

3 群 (コンピュータネットワーク) - 5 編 (通信品質)

3 章 品質制御技術

(執筆者: 山本 幹)[2011 年 1 月受領]

概要

通信品質を大きく左右する要因のうち、ネットワーク内部で発生する事象に、遅延、パケット遅延ジッタ、パケット損失などがある。遅延は、エンドツーエンド間で電気信号が伝搬するのに必要な伝搬遅延、パケット長をリンク容量で割った値で求められるパケット伝送遅延、ルータにおいてバッファリングされる待ち行列遅延で構成される。パケット遅延ジッタは、主にルータで被る待ち行列遅延がパケットごとに異なることで発生する。パケット損失は、主にルータでのバッファあふれがその原因である。リアルタイム情報の転送にあたっては、通信品質を左右するこれらの要因に、エンドホストもしくはネットワーク内部での対応が必要となる。本章では、品質制御技術として、エンドホストで対応する技術としてのリアルタイムアプリケーション品質制御、ネットワーク内部で対応する技術である、資源予約並びにキューイング制御、これらの制御技術の基本となる計測技術、を取り扱う。

現在広く普及しているインターネットでは一般にネットワークは、ベストエフォート型サービスを提供する。このため、遅延やパケット損失に対する一定水準の保証は全くなく、上記の通信品質劣化要因に対しては、エンドホストで対応する必要がある。3-1 節では、リアルタイムアプリケーションの品質制御と題し、これらエンドホストで対応する技術について詳しく説明している。インターネットの通信品質制御の甘さを改良すべく、いくつかのアプローチが試みられている。4-2 節では、その中の一つのアプローチである資源予約について詳しく説明している。具体的には、現在サービスが開始されている NGN で実際に適用されている DiffServ などを取り上げ、プロトコルや標準化について詳しく説明している。4-3 節では、別のアプローチとして、ネットワーク内のルータにおいて、待ち行列規律などを適切に制御することで通信品質制御を行う、キューイング制御を取り扱っている。ここで取り扱う技術も、一部の最新ルータでは実装されている機構である。4-4 節では、通信品質制御を行う際、もしくは通信品質制御の効果を評価する際に用いる、計測技術について詳しく説明している。

本章により、通信品質制御技術を、エンドホストでの対応、ネットワーク内部での対応と体系的に理解することが可能である。

【本章の構成】

3-1 節では、リアルタイム性を要求するリアルタイムアプリケーションに対する品質制御を説明している。3-2 節では、通信品質制御における資源予約について説明している。3-3 節では、通信品質制御のうちルータなどでの待ち行列での制御技術であるキューイング制御を扱っている。3-4 節では、通信品質制御を行ううえで、品質自体を評価する方法である計測技法について説明している。

3 群 - 5 編 - 3 章

3-1 リアルタイムアプリケーションの品質制御

(執筆者：戸出英樹)[2009年4月受領]

3-1-1 リアルタイムアプリケーションの利用形態

リアルタイムアプリケーションの利用形態に関して、通信接続の観点から、基本形態である1対1型に加えて、サーバを拠点とする放送、マルチキャスト配信に相当する1対多、更には、多地点TV会議システムやP2P情報配信などの多対多型の形態が存在する。リアルタイムアプリケーションの品質制御技術としては、基本的には1対1型に注目すればほかの利用形態でも同様の議論が成立すると考えられるが、階層型FEC技術など、1対多型通信でその利点が際立つものも存在する。一方、情報の流れの観点からは、1方向型と双方向(対話)型の形態が存在し、1方向型は、受信側で多少のレイテンシを許容することにより、受信側でのバッファリング時間を利用した品質改善が可能となる。これに対し、双方向型は、当該レイテンシの許容量がより厳格となるため、一般的にはより厳しい制約条件になる。

3-1-2 品質制御対象とQoSパラメータ

品質制御の対象としては、ネットワークQoS指標とアプリケーションQoS指標が挙げられる。ネットワークQoSの代表的な指標としては、帯域幅、遅延、パケット遅延ジッタ、パケット損失などが挙げられる。帯域幅は、エンドツーエンドのネットワーク内パス上の利用可能な転送レートを意味し、遅延は、データが宛先端末まで転送されるのに要する時間である。一方、パケット遅延ジッタはルータ内のバッファリングや経路の変化、中継機器の障害などにより発生し、大きな変動は受信側のPlaybackバッファで補償する必要がある。遅延に関して、情報は受信側の再生タイミングまでに到着する必要があるため、期限を越えた情報はパケット損失と等価とみなされる。パケット損失は、網内の輻輳に起因して発生する。絶対的な廃棄率のみならず、パケット損失の発生パターンや継続期間(バースト性)がコーデックやアプリケーションの設計者が開発するに当たっての重要な指標となる¹⁾。

3-1-3 圧縮技術に応じた制御

代表的な動画圧縮技術にMPEG(Moving Picture Experts Group)が挙げられる。MPEGでは、I(Intra coded)、P(Predictive coded)、B(Bidirectional predictive coded)ピクチャの3タイプの画像が存在している。映像をMPEGデータに符号化する場合、まずIピクチャが符号化される。次にIピクチャからの画像予測によりPピクチャが符号化される。最後に前後のIまたはPピクチャからの予測によりBピクチャが符号化される。このような予測符号化によって大幅なデータ圧縮が可能となるがその反面、MPEGデータを転送する場合、予測に使われるIピクチャやPピクチャに対しパケット損失が発生するとクライアント側の画質に大きな影響を及ぼすため、損失を起こさないように注意する必要がある。

MPEGのデータ構造は階層構造が規定され、画面のサイズ、表示周期、ビットレートを含むシーケンス層、予測符号化のための複数の画面データの集まりであるGoP(Group of Picture)層、画面のピクチャタイプを含むピクチャ層、1枚の画面を複数の領域に分割するスライス層、DCT(Discrete Cosine Transformation)の単位となるマクロブロック・ブロック層がある^{2,3)}。このような階層構造の規定により、MPEGは様々なデータに対応することができる。

リアルタイムアプリケーションの情報伝送に際し、MPEG などの圧縮方式を利用する場合、圧縮技術の原理・特徴を利用してネットワーク状況に適応させることが可能である。

送信レートは、異なる量子化レベルと符号化レートを使用することにより変更可能である。また、DCT の係数は異なる優先度をもつ二つのレイヤに分離して伝送することが可能である。ベースレイヤは重要なビデオ情報を伝送し、付加レイヤは更なる品質改善のために利用される。ネットワーク情報のフィードバックから、より多くの帯域が利用可能ならば、より多くのデータがベースレイヤに割り振られる。逆に、帯域が少ない場合は、ベースレイヤ内のデータの割り振りは減らされる。輻輳時には、ネットワーク内で低優先レイヤが優先的に廃棄される。

干渉トラヒック発生時に MPEG 再生の品質を維持するための高能率転送手法として、パラメータ変更制御、並びにフレーム識別に基づくフレーム間引き、シャッフル転送、スムージングを示す。

3-1-4 パラメータ変更制御

パラメータ変更制御とは、現在のネットワーク状況を反映したフィードバック情報に応じて情報圧縮アルゴリズムの各種パラメータを適応的に変更することによりエンコーダから発生するトラヒックのレートを調整する技術である。

ビデオのレートシェーピングにおける変更対象のパラメータとしては、例えば以下のものが挙げられる⁴⁾。

Refresh rate: ビデオエンコーダにより符号化されるフレームレート。Refresh rate が下がれば、エンコーダの出力レートが下がり、品質が低下する。

Quantizer: 符号化されている DCT 係数の数。Quantizer を増やせば、符号化される係数の数が減り、画像が粗くなる。

Movement detection threshold: フレーム間符号化の際に移動検知の基準となる閾値。フレーム間符号化では、DCT が信号値の差異に対して適用される。本閾値は、前のフレームと異なったものと検知されるブロック数を制限する。この閾値が増加すればエンコーダの出力レートが減少する。

フィードバック情報としては、パケット損失情報や受信バッファの占有レベルなどが使用される。

パケット損失率を利用する場合、レシーバは定期的に現在のパケット損失率を返信する。例えば、ビデオエンコーダのレートを動的に制御するアルゴリズムとして、次のような AIMD (Additive Increase Multiple Decrease) 機構が挙げられる⁵⁾。なお、GAIN, INC は増減の度合いを調整するパラメータである。

```
if(平均パケット損失率 > 許容パケット損失率)
then
max_rate = max(max_rate/GAIN, min_rate)
```

```
else
max_rate = max(max_rate+INC, min_rate)
```

3-1-5 フレーム識別転送

フレーム識別転送とは、MPEG データをフレームごとに識別（フレーム内のピクチャタイプ、フレーム番号、GoP 内のフレーム位置を識別する）し、パケット化して転送する方式である（図 3・1）。転送処理はフレームごとに行われるため、パケットごとにフレームタイプに応じたフラグを記録することでパケットの区別が可能になる。なお、フラグとしては、直接的なフラグ情報に加えて、RTP (Realtime Transfer Protocol)⁶⁾のタイムスタンプ情報の利用も該当する。またフレーム識別により、パケット紛失の際、紛失したフレームの特定が容易となる。しかし、フレームを識別する制御がサーバ、クライアント両方で必要になり、識別のための処理は増加する。

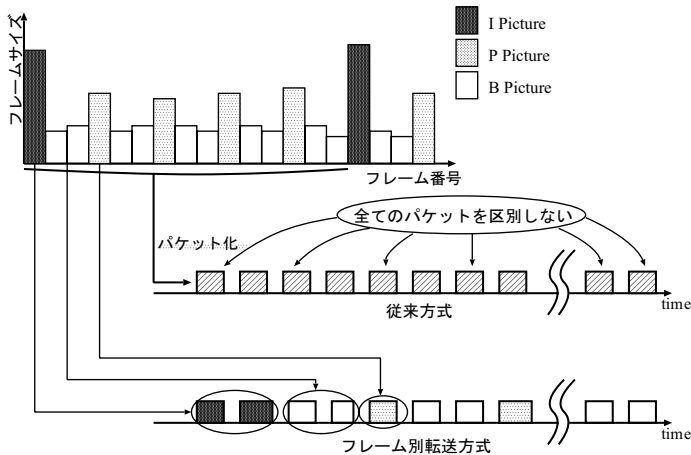


図 3・1 フレーム識別転送

3-1-6 フレーム間引き

フレーム間引き (Frame Dropping)⁷⁾とは、サーバからの転送に先立ち、サーバ - クライアント間のトラフィック状況を調べ、その状況に応じてあらかじめサーバ側で転送すべきフレームを適応的に間引くことにより、MPEG データの転送量を減少させ、輻輳を回避する制御方式である。

