

## InternetWeek2000 チュートリアル

### Diffservの仕組みと動向

長 健二郎

ソニーコンピュータサイエンス研究所  
kjc@csl.sony.co.jp

## Diffservの概要

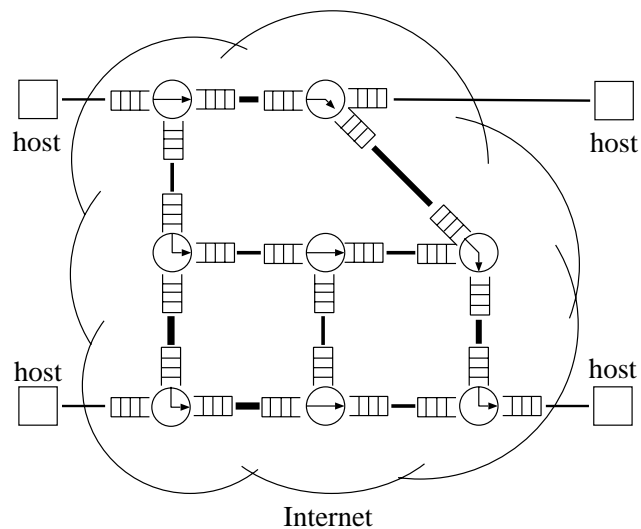
- 簡単な仕組みで粗粒度のクラス別QoSを実現する枠組
  - ネットワークの入口でクラス分け
    - ▷ IPヘッダ内にクラス識別子を書き込む
  - ネットワーク内部ではクラス別の優先制御
    - ▷ スケーラビリティへの考慮
- 標準化
  - IPヘッダ内のDSフィールド
  - コンポーネントの記述
- サービスの実現方法はISPの裁量
  - コンポーネントの組み合わせ方
  - プロビジョニング

## チュートリアルの内容

- DiffServの背景
  - DiffServを理解するにはQoS技術の知識が必要
- DiffServのアーキテクチャ
  - 標準化された部分と標準化中の技術
- DiffServの課題と動向

## インターネット

- パケットスイッチング
- ベストエフォート・サービス



## インターネットの利点と問題点

### □利点

- 使われていない帯域の有効利用（統計多重）
- 簡単な配送系（エンド・エンド・モデル）
  - ▶IP: パケット配送のみ
  - ▶TCP: エンド・エンドで通信を実現

### □問題

- 輻輳の発生
  - ▶パケットはだまって捨てられる
- 限度のない品質低下の可能性

## QoSへの要求

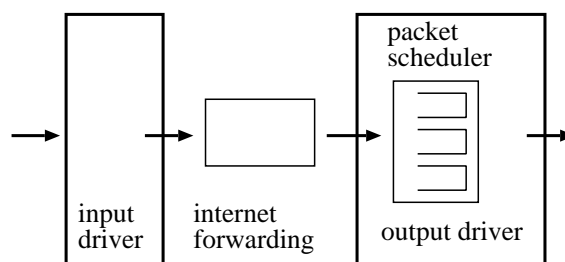
- 90年代に入ってインターネットが爆発的に普及
- リアルタイムアプリケーションの要求
  - 音声、動画通信
- ビジネス向けの高品質サービスの要求
  - インターネットを使ったビジネス
- 電話網に匹敵するQoSの要求
  - 電話網とIP網の逆転現象
    - ▶データ通信量が音声通信量を追い抜く現実
  - 従来：電話網上にIPネットワークを構築
  - 今後：IPネットワーク上に電話網を構築

## QoSとは何か

- 定量的に表現できる通信品質
  - 帯域、遅延、ジッタ、パケット損失率
- 優先制御によって実現される
  - 複数のコンポーネントの組み合わせ
    - ▷ アドミッション制御
    - ▷ シェーピング / ポリシング
    - ▷ パケットスケジューリング
    - ▷ バッファ管理

## QoSの実現

- 出力キューでの優先制御
  - 仮定
    - ▷ ルータのフォワーディングより回線の方が遅い
  - 回線の高速化による状況の変化
    - ▷ ルータ性能がボトルネックの場合、別アプローチが必要
    - ▷ しかし、広域網のエンド・エンドでは依然回線がボトルネック



## QoS技術要素

- アドミッション制御 (admission control)
  - 動的な資源配分の判断
- クラシファイア (classifier)
  - 到着パケットを対応するグループに分ける機構
- 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す
  - 規定以上の入力がないか監視
- パケット・スケジューラ (packet scheduler)
  - 各グループに応じたパケットの送出

## アドミッション制御

- セットアップ・プロトコル (signaling)
  - パス上の資源を確保する
- 資源が不足すると事前に失敗する
  - エラーが返る
- 資源の解放 (特に故障時)
- 関連技術
  - ポリシー、ルーティング、課金

