インターネットのトラフィック制御
— QoS の仕組みと技術課題 —

長 健二朗
(株) ソニーコンピュータサイエンス研究所

要 旨

QoS の要求を実現するために，従来のインターネットの配送モデルは大きく変化しようとしている．これからのインターネットでは，キューイングを中心とするコンポーネントを組み合わせて，帯域割り当て，遅延制御，輻輳回避を実現する．本稿ではインターネットのトラフィック制御の仕組みをキューイング技術を中心に解説する．

1　インターネットの配送系

インターネットはベストエフォート (best-effort) 方式のパケット配送を基本としている．ユーザからみた有効帯域や遅延は回線の混み方によって変化し，混雑がひどくなるとパケットは途中で捨てられてしまう．パケットの損失を検出し再送によって通信の信頼性を確保するのはエンドホスト間の役目である．

パケットは通信回線を繋いでパケット交換を行なうルータが配送する．ルータの主な仕事は経路管理とパケット配送である．ルータは，隣接するルータと経路情報を交換し，パケットの宛先から次の中継点を決定する経路表を管理する．ルータはこの経路表を引くことによって，受信パケットを次の中継点に配送する．パケットは通信回線の速度でしか送り出せないので，すでに送出中のパケットがある場合，次のパケットはルータの送出バッファのキューに並べられ，バッファが溢れるとパケットは廃棄される．インターネットにおけるパケット損失の主な原因はこのキュー溢れである．

インターネットの成功の大きな要因は，パケット配送系に信頼性を求めない簡単な仕組みを採用して，ネットワークを容易に拡大したり相互接続できるようにしたことである．しかしながら，インターネットの普及に伴って，音声や画像のリアルタイム通信や料金別の通信品質の差別化など配送系への QoS(Quality of Service) の要求が高まり，従来の簡単な配送モデルが大きく変わろうとしている．

図 1 はインターネットの配送系を，通信回線と経路選択とキューでモデル化したものである．配送系を改善するには，通信回線の容量アップや新規敷設，経路の工夫，キューイングの工夫の 3 つの選択がある．本稿ではパケット配送の差別化を実現するキューイングに焦点をあてる．

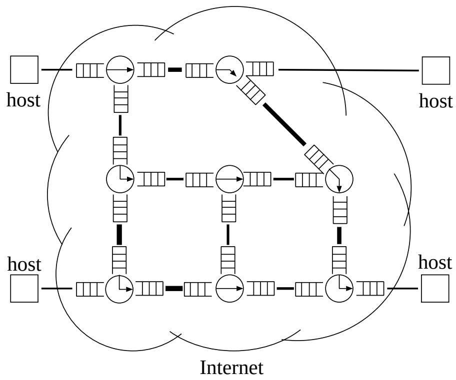

図 1: インターネットの配送系モデル

2　キューイングの仕組み

もっとも一般的で簡単なキュー構造は FIFO(First-In First-Out) キューである．パケットは到着順に送出され，キュー長が制限値を越えると到着パケットが廃棄される．ところが，キュー構造，パケットスケジューリング，パケット廃棄方法を工夫することによって，帯域割り当て，遅延制御，輻輳回避などが可能となる．これらの仕組みは総称してキューイングと呼ばれる．

図 2 は一般的なキューイングの構成である．キューイングは出力インターフェイスで行なわれ，キュー構造，クラシファイア，ドロッパ，スケジューラなどから構成される．キュー構造は通常複数の内部キューの組み合わせで構成される．クラシファイアは，パケットヘッダ内の情報をもとに到着パケットを対応する内部キューにマップする．パケットヘッダ情報には，送信アドレス，宛先アドレス，送信ポート番号，宛先ポート番号，プロトコル識別子などがある．共通属性を持つ一連のパケットはフローと呼ばれる．ドロッパは選択的にパケットを

