インターネットにおける通信品質とその制御

村田　正幸

1. はじめに

インターネット上で展開されるアプリケーションやサービスの発展，それに伴う電話サービスを中心とする電気通信網の衰退は，ネットワークが提供すべき通信品質（Quality of Service: QoS）の考え方にも大きな影響をもたらしつつある．本稿では，まず，ネットワークがQoSを提供する枠組みとして，IntServ，DiffServアーキテクチャについて簡単に述べる（節2，3）．次に，回線交換原理に基づくMPLSネットワークとその上でトラヒックフローを制御するトラヒック工学について述べる（節4）．以上は，ネットワーク側が積極的に品質制御を行う例である．一方で，ネットワークが主体となって行う品質制御は最小限にとどめ，伝送技術の進展にあわせてネットワーク資源（回線，ルータ）を増強していけばよいという考え方もある．これは，これまでのインターネットの考え方そのものである（節5）．その場合には，現在のネットワークの利用状況を同定し，必要な箇所に資源を投下するための品質測定技術が重要な研究課題になる．節6ではそのような例を述べる．また，トラヒック制御技術の観点から，特に本学会の読者に興味を持っていただけるような最近の話題を紹介する．

なお，技術的内容については多くの論文や解説もあるので，引用を適宜行うこととし，本稿では特に，これらの技術が必要とされた背景と現状，問題点などを中心に，筆者なりの考えもあわせて述べていきたい．

2. QoS保証の実現：IntServ

近年，研究開発が活発に行われたATM（Asynchronous Transfer Mode：非同期通信モード）通信方式[1]は，当初標榜されていたような究極の通信方式としての普及には至らなかった．しかし，そこでのQoS制御の考え方は，その後のインターネットアーキテクチャにも大きな影響を与えた．すなわち，IntServ（Integrated Services）アーキテクチャ[2]の登場である．IntServの主たる目的は，従来の電気通信網で提供されてきたQoS保証のしくみをいかにインターネットにおいて実現するかであった．

もともとインターネットにおいては，ネットワーク層プロトコルであるIPが経路到達性（reachability）を保証し，エンドホスト間のトランスポート層プロトコル（TCP）による再送制御によって初めてパケットを失うことなく送受信できる．ネットワーク内，すなわち，IPにおいてはQoSを保証するという考え方はなく，IPがベストエフォートサービスであるといわれるゆえんでもある．一方，IntServでは，ネットワークのマルチメディア化，すなわち，従来のデータ系サービスに加えて，実時間系サービス（音声，動画像）をも提供することを企図した．品質の良い実時間系サービスを提供しようとすれば，QoS保証の考え方が必要になる．ただし，ここでいうQoS保証とは，例えば，エンドホスト間に設定されたコネクションに対して，64 kbpsの帯域や50 msec以内のエンド間遅延を絶対的に保証することを意味する．実時間系音声会話を従来の電話網における電話サービスと同じように実現しようとすれば，ネットワーク内においてコネクションごとに資源を確保する必要がある[1]．すなわち，エンド間コネクションが経由する経路上の各回線において一定の帯域を確保しなければならない．いったん帯域が確保されれば，エンド間遅延の保証も可能になる[3]．

エンドホスト間コネクションに対してネットワーク資源を確保するためには，以下の二つが考えられる．

(1) 本節で述べるIntServアーキテクチャのようにIP自体を拡張する

(2) 節4で述べるMPLSのように下部ネットワークがその機構を提供する

むらた　まさゆき
大阪大学 サイバーメディアセンター
〒560-0043 豊中市待兼山町1-32

2.1 IntServの3要素

エンド間コネクションに対するQoS保証の本質は，回線交換原理のエミュレーションである．すなわち(図1参照)，

(1) 通信を開始する前にまずシグナリングプロトコルによってコネクション設定を行い，エンドホスト間の経路上のすべての回線において必要な帯域を確保する．IntServにおいてはRSVP (Resource Reservation Protocol)[4]が規定されている．RSVPは，コネクションの帯域要求情報を運ぶことによって各ルータに対して帯域確保を促す．