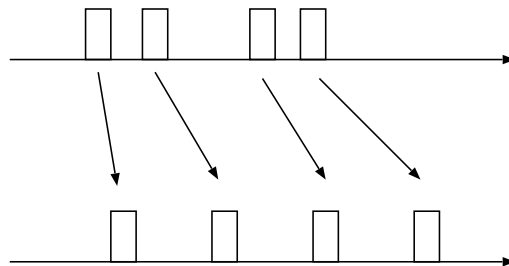
## シェーピング / ポリシング

### □シェーピング (送り側のメカニズム)

- バーストを一定のレートにならす
  - ▶ジッタを減少
  - ▶ポリシングに適合するよう調整

### □ポリシング (受け側のメカニズム)

- 規定以上の入力がないか監視
  - ▶規定以上の入力は廃棄



## クラシファイア

### □パケットをクラス分けする機構

- IPでは5つ組が使われてきた
  - ▶src\_addr, dst\_addr, src\_port, dst\_port, proto
  - ▶(ファイアウォールのパケットフィルタと同様)
- ワイルドカード
  - ▶検索コストが高い
- DiffservではTOSフィールドを利用
  - ▶クラシファイアの簡素化

## パケット・スケジューラ

### □ キューイング方式

- Priority Queueing
- WFQ (Weighted Fair Queueing)
- CBQ (Class-Based Queueing)

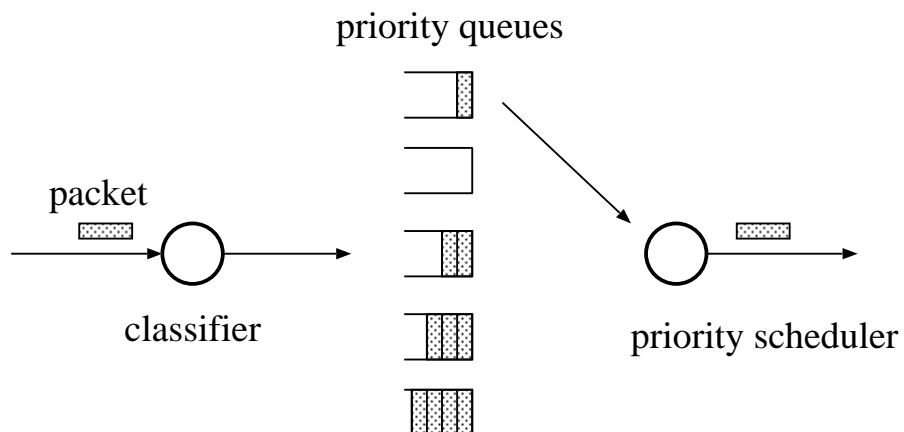
### □ バッファ管理

- Drop-Tail/Drop-Head/Drop-Random
- RED (Random Early Detection)

## Priority Queueing

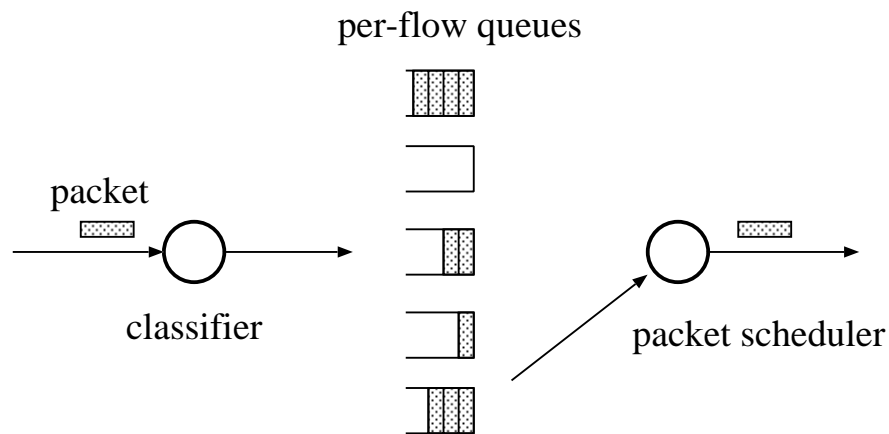
### □ 優先スケジューリング

- 単純な機構でリアルタイム性の保証
- 低優先度クラスがスタンプする可能性



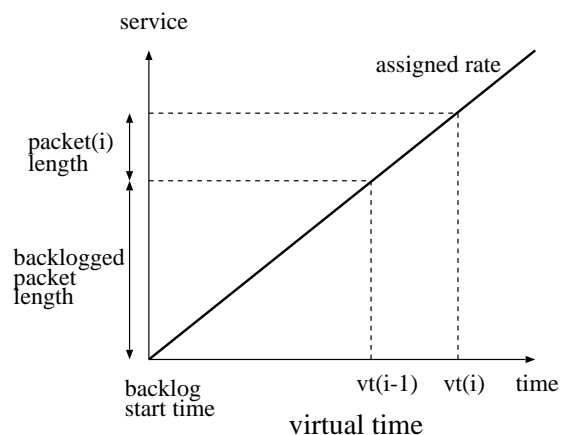
## WFQ (Weighted Fair Queueing)