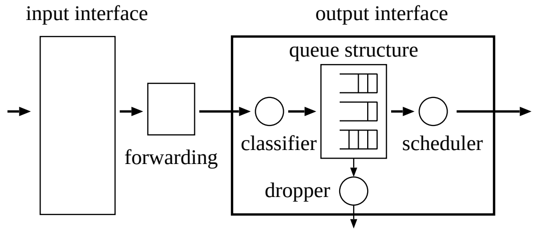

図 2: 一般的なキューイングの構成

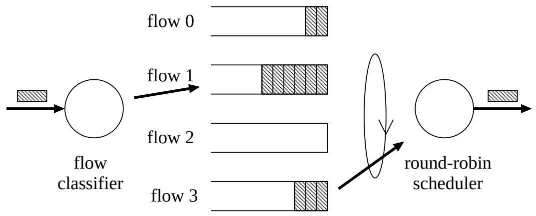

図 3: フェアキュー

廃棄する仕組みであり，スケジューラは次にパケットを送出すべき内部キューを選ぶ．

キューイング機構は処理能力の低いルータへの実装や高速ルータでのハードウエア化を考慮した簡単な構造が望ましいが，その設計は処理の負荷，メモリ消費，制御の精度等のトレードオフとなる．また，キュー操作のオーバヘッドは，パケットのフォワーディング性能に直接影響するが，その程度は回線速度やルータの処理能力に依存する．さらに，IP パケットは可変長なので，ATM 等の固定長方式に比較して制御構造が複雑になってしまう．このように設計上のさまざまなドレードオフが存在し，現在までに多くのキューイング方式が提案されてきている．

2.1 帯域割り当て

キューイングの第一の役割は帯域割り当てである．ここでは，フェアキューイング (Fair Queueing) を例にあげて動作を説明する．フェアキューイングは図 3 に示すように，フロー毎に独立したキューを割り当て，ラウンドロビンで各キューからパケットを送出する方式である．簡単のためパケットサイズが一定と仮定すると，フローが N 個ある場合，各フローは全帯域の 1/N の割り当てが保証され，使われていない帯域は他のフローに公平に分配される．帯域の割り当てによって，他のトラフィックの影響を遮蔽することもできる．

パケットサイズを考慮する場合は，各ラウンドでパケットを持つ内部キューに一定のトークンを与え，パケットサイズ分のトークンが溜ったキューをスケジュールすることで実現できる．また，与えるトークンの量を変えることによって，フロー毎に割り当て帯域の重み付けが可能な WFQ(Weighted Fair Queueing) になる．

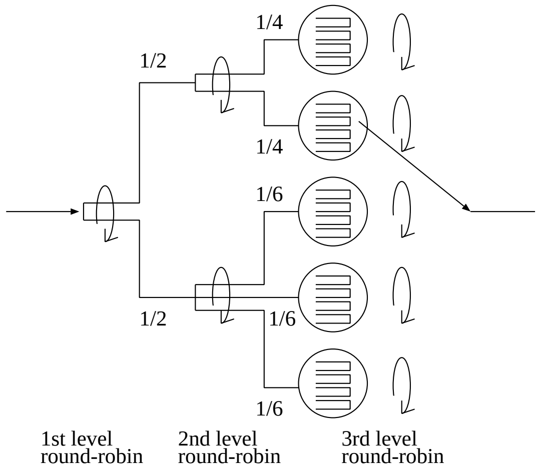

図 4: 階層化リンク共有

フェアキューイングの実装上の問題はフローの数だけ内部キューが必要になることである．フローの数を予測することは困難であり，また，バックボーンでは膨大な数のフローが存在する可能性もある．実際には，固定数のキューに対して，フローをハッシュ関数を使ってマッピングするなどの近似手法がとられる．

また，スケジューラを階層的に組み合わせることによって，帯域を組織構成等に従って柔軟に分配することもできる．図 4 にフェアキューイングのラウンドロビン・スケジューラを 3 段階のツリー構造に組み合わせた構成を示す．例えば，一段目で組織別に帯域を割り当て，二段目で部門別に，三段目でユーザごとに割り当てる構成をとると，ある部門にユーザが増えても他の部門の割り当てに影響を与えない運用ができる．

2.2 遅延制御

キューイングの第二の役割は遅延の制御である．遅延の制御は帯域制御より難しい．帯域はその定義から常に平均値で表されるのに対して，遅延は一般に最大値で表されるためワーストケースの計算が必要になる．また，帯域保証は中間ルータの数が増えても変わらないが，遅延保証は各中間ルータでの遅延を加算することになる．

ルータで発生する遅延には，ストア・アンド・フォワード遅延，ルーティング遅延とキューイング遅延がある．ほとんどのルータは，パケット全体を受信してからフォワード操作をするストア・アンド・フォワード方式なので，（パケットサイズ / 回線速度）で表されるストア・アンド・フォワード遅延が生じる．ルーティング遅延は，受信パケットを経路表を引いて送信バッファに送