(2) 各ルータは，RSVP制御情報に従ってコネクションのための帯域を確保する．各コネクションの必要帯域が異なる場合，それぞれの帯域による重み付けも考えておく必要がある．このようなスケジューリングを行う方式としてWRR (Weighted Round Robin) 方式がある．また，絶対的な遅延時間を保証するWFQ (Weighted Fair Queueing) もある[5]．

(3) ネットワークに帯域を確保してもらった限りは，コネクションはその帯域を超えてネットワークにデータを流入させてはならない．エンドホストにおけるリーキーバケット方式 (leaky bucket algorithm)[5]がその役割を担う．

2.2 IntServの問題点

IntServアーキテクチャは，回線交換のしくみをインターネットに持ち込もうとしたものであった．しかし，いくつかの重要な問題点が指摘され，実用には至っていない．代表的な問題はスケーラビリティに関する問題である．インターネットにおいてスケーラビリティは重要な概念であり，将来的な拡張性を常に確保しておく必要がある．

(1) コネクション数に関するスケーラビリティ

各ルータでWFQなどのパケットスケジューリング方式を動作させるには，送受信IPアドレス，ポート番号の組による各コネクションの識別，および，コネクションごとのパケット処理が必要となる．

(2) ホップ数に関する問題

RSVPを用いるには，エンドホスト間の経路上のすべてのルータがRSVPシグナリングを処理できなければならない．ネットワーク規模が大きくなればなるほど，その処理オーバーヘッドは増大する．

(3) ディプロイメント (deployment) に関する問題

エンドホスト間の経路上のすべてのルータがIntServアーキテクチャを提供していないと，コネクションに対するQoS保証が実現できず，アーキテクチャ自体が破綻する．このような仮定を，運用されているネットワークに対して置くことの問題は明らかである．

また，IntServアーキテクチャを実現するためにはネットワーク資源の管理が必要になる．すなわち，各ルータにおいてアクティブなコネクションに関する情報を維持する必要がある．これは，いったんルータに故障が発生した場合の障害回復手続きが煩雑なものになることを意味する．RSVPにおいては，電話網における帯域確保の方法とは異なり，ソフトステート (soft state) による帯域確保を行っている．すなわち，定期間隔時間ごとに帯域確保の制御メッセージを流し，制御メッセージがない場合にはルータは帯域を解放することによって，システムの耐故障性の向上に注意が払われている．しかし，その場合でも，ネットワーク内にステート情報は残る．そもそもインターネットが発展してきた理由には，その分散志向，単純な構成と制御，その結果としての低コスト化指向にあり，IntServのようなアーキテクチャをインターネットに導入すること自体に矛盾がある．

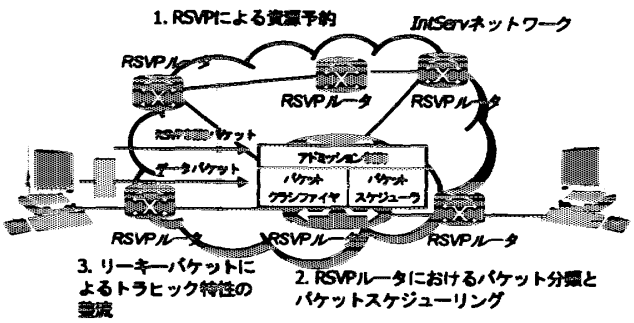

図1 IntServアーキテクチャ

3. QoSの差別化：DiffServ

IntServアーキテクチャにおける問題点の克服を試みたのがDiffServ (Differentiated Services) である[6]．DiffServでは，IntServのようにQoSを保証するという考え方はあきらめ，QoSの差別化が意図された．例えば，パケット単位の優先権制御をルータに導入すれば，高い優先権を与えられたパケットは，低い優先権を与えられたパケットより，確実に速く転送される．

DiffServ アーキテクチャの標準化が急がれた背景には，インターネットの商用サービス化によるところも大きい．すなわち，インターネット接続サービス提供者（ISP）は，ベストエフォートネットワークに対する単なる接続サービスだけでなく，そのQoSに対して他のISP業者との差別化を図る必要があった．そのため，実現困難なIntServよりも，QoSに関して不十分であってもより実現が容易なDiffServに期待が集まった．