先述のとおり、MPEG はフレーム間で予測を行うことにより情報を圧縮しているため、不用意にフレームを間引くことはできない。そこでほかのフレームに影響の出ない B ピクチャの入ったフレームを優先的に間引くことで、映像品質への影響を極力少なくしたフレーム間引きを行うことができる。

MPEG のデータ構造は階層構造である。したがって、ある特定のフレームを間引くためには、MPEG 階層のピクチャ層以下を削除するだけでよい。しかし、フレームを間引いて再生すると、間引いた分、その後のフレームが前に詰められ、その結果、再生が早送りになる可能

性がある。そこで、サーバはあらかじめクライアントにフレームを間引く旨を含む情報を送る必要がある。クライアント側はその情報を基にフレームレートを調節し、映像を再生する。

3-1-7 シャッフル転送

MPEG フレームが IP パケットにフラグメント化される場合、フレームサイズはパケットサイズよりも大きいため、連続して同じピクチャタイプが出力される。このとき、パケット損失が起きますと、同一ピクチャ情報のバースト誤りの生ずる可能性が高くなる。そこで、サーバの送信バッファ内でパケットの送信順序をシャッフルし、クライアント側でシャッフルされたパケットを元の順序に戻す操作を行うことで、バースト誤りを防止することができる(図 3・2)。

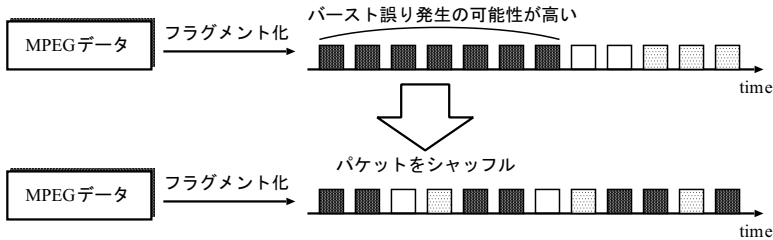


図 3・2 シャッフル転送

シャッフルされたパケットを受信側で元の順序に戻す操作は、フレーム転送時にパケットに記録されたフレーム番号とパケット番号(どのフレームの何番目のパケットか)を基にして行われる。

3-1-8 スムージング

スムージング (Smoothing)⁴⁾は、レート変動を和らげる方法として有効である。本制御方式の主要なアイデアは、クライアントに十分なバッファスペースがあるとき、後の表示に必要な大きなフレームを先に伝送することである。

ライブ型(非対話型)ビデオに対しては、スライディングウィンドウに基づくバッファ制御が利用される。SAVE (Smoothed Adaptive Video over Explicit rate networks)⁸⁾と呼ばれる方式では、少数のフレームがウィンドウの範囲内でバッファリングされ、ネットワークからのフィードバック情報を用いて MPEG エンコーダの quantizer 値を動的に変更する。

一方、蓄積型ビデオに対しては、送信側でビデオトラヒックを平滑化するために、事前にビデオ情報を利用する。帯域スムージングは、VoD (Video on Demand) アプリケーションにおける圧縮ビデオのバースト性を軽減することができる。

(1) オンラインスムージング制御

ウィンドウ型オンラインスムージングは、ウィンドウの適用法の観点から、ホップバイホップ型とスライディングウィンドウ型に分類される。ホップバイホップ型では、一定個数のフレームからなるウィンドウ単位で独立に転送間隔が整形される。そのため、ウィンドウ境界をまたがってデータを先読みすることができない。一方、スライディングウィンドウ型は、

ウィンドウを一定の幅でシフトしつつアルゴリズムを起動しスムージングを行う。そのため、アルゴリズムの適用頻度が多くなり結果的に性能は上がるが処理負荷が高くなる。

(2) プロアクティブなバッファ制御

本制御は、蓄積ビデオを対象としたレート規制型の帯域スムージングである。与えられたレートの制約条件に対してバッファ利用率の最小化を目指す。RCBS (Rate Constrained Bandwidth Smoothing) アルゴリズム⁹⁾では、映像フレーム系列を映像の終了側から逆順に検索し、制約レート以上を要求する大きなサイズのフレームがプリフェッチ (prefetch) される。これらのプリフェッチ分はフレーム系列前方の小さなフレームのギャップを埋めるように転送される。

3-1-9 適応同期

同期問題は、クロック周波数のドリフトやネットワーク遅延・パケット遅延ジッタに起因する。同期問題には、メディア内同期とメディア間同期の問題がある。適応同期 (Adaptive Synchronization) 技術とは、実現性を鑑みてシステム全体で共通のクロックを必要とせず、最適な遅延と QoS 要求を満たすためのバッファリングを提供する技術である。

基本アイデアの一例を以下に示す。パケットを wait, no-wait, discard のカテゴリに区分し、wait パケット内のパケットはある所定の時間後に表示され、no-wait パケットは直ちに表示される。一方、discard カテゴリのパケットは廃棄される。基本アルゴリズムは以下のとおりである¹⁰⁾。送信側はタイムスタンプをパケットに付与する。受信側は、playback クロック (PBC) と wait, no-wait, discard カテゴリのための 3 カウンタを管理する。パケットが早期に到着し十分な wait パケットが受信されると、PBC がインクリメントされる。一方、no-wait もしくは discard パケットが閾値を超えて受信されると、PBC はデクリメントされる。これにより、適応的にクロック速度が調整される。メディア間同期には、グループ単位の PBC を適用し、同期される複数ストリームの中で最も遅いものを元にクロックを増減させることで対応可能である。

3-1-10 誤り制御

マルチメディアアプリケーションは、パケット損失率の変化に適應する必要がある。転送レートを調整するだけでなく、損失したデータを復元させる機能、つまり誤り制御 (Error Control) も必要である。

誤り制御は大別して以下の 2 通りに分類できる。

- ARQ アプローチ (Closed loop): 受信側が送信側に廃棄パケットの再送を要求する。エンドツーエンド遅延を極端に増加させるため、基本的にはインターネット環境に不向きである。高速 LAN など RTT が小さい環境には適している。
- FEC アプローチ (Open loop): ソースがあらかじめ冗長な情報を伝送し、受信側では、その冗長な情報転送分をパケット損失からの回復に利用する。

3-1-11 ARQ (Automatic Repeat Request)

リアルタイムアプリケーションにおいても、受信側バッファでの再生タイミングまでの時間に余裕がある場合や、対象ネットワークが小規模でありエンドツーエンド遅延が小さい場合などは、優先再送を利用した品質制御が可能である。このようなアプローチはビデオスケジューリングとも呼ばれる。基本アイデアは、送信側において未伝送のペケットや再送が必要なペケットの中から、受信側で再生するための時間的期限を考慮しつつ、最も映像品質の向上に寄与しているフレームのペケットを優先的に転送することである。

単純な方法としては、フレーム種別（厳密には、GOP 内の位置）ごとに優先度に応じた閾値を設け、優先順序を規定する方式¹¹⁾や、期限を超えて到着するペケット数の期待値をひずみとみなして最小化するスケジューリング¹²⁾、ペケット損失による品質劣化の度合いをひずみとみなして最小化するスケジューリング¹³⁾¹⁴⁾などがある。

3-1-12 FEC (Forward Error Correction)

(1) FEC の分類

FEC は、実データにあらかじめ冗長な符号や誤り訂正符号を付与し、転送中に誤りが生じた場合にはクライアントが自発的に誤り訂正を行う方式である。FEC を用いることで、途中でペケットが廃棄された場合でも、冗長ペケットによりデータを復元することができる。FEC の種類としては、オリジナルペケットに対して、FEC 冗長ペケットを付与して送信し、途中でペケットが廃棄されてもクライアント側でペケットを復元できるようにする pro-active FEC 方式と、サーバは基本的にオリジナルペケットのみを送信し、ペケット廃棄が発生した場合に、クライアントは廃棄されたペケット個数を通知し、サーバは各クライアントの廃棄ペケット個数から最大廃棄ペケット数分の FEC 冗長ペケットを生成して再送する reactive FEC がある。IETF においても、FEC を有効な制御方式の一つとして挙げており、ガロア体 (Galois Fields) と Reed-Solomon 符号を用いたエンコード・デコード処理やソフトウェア実装など、現在、様々な方法で FEC の実装が行われている¹⁵⁾。reactive FEC は 1 対多型の ARQ 方式の一形態と見ることができ、以後、pro-active FEC を前提として説明する。

(2) FEC の実現方法

IP ネットワークにおける FEC の実現方法について述べる。FEC エンコード・デコードの実現には、図 3-3 のようなブロックを利用する。

まず、FEC エンコードを行うため、オブジェクト（データ）をブロック（ソースブロック）単位に分割する。次にソースブロックを FEC エンコードのためのソースシンボルという単位に分割する。図 3-3 ではオブジェクトを二つのソースブロックに分割し、更にそれぞれのソースブロックを k 個のソースシンボルに分割している（図 3-3 の例では $k = 8$ ）。FEC エンコードにはソースシンボル k 個とパラメータ n を入力する。このパラメータ n によって、出力されるエンコードシンボルの数が変化する。つまり k 個のソースシンボルに対し、 $n - k$ 個の冗長シンボルが生成される（図 3-3 の場合は 8 個のソースシンボルに対し $10 - 8 = 2$ 個の冗長シンボルが生成されている）。FEC 符号化する際の条件としては、冗長符号の生成のため、シンボル長は一定にすることが望ましい。可変長のシンボルに対して FEC 符号化を行うこともできるが、シンボル長をそろえるためにエンコード時にデータのパディング、またデコード後にパディング部分を取り除く必要がある。