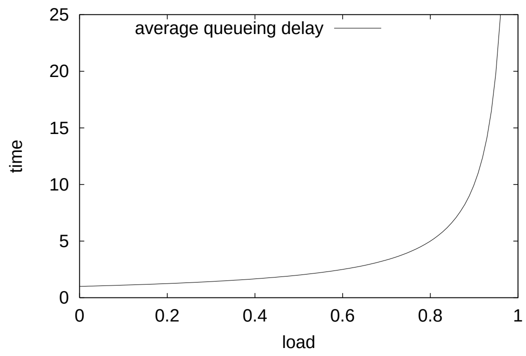

図 5: 回線使用率とキューイング遅延

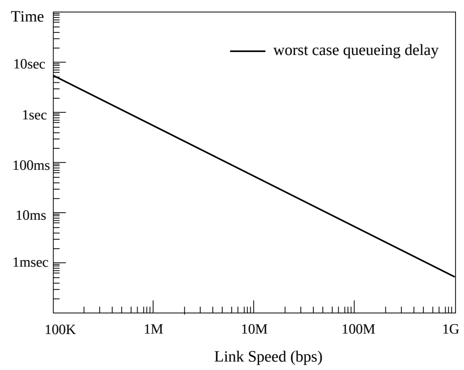

図 6: 回線速度とキューイング遅延の関係

るのに要する処理時間で，経路表の大きさとルータの処理能力に依存する．これに対して，キューイング遅延は，他のパケットによって出力キューで待たされる時間で，この値は回線が混んでくると大きくなる．

ルーティング遅延は経路表が大きい場合に問題となるが，その値がストア・アンド・フォワード遅延より小さければ，パイプライン効果でひとつ前のパケットの送出時間に隠れてしまう．

キューイング遅延は，一時的に回線容量を越える量のパケットが集まって発生する．速い回線から遅い回線への接続点や，複数の回線から特定の回線へパケットが集中するボトルネックの入口で起こる．図 5 はパケットがポアソン分布にしたがった間隔で到着すると仮定して，平均回線使用率に対するキューイング遅延の期待値を表したものである．パケットが等間隔に到着する場合はキューは常に空であるが，到着間隔にばらつきがある場合は，平均回線使用率の増加に伴い徐々にパケットの到着がオーバーラップする確率が増えていき，ある値を越えると急速に遅延が増大する．これはキューが到着間隔のゆらぎを吸収できなくなるためで，交通渋滞の発生と同様な現象である．

音声通信などのアプリケーションの遅延要求は一定であるが，キューイング遅延は回線速度に反比例するので，回線速度が遅くなると遅延の制御が必要になる．図 6 は回線速度とキューイング遅延の関係を示す．ここでは，最大キューイング遅延として，1500 バイトのパケットが 50 個キューにある場合のキューイング遅延をプロットしている．例えば，500msec の遅延保証要求がある場合，100Mbps の回線でのキューイング遅延は 6msec にしかならないので遅延制御しても効果が少ないが，1Mbps の回線では 600msec の遅延になるので遅延制御が必要となる．

最も簡単な優先制御方式はプライオリティキューである．プライオリティキューは優先度の異なる複数のキューを持ち，スケジューラは常に優先度の高いキューからパケットを処理する．低優先度のトラフィック量にかかわらず，高優先度のパケットが送出されるため，高優先度パケットのキューイング遅延が保証できる．

高優先度パケットのキューイング遅延は，低優先度パケットの送出中に高優先度パケットが到着して送出終了待ちが起こるために発生する．最大値は最大パケット長の低優先度パケットを送り始めた直後に高優先度パケットが到着した場合である．

また，高優先度パケットの送信者は，自身の高優先度パケットが複数重なってキューイング遅延を生じないように，間隔をあけて送信する必要がある．送信間隔を一定に保つ機構をシェーピング (shaping) と呼ぶ．ネットワーク側では，規定どおりのシェーピングがなされていることをネットワークの入口で監視して，規定以下の間隔で到着するパケットを廃棄することで過剰な流入を防止できる．この仕組みをポリシング (policing) と呼ぶ．

たとえ高優先度パケットの送信がシェーピングされていても，高優先度パケットの送信者が複数存在する場合には，高優先度パケットの競合でキューイング遅延を生じる．その最大遅延は，他のすべての送信者が最大パケット長の優先パケットを送った直後に自分が優先パケットを送る場合である．

また，フェアキューイングの場合も同様にシェーピングされたフローのキューイング遅延の最大値が計算できる．キューイング遅延が最大になるのは，自分のラウンドが終った直後にパケットが到着して，他のすべてのキューから最大長のパケットが送出される場合である．

実際の機器では，キューの先にさらにハードウエアのバッファがあるので，その分のキューイング遅延も計算する必要がある．また，キューイング遅延以外の遅延計算は，パケットあたりのオーバーヘッドは一定とみなせると仮定して，定数を使って近似することが多い．