- フローごとに独立したキューを割り当てる
- 重み付きラウンドロビン・スケジューリング
  - 他のフローの影響を一定以下に押えることが可能
  - フローの数だけキューが必要
    - ▷ 実装には何らかの近似法が使われる



## WFQ: より理論的な実現方法

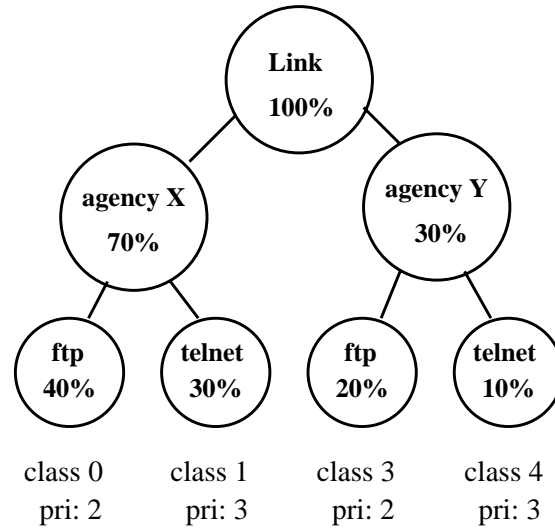
- 流体モデルの近似
  - 各仮想キューの割り当て帯域、バックログ・バイト数の状態を保持
  - パケット到着時にデッドラインを計算
  - デッドライン順にキューを管理





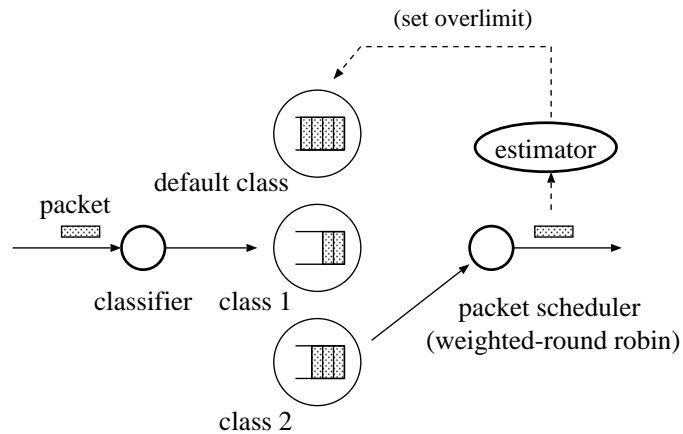
## 階層的リンクの共有

- 集約されたフローを階層的に管理
  - ▷余剰帯域の分配を制御



## CBQ (Class-Based Queueing)

- 階層的リンクの共有を実現
- 非ワークコンサービング

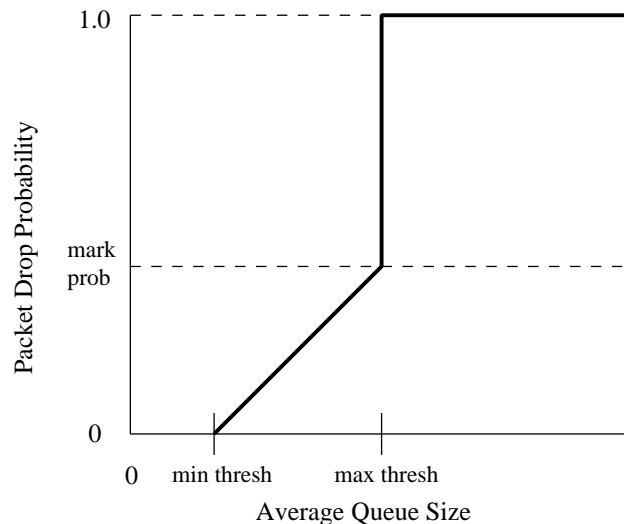


## RED (Random Early Detection)

- 平均キュー長に応じた確率でパケットを廃棄
  - 同期現象の回避
  - TCPがキューが溢れる前に流量をコントロールできる
    - ▷ キュー長を短く保つ
    - ▷ キューイング遅延を小さく保つ
  - 平均キュー長を使うことで短いバーストを許容
  - バッファ占有率に応じたフェアな廃棄
  - 廃棄のかわりにマークする
    - ▷ ECN (Explicit Congestion Notification)
- 欠点
  - パケット損失に应答しないトランスポートには無防備
    - ▷ RED Penalty-box

