

インターネットにおける QoS 制御技術 ~ Diffserv

長 健二郎 (ソニーコンピュータサイエンス研究所)

1999 年 12 月 14 日

Internet Week 99 パシフィコ横浜

(社) 日本ネットワークインフォメーションセンター編

この著作物は、Internet Week 99 における長 健二郎氏の講演をもとに当センターが編集を行った文書です。この文書の著作権は、長 健二郎氏および当センターに帰属しており、当センターの同意なく、この著作物を私的利用の範囲を超えて複製・使用することを禁止します。

©1999 kenjiro Cho, Japan Network Information Center

目次

1	概要	1
2	DiffServ 登場の背景	1
3	QoS とは何か	1
4	トラフィック制御理論と既存技術	6
5	IntServ と RSVP	9
6	DiffServ への流れ	11
7	DiffServ のしくみ	16
8	サービスの構築	19
9	DiffServ の課題	20
10	まとめ	21
11	参考資料	21

1 概要

この講演では、DiffServ について、次の順番で説明していきます。

- DiffServ 登場の背景
- DiffServ、および関連するアーキテクチャ
- DiffServ サービスの構築
- 今後の課題

2 DiffServ 登場の背景

パケットスイッチングを元にしたインターネット最大の特徴は、ベストエフォートであると言うことができるでしょう。すなわち、統計多重によって使われていない帯域を有効活用しており、エンド - エンドでの通信は保証されていません。このようなネットワークの問題として、輻輳 (congestion) が発生する恐れがあり、輻輳が発生した場合には際限なく品質が低下する可能性が挙げられます。

90 年代に入って、インターネットが爆発的に普及してきた時に、リアルタイムアプリケーション (音声や動画) や、ビジネス向けの高品質なサービスに対する要求が高まってきました。また、データ通信が音声通信を量的に上回るようになり、電話網の上に IP ネットワークを構築するよりも、IP ネットワークの上に電話網を構築した方が効率的なことから、電話網に匹敵する品質をインターネットで実現することが必要となってきました。

3 QoS とは何か

ネットワークサービスの品質、すなわち QoS (Quality of Services) とは、通信の品質を定量的に表現するものと言えます。つまり、通信の帯域速度、パケットの遅延時間やジッタの量、パケットの損失率等です。

3.1 QoS のしくみ

通信品質を定量的に一定範囲に収めるためには、次のような優先制御の技術が使われます。

- アドミッション制御 (admission control)

資源の動的な配分方法を判断することです。電話網で言えば、発呼時に発信元から着信先までのネットワークパスを確保することに相当します。

動的にアドミッション制御を行う場合には、セットアッププロトコルでパス上の資源を確保します。ここで、資源の確保に失敗すると、エラーを返します。確保した資源は、通信の終了時ならびに故障時に、正しく解放することが重要です。

アドミッション制御に関連して、パケットの経路を決定するポリシーや、課金の技術も必要となります。

- クラシファイア (classifier)

到着したパケットをグループに分類する機構です。

通常の IP パケットでは、ソースアドレス、ソースポート、デステネーションアドレス、デステネーションポート、プロトコルの 5 つの値を使って、パケットのクラス分けを行っていました(ファイアウォールのパケットフィルタと同様)。このような方法でクラス分けを行うと、検索コストが高ついてしまいますので、DiffServ では IP ヘッダの中の TOS (Type Of Services) フィールドを定義して利用することが行われています。

- シェーピング (shaping) / ポリシング (policing)

バーストを一定のレートにならず (シェーピング)、あるいは、規定以上の入力がないかを監視する (ポリシング) 機構です。

シェーピングは、バーストをならしてジッタを減少させ、トラフィックが一定量になるように働きます。シェーピングでは、規定以上の入力を監視して、それを超えるものを破棄する働きをします。通常、ポリシングを行っているネットワークには、シェーピングを行ってならしたトラフィックを流し込むために、これらの技術を同時に使用します。

- パケットスケジューリング (packet scheduling)

パケットを、属するグループに応じたレートで送出する機構です。パケットをキューイングする機構と、バッファ管理を行う機構に分けて考えることができます。QoS を実現する基本的な技術ですから、代表的な方法について引き続き解説します。

3.2 キューイング方式

3.2.1 Priority Queuing

最も基本的な方法で、優先度に基づくクラス毎にキューを持ち、優先度の高いものから出力していく方法です。単純な機構で、(優先度の高いクラスの)リアルタイム制を保証できますが、優先度の低いクラスがスタープ(枯渇)してしまう可能性があります。

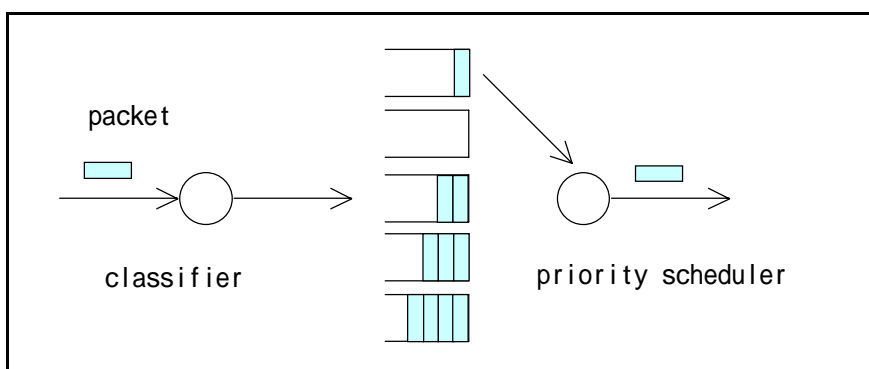


図 1 : Priority Queuing

クラシファイアによってクラス分けされたパケットは、優先度別にキューに入れられます。スケジューラは、優先度が高いキューから順にパケットを取り出して出力していきます。

3.2.2 WFQ (Weighted Fair Queuing)

フロー(たとえばTCPセッション)毎にキューを割り当てて、フロー毎に公平なキューイングを行う方法です。他のフローの影響を一定以下に抑えることができますが、フローの数だけキューが必要となります。