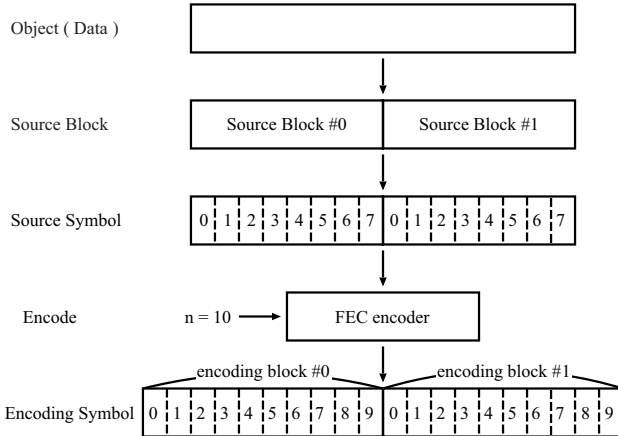


図 3・3 FEC エンコード例

生成されたシンボルの識別のためにフラグと ID をヘッダに付与する。フラグにより、そのシンボルがソースシンボルか冗長シンボルかを決定し、ID 付与により、番号を管理する。例えば、識別情報としては、[ブロック番号, フラグ, ID] が挙げられる。フラグが 1 の場合はソースシンボル, 0 の場合は冗長シンボルを表す。図 3・3 は $k = 8, n = 10$ の例であるので、各クライアントは、サーバから配信される同一ブロックの 10 個の packets のうち任意の 8 個を到着順序にかかわらず受信できれば、元の情報を復元することができる。

3-1-13 階層型マルチキャスト配信

(1) DCT を用いたデータ階層化

階層化マルチキャストなど、各端末の受信能力に応じた視聴環境を提供する方式では、別々のマルチキャストアドレスでデータを送信するため、サーバで配信データの階層化を行う必要がある。階層化方式は特に規定されていないので、様々な方式を用いることが可能であるが、基本レイヤのみでデータ再生可能であることが前提となる。

例として MPEG-2 でのデータ階層化について述べる。MPEG-2 の階層化方式は MPEG-2 国際標準³⁾として、データ分割 (data partitioning), SNR スケーラビリティ (SNR scalability), 空間的スケーラビリティ (spatial scalability), 時間的スケーラビリティ (temporal scalability) の四つのコーディング方式が用意されている。以下、データ分割の実現方式として、離散コサイン変換 (Discrete Cosine Transform: 以下 DCT) を用いたデータの階層化について述べる。

原画像 ($n \times n$ 行列): $f(x, y) \rightarrow F(u, v)$ に変換する DCT の変換式は以下ようになる (ただし, x, y, u, v 共に, $0 \sim n-1$ までの整数)。

$$F(u, v) = \frac{1}{\sqrt{2n}} C(u) C(v) \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \cdot \cos \frac{(2x+1)u\pi}{2n} \cos \frac{(2y+1)v\pi}{2n}$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & (x = 0) \\ 1 & (x \neq 0) \end{cases}$$

DCT 画像 $F(u, v)$ は、 $F(0, 0)$ が最も低い周波数を表し、 $F(n-1, n-1)$ が最も高い周波数となるため、図 3・4 のように原画像は左上部分が低周波数成分、右下部分が高周波成分に変換される。低周波数成分は画像への影響度が高く、高周波成分は画像への影響度が低いいため、データ分割を行う際、図 3・4 のように DCT の位置により左上部分を基本レイヤにし、右下部分を拡張レイヤにする。このような手法を用いることで、基本レイヤであるレイヤ 1 のみで映像再生が可能であり、拡張レイヤを追加することにより更に品質が向上されるというデータの階層化が可能となる¹⁶⁾。

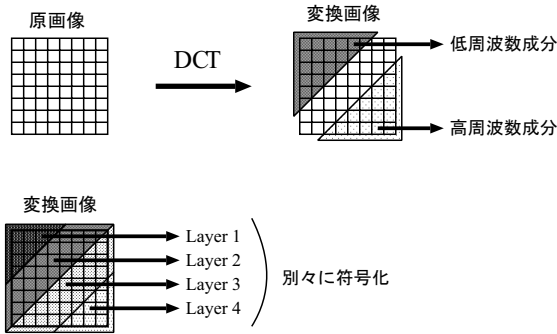


図 3・4 データの階層化

階層化されたデータを再生する場合、基本レイヤのみ受信している状態では映像再生に問題はないが、基本レイヤと差分レイヤを同時に受信している場合、基本レイヤと拡張レイヤの情報を結合する必要がある。よって、差分レイヤにも基本レイヤと同様のヘッダ情報（シーケンスヘッダ、GOP ヘッダ、ピクチャヘッダ）を付与しなければならない。

(2) 階層符号化

Scalable Video Coding (SVC)¹⁷⁾ は階層符号化方式として使用されている。SVC は H.264 / MPEG4 AVC の拡張である。SVC の目的は、ビットストリームレベルでのスケーラビリティを提供するビデオコーデックの開発にある。

SVC の基本符号化方式は、空間、時間、SNR の幅広いスケーラビリティを達成するため、階層化されたビデオコーデックとして区分されている。例として、時間スケーラビリティは階層 B ピクチャを利用して実現される。階層 B ピクチャでは、I/P ピクチャ（第 0 層と位置づける）を用いて（第 1 層）B ピクチャが作成されるが、その B ピクチャと I/P ピクチャから、（第 2 層）B ピクチャが新たに生成される。同様にして、（第 0, 1, 2, …, i 層）のピクチャから、第 $i+1$ 層の B ピクチャが生成される。サービスの際には、ユーザの希望する時間解像度のレベルに合わせて第 0 ~ i 層のピクチャを送ることにより、時間解像度のスケーラビリティが実現される。

また、SVC において階層化に関するよりフレキシブルな符号化を実現するために、Fine Grain Scalable (FGS)¹⁸⁾ 符号化が提案されている。FGS は、異なる量子化ステップサイズ

の基本レイヤと拡張レイヤに階層化された符号化ストリームを構成する方式である。拡張レイヤにおいては、符号化ストリームが、復号画像の品質への寄与が高い情報から順に連続的に階層化されている。受信側では基本レイヤ情報さえ保持すれば再生可能となる。そのため、基本レイヤのビットレート以上、拡張レイヤのビットレート以下の範囲であれば、受信環境にあわせた符号化ストリームを生成することが可能となる。

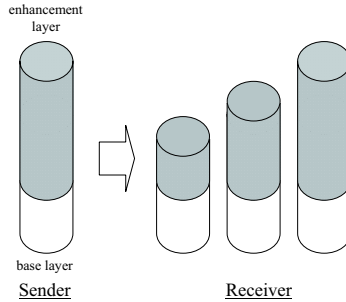


図 3・5 FGS の概観

FGS の特徴は、拡張レイヤのどの位置で切っても符号量に応じた品質の動画像が取り出せることである (図 3・5 参照)。これを実現するために拡張レイヤでは、変換符号化後の変換係数のコーディング順を変更している。つまり、マクロブロックをそのままスキャンするのではなく、マクロブロックをいくつかのパスに分割して、そのパスごとにスキャンする (図 3・6)。これにより、柔軟なレートに対応した品質を提供することが可能となる。

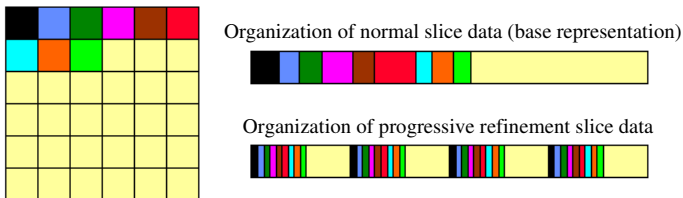


図 3・6 変換係数のコーディング順の変更

参考文献

- 1) M.S.Borella, D.Swider, S.Uludag, and G.B.Brewster, "Internet packet loss: Measurement and implications for end-to-end QoS," Proc. Int. Conf. on Parallel Processing, Aug.1998.
- 2) ISO/IEC 11172-1/2/3, "Information technology-coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s" 1993.
- 3) ISO/IEC 13818-2/3, "Information technology-generic coding of moving picture and associated audio" Mar. 1994.
- 4) B. Vandalore, W.Feng, R.Jain, and S.Fahmy, "A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia," OSU(Ohio State University) Technical Report, OSU-CISRC-5/99-TR14.

- 5) J.C. Bolot and T. Turletti, "A rate control mechanism for packet video in the internet," Proc. IEEE Infocom, Nov. 1994.
- 6) H. Schlzrinne, S. Casner, R.Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," audio-video transport working group, RFC1889, Sept. 1987.
- 7) 橋本浩二, 知念正, 佐藤純, 柴田義孝, "圧縮ビデオデータ転送のためのパケットおよびフレームレート制御法," 情報処理学会論文誌, vol.39, no.2, pp. 339-341, Feb. 1998.
- 8) N.G. Duffield, K.K. Ramakrishnan, and A.R. Reibman, "SAVE: An algorithm for smoothed adaptive video over explicit rate networks," Proc. IEEE Infocom, Apr. 1998.
- 9) W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video," Proc. SPIE Multimedia Networking and Computing, pp.316-327, Nov. 1997.
- 10) C.Liu, Y.Xie, M.J. Lee, and T.N. Saadawi, "Multipoint multimedia teleconference system with adaptive synchronization," IEEE JSAC, vol.14, no.7, pp.1422-1435, 1998.
- 11) S.H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," Proc. PacketVideo Workshop'02, Pittsburgh, USA, Apr. 2002.
- 12) S. Wee, W. Tan, J. Apostolopoulos, and M. Etoh, "Optimized video streaming for networks with varying delay," Proc. IEEE International Conference on Multimedia & Expo'02, Lausanne, Switzerland, Aug. 2002.
- 13) J.O. Kim, H. Tode, and K. Murakami, "Feedback-based Adaptive Video Streaming over Lossy Channels," IEICE Trans. Commun., vol.E87-B, no.10, pp.3076-3084, Oct. 2004.
- 14) P.A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," IEEE Trans. Multimedia, May 2005.
- 15) L. Rizzo and L. Vicissano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques," Proc. The Fourth HPC '97 IEEE Workshop, June 1997.
- 16) F. Ruijin, L.B. Sung, and A. Gupta, "Scalable Layered MPEG-2 Video Multicast Architecture," IEEE Trans. Consumer Electronics, vol.47, no.1, pp.55-62, Feb. 2001.
- 17) "HHI presentation of the Scalable Extension of H.264/AVC," http://ip.hhi.de/imagecom_G1/savce/index.htm
- 18) F. Wu, S. Li, and Y.-Q. Zhang: "A framework for efficient progressive fine granularity scalable video coding," IEEE Trans. on Circuits and Systems for Video Technology, vol.11, no.3, pp.332-344, Mar. 2001.

