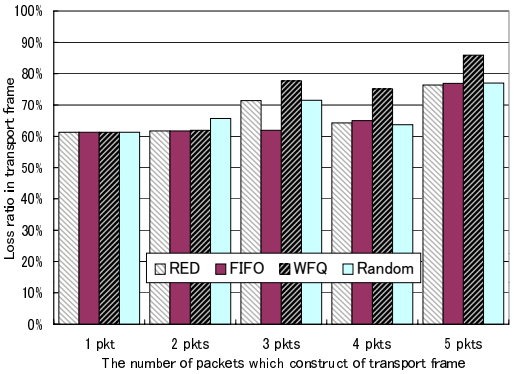

図 12 5 Mbps の CBR を 5 本流したときの TDU の欠損率
Fig. 12 The TDU loss ratio in five 5 Mbps-CBR flows.

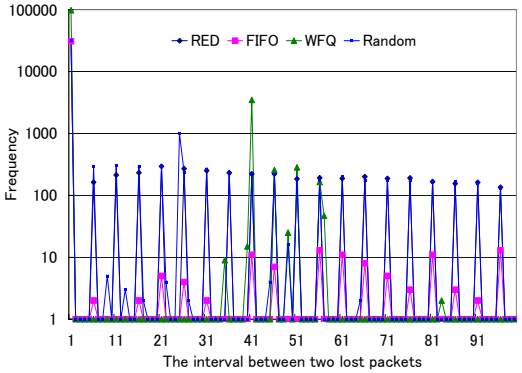

図 11 欠損したパケットの間隔
Fig. 11 The interval between two lost packets.

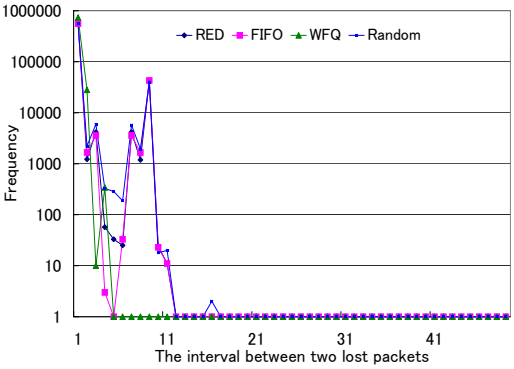

図 13 欠損したパケットの間隔
Fig. 13 The interval between two lost packets.

入力速度は 25 Mbps で出力速度は 10 Mbps であるため，15 Mbps のフローが欠損することとなる．TED を利用しない実験結果である図 5 と比べると全体的に TDU の欠損率が減少していることが分かる．FIFO を用いた場合，TED を利用しなくても過負荷状態ではほぼ連続的にパケットが破棄される．そのため，TED を利用しても TDU の転送性能はそれほど変化しなかった．RED，WFQ，Random Drop FIFO に関しては最大 30%近く TDU 欠損率を減少させた．

図 13 のグラフは欠損したパケットの間隔を示している．図 6 と比較すると RED，WFQ，Random Drop FIFO のグラフは明らかに異なる特徴を示している．どのグラフも FIFO と近似したグラフとなった．したがって RED，WFQ，Random Drop FIFO では TED を利用すると連続的な欠損が増加することが分かる．

4.3 欠損特徴の分析

TED を利用した場合と，しない場合でパケットの欠損特徴を解析するための実験をした．キューイング方式には FIFO を利用して，2 Mbps の CBR フローを 5 本同時に送信した．3 個のパケットから構成される TDU を 1 万個送信した．欠損パケットの特徴を調べるために欠損パケットのシーケンス番号を取得した．

表 1 は，同一のデータに対する解析である．粒度を変更させて，各々の粒度での欠損パケットの平均と標準偏差を示している．欠損率は TED を利用したときと，利用しないときでつねに 6.7%であった．TDU の欠損率は，前章の実験で示したとおり TED を利用しない場合は 17%，TED を利用した場合は 7%と TED を利用すると 10%近く向上する．ここで特徴的であるのは TED を利用した場合，粒度を大きくしても標準偏差が大きくならないことである．これは TED を利用することである程度，ランダムなパケットの欠損を回避していることを示している．

表 1 平均欠損数と標準偏差
Table 1 The average drop and standard deviation.