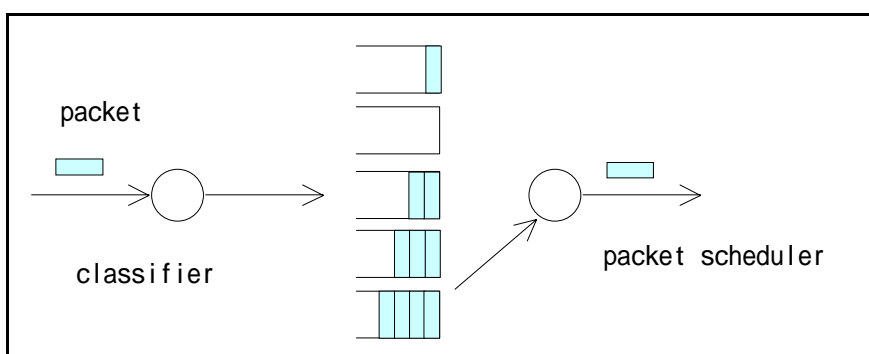


図 2 : WFQ (Weighted Fair Queuing)

クラシファイアは、フロー毎のキューにパケットを納めます。スケジューラは、ラウンドロビン方式で、それぞれのキューから公平にパケットを出力します。これにより、それぞれのフローは、少なくとも1周に1回は出力のチャンスを得ます。

3.2.3 CBQ (Class-Based Queuing)

フローを階層的に制御するために、クラス毎に規定した使用量に応じたトラフィック制御を行うための方法です。

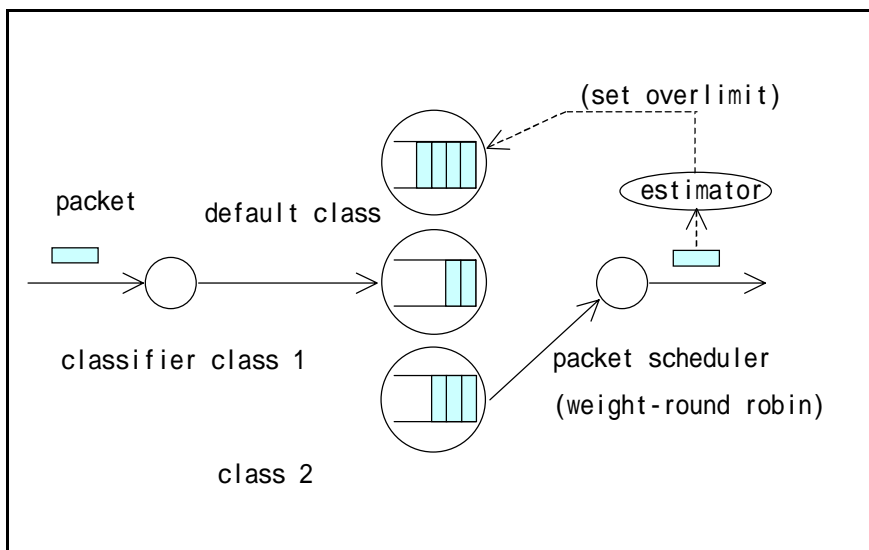


図 3 : CBQ (Class-Based Queuing)

スケジューラの出力は、クラス毎にエスティメータによって測定され、クラスに設定された上限値を超えた場合には、キューに対するペナルティとなります。スケジューラは、そのペナルティを考慮したラウンドロビン方法によって出力を行います。これにより、クラス毎に「出力の % まで」といった、定量的な利用量を定義できるようになります。

3.3 バッファ管理

3.3.1 Drop-Tail / Drop-Head / Drop-Random

キューが溢れた時に、溢れたパケットを破棄するための機構には、いくつかの種類があります。Drop-Tail とは、いわゆる FIFO バッファで、溢れているキューの末尾、すなわち最後のものを破棄する方法です。Drop-Head とは、逆に先頭のものを破棄する方法で、TCP の再送メカニズムからは、パケット溢れが早期に検出できるというメリットがあります。また、キューにあるパケットをランダムに破棄する、Drop-Random という方法も考えられます。

3.3.2 RED (Random Early Detection)

ウィンドウによって流量をコントロールしている TCP では、いくつかのフローにおいて同時にパケットの破棄が起きた場合に、それぞれのフローが同時に送出手の停止と再開を行って、効率的な送信ができなくなってしまうという同期現象が起こり得ます。RED は、これを回避して、バッファが溢れる前に流量をコントロールし、キューの長さをできるだけ短く保つために、キューの長さの平均値に応じた確率でパケットの破棄を行う方法です。平均キューサイズの上限值と下限値を決めておき、図 4 のような確率でパケットを破棄します。

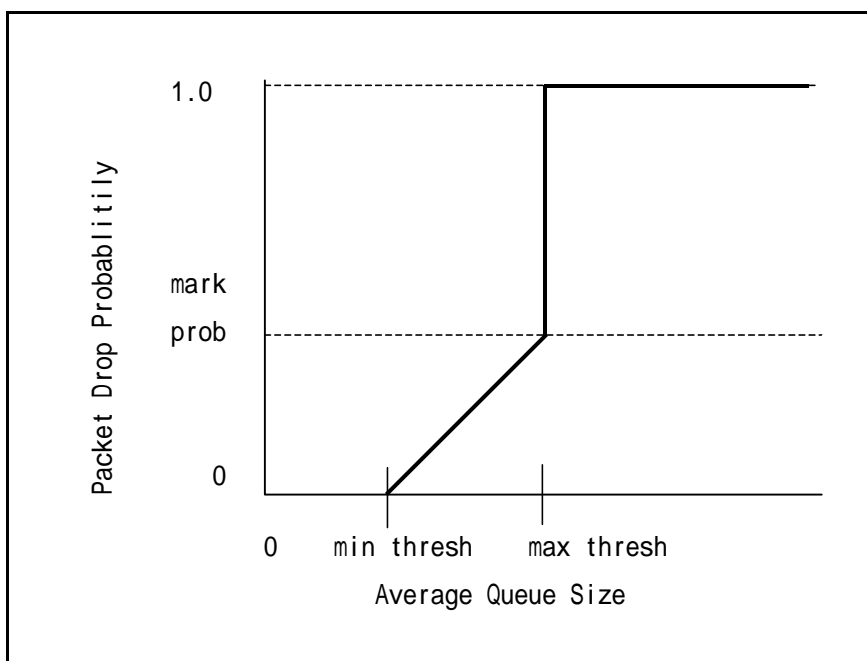


図 4 : RED のパケット破棄

これにより、バッファの占有率に応じた公平な破棄となりますし、瞬間的な短いバーストを許容することもできます。ただし、パケットの破棄が起きても流量を絞らないフロー（たとえば UDP のストリーム等）には無防備です。