## REDパケット廃棄確率

- 平均キュー長が2つのスレシユホールドの間であれば
  - ▷ 確率的にパケットを廃棄



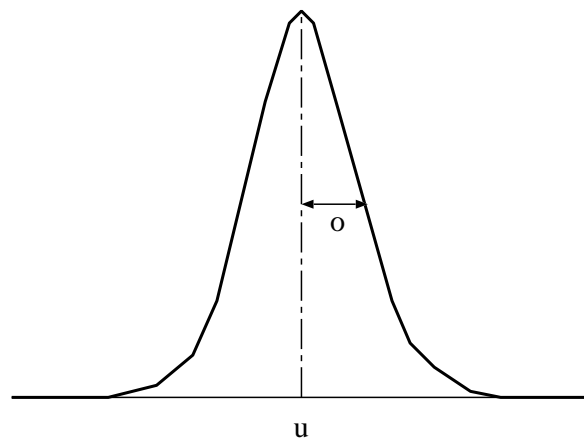
## トラフィックの理論モデル

### □ キューイング理論

- ARPA ネット時代に確立
- 統計的な解析
- 電話網（回線交換）への応用の成功、発展
  - ▶ 着呼、通話時間はポアソン分布
- データ通信では応用が困難
  - ▶ 可変長パケット
  - ▶ バースト的な通信パターン

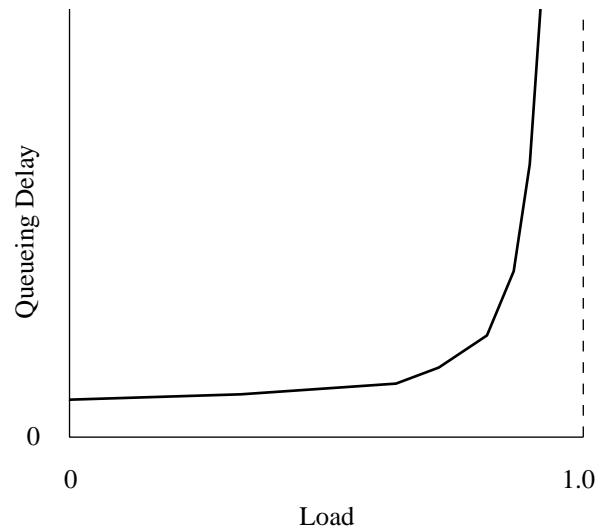
## ポアソン分布

- 数学的に取り扱い易い
  - ▶ 平均値が単一パラメーター
- 平均値を中心とした出現確率
  - ▶ 平均値から離れると急速に確率が減少



## キューイング理論

- キューイング・システムの挙動
  - ▷負荷があるポイントを越えると急速に効率悪化



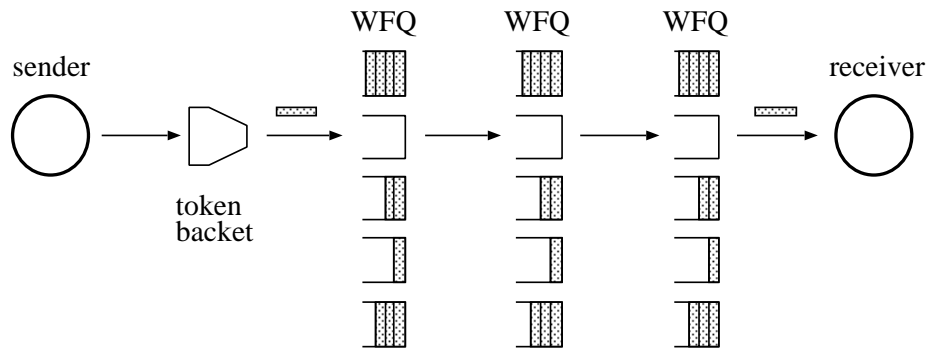
## 遅延保証の理論

- Parekhがパケット交換で遅延保証ができることを解析的に証明
  - アドミッション制御
  - 送出量の制限
  - パケットスケジューリング

## Parekh's Model

### □ トークンバケットとWFQの組み合わせ

- 最大遅延保証が可能なことを証明



## Parekhの最大遅延計算

- バースト遅延 + 自フローによる遅延 + 他フローによる遅延

$$D_i = \frac{b_i}{g_i} + \frac{(h_i - 1) l_i}{g_i} + \sum_{m=1}^h \frac{l_{max}}{r_m}$$

$D_i$  delay bound for flow  $i$

$b_i$  token bucket size for flow  $i$

$g_i$  weighted rate for flow  $i$

$h_i$  hop count for flow  $i$

$l_i$  max packet length for flow  $i$

$r_m$  bandwidth at hop  $m$

## QoS保証実現へのアプローチ

- データ通信と音声通信の統合ネットワーク
  - 電話網の拡張
    - ▷ ISDN、ATM (Broadband ISDN)
  - インターネット網の拡張
    - ▷ IntServ、DiffServ