3 群 - 5 編 - 3 章

3-2 資源予約

(執筆著者：長谷川輝之)[2011 年 1 月受領]

インターネットに代表されるパケット交換網で音声や映像などのリアルタイム通信を実現するための品質制御技術として、パケット転送に先立ち帯域等のネットワーク資源を予約するアプローチがある。本節では、インターネットにおける資源予約の概要を述べ、具体的なアーキテクチャである IntServ (Integrated Services) と、これを実現するプロトコルである RSVP (Resource ReSerVation Protocol) について解説する。

3-2-1 インターネットにおける資源予約

インターネットでは、通信経路上のルータがパケットをホップバイホップで次ノードに転送し宛先ホストまで配送する。ルータ間の転送はベストエフォートであり、ルータに出力回線帯域を上回る入力があればパケットの損失やキューイング遅延が発生する。回線交換網と異なりエンドツーエンドでのパケット流量制御が困難なことから、インターネットにおける帯域と遅延、パケット損失率などの通信品質 (QoS: Quality of Service) の扱いについて、1980 年以前より様々な検討が行われてきた。1979 年には MIT を中心に音声通話や音声会議を目的としたプロトコル ST¹⁾ (Stream Protocol) が開発された。1990 年には ST を改良した ST-II が RFC1190²⁾ として標準化され、IP バージョン 5 が割り当てられている。ST-II は①資源予約と②データ転送から構成される 2 段階の通信モデルを採用している。具体的には第 1 段階として、SCMP (ST Control Message Protocol) により、送信ホストから受信ホストに向かってリアルタイムデータの転送に必要なネットワーク資源をホップバイホップで交渉並びに予約し、片方向の通信路 (ストリーム) を確立する。資源予約のパラメータとして、ストリームの要求帯域や許容遅延を含む Flow Specification (FlowSpec) が規定されている。次に第 2 段階として確立したストリームを利用して QoS を保証したデータ伝送を行う。図 3・7 に ST によるデータ転送の概念図を示す。

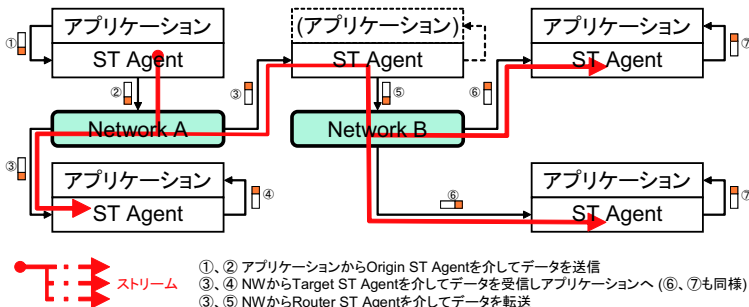


図 3・7 ST によるデータ転送の概念図

ST は、インターネットにおける資源予約という概念をいち早く導入したプロトコルである。1995 年には IBM が中心となり ST2+ (RFC1819) に更新されたものの、①当時 BSD 系の

OS で広く実装されていた IP バージョン 4 とは異なるプロトコル体系であること、②SCMP メッセージの転送に信頼性を要求するハードステート型プロトコルであり実装が複雑化すること、③予約した資源を担保するための転送制御手法が未確立であったことなどの理由により普及には至らなかった。その一方で、IP バージョン 4 (後継であるバージョン 6 を含む) のインターネットでリアルタイムとベストエフォートのアプリケーションを統合的に提供する機運が高まり、後述する IntServ アーキテクチャや RSVP が策定された。IntServ や RSVP にも ST-II の資源要求の枠組みである FlowSpec の概念は受け継がれている。

3-2-2 IntServ (Integrated Service)

1994 年 3 月の IETF 会合で設立された IntServ WG (Working Group) では、ST-II の標準化がプロトコル実装先行で進められた反省から、統合サービス基盤として IP バージョン 4 のインターネットを拡張するための要求機能の整理とアーキテクチャ確定に注力することとなった。IntServ WG は、8 件の RFC (RFC2210-2216, 3360) を発行し 2000 年 12 月に終了している。WG での具体的な検討項目は以下のとおりである³⁾。

- QoS の定義とサービスの規定、資源要求パラメータの具体化。
- ネットワークエレメント (ホスト, ルータ, サブネット) に対する要求機能と実装フレームワークの規定。

IntServ も、「経路上の全ノード (ホスト, ルータ)」で通信に先立ち FlowSpec に基づく資源予約を行うアーキテクチャであり、資源予約のための①シグナリングと②受付 (アドミッション) 制御、及び、トラフィック監視や制御のための③分類識別 (クラシファイ) と④適合性検証 (ポリシング)、スケジューリングとシェーピングなどを構成要素とする (3-2-3 節図 3・10 参照)。

サービスとしては、従来のインターネットで提供するベストエフォートサービスに加えて、Guaranteed Service⁴⁾と Controlled-Load Service⁵⁾との 2 種類が規定されている。図 3・8 に各サービスの特徴をまとめる。

Guaranteed service	Controlled Load service	Best Effort service
<ul style="list-style-type: none"> - 帯域の確保 - 数学的厳密性のあるエンド・エンド最大遅延保証 - 適合パケットに対するキューイングロスが無いことを保証 	<ul style="list-style-type: none"> - 負荷の少ないBest Effortサービスと同等のQoS - 適合パケットに対するキューイングロス及び極端な遅延が殆ど発生しない 	<ul style="list-style-type: none"> - QoSは保証しない
TSpec, RSpec	TSpec	N/A

図 3・8 IntServ における各サービスの特徴

(1) 資源予約パラメータ

FlowSpec は TSpec (Traffic Specification) と RSpec (Service Request Specification) に分類される．ホストやルータが送受信するトラフィックの特性を示すパラメータ (平均レート，ピークレートや最大バーストサイズなど) は TSpec で記述し，サービスに要求する QoS のパラメータ (帯域や遅延など) は RSpec で記述する．

TSpec は RFC2215 ⁶⁾ に一般的なパラメータのひな形が規定されており，トークンバケツモデルである TOKEN_BUCKET_TSPEC パラメータが Guaranteed Service と Controlled-Load Services 共通で利用されている．一方，RSpec は Guaranteed Service のみが利用し，具体的なパラメータはサービス仕様である RFC2212 に規定されている．TSpec 構成要素 (r, b, p, m, M) と RSpec 構成要素 (R, S) の概要を含む IntServ のアーキテクチャを図 3・9 にまとめる．

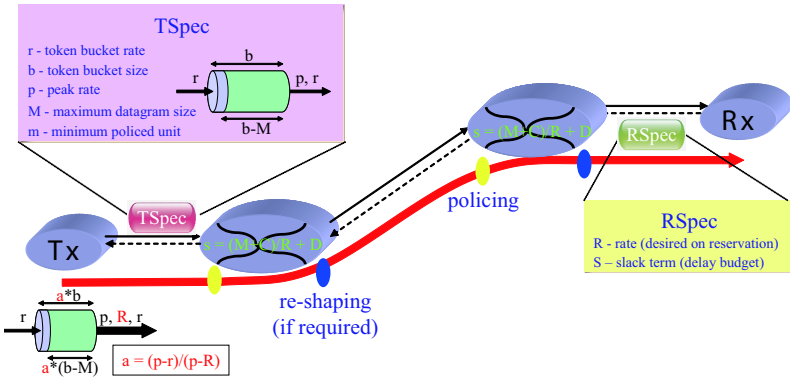


図 3・9 Intserv アーキテクチャと資源予約パラメータ

(2) Guaranteed Service [RFC2212]

Guaranteed Service サービスは，資源予約を行ったトラフィックに対して，伝送帯域並びにルータで付加されるキューイング遅延の最大値をエンドツーエンドで厳密に保証した，損失のないパケット転送を提供する．具体的な転送原理は Parekh のモデル⁷⁾に基づいており，ホストから TSpec に従って送信したパケットに対して，各ルータが WFQ (Weighted Fair Queuing) などの「流体モデルで近似可能な」スケジューリングを行うことで，エンドツーエンドでの最大キューイング遅延 D_{max} を保証する．RFC2212 が保証する D_{max} は以下のとおりである．

$$D_{max} = \begin{cases} (b - M)/R * (p - R)/(p - r) + (M + C_{tot})/R + D_{tot} & (p > R \geq r) \\ (M + C_{tot})/R + D_{tot} & (R \geq p) \end{cases}$$

ここで C_{tot} と D_{tot} は経路上の各ルータの付加遅延 C と D をそれぞれ合計した項である．予約帯域 R に依存する項は C ，しない項は D で表現する．例えば WFQ のスケジューリング遅延は C ，MAC レイヤのスロット待ち時間や経路計算処理による転送中断時間は D に含める．

Guaranteed Service では，FlowSpec として TSpec と RSpec の双方を用いる．資源予約時

に、経路上の各ルータは自身が付加する遅延 s を計算し順次 RSpec を更新していく。通常は各ルータで S から s を減算するが、 S に余裕がなければ R を削減することもあり得る。パケット転送に際しては、ルータで、受信トラヒックに対して TSpec に基づくポリシングを行うだけでなく、必要に応じて TSpec に従った再シェーピングを行うことが想定されている。

以上のように、Guaranteed Service は、流体モデルスケジューリング適用を前提に、厳密な受付制御とポリシング及び再シェーピングを用いて QoS を提供するモデルとなっている。

(3) Controlled-Load Service [RFC2211]

Controlled-Load Service は、資源予約を行ったトラヒックに対して、低負荷なベストエフォートネットワークと同等のパケット転送特性をネットワーク負荷によらず提供する。ルータの実装容易性も考慮し、前述の Guaranteed Service のように厳密な QoS 保証は行わずに、ホストから TSpec に従って送信したパケットについて、「パケットが損失なくかつ極端に遅延せず」に到着する状況をはば確実に実現する緩やかな QoS 保証の提供を目指している。ここで、ほぼ確実に実現するとは、パケット損失並びにバースト時間 ($= M/r$) を超えるような遅延がほとんど発生しないことを意味する。よって、アプリケーションが軽微な品質劣化に適應することが前提となっている。

Controlled-Load Service では、FlowSpec として TSpec のみを交換し RSpec は利用しない。資源予約時の受付制御では、各ルータで「經由する全フローの TSpec の合計」が自身の提供資源を超過しないようにフローを受け付ける保守的な運用が考えられるが、これに加えて、実効的な資源使用状況に応じたオーバブッキング運用も認めている。パケット転送に際しては、各ルータは受信したトラヒックについて TSpec に基づくポリシングを行い、超過分はベストエフォートとして扱う。ルータでのトラヒックの再シェーピングは行わない。