4 トラフィック制御理論と既存技術

QoS を理解するために、まず、トラフィックの論理モデルについて簡単に振り返っておきましょう。

4.1 キューイング理論

ARPANET 時代に、パケットの挙動を統計的に解析することが確立されました。元来は、電話網（回線交換）における着呼や通話時間がポアソン分布することを利用したものでした。パケットを使ったデータ通信においては、パケット長が可変であることや、バースト的な通信パターンのため、正確な分析は困難です。

キューイングシステムでは、一定間隔で到着する負荷の場合には効率の低下はありませんが、到着間隔にゆらぎがある場合には、負荷がある段階（上限の 7 ~ 8 割）を超えると急激に効率が悪化することが知られています。

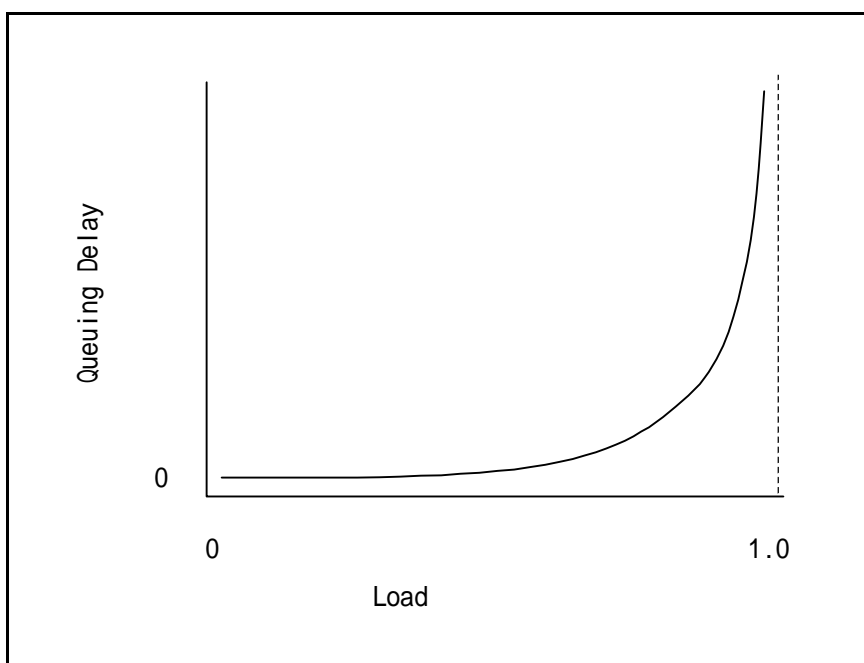


図 5：キューイング理論

4.2 遅延保証

80年代の半ばに、Parekh が、パケット交換において遅延保証ができることを解析的に証明しました。QoS は、この理論を基に発展を始めたと言っても良いでしょう。Parekh は、アドミッション制御と、送出量の制限(シェーピング)、パケットスケジューリングを組み合わせれば、遅延時間を一定以内に抑えられることを示しました。

Parekh のモデルでは、トークンバケット(シェーピングの一種)と WFQ を組み合わせて使用しました。トークンバケットによって、一定レート以下にシェーピングされたトラフィックは、WFQ によるキューイングシステムを何段か経由しても、WFQ によって最大遅延時間が保証されますので、エンド - エンドでも遅延時間が保証できることになるというものです。

このように、パケット交換においても遅延保証ができることが証明されたことにより、データ通信と音声通信を統合したネットワークが考えられるようになってきました。電話網を拡張してデータ通信を行う技術として ISDN や ATM が、データ通信網を拡張して音声等の QoS を必要とする通信を行う技術として IntServ や DiffServ が登場してきます。

コラム：ポアソン分布

平均値を中心とした出現確率を持つ分布。平均値から離れるほどに、急激に確率が減少するものです。

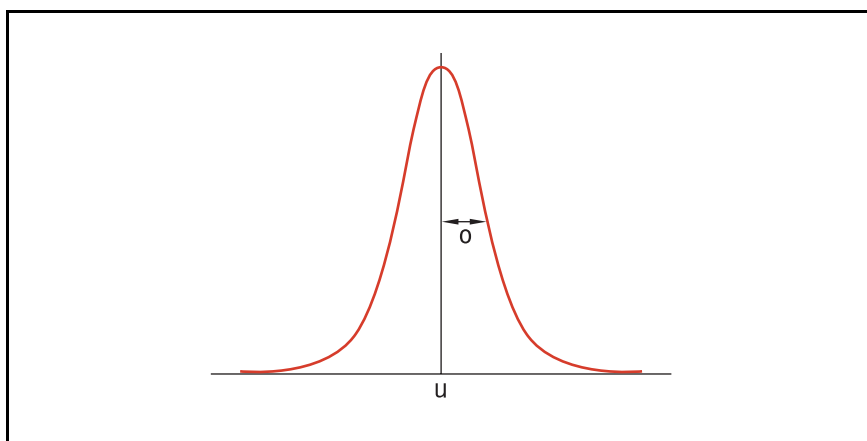


図 6：ポアソン分布

4.1 ATM

この講演の本題であるDiffServにも影響を与えていますので、ここでATMの技術について解説しておきます。

ATMの次のような特徴は、DiffServ等にも生かされています。

- 固定長のセルを使用している。
固定長のセルを利用するため、遅延保証を計算しやすくなっています。
- ホップ毎にVCを書き換える。
スイッチ間でローカルなIDを受け渡すことで、スケーラブルなネットワークを実現しています。
- 網とユーザネットワークが分離されている。
網の境界で制限を行うことで流入量を監視する等、責任境界が明確になっています。
- さまざまなサービスクラスがある。
CBR (Constant Bit Rate)、UBR (Unspecified Bit Rate)、VBR (Variable Bit Rate) 等、さまざまなサービスクラスが定義されています。
- 管理されたネットワークが前提である。
正しく設定され、ポリシーがしっかりしているネットワークを前提として、プライオリティスケジューリングが簡単に有効となります。