	Average Drop (packets)		Standard Dev. (packets)	
Granularity	FIFO with TED	FIFO	FIFO with TED	FIFO
50	3.35 (6.7%)	3.34 (6.7%)	1.64	1.34
100	6.74 (6.7%)	6.68 (6.7%)	1.55	1.30
500	33.42 (6.7%)	33.49 (6.7%)	3.40	3.16
1000	67.20 (6.7%)	66.88 (6.7%)	4.39	6.08
1500	100.63 (6.7%)	100.05 (6.7%)	5.78	7.42
2000	134.16 (6.7%)	133.50 (6.7%)	5.37	9.29

5. TED の性能に関する議論

TED を利用することでネットワーク層でのパケット分割の問題を回避しトランスポート層のフレームを効率的に扱うことが可能となる．本章では TED の性能に関する議論を行う．

5.1 IP 層はどの層のフレームを考慮した制御をするべきか

TED の本質的な目的はアプリケーションに対して高い性能を提供することである．しかし TED ではあえて，アプリケーションデータの境界線を厳密に調べる手法を用いていない．ネットワーク層で扱われるデータは，ネットワーク層で利用されているプロトコルのみを利用して処理を行うべきであるという考えをもとにに設計されている．たとえトランスポート層のポート番号やアプリケーションで独自の識別子を利用してネットワークでアプリケーションデータの境界線を見極めたとしても新しいアプリケションやトランスポート層を持つフローでは，その実装は動作しなくなる．また IPsec [7]などの新しい技術により IP は将来的に上位層のデータを解析することができなくなる可能性もある．

送信ホストが ADU を IP Fragment させて送ることにより TED を実装したルータは IP ヘッダ以外の情報を利用せずに ALF（Application Level Framing）[8] を実現することが可能となる．

5.2 IP vsersion 6 ネットワークへの適応

IPv6 ヘッダでは，IPv4 における `Identification` 領域と `offset` 領域が削除されているため，TED では IPv4 パケットと IPv6 パケットを異なる処理で扱う必要がある．

IPv6 では経路検索 MTU [9]を用いて途中経路における IP Fragment を送信ホストがあらかじめ回避する仕組みを取り入れている．しかしこれはルータ内での IP Fragment の回避で IPv6 においても送信ホスト内での IP Fragment を避けることはできない．そのため，断片ヘッダという拡張ヘッダを IP ヘッダとペイロードの間に挟むことができるように規定している．この断片ヘッダは IPv4 の `Identification` と `offset` 領域に価する．拡張ヘッダを持たないパケットは分割されていないと見なすことができる．

6. 関 連 研 究

1990 年に提唱されていた ALF [8]の考え方に基づいて現在までに，いくつかの研究が行われてきた．特に ATM（Asynchronous Transfer Mode）ネットワークにおいて扱われる固定長セルは 53 byte と上位層が扱うデータの単位と比べて非常に小さい．そのため，ALF を応用した研究がさかんに行われている．その 1 つである EPD（Early Packet Discard）[10]は PPD（Partial Packet Discard）[11]を改良しキューのオーバフローを起こさせずに高い性能を提供する．TED も EPD 同様にバッファに閾値を持たせているが，credit という概念を利用しているために全体的な欠損率を変更させずに制御することができる．また IP ネットワークでの利用を可能とし，種類の異なる複数のキューイング方式と容易に組み合わせて利用することができる．

FIPD（Frame-Induced Packet Discarding）[12]はビデオデータを対象として ATM 上で無駄な cell を破棄する手法である．FIPD では，ビデオ送信元自体が `frame-identifier` ビットを付加する必要がある．そのビットをもとにルータはフレームの境目を識別する．TED では FIPD のように送信側からのコーディングの情報は必要とせずに IP ヘッダの情報のみを利用するため，スケーラビリティの点で有利である．