ここで述べた遅延計算はあくまで決定論的手法を使っ

たワーストケースの値で，実測される遅延はこれよりずっと小さい値になる．逆にいうと計算で求められる遅延の最大値は非常に大きな値になってしまう．そこで，統計的手法を用いて，エラー率何％以下で遅延を保証する統計的遅延保証を使うこともある．例えば，図5のキューイング遅延の期待値からわかるように，プライオリティキューの高優先度パケット同士の競合による遅延は，高優先度パケットが全体に占める割合が少なければ十分に小さいと予想できる．一般に統計的遅延保証を使えば決定論的遅延保証より大幅に小さい値を得られるが，統計解析には仮定条件があるので注意が必要である．例えば，パケット到着間隔がポアンソン分布すると仮定して計算された統計的遅延保証は，バースト的なトラフィックには適用できない．

2.3 輻輳回避

キューイングの第三の役割は，TCP等の上位層と連動して，輻輳回避を行なうことである．トラフィック制御はキューイングだけでなく，エンドホスト間のフロー制御，容量設計，課金方法などの要素が含まれる．これらは異なる時間粒度で機能し，互いに補間関係にある．

キューイングの時間粒度はパケット送出時間で，この時間粒度で起こるバースト的な集中を制御するのに効果がある．一方，エンドホスト間のフロー制御はより大きな時間粒度でフローの転送レートを調整する．フロー制御の重要な役割は，パケットのバーストサイズを小さく保って，キューイングが制御できる範囲に収めることである．逆にキューイングはフロー制御に適切に輻輳情報を伝える役目がある．つまり，トラフィックの制御はキューイングだけでできるものではないし，単に回線容量を増やすだけで解決できるものでもない．今後回線容量の格差が拡大するにしたがって，フロー制御とキューイングの連動がますます重要になってくる．

現状のインターネットでは，トランスポート層に輻輳を知らせるには，パケットを廃棄するしかない．TCPはパケット損失を輻輳情報として捉えてパケット送出量を調整する．単なるFIFOキューではキュー長が制限値に達するまでパケット廃棄は起こらないが，一度制限値に達すると続けてパケットが廃棄される．このため，すべてのTCPが同時にパケット損失とスロースタートを繰り返す同期現象が起こり，スループットが振動する可能性がある．RED(Random Early Detection)[3]は，平均キュー長に応じた確率でパケットを廃棄する仕組みで，キュー長が限界に達する前にTCPに輻輳の始まりを伝えることによって同期現象を回避し，かつ，平均キュー長を短く保つ効果がある．

3 キューイングの応用

将来，大容量回線が容易に利用できるようになれば，輻輳は起こらなくなるので，QoS制御の必要はなくなるという意見もある．しかし，QoS制御は有無の二者択一ではなく，どの程度余裕を見込みどの程度制御するかというバランスの設計である．確かに，回線容量がトラフィックのピーク値を十分上回る程余裕があれば制御の必要はない．その一方で，余裕が全くない状態で多くの要素が複雑に絡む制御を行なうのは非常に難しい．したがって，実際にはある程度余裕をもった資源配分(provisioning)のもとで，ある程度の制御を行なうわけで，コスト効率，運用の容易性，将来の拡張性等を考慮したバランスポイントを見つけることが重要となる．

ボトルネックの入口にキュー管理を導入することは，系の中で資源余裕の足りない部分に制御の割合を増やすことで系全体のバランスをとる一つの手段と見ることもできる．キュー管理によって見かけ上の輻輳開始点がずれるので，回線容量を増やすのと同様な効果がある．一方，回線使用率に適正な余裕を持つネットワークでは，キュー管理は輻輳時や設定ミスに備えた防備と見ることもできる．このようなネットワークでは，通常はキュー制御の効果はほとんど分からないはずである．

キューイングの役割をネットワーク全体で設計することも重要である．例えば，単純なプライオリティキューでは，高優先度のトラフィックが増えると通常のトラフィックがサービスを受けられなくなる．しかし，ネットワークの入口で適正なポリシングをして優先パケットの流入量を調整すれば，単純なプライオリティキューを採用しても問題は発生しない．その一方で，一箇所のポリシングの設定ミスがネットワーク全体に被害を与えるという運用上のリスクも考慮することが必要である．

3.1 IntServとDiffServ

エンドホスト間の通信が中間ルータでの優先制御を受けるためには，通信に対して優先制御を指定する仕組みが必要である．インテグレーテッド・サービス（以下IntServ）とデファレンシエーテッド・サービス（以下DiffServ）はインターネットにおいて優先制御を指定する枠組である．いずれの場合もトラフィック制御部にはキューイングによる優先制御が用いられる．