これまでは、網中心の設計思想や、ATMの想定要件を上回るインターネットのバースト的なトラフィック、電話網屋さんとインターネット屋さんの文化の違い等から、インターネットではATMの技術があまり利用されてきませんでした。しかし、最近では、ATMのラベルスイッチング技術をインターネットにもうまく活用するようになってきています。

4.2 IPパケットのTOSフィールド

IPパケットの第2オクテットにある、8ビットのTOSフィールド(図7)をうまく使ってQoSを実現しようというのがDiffServの出発点でした。

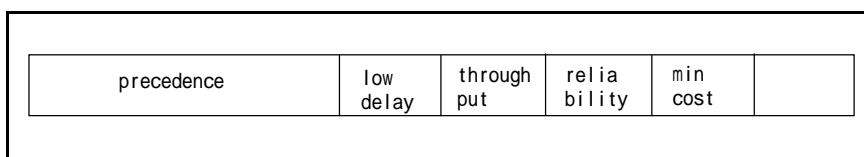


図7：TOSフィールドの意味

Precedence の 3 ビットを使って優先順位を指定し、次の 4 つのフラグを使ってパケットの取り扱いを指定することを目標としていました。

low delay	遅延を最小とする。
throughput	スループットを最大にする。
reliability	信頼性を最大にする。
min cost	コストを最小にする。

しかし、この定義が曖昧だったことと、管理するメカニズムが存在しなかったために、事実上はあまり使用されず、ベンダーが独自に利用するといった用途に限られていました。

DiffServ は、この TOS フィールドを、組織間での利用を考慮して拡張したものであるという考え方も可能です。

5 IntServ と RSVP

DiffServ の前身として、89 年頃から研究が始まった IntServ と RSVP を紹介しておきましょう。

IntServ (Integrated Service) は、QoS パラメータの指定方法や、交換のフォーマットを定義するものです。それを具体化したプロトコルとして RSVP (ReSerVation Protocol) が定義されました。すなわち、RSVP は IntServ パラメータを交換して QoS のサービスを実現する、資源予約プロトコルです。

5.1 IntServ

IntServ には、2 つのサービスモデルがあります。1 つは、「Guaranteed QoS Control サービス」という、論理的な QoS サービスを実現しようとするモデルです。Guaranteed QoS Control サービスを実現することは困難であったため、現実的なモデルとして「Controlled-Load サービス」が登場しました。これは、適応型のアプリケーションを想定して、実際には混雑しているネットワークの上であっても、負荷の低い(トラフィックが少ない)ネットワークと同じ動作を実現しようとするものです。

IntServ では、トークンバケットと言うシェーピングのしくみを使って短いバーストを許容することで、TCP との親和性を高めています。

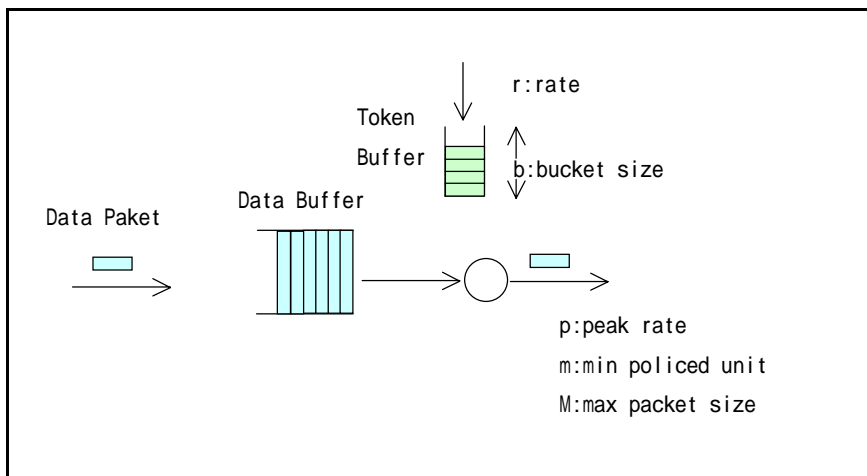


図 8 : トークンパケット

これは、データのバッファとは別に一定のレート r でトークンが追加されるバッファを考え、トークンバッファにトークンがある場合に限り、データバッファからパケットを送出できるように制限するものです。トークンが入ってくるレート r が最大のレートを、トークンバッファの大きさ b が許容できるバーストの大きさを制御することになります。

これらのパラメータは、IntServ において重要な意味を持っています。

5.2 RSVP

RSVP の特徴は次のようにまとめられます。

- ソフトステートである。

電話網のように途中の装置が状態を保持しているものではなく、状況に応じて状態を常に更新し続けるシステムとすることで、自動的に障害への対応が行われるシステムとなっています。