以上のように、Controlled-Load Service は、比較的単純な受付制御とポリシングのみを用いて緩やかな QoS を提供するモデルとなっている。

3-2-3 RSVP (Resource ReSerVation Protocol)

IntServ WG と同時期に設立された RSVP WG により、IntServ に基づく資源予約を実現するシグナリングプロトコルとして RSVP の標準化が進められた。RSVP WG は、10 件の RFC (RFC2205-9, 2745-7, 2961, 3097) を発行し 2001 年 5 月に終了している⁸⁾。RSVP ノードの機能ブロック構成を図 3・10 に示す。以下、RFC2205⁹⁾と Intserv WG で規定された RFC2210¹⁰⁾を中心にプロトコル概要を説明する。

1. 送信ホストから受信ホストに至る片方向フローの資源予約を受信側から行う。すなわち、資源予約のための Resv メッセージは受信側から送信側に向けて発信する。
2. Resv メッセージ発信に先立ち、送信側から受信側に向けて、送信トラヒックの特性である SENDER_TSPEC オブジェクトを含む Path メッセージを広告する。SENDER_TSPEC は経路上のルータでは変更されない。
3. 経路上のルータは、Path メッセージの ADSPEC オブジェクトを新規に生成（あるいは更新）して、利用可能なサービスや遅延と帯域の見積等、受信側での資源予約パラメータ決定に必要な情報を付加する。

4. 受信ノードは, Path メッセージの内容に基づき資源予約パラメータである FLOWSPEC オブジェクト (Receiver.TSpec, RSpec) を決定し, これを含む Resv メッセージを生成し発信する. なお, RSpec は Guaranteed Service のみで利用する.
5. 着 IP アドレス・プロトコル ID・着 TCP/UDP ポート (あれば) の三つ組 (3-tuple) が共通のフロー集合を RSVP セッションと呼ぶ. セッションの発 IP アドレス (複数可)・発 TCP/UDP ポート (あれば) は, Resv メッセージ内の FILTERSPEC オブジェクトに記述する. FILTERSPEC と FLOWSPEC の組をフロー記述子と呼ぶ.
6. 経路上の各ルータは, 隣接ノードから Resv メッセージを受信すると, ポリシー制御機能やアドミッション制御機能に受付可否を確認する. 受付可であればクラシファイアやスケジューラの動作パラメータを設定する.

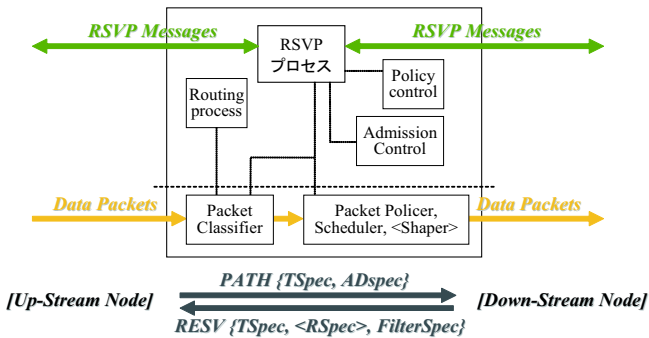


図 3・10 RSVP ノードの機能ブロック構成

エンドツーエンドで Path と Resv を交換し資源予約が完了すれば, 対応するフローの QoS が確保される. RSVP セッションの情報は, 通信が継続する間, 経路上の各ノードで保持する必要がある. RSVP はソフトステート型プロトコルであり, ノード間で定期的に Path と Resv メッセージを交換することでセッションを維持する.

RSVP のメッセージフォーマット概要を図 3・11 に示す. メッセージのタイプ (Path や Resv など) や長さを指定する「共通ヘッダ」とタイプごとのパラメータ (FLOWSPEC や FILTERSPEC など) を記述する「オブジェクト」から構成される

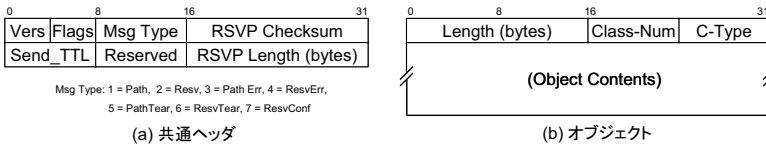


図 3・11 RSVP メッセージフォーマット概要

上述のように、RSVP は受信者主導で資源予約を行っているため、マルチキャストとの親和性は高く、分岐ルータでの Resv メッセージの結合や、経路ループを防ぐための SCOPE オブジェクトなど、マルチキャストサポートに必要な各種仕様が規定されている。

3-2-4 まとめ

本節では、インターネットにおける品質制御技術の一つである資源予約アプローチについて、RSVP と IntServ を中心に説明した。RSVP や IntServ は、ルータにフロー単位の資源管理を要求するといったスケラビリティの問題を抱えていることから、インターネットでは普及しなかった。一方で、MPLS (Multi Protocol Label Switching) パスの帯域確保を行うシグナリング手段として拡張され (RSVP-TE: Traffic Engineering)、ネットワーク規模をコントロール可能な通信キャリア網内では広く活用されている。

参考文献

- 1) J. Forgie, "ST - A Proposed Internet Stream Protocol," IEN (Internet Experiment Note) -119, Sept. 1979.
- 2) C. Topolcic, Ed., "Experimental Internet Stream Protocol, Version 2 (ST-II)," RFC1190, Oct. 1990.
- 3) <http://www.ietf.org/html.charters/OLD/intserv-charter.html>.
- 4) S. Shenker, et al., "Specification of Guaranteed Quality of Service," RFC2212, Sept. 1997.
- 5) J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC2211, Sept. 1997.
- 6) S. Shenker, et al., "General Characterization Parameters for Integrated Service Network Elements," RFC2215, Sept. 1997.
- 7) A.K. Parekh, et al., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Trans. On Networking, vol.1, no.3, pp.344-357, June 1993.
- 8) <http://www.ietf.org/html.charters/OLD/rsvp-charter.html> (注: RFC2205 は削除されている)。
- 9) R. Braden, Ed., "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification," RFC2205, Sept. 1997.
- 10) J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC2210, Sept. 1997.

3 群 - 5 編 - 3 章

3-3 キューイング制御

(執筆者: 川原亮一) [2011 年 1 月受領]

ある出力リンクに対して、複数のフローからのパケットが入力される場合、このリンク速度 (= リンク帯域) の効率的な利用やフロー間の公平性、フロー種別に応じた優先制御を実現するための手段として、その出力リンク待ちバッファ (キュー) において様々なキューイング制御技術がある。ここで、フローの定義としてはいろいろなケースが考えられるが、例えば [発信元 IP アドレス, 着信先 IP アドレス, 発信元ポート番号, 着信先ポート番号, プロトコル] の五つ組を同じくするパケット群をフローとすることが考えられる。このとき、TCP コネクションは 1 本のフローに対応する。別のフローの定義としては、例えば同一アプリケーションからのトラフィックを一つのフローとする場合も考えられる。

キューイング制御は大きく二つに分けられ、(i) フローごとにキューを分けて各キューからのパケット取り出しルールを与えるパケットスケジューリング技術と、(ii) キューは複数フロー間で共有として、そのキューへパケットが到着したときにキューの状態やそのパケットの種別に応じてパケットを選択的に廃棄するパケット選択廃棄技術がある。(i) については、PQ (priority queueing)、WFQ (weighted fair queueing)¹⁾、DRR (deficit round robin)²⁾などが有名である。(ii) については、RED (random early detection)³⁾、RIO (RED with In and Out)⁴⁾などがある。また、別のアプローチとして、(i) のようにフローごとにパケットスケジューリングを行う代わりに、スケーラブルにフロー単位でのトラフィック制御を実現する方法として、CSFQ (core stateless fair queueing)⁵⁾と呼ばれるキューイング制御も提案されている。

以下で、各方式について説明する。なお、各方式の詳しい解説は文献 6) を参照頂きたい。

3-3-1 スケジューリング方式

(1) Priority queueing

PQ について説明する前に、FIFO (first-in first-out) について述べる。FIFO では、フロー間で共有のキューが一つあり、すべてのフローのパケットを到着順にキューの先頭から出力する。これに対し、PQ では、図 3・12 のように複数のキューを用意する。PQ では、フローが送信するパケットに優先度を示すプライオリティ値を記入する。一方、パケットを受信するノードではパケットの優先度ごとにキューを用意する。図 3・12 の例では、三つの優先度があるとし、“1” が記されているパケットを最高優先度、“2” のパケットを次の優先度、“3” のパケットを最低優先度であるとしている。ノードは常に最高優先度のキューにあるパケットから送出する。そのキューが空であれば、次の優先度のキューにあるパケットを処理する。こうすることにより、最高優先度のパケットに対しては、ほかの優先度のトラフィックの影響をほとんど受けずに出力リンク速度でパケットを送出可能となる。利用例としては、遅延やパケット損失率に敏感な音声パケットを高優先、データパケットを低優先にすることが考えられる。

(2) Deficit round robin

PQ では低優先パケット転送に対する不公平性を生ずる可能性があるが、ここで紹介する DRR は、フロー間公平性を考慮している。基本的な考え方は、あらかじめ定めた重み w_i

($\sum_i w_i = 1$) に従って各キュー i からパケットをラウンドロビンで取り出すことにより、各キューからの出力レートが出力リンク速度の $w_i / \sum_i w_i$ となるようにする。つまり、重み付けされた公平な帯域割当を実現している。具体的なパケットの取り出し方は以下のとおりである。まず、各キュー i に対して、重み w_i に比例する量である Quantum Size QS_i [byte] というものを定める。図 3・13 のように、今、各キューに出力待ちのパケットが格納されているとする。ここで各パケットの数値はパケットサイズを表す。今、キュー 1 からパケットを取り出そうとしているとする。そのとき、キュー 1 に割り当てられる deficit counter DC_1 を $DC_1^{new} = DC_1^{old} + QC_1$ とする。ここで $DC_1^{old} = 0$ だったとし、 $QC_1 = 500$ とすると、 $DC_1^{new} = 500$ となる。次に、キュー 1 から DC_1 の範囲内でパケットを送出する。この例では、キューの先頭から二つ目のパケットまで ($100 + 100$ [byte]) が取り出される。その後、 $DC_1 = 500 - 100 \times 2 = 300$ とする。これを繰り返すことにより、可変長のサイズを持つパケットに対して重み付け帯域割当を実現する。