7. まとめと今後の課題

近年，インターネットの普及にともない，ユーザのサービス要求が高まりつつある．サービスの高品質化にともないネットワークアプリケーションが扱うデータ量は増加してきた．このような状況ではネットワークとアプリケーションで扱うデータの単位の差がより拡大していくこととなる．

IP 層においてアプリケーションデータを考慮した処

理をする ALF は長年の課題である．ルータが IP ヘッダ以外の情報をもとにパケットを制御することはスケーラビリティの点で不利である．また同時にルータのパケット中継の作業を複雑にしてしまう恐れがある．

本論文では FIFO，WFQ，RED，Random Drop FIFO を利用したルータが輻輳状態においてどのようにパケットを破棄するのかを解析した．さらにネットワーク層でのパケットの破棄がランダムになると，上位層でのデータがフラグメントされていた場合，大きな性能低下が起こることを示した．我々は，この問題を解決するために TED を提案した．TED では IP ヘッダの情報から上位データの境目を識別して無駄なパケットを送信しないように効率的な処理を行う．TED はパケット破棄の順序を変更するだけで全体のパケット欠損率は変化させない．そのため，トラフィックシェーピングなどの技術に応用することが可能である．また，アプリケーションが IP Fragment を積極的に活用することにより，TED はアプリケーションデータの単位でパケットを破棄することも可能となる．

今後の課題としては，フローの数が増加したときの TED の挙動などをシミュレータなどを利用して解析する必要がある．また実際に音声動画といったデータを流したときのアプリケーションに対する性能評価も今後の課題である．

参 考 文 献

1) Postel, J.: Internet Protocol—DARPA Internet Program Protocol Specification, RFC 791 (1981).
2) Demers, A., Keshav, S. and Shenker, S.: Analysis and simulation of a fair queueing algorithm, *Journal of Internetworking Research and Experience* (1990).
3) Floyd, S. and Jacobson, V.: Random Early Detection for Congestion Avoidance, *IEEE/ACM Trans. Networking*, Vol.1, No.4 (1993).
4) Stoica, I., Shenker, S. and Zhang, H.: Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks, *Proc. ACM SIGCOMM'98* (1998).
5) Cho, K.: A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers, *Proc. USENIX'98* (1998).
6) Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, RFC 1889 (1996).
7) Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC 2401 (1998).
8) Clark, D.D. and Tennenhouse, D.L.: Architectural Considerations for a New Generation of Protocols, *Proc. ACM SIGCOMM'90* (1990).
9) Mogul, J. and Deering, S.: Path MTU Discovery, RFC 1191 (1990).
10) Romanow, A. and Floyd, S.: Dynamics of TCP Traffic over ATM Networks, *Proc. ACM SIGCOMM'94* (1994).
11) Armitage, G. and Adams, K.: Packet Reassembly During Cell Loss, *IEEE Network* (1993).
12) Ramanathan, S., Rangan, P. and Vin, H.: Frame-Induced Packet Discarding: An Efficient Strategy for Video Networking, *4th International Workshop on Network and Operating System Support for Digital Audio and Video* (1993).

(平成 12 年 12 月 19 日受付)
(平成 13 年 2 月 1 日採録)

田村 陽介（学生会員）

平成 11 年慶應義塾大学大学院政策・メディア研究科修士課程修了．同年同大学院博士課程．主として TCP/IP のプロトコル処理効率化に関する研究に従事．IEEE 会員．

戸辺 義人（正会員）

平成 13 年慶應義塾大学プロジェクト助教授．主として，計算機通信における QoS 制御の研究に従事．IEEE，ACM，日本ソフトウェア科学会各会員．

徳田 英幸（正会員）

昭和 50 年慶應義塾大学工学部管理学科卒業．昭和 58 年カナダ・ウォータールー大学計算機科学科博士課程修了．Ph.D. in Computer Science．同年米国・カーネギーメロン大学計算機科学科勤務．平成元年慶應義塾大学環境情報学部助教授兼務，平成 8 年同教授，平成 9 年慶應義塾大学常任理事，現在に至る．分散システム，OS，ネットワーク等の研究に従事．情報処理学会 OS 研究会主査．IEEE，ACM，日本ソフトウェア科学会各会員．