## ATMの特徴

- 固定長セル（遅延保証に有利）
- ホップごとにVCを書き換える
- 網とユーザネットワークの分離
- ネットワークの入口で流入量を監視
  - ポリシング
- サービスクラス
  - CBR, UBR, VBR, ...
- しっかり管理されたネットワークが前提
  - 正しい設定、ポリシング
  - 簡単なプライオリティ・スケジューリング

## ATMとインターネットとの整合の問題点

- 網からのアプローチ
- バースト的なトラフィック
  - 当初の想定より大きなバッファが必要
- 文化の違い？
  - 使う前に仕様が膨れ上がる
  - 上位層、制御系がどんどん複雑になっていく
  - 遅延、エラーへのこだわり

## ラベルスイッチング技術

- ATMとインターネットを融合する技術として登場
  - ATM、フレームリレイ
- レイヤ2とレイヤ3の統合
  - ホップごとにタグを書き換える
- 2つの方式
  - トラフィック・ドリブン
  - トポロジ・ドリブン
- TE(トラフィックエンジニアリング)技術として再注目されている
  - QoSから経路管理へ

## IntServ/RSVP

- インターネットでのQoSの実現
  - 研究 89 ~ IETF 94 ~
- IntServの標準化
  - IntServ
    - ▶ QoSパラメータの指定、交換のフォーマット
  - RSVP (ReSerVation Protocol)
    - ▶ IntServモデルを実現する資源予約プロトコル
    - ▶ IntServパラメータを交換してサービスを実現

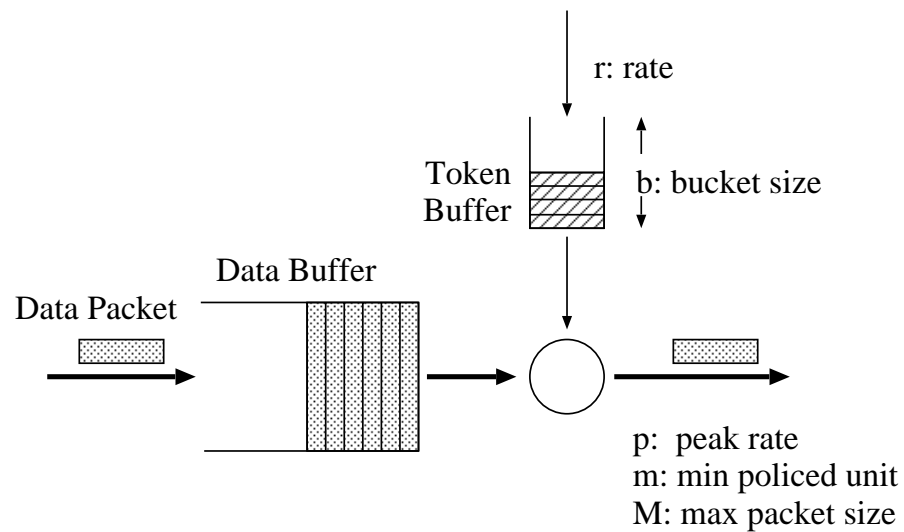
## IntServの特徴

- 2つのサービスモデル
  - Guaranteed QoS controlサービス
    - ▶ 従来の意味でのQoS保証
  - Controlled-Loadサービス
    - ▶ 低負荷のネットワークをエミュレート
    - ▶ 適応型アプリケーションを想定
    - ▶ LANでなら動作するAppをWANでも動くようにする
- トークンバケットによるパラメータの指定
  - 短いバーストを許容する
    - ▶ TCPとの親和性



## トークンバケット

- トークンバケット・パラメタ [  $r$ ,  $b$ ,  $p$ ,  $m$ ,  $M$  ]



## RSVPの特徴

- ソフトステート
- レシーバ主導
- マルチキャストのサポート
  - 予約のマージ
- 非IntServノードの透過
- ルーティングプロトコルからの分離
  - ルーティング情報をもらって利用
- 実装、運用依存部の分離
  - アドミション制御
  - トラフィック制御
  - ポリシ制御

## RSVPの問題

- スケーラビリティ
  - 中間ルータにフローごとの状態が必要
  - バックボーンでは状態数が膨大になる
- 技術主導で進み、現状とのギャップ
  - メカニズムが複雑すぎる
  - システム管理が困難
  - ビジネスモデルとの整合
    - ▷ 課金、コスト
- シグナリングは本質的に難しい
  - アドミッション制御
    - ▷ 動的な状態管理
  - 動的な資源予約
  - エラー処理
  - システム管理