3.1 DiffServの概要

DiffServにおいては，まず，コネクション数に関するスケーラビリティの問題に対応するため，コネクションの集合に対してクラスという概念を導入し，クラスごとにQoSが異なるような制御を施す．そのために，IPパケットのヘッダ部のTOS（Type of Service）フィールドをDSCP（Differentiated Services CodePoint）フィールドと読み替え，DSCPフィールドの値によって異なるQoSサービスを適用する．具体的には，ネットワークの入口に位置するルータ（エッジルータ）がDSCPの値をクラスごとに設定し，ネットワーク内部の各ルータはクラスごとに異なるサービスを施すようなパケットスケジューリングを行う．なお，TOSフィールドはIPv4でもともと規定されていたが，実際には用いられていなかったフィールドである．IPv6の場合には，トラヒッククラスフィールドを読み替える．

エッジルータがDSCPをどのように設定するかはネットワーク管理ポリシーによって決定される．すなわち，シグナリングプロトコルは不要で，ISPがあらかじめ決定しておくものである．ネットワーク内ルータはそれぞれ単独にパケットスケジューリングを動作させればよいので，コネクションごとのネットワーク資源の管理が不要になる．各ルータでは個別フローに関する情報を維持する必要はない．その代わりに，個別パケットがその処理に関する情報をDSCPフィールドによって運んでくる．それぞれのルータはその情報に従ってパケットスケジューリングを行う．これをPHB（Per Hop Behavior）と呼ぶ．その結果，エンドホスト間の経路上の少なくとも一つのルータにこの機構を導入すれば，それだけ分の品質の差別化は可能になる．

DiffServの特徴をまとめると以下のようになる．

(1) シグナリングプロトコルは不要で，各ルータはコネクションごとの情報を維持する必要はない．したがって，IntServと比較してスケーラビリティを有する．

(2) ルータは，DSCPフィールドに従ってPHBを適用する．そのため，IPアドレスの識別以外にDSCPフィールドに関する処理も必要になるが，DSCPフィールドはIPヘッダ内にあるので高速処理も可能である．

DiffServサービスクラスとして，以下の三つが規定されている．なお，それぞれのサービスクラスに対するルータのPHB処理は，それぞれ，EF（Expedited Forward），AF（Assured Forward），DF（Default Forward）と呼ばれる．

(1) プレミアムサービス（Premium Service）：固定帯域を優先して割り当てる．例えば，10 Mbpsの回線に対して，あるクラスに10 Mbpsの固定帯域を割り当てるためには，そのクラスのパケットに対してのみプレミアムサービスを適用すればよい．

(2) タイアードサービス（Tiered Service）：ベストエフォートサービスより「まし」なサービスを提供する．3ビットの情報（Drop Precedence）によりパケットを棄却する時の優先度を決定する．

(3) ベストエフォートサービス（best-effort service）：残りの資源を用いて，パケットフォワーディングを行う．

3.2 DiffServの問題点

ルータにおいてPHBをいかに実現するかは標準では規定されていないが，上記サービスを実現するために各ルータに具体的なパケットスケジューリング方式を実装する必要がある．簡単な例としては，HOL（Head of Line）優先権処理が挙げられる．バッファ内にクラス1のパケットが蓄積されていれば，必ずそのパケットからサービスする，クラス1のパケットがない時のみクラス2のパケットをサービスするというものである．このしくみを利用すれば，クラス1とクラス2の差別化が可能になる．ただし，これはあくまでネットワーク側の論理である．クラス1に属するユーザが経験する品質がクラス2のそれより良いということが，果たしてクラス1のユーザにわかるだろうか？　高いクラスのユーザは，低いクラスのユーザより良いサービスを受けているということがわからなければ納得しないだろう．これがDiffServにおける根本的な問題である．優先権制御については，これまでも多くのネットワークにおいて提案がなされてきた．例えば，LANのアクセス制御（CSMA/CD方式，ト