- IP マルチキャストと親和性が高い。

マルチキャストを意識して、データの受信側が主導権を持って制御を行うシステムです。また、途中のノードで資源予約をマージする点もマルチキャストと類似しています。

- ルーティングプロトコルから分離している。

ルーティングプロトコルが確立したパスの上に、独立して資源を予約していくシステムとなっています。

- 実装・運用に依存する部分が分離されている。

アドミッションを制御する部分、トラフィックを制御する部分等、実装が難しい部分が抽象化されており、プロトコルからうまく分離されています。

ただし、RSVP は次のような理由からあまり利用されませんでした。

- スケーラビリティがない。

中間ルータにおいて、フロー毎のステートが必要で、バックボーンにおいてはステート数が膨大となってしまうためスケールしません。

- 技術主導であり、現状とのギャップが大きかった。

メカニズムが複雑すぎる、システム管理が困難、課金やコスト等のビジネスモデルとの整合がない等、実際の運用には多くの問題点がありました。

- シグナリングが本質的に難しい。

動的な状態管理を行うアドミッション制御は、本質的に解決が難しい問題をいろいろと持っていることがわかり、DiffServの開発へと繋がっていきました。

6 DiffServ への流れ

IntServ/RSVP への疑問や、マーケットからの要求によって、簡単な方法ですぐに ISP が使用できる標準の必要性が高まってきました。実際のサービスは、スケーラブルで、ISP のビジネスモデルにマッチしたものであることが求められます。また、IP パケットの TOS フィールドを再定義して利用しようという気運が盛り上がってきていました。

そこで、これまでの理論的な最悪値を積み上げ計算していく従来の QoS のアプローチとは異なり、広帯域の回線や、より高機能なエンドシステムを用いた厳密な制御を行う必要のない、現実的な QoS 実現に向けてのアプローチが強くなってきました。従来の IntServ/RSVP が「制御するか、しないか」の二者択一的であったのに対して、厳密に制御しなくても、そこそこの実用的に動作する provisioning を重視する考え方に変化してきたと言えることができるでしょう。

6.1 DiffServ の基本

DiffServ は次のようなアイデアを基にして実現されています。

- 相対的な QoS の利用

絶対的な QoS ではなく、実現が容易でインターネットに向いている、通信品質が異なるクラスを定義して、それらの間の相対的な品質を制御する CoS (Class of Service) を実現しようとするものです。絶対的な QoS と、相対的な QoS は混在して使用することができます。

- プロビジョニング

余裕を持った資源配分を、コスト効率や運用の容易性、拡張性等の面から、厳密な制御との間でバランスさせる考え方です。より現実的なものを実現できる反面、論理的な解析が困難になってきています。

- TOS フィールドの再定義 (図 9)

TOS フィールドを、6 ビットの DS (Differentiated Service) フィールドとして定義し直し、DSCP (DiffServ Code Point) という、DS ドメインにローカルな値を納めて利用します。少数の DSCP は、標準として互換性を持たせています。

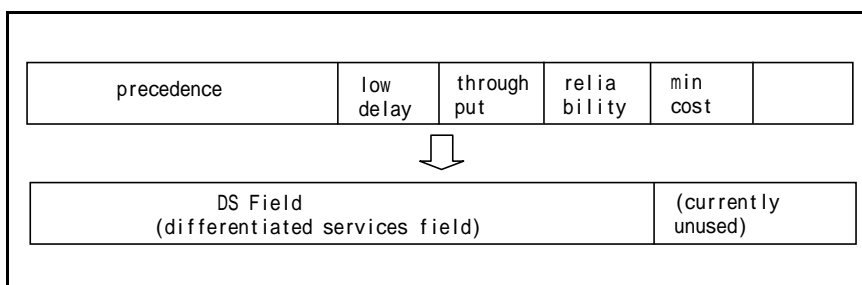


図 9 : TOS フィールドの再定義

なお、IPv6 では、同じく 6 ビットの Traffic Class フィールドが定義されています。

- DS ドメイン

閉じたネットワークにパケットが流入した時にクラス分けを行って DSCP を書き込み、そのネットワークの内部では DSCP のみを参照して優先制御を行います。ルーティングにおける AS ドメインと同様に、2 階層構造を持ったピアリングモデルとすることで、DS ドメイン内が簡単になると共に、シグナリングが不要となります。

- PHB (Per-Hop Behavior)

DS ドメインに属するすべてのホップにおいて、外部から観測されるフォワーディングの挙動を定義したテーブルを用意し、パケット毎の DSCP からそのテーブルをルックアップすることで、パケット毎の挙動を制御します。集約したフローのみを扱うことでスケーラビリティを実現します。

- サービスモデル

ユーザとの契約、あるいは DS ドメイン間の契約によってネットワークの設定を行うサービスモデルとなっています。メカニズムを実現するコンポーネント、ポリシー、プロビジョニングを組み合わせ、契約を満足するサービスを、ISP の裁量によって実現できます。

このように、さまざまなアイデアを使って実現されている DiffServ は、TOS フィールドの値を DS フィールドにマップすることで、既存技術との互換性を保っています。また、ラベルスイッチングを使って DiffServ を実現する方法も考慮されています。DiffServ と IntServ/RSVP を独立して使用することもできますし、ドメインのエッジのみで IntServ/RSVP を利用し、DiffServ を使ってコアネットワークをトンネリングすることで、IntServ/RSVP のスケーラビリティを向上させる方法も考えられています。

6.2 アーキテクチャへの要求

DiffServ の開発が始まった時には、その開発目標として次のようなものがあげられていました。

- 複数のネットワークに跨る幅広いサービスやポリシーに利用できること。
- 特定のアプリケーションに依存しないこと。
- 既存のアプリケーションを変更する必要がないこと。
- シグナリングに依存しないこと。
- スタティックで簡単な構成が可能で、シグナリングに依存しないこと。
- 簡単な forwarding behavior (PHB) のみを規定し、ルータが単純なもので済むこと。
- コアネットワークにおいては、フローやユーザ毎の状態を持たず、簡単なクラシファイアのみで集約されたフローを扱うこと。
- DiffServ に対応していない機器とも相互運用でき、段階的な導入が可能であること。

このような要求を実現するために、ネットワークの入り口でクラシファイとコンディショニング（後述）を行って DSCP を設定し、コアネットワークでは DSCP に対応したホップ毎のフォワーディングを行うというアーキテクチャとなりました。