IntServは，パケット交換網に優先制御を導入して，音声や画像等のリアルタイム通信を統合する目的で1994年からIETF(Internet Engineering Task Force)での標準化が始まった．IntServではエンドホスト間でQoSパラメータを指定して資源予約をして通信するモデルがとられている．中間ルータでは，指定されたQoSパラメータに従って，予約の受付制御(Admission Control)

を行ない，動的に資源予約を実行する．IntServ では QoS パラメータとして測定可能な通信品質を定義し，資源予約プロトコル RSVP[6] が開発された．

RSVP の開発により QoS 実現の技術的理解は大きく前進したが，現状のインターネットとのギャップは大きく普及は進んでいない．RSVP の問題点として，仕組みが複雑で実装が困難である，中間ルータでフロー毎の状態の保持が必要でスケーラビリティに欠ける，プロバイダにとってコスト効率が悪くビジネスとして導入が難しい等が上げられている．

DiffServ は，IntServ の経験と反省をもとに，より簡単で柔軟な QoS の実現手段として提案され，1997 年より標準化が進行している．DiffServ ではネットワークの周辺部のエッジルータと内部のコアルータを分離している．エッジルータは個別フロー毎の複雑な処理を行ない，パケットを少数のクラスに分類してクラス識別子をヘッダに書き込む．コアルータはこのクラス識別子のみを参照し，クラス別に集約されたフローに対しクラス別に定義された制御を行なう．

DiffServ はあくまでもクラス別のサービスを提供する枠組である．どのようなコンポーネントをどのように組み合わせてどのようなクラス別サービスを実現するかは，原則としてプロバイダの裁量に任される．

IntServ と DiffServ の考え方の違いは，前述した制御と資源配分の余裕設計のバランスのとらえ方である．IntServ は制御を中心に考えられており，資源余裕がなくても正しく動作するよう設計されている．これに対して，DiffServ は適正な資源余裕が準備されるという前提で制御機構を考えようとしている．よって機器構造が簡単になる反面，適正な運用に依存する仕組みとなる．

3.2 ALTQ

インターネットの技術開発には実証研究が重要な意味を持つ．しかし，研究ネットワークの構成要素も商用ネットワーク機器への依存度が高まり，製品化されないと実証研究が進まない状況が出てきている．筆者は研究利用できるキューイング実装の必要性を感じ，1997 年より PC UNIX ベースで動作する ALTQ (Alternate Queueing)[1] というプラットフォームの開発を行なってきている．ALTQ はさまざまなキューイング方式を実装するためのフレームワークで，現在 6 種類のキューイング方式が実装され，RSVP や DiffServ にも対応している．ALTQ は世界中の研究機関で利用されるようになってきており，米国の Internet2 や NLANR，日本の JGN といった次世代インターネット実証実験にも使われている．

4 今後の課題

インターネットにおける優先制御は，まだ始まったばかりで多くの課題を抱えている．理論的には，優先制御のために必要な個別のコンポーネントやその挙動に関してはある程度研究が進んできているが，これらが組み合わされてできる系としてのネットワークをとらえることはできていない．運用面では，トラフィックのモニタリング技術，障害対応技術などの進歩が，管理者の運用技術のレベルアップと共に欠かせない．

IntServ の個別フロー，なかでも固定レートの通信に対するキューイングの挙動やサービスの特性に関してはある程度理解されているが，その反面，実現が容易でないことも分かってきている．いっぽう，DiffServ は比較的容易に実現できると考えられているが，集約フローや資源余裕の理論モデルが確立されていないので解析的なアプローチは難しく，実証的なアプローチによる試行錯誤が必要になっている．DiffServ はようやくフレームワーク部分の標準化が一段落して，これから実証実験が始まる段階でしかない．また，異なる DiffServ ドメイン間の調整，受信者によるクラス指定，DiffServ をコアにした RSVP の統合，ラベルスイッチとの連動なども今後の課題である．

なお，キューイングや QoS 技術について興味を持つ読者は文献 [2][4][5] などを参照されたい．

参考文献

[1] K. Cho. A Framework for Alternate Queueing. In *Proceedings of USENIX 1998*, pp.247–258, June 1998.

[2] P．Ferguson and G．Huston. *Quality of Service.* Wiley，1998.

[3] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transaction on Networking*, 1(4):397–413, August 1993.

[4] S．Keshav. *An Engineering Approach to Computer Networking.* Addison-Wesley，1997.

[5] C．Partridge. *Gigabit Networking.* Addison-Wesley，1994.

[6] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A new resource reservation protocol. *IEEE Network*, 7:8–18, September 1993.