ークンリング方式）などである．しかし，それらが必ずしも成功していないことが，このような制御の限界を物語っている．

先に述べたWRR方式をクラス単位に適用すれば，クラスごとに帯域を割り当てることも可能になる．すなわち，コネクションごとの品質を保証するしくみは困難であるが，クラスに対して帯域を確保することは可能である．そのためには，DiffServにおいてどのようなポリシーに従ってネットワークを運用していくかを管理する必要がある．DiffServにおいてはエッジルータがDSCPを指定するが，例えば無制限にすべてのエッジルータがプレミアムサービスを指定すれば，DiffServの論理はただちに破綻する．破綻しないためには，エッジルータにおいて各サービスの割合をあらかじめ決定しておく必要がある．DiffServでは，ドメイン内での資源管理を行うサーバを帯域ブローカ（Bandwidth Broker）と呼んでいる．DiffServドメイン内の帯域管理は，集中的に行わなければならない．また，それゆえに，ドメインをまたがったQoS差別化を実現しようとすれば，ドメイン間の連携が必要になる．これがSLA（Service Level Agreement）の考え方である．しかし，これらも本来のインターネットの考え方とは異なるものである．

以上のようにIntServやDiffServの導入が図られた背景には，テレコムキャリアのデータコムビジネス（インターネット）への参入が大きいと考えられる．従来の電気通信網におけるビジネスモデルでは，ネットワーク提供者はユーザとQoSに関する契約を行い，QoS保証を含めたサービスを提供することによって収入を得る．このような考え方をデータコムに持ち込もうとすれば，当然，QoS保証，それが困難であればQoS差別化，ということになる．しかし，その実現に課題が多く残されているのは上述のとおりである[7]．

4. MPLSとトラヒック工学

IntServ，DiffServはそれぞれ，コネクションレベルでのQoS保証，パケットレベルでのQoS差別化に関するものであるが，最近では，パスレベル（パスについては後述）においてトラヒックフローを制御するという考え方が出てきた．背景には，MPLS（Multi-Protocol Label Switching）ネットワークの登場がある[8]．MPLSは，ATMやWDM（Wavelength Division Multiplexing）などの回線交換型ネットワークを下位インフラとして，その上にIPネットワークを構築するものであり，ネットワーク層とデータリンク層の中間に位置する．データリンク層が隣接ノード間の回線を用いたデータ転送を規定するのに対して，MPLSでは複数ノードをまたがったドメインのエッジノード間に直結の仮想回線（これをパスと呼ぶ）を設定し，論理的なネットワークを構築する．エッジノードに到着したIPパケットは，その目的アドレスに従って適切なパス上に流される．論理ネットワーク上にどのようにパスを設定し，どのパスにIPパケットを流すかは経路制御に属するものであるが，あるコネクション専用のパスを設定し，その帯域を他のコネクションには使わせないようにすれば，コネクションに対する帯域保証も可能になる．最近，このようなMPLSにおける経路制御に関する諸技術を特にトラヒック工学（Traffic Engineering）と呼んでいる．

4.1 MPLSの概要と問題点

MPLSネットワークへの入口のルータ（Ingress LSR：Ingress Label Switching Router）では，到着するIPパケットの宛先アドレスを見て，適切なラベルを与える（図2参照）．MPLSネットワーク内のルータ（Core LSR）では，IPアドレスではなく，ラベル（例えば，ATMにおけるVPI/VCI，WDMにおける波長番号）に基づいてフォワーディングを行う．フォワーディング情報はテーブルによって維持され，経路制御プロトコル（後述）によってテーブルを更新する．MPLSネットワークの出口ルータ（Egress LSR）に到着すると，パケットのラベルを取り除く．

MPLSの利点として，フロー集約，ラベルへのマッピングによるさまざまな経路制御プロトコルのサポート，固定長ラベルによるパケットフォワーディング処理の高速化などが挙げられるが[8]，ここで重要なことは，経路を明示的に指定することができる