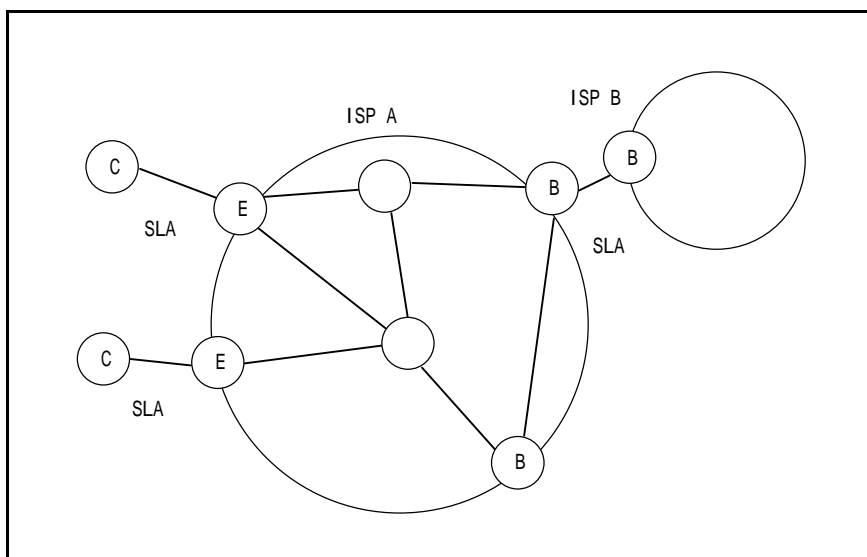


図 10 : DS ドメイン

すなわち、共通のポリシーで管理されるドメインの内部ノードは、共通の PHB セットが設定され、共通の DSCP を使用して運用されます。通常は、その内部の管理に責任を持つ組織や部門、ISP 等が DS ドメインの単位となります。なお、ピアリングルールが確立していて、相互運用が可能な連続した DS ドメインを「DS サービス領域（リージョン）」と呼ぶことがあります。

DS ドメインのエッジノードでは、ユーザ毎、状態毎の個別処理が行われます。入り口ではトラフィックコンディショニングが行われますし、出口ではピアリング契約に応じたシェーピングや DSCP の書き換えが行われます。対して、コアノードでは、集約されたフローのみを処理します。複雑な処理をすべてエッジノードで行うため、コアノードのスケラビリティが高まります。

6.3 DiffServ の標準化

DiffServ は IETF (Internet Engineering Task Force) で標準化作業が行われています。1997 年 8 月に Munich で開かれた会合で、IntServ ワーキンググループが DiffServ BOF を開催したのが始まりです。1998 年 3 月には、IETF 内に DiffServ ワーキンググループが設立され、大学や政府、ベンダー、ISP の積極的な協力によって、急速に標準化が進んでいます。

97年のDiffServ BOFで提案されたDiffServの原型が、現在のDiffServに大きな影響を及ぼしています。まず、それらの提案を解説しておきましょう。

- Premium Service Model

V. Jacobson（当時ローレンス・バークレー研究所、現在シスコ・システムズ）が提案した、現在のEF PHBの原型です。優先パケットが使用する帯域を十分に小さくして（10%以下）そこで単純なスケジューリングを行うと、入り口での適当なポリシングによって、低遅延が保証されるというものです。集線によってジッタが増加しますが、余剰帯域を利用しつつ、優先パケットが使用する帯域を仮想的な専用線として利用することができます。基本的には、ATMにおけるCBRと同じ考え方を提案したものでした。

- Drop Precedence Model

D. Clark（MIT）が提案した、現在のAF PHBの原型です。REDを改良したRIO（RED with IN and OUT）を使って、各パケットに契約内/外のマークを付けておき、輻輳時には契約外のパケットから破棄することで最低帯域保証サービスを実現しようとするものです。

なお、RIOとは、契約の内外によって独立したREDパラメータを与えて、輻輳時には契約外のパケットから先に破棄することで、契約内のパケットが破棄される確率を抑えるものです。

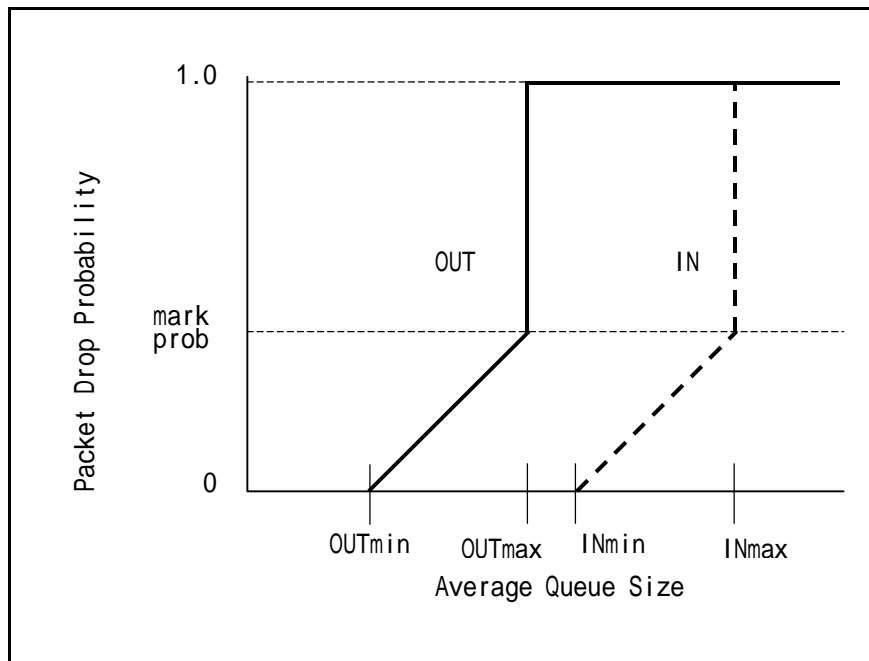


図 11 : RIO の破棄確率

- Cisco's CoS

シスコ・システムズのルータにおいて実装されている Precedence フィールドの利用方法を、F. Baker が説明したもので、Class Selector PHB の原型となっています。Precedence フィールドの値を使って、クラス毎に相対的な優先順位を付けるもので、7 段階に拡張した RIO である WRED (Wehghted RED) を利用しています。