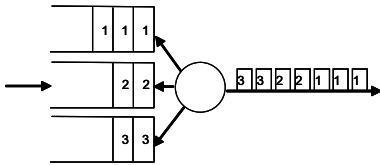


図 3・12 Priority queueing

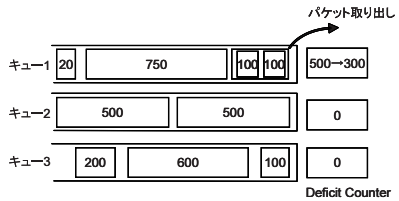


図 3・13 Deficit round robin

(3) Weighted fair queueing

WFQ においても、各キューから重み w_i でパケットを取り出すことにより公平性を実現している。WFQ では、まず、仮想的にパケットサイズが無限小であるとしたときの理想的なスケジューリングを考える。図 3・14 のように、フロー 1, 2, 3 からパケットが到着したとする (簡単のためすべてのパケットサイズを 1500 [byte] とする)。また出力リンク速度を 1500 [byte/s]、重みを $w_1 = 0.5$, $w_2 = 0.25$, $w_3 = 0.25$ とする。このとき、理想的なパケットスケジューリング (帯域割当) は本図の“理想帯域割当”のようになる。すなわち、理想スケジューリングでは、ある時点において、送出すべきパケットが存在するフローに対して、 $w_i / \sum_i w_i \times 1500$ [byte/s] の帯域を割り当てている。WFQ では、キューにあるパケットの中で、この理想的なパケットスケジューリングで処理が最も早く完了する (完了済みのものを含む) パケットを選択して出力処理を行う。その結果、図 3・14 での“出力”のようにパケットが出力される。

(4) Class-based queueing

ここでは、WFQ や DRR のようにフローごとのスケジューリングを行わずに、クラス単位でスケジューリングする方法 CBQ (Class-based queueing)⁷⁾ について説明する。図 3・15-(a) を例に説明する。まず、組織 A と B というクラスが存在し、それぞれに出力レートの 50% ずつを割り当てる。次に、各組織の中に、リアルタイムとデータという 2 種類のフローを設ける (例えば前者は音声・ストリーミング、後者はそれら以外の通信)。また各フローへの割当を図 3・15-(a) のようにする。CBQ では、組織 A が割当レートを使い切れない場合にかぎ

り、組織 B がその分余分に帯域を使用できるようにスケジューリングする。前節までで紹介したスケジューリングでは、組織 A 中のリアルタイムフローが帯域を残せば、組織 A のリアルタイムフロー以外のすべてのフロー（組織 B のフローを含む）が公平にその余りの帯域を使用できた。これに対して、CBQ ではフローの間にクラスという構造を入れて、未使用帯域はまず同一クラス内のフローで配分し、更に余裕がある場合にかぎり、ほかのクラスへまわすという考え方である。

上記を実現するため、CBQ ではクラスごとのトラヒックを監視・制御する機構を設けている。CBQ では、実際の転送レートが割当てレート以下であり、かつある期間持続してキューに送信すべきパケットが滞留しているようなフローまたはクラスを救済する。このようなクラスを非充足クラスと呼ぶ。非充足クラスが存在するということは、実際の転送レートが割当てレートを上回るクラス（超過クラス）が存在し、CBQ ではこのようなクラスを検出・制御する。例えば図 3-15-(b) のように、組織 A のリアルタイムフローと組織 B のそれが超過クラスになっているとする。この図で、色付きの円が実際のレートの大きさを表しており、その内側の円が割当てレートを表している。また、組織 B のデータフローが非充足となっているとする。この場合、組織 A 全体として超過クラスでなければ規制の対象にはならず、組織 B のリアルタイムフローのみが規制の対象となる。

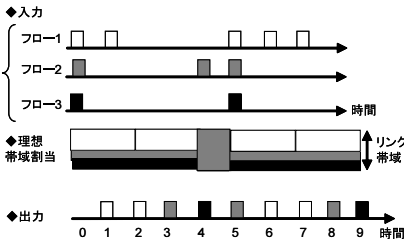


図 3-14 Weighted fair queueing

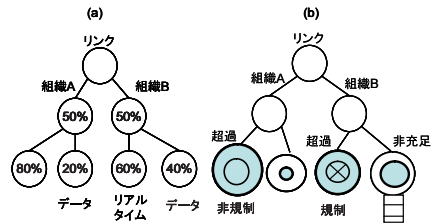


図 3-15 Class based queueing

3-3-2 パケット廃棄方式

(1) RED

フローごとあるいはクラスごとにキューを設けず、FIFO のような共有キューが一つあり、そのキューの入り口でのパケット廃棄法を工夫する手法の一つとして、RED がある。

通常はトラヒックが集中するとキュー長が増大し、キューが満杯となってパケットがあふれる。これを Drop Tail と呼ぶ。ここで TCP フローを集約する場合に着目し、Drop Tail ではどのような問題が生ずるかを考える。TCP では、輻輳発生によりパケットが廃棄されると、送信ホストがデータ送信レートを抑制する。したがって単純な Drop Tail では以下の問題が生ずる可能性がある。(1) 輻輳検出の契機はキューあふれであり、そのときは既に輻輳が発生しているため、輻輳回避にはならない。(2) パースト性の高いトラヒックは、送信レートが低くても、キューあふれを生ずる可能性が高く、結果として減速対象となりやすく公平性を欠く。(3) 輻輳が進むとほとんどすべてのフローが減速対象となり、ネットワークの負荷が急激に減少し、リンク帯域の利用効率が悪くなる。この現象を global synchronization と呼ぶ。

そこで、RED では、パケットを早めに廃棄して輻輳を事前を回避し（上記問題 (1) への対応）、かつランダム廃棄により上記問題の (2)(3) を解決しようとしている。RED の動作概要は以下のとおりである。まず、キューの平均長を算出して輻輳の度合いを推定する。具体的には、平均キュー長を $avgQ$ としパケット到着ごとに

$$avgQ^{new} = (1 - w_q)avgQ^{old} + w_q \times Q \quad (3-1)$$

と更新する。ここで Q はその時点でのキュー長であり、 w_q はあらかじめ定めるパラメータである（e.g., $w_q = 0.002$ ）（厳密には、キューが空の場合には、空であった時間の長さに応じて $avgQ$ を小さく補正している³⁾）。

次に、平均キュー長によって定まる確率でパケットを廃棄する。廃棄確率 p_b は平均キュー長に関する単調非減少関数であり、以下で計算される（図 3-16 参照）。

$$p_b = p_{max}(avgQ - min_{th}) / (max_{th} - min_{th}) \quad (3-2)$$

ここで、 p_{max} , min_{th} , max_{th} はあらかじめ定めるパラメータであり、例えば $p_{max} = 1/50$, $min_{th} = 5$ パケット, $max_{th} = 15$ パケット, である。一方、最後にパケットが廃棄されてからキューに収容されたパケット数をカウントしておきその値を $count$ とし、現在の平均キュー長が $min_{th} \leq avgQ \leq max_{th}$ であれば、確率 $p_b / (1 - count \times p_b)$ でランダムに廃棄する。もし $max_{th} \leq avgQ$ であれば、必ず廃棄する。

(2) RED with In and Out

RED の拡張版の一つとして、RIO (RED with In and Out) と呼ばれるものがある。この方式では、ネットワークの入り口において、フローごとにトラヒックのポリシングを行うことを前提としている。すなわち、フローごとに、あらかじめ定めた許容送信レート以内でパケットが送信されていればそのパケットには適合のマーク (in) を、そうでなければ不適合のマーク (out) を付与する。

一方、ネットワーク内の各ノードでは、in, out パケットにそれぞれ異なるパケット廃棄確率を割り当てる。ここで、out パケットの廃棄確率を、in パケットの廃棄確率より大きくするように設定することにより、in パケットを多く含むフローの出力レートを大きくし、各フローに許容レート以上のスループットを保証しようとしている。

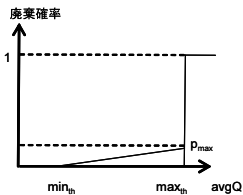


図 3-16 RED でのパケット廃棄確率

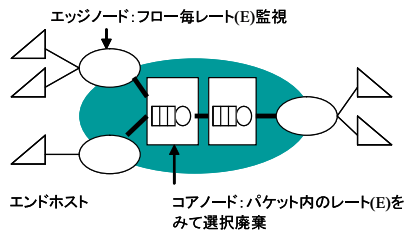


図 3-17 CSFQ

3-3-3 スケーラブルなキューイング制御: CSFQ

基本的な考え方は、ネットワークの境界に位置するエッジノード（多重フロー数が少ない）と、内部に位置するコアノード（多重フロー数が膨大）で協調動作をすることにより、スケーラブルにフロー単位でのトラフィック制御を実現している（図 3・17 参照）。具体的には、エッジノードにおいてフローごとの送信レートを監視し、その結果をパケットヘッダ内にマークし、コアノードではフローごとの状態は管理せずに、パケットヘッダ内の情報のみをみて、パケットを受け付けるか否かを判定する。

CSFQ の動作は以下のとおりである。コアノードにおいて、目標レート TR とパケットにマークされている送信レート E を比較してパケットの選択廃棄を実施する。ここで目標レートの意味は、もし現在通信中のフローが N 本存在し、出力リンク速度が C であれば、 C/N （帯域を均等割りする場合）が目標レートとなる。 TR を決定するために、パケット到着レート A とパケット受付レート F （到着したパケットのうち、実際に受け付けられた分）を測定する。具体的には、パケット到着ごとに

$$A^{new} = (1 - e^{-T/K}) \times l/T + e^{-T/K} \times A^{old} \quad (3 \cdot 3)$$

により A を更新する。ここで、 T は現在の到着パケットと最後に到着したパケットの到着間隔、 l はパケットサイズ、 K はあらかじめ定めるパラメータ（e.g., $K = 0.1$ s）である。 F についても同様の手順で更新する。 A と F を用いて、 TR を以下のように計算する。もしあらかじめ定めた期間 T_c の間、 $A > C$ （ C は出力リンク速度）ならば、 $TR^{new} = TR^{old} \times C/F$ とし、そうでなければ、最後に TR を更新してから観測されたパケット送信レート E の最大値を TR^{new} に設定する。

この TR を用いて、パケットの受付可否判定を行う。到着パケットの送信レート E が TR より小さければそのパケットを受け付ける。もし $E \geq TR$ であれば、確率 $1 - E/TR$ で廃棄する。

こうすることにより、各フローの送信レートを目標値となるように制御している。同様のアプローチとして、RFQ (rainbow fair queueing)⁸⁾ や執筆者らの検討⁹⁾ などがある。

参考文献

- 1) A. Parekh, R. Gallager, "A generalized processor sharing approach to flow control in integrated services network: the single node case," IEEE/ACM Transaction on Networking, vol.1, no.3, June 1993.
- 2) M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," ACM SIGCOMM 95, Sep. 1995.
- 3) S. Floyd, and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. on Networking, vol.1, no.4, August 1993.
- 4) D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," IEEE/ACM Trans. on Networking, vol.6, no.4, pp.362-373, August 1998.
- 5) I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high-speed networks," ACM SIGCOMM 98, pp.118-130, 1998.
- 6) 戸田, "ネットワーク QoS 技術," オーム社.
- 7) S. Floyd, V. Jacobson, "Link-sharing and recourse management models for packet networks," IEEE/ACM Trans. on Networking, vol.3, no.4, August 1995.