## DiffServの登場の背景

- 商用ISPからの要求
  - ベストエフォートより高品位のサービスを提供する
    - ▷ すでに非標準な方式で実施されていた
- IntServ/RSVPへの疑問
- 簡単な方式ですぐにISPが使える標準の必要性
  - 簡単な仕組み
  - スケーラブルな構造
  - プロバイダのビジネスモデルにマッチすること
- TOSフィールドを再定義して利用する

## DiffServのアイデア

- 相対的なQoS (CoS: Class of Service)
- プロビジョニング
- TOSフィールドの再定義
- 柔軟なサービスモデル

## 相対的なQoS (CoS: Class of Service)

- 絶対的なQoS: 遅延、帯域等の絶対値で指定
- 相対的なQoS: 通信品質の異なるクラス
  - 実現が容易
  - インターネット通信にむいている
  - 厳密な定義や理論解析は困難
- 絶対的なQoSと相対的なQoSは混在できる

## プロビジョニング

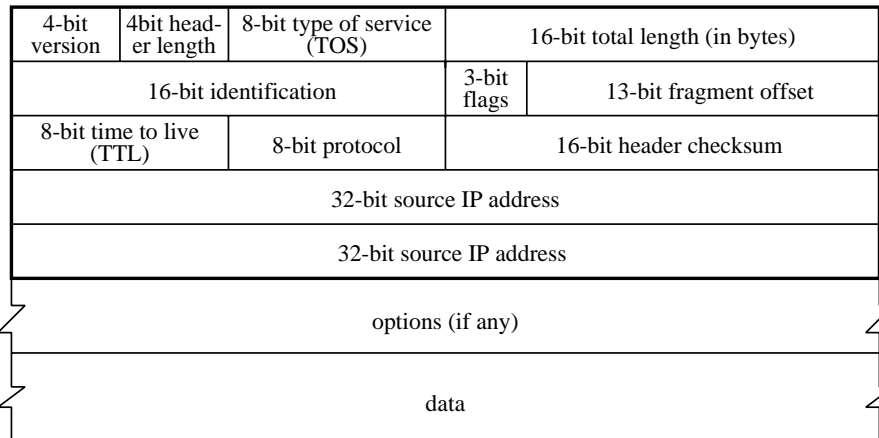
- 資源の余裕配分の重要な役割
- 制御と余裕資源配分のバランス
- バランスポイント
  - コスト効率
  - 運用の容易性
  - 将来の拡張性
- 問題点
  - 理論解析が困難

## QoSに対する考え方の変化

- 従来のQoSアプローチ
  - 理論的最悪値の積み上げ計算
    - ▷ 大きくなり過ぎる
    - ▷ 実用上あまり有効でない
  - QoS制御するかしないかの二者択一
- より現実的なQoSアプローチ
  - ルーズな制御
    - ▷ 広帯域な回線が利用可能
    - ▷ 厳密な制御の必要性が減少
  - 賢いエンドシステムを想定
    - ▷ 監視より情報のフィードバック
  - QoS制御の幅広い選択肢
    - ▷ プロビジョニングも重要な要素
    - ▷ DiffServは中間的なQoS制御

# TOSフィールドの再定義

## ○IPv4ヘッダ



# TOSフィールド (1)

## ○IP precedence (3bits)

▷0-7の優先順位

## ○TOS (4bits)

▷低遅延、広帯域、高信頼性、低コストの指定

|            |           |            |             |          |  |
|------------|-----------|------------|-------------|----------|--|
| precedence | low delay | throughput | reliability | min cost |  |
|------------|-----------|------------|-------------|----------|--|

## TOSフィールド (2)

### □例

- precedence: ルーティングプロトコルや機器の制御を優先
- telnet: 低遅延を指定

### □組織内での使用を仮定

### □現実にはあまり使われていない

- 定義があいまいで相互運用できない
- 正しい値が設定されている保証がない
- 悪用される可能性

### □DiffservはTOSフィールドの拡張とも考えられる

- 組織間での利用を考慮
  - ▷各ノードの役割を分離

## サービスモデル

### □ユーザ契約、ISP間契約

- 契約に従ったネットワークの設定
  - ▷SLA (Service Level Agreement)
  - ▷SLS (Service Level Specification)

### □コンポーネント

- 個々のコンポーネントは単にメカニズムを提供

### □サービスの実現

- ポリシ、プロビジョニング、コンポーネントの組み合わせ
- 契約を満足するサービスを実現するネットワークを構築

### □ISPの裁量の自由度の増加

- 新しいサービスメニューの提案が可能
- 工夫すればコストダウンできる
  - ▷さまざまなトレードオフ
- の力量が問われる

## Diffservの標準化 IETFでの活動

- 1997/08 Munich IETF
  - Int-serv WGが Diff-serv BOFを開いた
    - ▷ Premium Service Model
    - ▷ Drop precedence Model
    - ▷ Cisco's CoS
- 1998/03 IETF Diff-serv WG設立
  - 大学、政府、ベンダ、ISPの大物が協力
    - ▷ 急速に標準化が進んだ