IETF では、これらの提案を受けて、相互運用に必要な最小限の取り決めを実現するために、DiffServ ワーキンググループが作られました。そこでは、IPv4 の TOS フィールドと IPv6 の Traffic Class フィールドを DS フィールドとして定義するために RFC1394 を変更することや、実証実験のための枠組みとして標準的なアーキテクチャ (PHB) を規定することが話し合われました。

7 DiffServ のしくみ

7.1 DiffServ を構成するコンポーネント

DiffServ による QoS を実現するコンポーネントの役割を解説しましょう。ルータの入カインターフェイス側で、トラフィックコンディショニングを行い、出カインターフェイス側で PHB に応じた動作を行います。

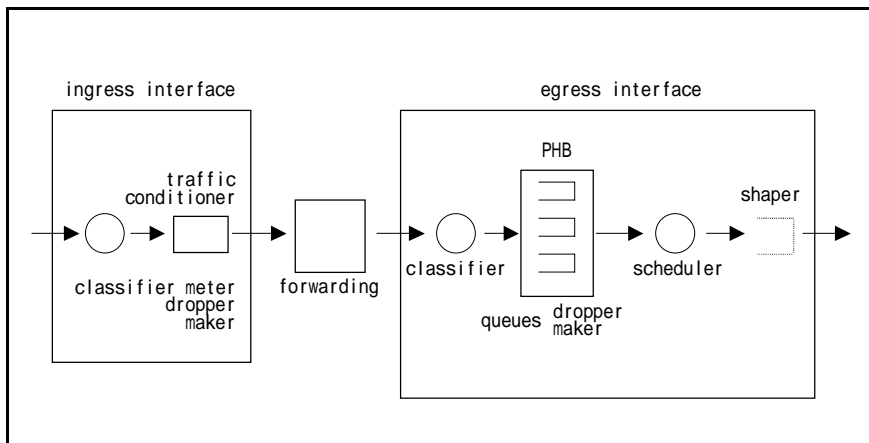


図 12 : DiffServ を構成するコンポーネント

入カインターフェイスにおいて、トラフィックコンディショニングを行う機能コンポーネントは、さらに次のように分類することができます。

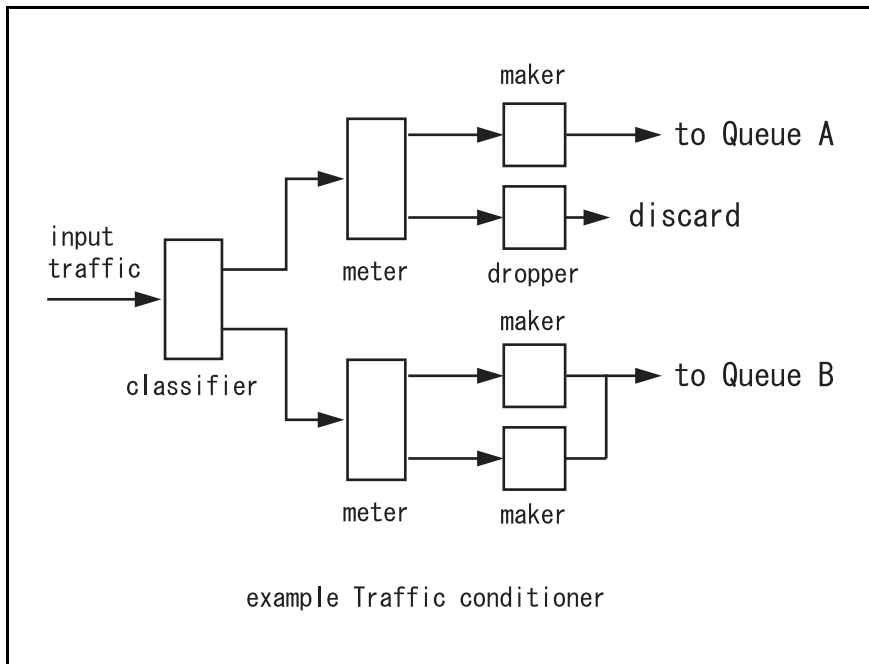


図 13 : トラフィックコンディショニング

- クラシファイア (Classifier)

パケットを対応するクラスに分類する働きをします。DS フィールドのみを参照してパケットを分類する「BA (Behavior Aggregate) クラシファイア」はコアノードで使用され、その他のパケットヘッダも参照してパケットを分類する「MF (Multi Field) クラシファイア」はエッジノードで使用されます。契約に基づくルール(プロファイル)によって、ルールにマッチする(契約値内)ものを in-profile、マッチしないものを out-of-profile と呼びます。

- メータ (meter)

クラシファイアが選択したパケットが、プロファイルに適合しているか否かを判断します。流量の平均値を見るものや、指定されたトークンパケットパラメータに一致するか否かを見るものがあります。トークンパケットパラメータでパケットを分類するメータを 2 段に組み合わせて、ピークレートを超えたものを赤、ピークレート未満で保証値を超えたものを黄色、保証値未満のものを青と、パケットを 3 種類に分類する「2-rate 3-color メータ」というものもあります。

- マーカ (maker)

メータによる分類結果を、DS フィールドに DSCP 値として書き込む働きをします。

- シェーパ (shaper)
パケットを遅延させて、プロファイルに適合させる働きをします。ただし、通常は入力側でシェーピングを行うことはあまりありません。
- ドロッパ (dropper)
メータによる分類結果に基づいて、パケットを破棄する働きをします。

7.2 PHB (Per-Hop Behavior)

それぞれのノードにおいて、外部から観測できる、パケットフォワーディング動作を記述したものです。たとえば、「最低帯域を保証する」、「ウエイトに比例した余剰帯域の分配を行う」といったように、特定の実装を指すものではありません。実際に PHB を実装するには、パケットスケジューリングを行う部分と、バッファ管理を行う部分が必要となります。