- 8) Z. Cao, Z. Wang, and E. Zegura, "Rainbow fair queueing: fair bandwidth sharing without per-flow state," INFOCOM2000, 2000.
- 9) Ryoichi Kawahara and Naohisa Komatsu, "A Scalable IP Traffic Control Method for Weighted Bandwidth Allocation per Flow," IEICE Trans. Commun., vol.E84-B, no.10, pp.2815-2829, Oct. 2001.

3 群 - 5 編 - 3 章

3-4 計測技術

(執筆者: 鶴 正人)[2011年1月受領]

一般に、大規模で複雑なシステムでは、「計測・解析」「モデル化」「設計・制御」が三位一体となって品質管理を支えている。情報通信サービスも例外ではなく、ネットワーク計測技術は不可欠な要素であるが、インターネットでの広域性、多様性・発展性、高速性、あるいは管理の分散性に起因して、その特性計測の手法や特性のモデル化は必ずしも十分に確立されておらず、様々な研究が進行中である。例えば、単純に必要な場所で必要なだけデータを収集することは困難または非効率であり、直接的・全体的な計測に頼らずに、間接的あるいは部分的な計測データから解析・推定を行う必要がある。また、個々の計測技術だけでなく、複数の組織、地域にまたがった協調的な計測あるいは計測情報の共有、更にそのための計測技術に関する標準化も重要である。

近年では、通信品質に非常に敏感な、あるいはデータ量が格段に多いアプリケーションが増え、また、無線や移動体通信のように状態変化が急な通信環境も一般的になってきたことから、計測・解析の「質」に関しても、多様で詳細な特性（例えば、RTTだけでなく片道遅延、平均だけでなく分散やパーセンタイル、トラヒックの総量だけでなく細粒度のフロー特性）の把握が必要になり、実時間性、継続性、あるいは高信頼性も要求されるようになってきた。例えば、計測に基づく制御のためには実時間性と信頼性が不可欠であり、また、Service Level Agreement (SLA) 監視に利用される場合は、契約違反の判断のための精度と信頼性が要求される。更に、インターネットにおけるセキュリティ問題の深刻さは増大する一方であるが、その検出や原因分析において、通信データの中身を含めた高度な解析を高速に実現する技術や広域で連携するシステムがキーになる。本節では、ネットワーク特性計測の一般論を説明した後、パスに沿った帯域幅の推定技術、及び高速ネットワークでの細粒度のトラヒックフロー特性の計測・推定技術の事例を簡単に紹介する。

3-4-1 ネットワーク特性の計測

(1) 計測の対象

計測対象は様々であるが、大別すると、「エンドツーエンドの通信品質・性能」と「ネットワーク内部（局所または大域的）の状態・特性」に分類できる。前者は、パス（エンドツーエンドで IP パケットが通過する経路）に沿った一般のトラヒックが時間平均的に経験する品質を指す場合と、特定のトラヒックフローが経験する品質を指す場合があり、ネットワーク利用者寄りの情報である。例えば、接続性、パケット損失（率やパターン）、パケット遅延やその変動、パケットの順序逆転、スループット、あるいは音声や動画の再生品質などがある。このようなエンドツーエンドの通信品質の基本的指標は、IETF の Working Group* の一つである IP Performance Metrics (IPPM) による標準化が進み、また、アプリケーションレベルの性能指標の検討も Performance Metrics for Other Layers (PMOL) などで始まっている。

一方、後者は、ネットワーク構成要素や内部区間、あるいはパスの状態・特性に関するものと、そこを流れるトラヒックの局所または大域的特性（量・種類など）に関するものがあり、

* <http://www.ietf.org/html.charters/wg-dir.html>

ネットワーク提供者寄りの情報である。IETF の Working Group である IP Flow Information Export (IPFIX) や Packet Sampling (PSAMP) などで標準化が進められている。

計測対象の多くは変動量であり、その統計的な振る舞いの把握が目標であるので、計測の時間粒度・間隔・期間などが重要である。一方、帯域幅、伝搬遅延、バッファ量のような確定量や、経路（トポロジー）のような短い時間では変化しないものを、それらを直接は知ることができない立場の観測者が推定したい場合もある。

(2) アクティブ計測とパッシブ計測

計測データを取得する形態として、人為的に試験パケットを送受信してその振る舞いを観測するアクティブ計測と、ネットワークを通過する実トラフィックの振る舞いを観測するパッシブ計測とがある。前者は試験パケットが通過したパスや区間の特性を知ることができる。最も基本的で普及しているツールとして、到達性を調べる ping、IP 経路を調べる traceroute、TCP や UDP のフローを流してスループットを調べる iperf などがあるが、パケットペア（接近した 2 個のパケット）、パケットトレイン（近接したパケット並び）などを用いる様々な計測・推定手法も開発されている。目的に応じて、必要なタイミングに必要な場所で、試験パケットの送信間隔やサイズを制御した技巧的な試験トラフィックを生成して計測を行える点が特長であるが、多量の試験パケットはネットワークにとって余計な負荷になる。

後者はネットワークにどのようなトラフィックが流れているかを知るためには不可欠であり、観測地点を通過する総量の計測だけでなく、パケットの中身を見てフローを識別し、フローに関する特性の計測・推定を行う。通常最も細かいフローの単位は、始点・終点 IP アドレス、始点・終点ポート番号、プロトコル番号の五つ組であり、目的に応じて、それらを集約した様々な粒度のフローも計測対象になる。更に、特定の双方向フローの観測からプロトコル解析も行うことができる。

各地点で計測された、上記のようなネットワーク品質・特性の情報やネットワーク内部資源の情報（構成、設定、状態）は、一般に別の場所（ネットワーク管理サーバ）に集めて解析・利用される。そのような情報収集・管理のためのプロトコルとして標準的には Simple Network Management Protocol (SNMP) が用いられる。

片道遅延の計測やその他の多地点での連携計測においては、2 点での時計（タイムスタンプ）の同期と精度が問題になる。一般にネットワーク上の機器の時計どうしは同期していないうえに進み方も違う。ネットワーク経由で計算機の時刻を同期させるためのプロトコルとして Network Time Protocol (NTP) が知られているが、高い精度は期待できない。一方、各計測点が GPS を使って正確な共通の時刻を維持することもコストが大きい。よって、ネットワーク経由での高精度な時刻同期や計測データの時刻補正なども研究されている。

広域ネットワークでは、多地点で計測されたデータの転送は、それ自身がネットワークに余計な負荷を懸ける。情報収集の実時間性、転送データ量、配置コスト等のトレードオフを解決するために、観測点の配置やデータ転送方式などの分散管理の研究がなされている。また、perfSONAR* のような多地点計測・計測情報共有のフレームワークも開発されている。

Cooperative Association for Internet Data Analysis (CAIDA)[†] は、1997 年に米国で設立

* <http://www.perfsonar.net/>

† <http://www.caida.org/>

された組織であり、ネットワーク計測技術の開発や普及に貢献してきたが、その Web サイトの Tools Taxonomy ページでは、Topology, Workload, Performance, Routing, Multicast, Measurement Infrastructures という分類で、フリーソフトから商用のハードウェア計測装置まで様々な計測ツールが紹介されている。

(3) 複数の計測データの相関

時間、空間、特性などの異なる視点から「複数の計測データの相関」を解析することで有用な情報が得られる可能性がある。まず、同じ場所 / 特性の時間的に異なる計測の比較（時系列）として、昔から、平滑化、フィルタ、予測の手法が研究されてきた。定常的な変動の中からの本質的な変化の検出（Change detection）も、ネットワークの定常性の把握や異常発生を検出などに重要である。ネットワーク全体にかかわる「異常」（DDoS など）の検知においては、単一の特性 / 場所だけでなく複数の特性 / 場所の同時観測による Change Detection によって精度が高まると期待できる。

同じ時間 / 特性の空間的に異なる計測の比較の代表例として、(狭義の)「ネットワークトモグラフィ」がある¹²⁾。トモグラフィ（断層撮影）とは、例えば、X 線を用いて複数の方向から透視して、切開せずに内部の三次元的な診断を行うことであるが、ネットワークトモグラフィも、この類推で名前が付けられた情報通信ネットワークにおける計測・推定の技術で、広くとらえると、直接観測することが困難なネットワークの局所的または大域的な状態・特性を、観測が容易な複数の（複合的）現象の観測結果から推定する手法と言える。典型的なネットワークトモグラフィの形態としては以下の二つがある。

- ネットワークを横断する複数のパスに沿った通信の挙動を計測して内部の局所的状態やトポロジーを推定する。特に複数のパスでのエンドツーエンド特性を同時に計測し、その相関から複数パスに共有される区間の特性を推定する。
- ネットワーク内部の複数点において通過する通信量を計測してネットワークを横断する大域的な通信フローの流量（単位時間当たりの通過量）を推定する。特に複数の地点での通過流量を同時に計測し、その相関から複数地点を縦断するフローの流量を推定する。