## Premium Service Model

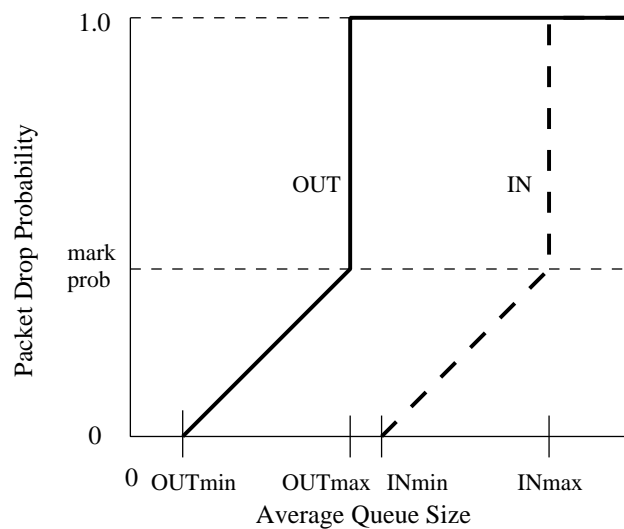
- EF PHBの原型
  - V. Jacobson (LBL) の提案
- 低遅延の保証
  - 単純な優先スケジューリング
  - 帯域割り当ては十分小さくする（例えば10%以下）
  - ポリシング
- 仮想専用線サービス
  - 専用線と同様に使える
    - ▷ 集約によりジッタは増加
  - 余剰帯域は利用可能
- 基本的にATMのCBRと同じ考え方

## Drop precedence Model

- AF PHBの原型
  - D. Clark (MIT)の提案
- RIO (RED with IN and OUT)
  - 各パケットに契約内 / 外のマークを付ける
  - 輻輳時には契約外パケットから廃棄
  - REDを拡張したRIOを提案
    - ▷ バッファ管理のみの簡単な構造
    - ▷ 順序入れ換えが起こらない
- 最低帯域保証サービス

## RIO (RED with In and Out)

- プロファイルに対してIN/OUTを判別
- IN、OUTパケットに独立したREDパラメータを与える
- 輻輳が起これるとOUTパケットから先に廃棄される





## Cisco's CoS

- Class Selector PHBの原型
  - F. Baker (Cisco)の提案
- CiscoのIP Precedenceの実装
  - クラスの相対的な差別化
  - WRED (Weighted RED): 7段階のRIO

## IETF DiffServワーキンググループ

- 相互運用に必要な最小限の取り決めの実現
  - ドメイン間、ベンダー間の相互運用
  - DSフィールドの規定 (RFC1394を更新する)
    - ▷ IPv4のTOSフィールド
    - ▷ IPv6のTraffic Classフィールド
  - 標準PHBを規定
    - ▷ アーキテクチャ
    - ▷ 運用のために必要な具体的な使用例
    - ▷ 実証実験開始のための枠組
  - やらないこと
    - ▷ 個別フローを特定するメカニズム
    - ▷ マーキングをサポートするシグナリング
    - ▷ エンド・エンド・サービスの定義
    - ▷ SLA (Service Level Agreement)

## DiffServアーキテクチャへの要求 (1)

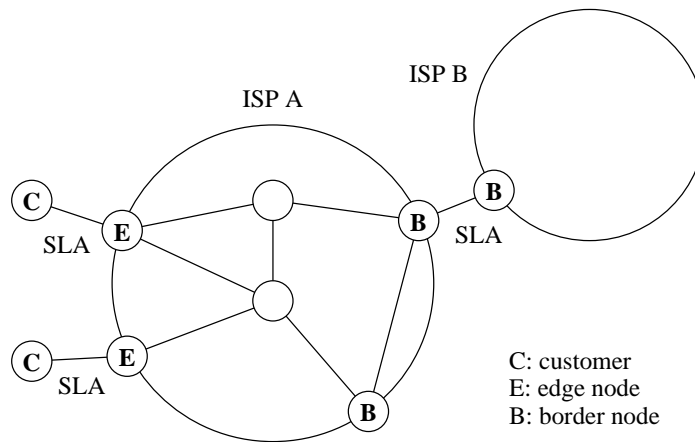
- 複数のネットワークにまたがる幅広いサービスやポリシーに利用できる
- 特定のアプリケーションに依存しない
- 既存のアプリケーションを変更する必要がない
- シグナリングに依存しない
  - staticで簡単な構成が可能