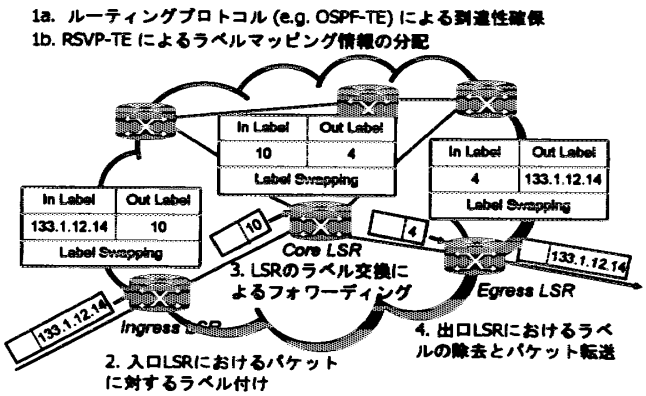

図2 MPLSネットワークアーキテクチャ

(explicit routing) 点である．その結果，経路制御によって効率的な資源利用が期待でき，またQoS制御も可能になる．

QoS制御は以下のようにして実現でき，これをトラヒック工学と呼んでいる．基本的には，帯域などのリンクに関する情報は，既存の経路制御プロトコルであるOSPF (Open Shortest Path First) を拡張したOSPF-TEを用いて収集する[9]．エッジルータはその情報によって制約付き最短経路アルゴリズムによって，LSRの明示的な経路を定める．求めた経路の情報はRSVPの拡張であるRSVP-TEを用いて経路上のルータでのラベル設定を行う．その結果，エッジルータ間で帯域の制約などを考慮した経路制御が可能になる．もちろん，MPLSが適用できる範囲は一つの閉じたドメインであり，QoSルーティングもその範囲内に限られる．すなわち，エンドホスト間のQoSが保証できたり，差別化できたりするわけではない．

5. QoSとEnd-to-End Argument

以上のようなアプローチに対して，その有用性を問う議論はもちろんある．特に，インターネットがこれまで発展してきた経緯を考えると，その分散性は重要な特徴であり，End-to-End Argumentと呼ばれるエンド間通信の原理は特に重要である[10]．これは，

(1) ネットワークは特定のアプリケーションに基づいて，あるいは，特定のアプリケーションのサポートを目的として構築してはならない
(2) エンドノードで実現できる機能はそのノードに任せ，関係するステート情報はそのノードにおいてのみ維持すべきである

というものである．

すなわち，通信機能はできるだけエンドノードにおいて実現すべきで，ネットワークはビットを運ぶことに徹すべきである，というものである．もちろん現実にすべてこのとおりというものではないが，基本的なアーキテクチャの原理として維持することが必要で，これらによって予測不可能なアプリケーションやサービスの出現に対応可能な，発展性のあるネットワークが構築できる．少なくとも現状のIntServやDiffServは，上述の原理に則ったものとはいいがたい．それよりも，次節で述べるような手法がより現実的であるといえる．

6. 輻輳制御とトラヒック計測技術

前節に述べた観点に基づけば，QoSに関する積極的な制御はやめて，ネットワーク技術の進展にあわせつつ，ネットワーク資源を常に余分に配するという考え方が成立しうる．そうすれば，QoS保証はされなくとも，ある程度のレベルの品質は提供される．実際，通信技術の発展により，ボトルネックはエンドホストに移行しつつあり，コネクションレベルでのQoS保証という考え方は捨てる，そのかわり，ネットワーク資源がボトルネックにならないように常に回線増強に心がける，という考え方には意味がある．その場合に鍵となるのは，トラヒック計測技術と輻輳制御である．すなわち，トラヒック計測技術によって現状のQoSを同定し，必要ならネットワーク資源増強のアラートとする．また，いったんネットワーク資源が与えられた時に，QoSを最大化するために輻輳制御を用いる．

6.1 トラヒック計測技術

現状のネットワークの通信品質を知ろうとすれば，トラヒック計測が必須である．トラヒック計測技術には，以下のように受動的測定 (passive measurement) と能動的測定 (active measurement) がある．

6.1.1 受動的トラヒック測定

受動的測定は，ルータ上であるいは回線上でモニターして，通過するパケットを調べるものである．例えば，パケット数とその長さをカウントすることにより，回線の利用率を知ることができる．その結果，回線利用率が高いことがわかれば，少なくともその回線部分におけるパケット転送遅延が増えていることを意味し，回線容量の強化の必要性が把握できる．また，より高位のプロトコル階層の振る舞いを調べることも可能である．その結果，例えばTCP層におけるフロー単位，あるいはアプリケーション層におけるコネクション単位の分析も可能になる．