前者では、内部の特定区間でのパケット損失率、遅延分散、遅延増分（絶対遅延から固定遅延を引いたもの、キューイングによる遅延相当）の分布などの量的特性や、損失や遅延の増大が多発しているかどうかの質的状态、始点から複数の終点への経路木の構造などを推定する。後者は、OD（Origin-Destination）トラヒック行列推定問題やトラヒック交流問題とも呼ばれ、どこからどこへ、どのくらい、どのようなトラヒックが流れているかを推定するが、その空間粒度は、POP 間、AS 間、ネットワーク間、ルータ間など様々であり、また対象トラヒックも、全パケットではなく、特定種類のパケットに絞ることも比較的容易である。

一方、同じ時間 / 場所の異なる特性の計測の比較によって、例えば、直接的な計測を行うとネットワークに過大な負荷をかけるスループット特性を計測負荷の軽いほかの特性（パケット損失、パケット遅延、可用帯域幅など）から推定することや、減多に起きない（間欠的な計測では把握が難しい）損失の発生状況をほかの特性計測から推定することが期待できる。

3-4-2 パスに沿った帯域幅と可用帯域幅の間接的推定

パスに沿った各リンク i の単位時間当たりの IP ユーザデータの最大転送可能量 (IP Capacity) をリンク帯域幅 C_i とする。IP パケット長に依存するので、MTU 長の IP パケットを用いた場合の値として C_i を定義する。パス帯域幅 C とは、パスに沿った各リンクの帯域幅の最小値： $C \stackrel{\text{def}}{=} \min_i C_i$ のことである。つまり、パス帯域幅は、ほかのクロストラフィックがない場合に、そのパスに沿ってトラフィックを流して得られる最大 IP スループットといえる。この最小帯域幅を持つリンクを narrow link と呼ぶ。一方、ある単位時間幅 τ を定めて、時間区間 $(t-\tau, t]$ 内でリンク i のキュー長が 0 であった時間の和を $a_i(t)$ と置いて、時刻 t でのリンク i の (τ 平均) 可用帯域幅を、 $A_i(t) \stackrel{\text{def}}{=} C_i \times a_i(t)/\tau$ で定義する。このとき、パス可用帯域幅 A とは、 $A \stackrel{\text{def}}{=} \min_i A_i$ のことである。この最小可用帯域幅を持つリンクを tight link と呼び、narrow link とは異なる場合がある。以下で、パスに沿ったアクティブ計測による、パス帯域幅の推定ツール (e.g., pathrate¹⁾, capprobe⁸⁾) とパス可用帯域幅の推定ツール (e.g., pathload⁷⁾, pathchirp¹⁰⁾) で使われる基本原理 (パケットペア法) を説明する。

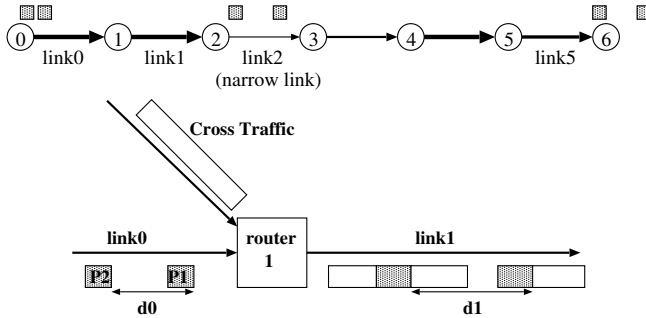


図 3-18 パケットペア法

パケットペア法では、図 3-18 (上) のようなパスを考えたとき、始点ノード 0 から時間間隔 d_0 で連続して長さ L の 2 個のパケット (先頭パケット P_1 の先頭ビットの送出時刻と後方パケット P_2 の先頭ビットの送出時刻の差が d_0 である) を終点ノードに向けて送出する。リンク i 上での間隔を d_i とすると、クロストラフィックの影響を無視できる場合は、 $d_i = \max(d_{i-1}, L/C_i)$ が成り立つので、初期値 d_0 を十分小さく取ると、最終リンク上での間隔は、

$$d_n = \max_i \left(\frac{L}{C_i} \right) = \frac{L}{\min_i C_i} = \frac{L}{C} \quad (3-4)$$

となり、終点ノードで受信間隔 d_n を観測すれば、パス帯域幅を $C = L/d_n$ で推定できる。

現実にはクロストラフィックが混在する。図 3-18 (下) は、ノード (ルータ) 1 でクロストラフィックが挿入される様子である。 P_1 が着いた後、 P_2 が着くまでの期間にリンク 1 に流入するクロストラフィックの量を X と置くと、もし、 d_0 が $(L+X)/C_1$ に比べて小さくて、その期間に一度もリンク 1 への出力キューが空にならなかった場合は、 $d_1 = (L+X)/C_1$ が成り立つ。リンク 1 の可用帯域幅を A_1 と置くと、 $E[X] \approx (C_1 - A_1)d_0$ という近似から、

$$E[d_1] \approx \frac{(C_1 - A_1)d_0}{C_1} + \frac{L}{C_1} \geq \frac{L}{C_1} \quad (3.5)$$

d_0 が $(L + X)/C_1$ に比べて大きく、また、クロストラヒックが定常的な場合は、

$$E[d_1] \approx d_0 \quad (3.6)$$

よって、一般に式 (3.4) が成り立たないので、終点で観測されたパケット間隔 d_n からパスの帯域幅を推定するためには、多数回のパケットペアの観測を行って、クロストラヒックの影響を統計的に除去する必要がある。

ところが逆に、これをパスの可用帯域幅の推定に利用する立場からは、観測データからこのクロストラヒックの影響を最大限抽出したい。図 3-18 (下) での入力パケット間隔 d_0 の値を縮めていくと (つまりノード 0 での送出レートを増加させると)、出入の間隔比 d_1/d_0 は、式 (3.5)、(3.6) より、はじめは変化ないが、途中から増大する。ここで、変化する点での d_0 を d^* と置くと、 $d^* = d_0 \approx d_1$ 、かつ、式 (3.5) が成り立つ臨界点であり、

$$d^* \approx \frac{(C_1 - A_1)d^*}{C_1} + \frac{L}{C_1}$$

となる。この式を解けば、 $A_1 \approx L/d^*$ 。これは多段の場合に拡張できて、始点でのパケット送出間隔 d_0 を調整しながら、終点でのパケット到着間隔 d_n を観測し、上記の臨界点の値 d^* を見つけることで、 $A = \min_i A_i \approx L/d^*$ が推定される。また、式 (3.5) より、 d_0 が小さい領域での $1/d_0$ に対する d_1/d_0 の増大はリニアであり、その傾き L/C_1 とパケット長 L とから、リンク 1 の帯域幅 C_1 も推定できる。

3-4-3 高速ネットワークでのトラヒックフローの計測

様々な粒度のフロー単位の情報はネットワーク管理に極めて有用であり、現実のネットワーク機器にも、NetFlow や sFlow などのルータベンダー主導のフロー情報計測 (サンプリングベース) の導入が始まっている。

高速ネットワークにおいては、膨大な量のパケットが通過するため、パッシブ計測によって単純に全パケットを収集、保持、あるいは他地点へ転送してフロー情報を取得・解析することは困難である。特に、細粒度フローの計測では、フロー数は膨大かつ事前には分からないので、例えば、10Gbps の高速回線速度で流れるパケットを収集してフローを識別する所まではできても、そのフローごとの情報をすべて高速アクセスメモリ上に保持できない。そこで、保持する必要のあるデータ量を減らすために、フィルタリング、集約⁽¹¹⁾、高度なサンプリング⁽³⁾ や情報圧縮⁽⁹⁾ などに関する手法が開発されてきた。

また、ほかに比べて流量が大きい “heavy hitter” フローを対象として、カウンタ付きハッシュ表⁽⁴⁾ やスライディングウィンドウ⁽⁵⁾ などを使って少ないメモリで情報を保持する手法も研究されている。この背景には、インターネットでは、少数の巨大なフロー (エレファントフロー) がトラヒック量的には全体の多くを占めるという性質 (フローサイズ分布の heavy tail 性) があり、その重要性は高い。

更に、複数点での観測をパケット単位で付き合わせて縦断するフローを解析 (追跡) するために、Trajectory Sampling⁽²⁾ のように各観測点で同一のハッシュ関数を使ってサンプリン

グすることで効率的に「同じ」パケットを複数点で観測できる手法も開発されている。

参考文献

- 1) C. Dovrolis, P. Ramanathan, and D. Moore, "Packet dispersion techniques and capacity estimation.," *IEEE/ACM Trans. Networking*, vol.9, July 2001.
- 2) N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation.," *IEEE/ACM Trans. Networking*, vol.9, no.3, pp.280-292, June 2001.
- 3) N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics.," In *Proc. ACM SIGCOMM*, pp.325-336, Karlsruhe, Germany, Aug. 2003.
- 4) C. Estan and G. Varghese, "New directions in traffic measurement and accounting.," In *Proc. ACM SIGCOMM*, pp.323-336, Pittsburg, Aug. 2002.
- 5) L. Golab, D. DeHaan, E. Demaine, A. Lopez-Ortiz, and J. I. Munro, "Identifying frequent items in sliding windows over on-line packet streams.," In *Proc. ACM SIGCOMM Internet Measurement Conference*, Miami, USA, Oct. 2003.
- 6) G. Iannaccone, C. Diot, I. Graham, and N. McKeown, "Monitoring very high speed links.," In *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, Nov. 2001.
- 7) M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput.," In *Proc. ACM SIGCOMM*, Pittsburg, Aug. 2002.
- 8) R. Kapoor, L. Chen, M. Gerla, and M. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique.," In *Proc. ACM SIGCOMM*, Portland, Aug. 2004.
- 9) A. Kumar, J. Xu, J. Wang, O. Spatscheck, and L. Li, "Space-code bloom filter for efficient per-flow traffic measurement.," In *Proc. IEEE infocom*, Hong Kong, Mar. 2004.
- 10) V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "Pathchirp: Efficient available bandwidth estimation for network paths.," In *Proc. PAM workshop*, La Jolla, California, Apr. 2003.
- 11) L. Yuan, C. N. Chuah, and P. Mohapatra, "Progme: towards programmable network measurement.," In *Proc. ACM SIGCOMM*, pp.97-108, Kyoto, Japan, Aug. 2007.
- 12) 鶴 正人, "ネットワークトモグラフィ技術の研究動向.," *オペレーションズ・リサーチ*, vol.53, no.6, pp.321-327, June 2008.