PHB を指定するコードポイントは、次のように定義されています。

- 000000 ~ Default PHB
最小限の資源を割り当てる条件があることを除いて、ベストエフォートを意味します。
- xxx000 ~ Class Selector PHBs (7 個)
IP パケットの Precedence にマップされて、互換の動作をします。実装アルゴリズムとして、WFQ、WRR、CBQ 等が使用できます。
- cccdd0 ~ AF (Assured Forwarding) PHBs (12 個)
ccc はクラスを、dd は drop precedence を意味します。4 つの独立したクラスに対して、それぞれ 3 つのドロップ Precedence を用意し、Precedence に応じた確率的な破棄を行います。
- 101110 ~ EF (Expedited Forwarding) PHB
低損失、低遅延、低ジッタの高優先度サービスを意味します。最低送出レートが保証される PHB を使用すること、さらに PHB が保証する最低送出レートよりも少ないパケットのみが流入することを保証するコンディショナを使用しなければなりません。実装アルゴリズムとして PQ、WFQ、WRR、CBQ 等が使用できます。
- xxxx11 ~ Experimental/LocalUse
- xxxx01 ~ Experimental/LocalUse
実験やローカルでの利用のために予約されています。

8 サービスの構築

ここで、DiffServ を使った QoS サービスをどう定義して実現するかを、例として示します。

8.1 EF を使った仮想専用線サービス

顧客との契約で、次のような項目を定義します。

- 接続がどこからどこまでか。
- トラフィックプロファイルとして、トークンバケットのレート r と深さ b の値。

トークンバケットのパラメータによって、in-profile パケットの遅延は msec でパケット破棄が % であることが示せます。なお、仮想専用線サービスですから、out-of-profile パケットは破棄します。

この場合に、ISP はエッジで次のような項目を設定します。

- クラシファイアでソースとディステネーションを見て、顧客のパケットを判断する。
- トラフィックコンディショナのメータに、トークンバケットのレートを設定する。
- in-profile パケットに EF DSCP をセットする。
- out-of-profile パケットは破棄する。

また、ISP は、この契約を実現するためのプロビジョニングとして、たとえば、DS ドメイン内の全てのルータで容量の 10% を EF 用にリザーブし、さらに、接続パス上のすべてのノードにおいて EF の合計がそのリザーブ量を超えないように販売する必要があります。

8.2 AF を使った最低帯域保証サービスの例

顧客との契約で、次のような項目を定義します。

- どこからの接続であるか。
- トラフィックプロファイルとして、トークンバケットのパラメータ r と深さ b の値。

トークンバケットのパラメータによって、in-profile パケットの遅延は msec でパケット破棄が % であることが示せます。out-of-profile パケットは、ベストエフォートとして扱われます。

この場合に、ISP はエッジで次のような項目を設定します。

- クラシファイアでソースを見て顧客のパケットを判断する。
- トラフィックコンディショナのメータに、トークンバケットのパラメータ（4つ）を設定する。
- メータの判定に応じて、異なる AF DSCP の値をセットする。

また、ISP は、この契約を実現するためのプロビジョニングとして、たとえば、DS ドメイン内のすべてのルータで容量の 50% を AF 用にリザーブし、さらに、実績ベースで保証値を下回る（緑の）パケットが破棄されないように販売する必要があります。

9 DiffServ の課題

ここまでの解説は、DiffServ ワーキンググループでの標準化がほぼ終了した事柄でした。決まっているのは枠組みだけであり、実装や運用は各ドメインに任されています。そのため、次のような点が問題となっています。

- DS ドメイン境界での再マーキング
ドメイン境界でのマッピングにおいて、等価な PHB が実現できるとは限りません。
- Bandwidth Broker
動的な資源配布を行おうとした場合には、ポリシーを管理するサーバーと、それに基づいてトラフィックを制御するルータが必要となります。ポリシーを管理するプロトコルの枠組みとして COPS プロトコルが着目されています。
- 受信者ベースのサービス
DiffServ は、送信者からのトラフィックしか制御できません。受信者主導の QoS を実現するには、一種のシグナリングを行うのが有効ではないかと言われてはいますが、まだアイデアだけで実体がありません。
- マルチキャスト
マルチキャストグループはダイナミックなので、プロビジョニングが困難です。マルチキャストについては、ほぼ手つかずの状態です。

- RSVP over DiffServ

エッジでのみ RSVP の処理を行い、コアネットワークを DiffServ でバイパスすることで、RSVP のスケーラビリティを高めようとするモデルですが、実際にうまく動作するかの実績がありません。

10 まとめ

従来はベンダーが機器の機能として QoS サービスを提供していましたが、今後は ISP の運用技術として提供されるようになって考えられます。

DiffServ の枠組みは固まってきており、対応製品も登場していることから、小規模なネットワークならば十分に利用可能となっています。しかし、大規模なネットワークにおける運用については不明な点が多く、実証実験を通じた経験の蓄積が必要でしょう。

11 参考資料

11.1 関連リンク

- IETF
<http://www.ietf.org/>
- IETF DiffServ ワーキンググループ
<http://www.ietf.org/html.charters/diffserv-charter.html>
- IETF IntServ ワーキンググループ
<http://www.ietf.org/html.charters/intserv-charter.html>
- IETF ISSLL ワーキンググループ
<http://www.ietf.org/html.charters/issll-charter.html>
- IETF RSVP ワーキンググループ
<http://www.ietf.org/html.charters/rsvp-charter.html>
- RSVP
<http://www.isi.edu/rsvp/>

- ALTQ

<http://www.csl.sony.co.jp/~kjc/software.html>

11.2 関連書籍

- Differentiated Services for the Internet. K. Kilkki.
ISBN 1-57870-132-5. 1999.
- Quality of Service. P. Ferguson and G. Huston.
Wiley, ISBN 0-471-24358-2. 1998.
- An Engineering Approach to Computer Networking. S. Keshav.
Addison-Wesley, ISBN 0-201-63442-2. 1997.
- High-speed Networks: TCP/IP and ATM Design Principles. W. Stallings.
Prentice Hall, ISBN 0-13-904954-1. 1998.
- Gigabit Networking. C. Partridge. (邦訳あり)
Addison-Wesley, ISBN 0-201-56333-9. 1993.