しかし，受動的測定はあくまで定点観測である．インターネットの重要な概念はエンド間通信であり，ユーザが感じる品質はあくまでエンドアプリケーション・ユーザの尺度に基づくべきである．また，インターネットは，そもそも緩やかなネットワークの集合体として規定され，各エンドユーザ，あるいはISPなどのネットワーク管理体であっても，インターネット全体のネットワークの負荷状況を知ることはおよそ不可能なことである．そのような場合に，QoSを測ろうとすれば，次節で述べる能動的測定が必要になる．

6.1.2 能動的トラヒック測定

例えば，あるエンドホスト間QoSとしてTCPレベルでのスループットを知りたければ，実際にデータ転送実験を行えばよい．そのような測定をシステマティックに行うためのツールが近年さかんに開発されている[11]．例えば，pathcharはエンドホスト間の各リンクの回線容量が測定できる．もちろん，pathcharによって回線容量がわかっただけではQoS向上には結びつかないが，エンドユーザが現状のネットワークの状況を知る一助となる．さらに，ISPが管轄下以外のネットワークの状況を知ることも可能になる．これらの測定手法の組み合わせにより，ボトルネックとなっている回線やルータを把握することができれば，ネットワーク資源の適切な増強に結びつけることができるようになる．

6.2 輻輳制御

上述のトラヒック工学を含む経路制御が数十秒からそれ以上のタイムスケールにおけるネットワークの資源管理を行うものであるのに対して，より短期的な輻輳制御を行うものがTCPである[12]．TCPでは，エンドホスト間のパケット流量を整流するフロー制御としてウィンドウフロー制御を採用しているが，ネットワークの輻輳状態に応じてウィンドウサイズを制御することによって輻輳制御を実現している．基本的には，送信側ホストは受信側ホストからのACKを受信するごとに（帯域が十分あると判断されるので）ウィンドウサイズを増加させ，ACKが返ってこなかった場合には（ネットワークが輻輳状態にあることを意味していると解釈して）それを減少させる．ここで注意すべきことは，ネットワーク自身（ネットワーク層）が輻輳制御を行うのではなく，エンドホストに位置するトランスポート層（TCP）が行うという点である．

以下では，最近の話題を三つ挙げる．

(1) TCPの公平性[13]

TCPパケットがネットワーク内でランダムに確率pで失われるものとし，RTTを平均ラウンドトリップ時間とした場合，TCPのパケット単位で正規化されたスループットは，

$$S=\frac{1.3}{RTT\times\sqrt{p}} \tag{1}$$

で与えられる．式(1)から明らかなように，TCPのスループットはRTTに反比例する．これは，ラウンドトリップ時間を経たACKを受け取るごとにウィンドウサイズを変更するためである．これを解決するためには，TCP自体の改良，ネットワーク側のサポートが考えられるが，前者についてはディプロイメントの問題がある．例えば，現在広く使われているTCP Renoの改良版としてTCP Vegasが提案された．TCP VegasはTCP Renoより性能，公平性の面で優れているという評価結果も得られているが，TCP Renoとの混在下ではTCP Renoのほうが急激なウィンドウサイズの変更を行うことでより高い性能を得る[14]．これでは誰もTCP Vegasに移行しようとは思わないだろう．一方，ルータでコネクション間の公平なサービスを行うためには，例えばコネクションから発生したパケットのラウンドロビンスケジューリングを行えばよいが，その場合にはステート保持が必須となる．

(2) TFRC (TCP-Friendly Rate Control)[15]

もう一つのトランスポート層プロトコルであるUDPでは輻輳制御を行わない．その結果，いったんネットワーク内で輻輳が発生すると，TCPはウィンドウサイズを小さくすることによって送出レートを下げるが，UDPはそのままパケットを送出し続けるため，TCPとUDPの間に著しい不公平が生じる．そのため，End-to-End Argumentの観点から，UDPも輻輳制御に参加すべきであるという議論があり，TFRCの定義を以下のように定めた．

> "A non-TCP connection should receive the same share of bandwidth as a TCP connection if they traverse the same path."

TCPスループットにタイムアウト時間T_0を考慮した場合には，

$$S=\frac{1}{RTT\sqrt{\frac{2p}{3}}+T_0\min\left(1,3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)} \tag{2}$$

で与えられるので，UDPもこの式に合うように送出レートを調整すべきであるという考え方である．現状でもストリーミング型アプリケーションは独自の輻輳制御を行っているが，それはあくまで自アプリケーションのQoSを最適化することを目的としており，今後はTFRCに基づく制御を行うことが要求されている．

(3) 輻輳制御の基礎理論

TCPコネクションは，ネットワーク輻輳の発生を，ACKを介して検知し，ウィンドウサイズを制御する．すなわち，フィードバックループを形成している．と

ころが，これまでのネットワークにおける基礎理論研究は待ち行列理論に関するものが中心で，フィードバック系を扱う理論がほとんどなかった．そのため，最近では制御理論を適用した基礎的研究も活発に行われつつある[12]．TCPに限らず，次節でも述べるように，エンドホストのネットワーク適応性が今後重要になると，このようなフィードバック系に関する研究がますます重要になるものと考えられる．

7. おわりに

本稿では，QoSに関して主としてネットワーク側サポートの現状を中心に述べてきた．しかし，今後は，ネットワーク側のQoSサポートを前提とするのではなく，エンドホストがネットワークの状態に適応しつつ，かつ，アプリケーションの特性に応じてよりQoSを高める機構がより重要となると考えられる．ストリーミング技術などはまさしくその例である．また，モバイルネットワークやP2P (Peer-to-Peer) ネットワークにおいてはネットワーク資源（回線や情報源）そのものの移動，変動もありうる．すなわち，IntServ，DiffServ，MPLSなどのようにそれらが半固定化されているような状況が想定し得ないネットワークも出現しつつある．そのため，今後のネットワーク研究に重要なキーワードとして，スケーラビリティ，モビリティのほかにエンドホストの適応性 (adaptability) も挙げられよう．

参考文献

[1] 村田正幸，“マルチメディア情報ネットワーク”，共立出版，1999.

[2] R. Braden, et al., “Integrated Services in the Internet Architecture: an Overview”, *RFC 1633*, June 1994.

[3] 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.

[4] R. Braden, et al., “Resource ReSerVation Protocol (RSVP)”, *RFC 2205*, Sept. 1997.

[5] 村田正幸 他，社会基盤としてのインターネット，岩波書店，Sept. 2001.

[6] S. Blake, et al., “An Architecture for Differentiated Services”, *RFC 2475*, Dec. 1998.

[7] G. Huston, “Next Steps for the IP QoS Architecture”, *RFC 2990*, Nov. 2000.

[8] U. Black, MPLS and Label Switching Networks, Prentice Hall PTR, 2001.

[9] 佐藤昌平，吉田万貴子，“次世代インターネットとトラヒック工学”，電子情報通信学会，Vol. J 85-B, No. 6, pp. 875-889, June 2002.

[10] D. Clark, and M. Blumenthal, “Rethinking the Design of the Internet: The End-to-end Arguments vs. the Brave New World”, Telecommunications Policy Research Conference, Sept. 2000.

[11] “Internet Tools Taxonomy”, available at http://www.caida.org/tools/taxonomy/, 2002.

[12] 村田正幸，“インターネットにおける輻輳制御とトラヒック特性”，システム/制御/情報学会誌，Vol. 43, No. 3, March 1999.

[13] “Networking Lab: California Institute of Technology”, available at http://netlab.caltech.edu, 2003.

[14] G. Hasegawa and M. Murata, “Survey on Fairness Issues in TCP Congestion Control Mechanisms”, *IEICE Trans. on Commun.*, Vol. E 84-B, No. 6, pp. 1461-1472, June 2001.

[15] “The TCP-Friendly Website”, available at http://www.psc.edu/networking/tcp_friendly.html, 2002.