## DiffServアーキテクチャへの要求 (2)

- 簡単な構造のforwarding behaviorのみを規定
  - ルータのコストを上げない
  - 将来の高速ルータ設計の障害にならない
- コア・ネットワークでは
  - マイクロフローやユーザごとの状態を持たない
  - 集約フローのみを扱う
  - 簡単なクラシファイア (BAクラシファイア)
- DiffServ非対応機器との相互運用
- 段階的な導入が可能

## アーキテクチャモデル

- ネットワークの入口
  - クラシファイ、コンディショニング
  - DSCPを設定 (behavior aggregate)
- コア・ネットワーク
  - DSCPに対応したPHBによるフォワーディング



## DSドメイン

- インターネット・ピアリング・モデル
  - 2階層構造
  - 各DSドメインは内部の管理に責任
- 閉じたネットワーク
  - すべての流入パケットが監視可能
    - ▷ パケットがDSドメインに入る所で
      - 少数のクラスに分ける
      - 対応するDSCPを書き込む
    - ▷ DSドメイン内部で
      - DSCPのみを参照して優先制御
- DSドメイン内部のノードは
  - 共通のポリシーで管理
  - 共通のPHBのセットが設定
  - 共通の...を使用して運用

## DSサービス領域（リージョン）

- 相互運用が可能な連続したDSドメイン
  - ピアリングルールが確立している
    - ▷ 共通なDSCP、PHBの使用
    - ▷ DSCPのマッピングが設定されている

## エッジノードとコアノード

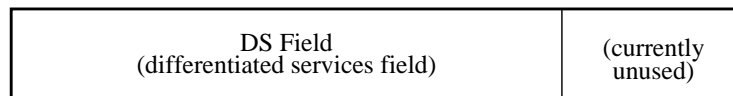
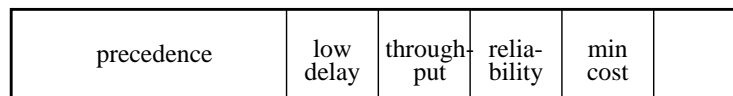
- DSドメイン
  - エッジノードとコアノードで構成される
- エッジノード
  - 個別フローの処理（機能優先）
    - ▷ ユーザごとの処理、状態保持
  - 入口処理
    - ▷ トラフィック・コンディショニング
  - 出口処理
    - ▷ ピアリング契約に応じたシェーピングや再マーキング
- コアノード
  - 集約フローのみ処理（性能優先）

## DSフィールド

- TOSフィールドを再定義
  - 内6ビットを使用
- DSCP (DiffServ Code Point)
  - 個々のDSフィールド値
- 同時使用はたかだか64個
  - DSドメインにローカルな使用
    - ▷ 拡張性
  - 少数の標準DSCP
    - ▷ 互換性

## DiffServe CodePoint

- TOSフィールドをDSフィールドに再定義
  - ▷ 2ビットはECNのために残されている
  - ▷ IPv6のTraffic Classフィールドにも適用



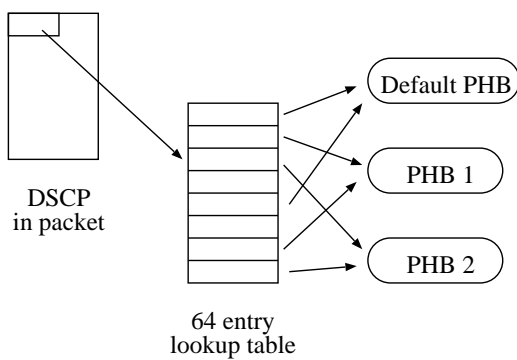
## PHB (Per-Hop Behavior)

### □ フォワーディングのメカニズムの記述

- 外部から観測できる挙動
  - ▷ 実装非依存
- 集約したフローのみを扱う
  - ▷ 限られたDSCPスペース
  - ▷ スケーラビリティ

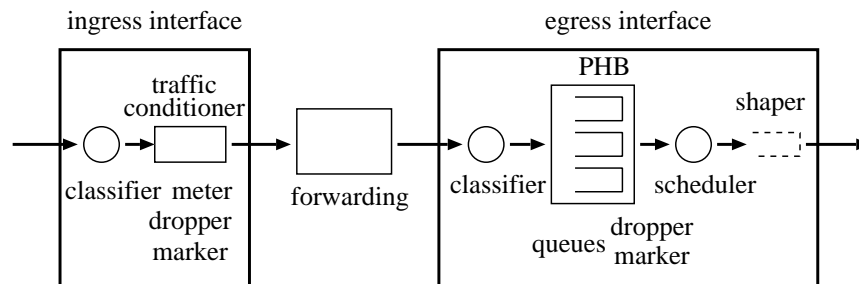
## PHBの選択

- DSCPからテーブルを引いて対応するPHBを得る
  - ▷ 64個のテーブルエントリ
  - ▷ 複数のDSCPが同じPHBにマップ可能



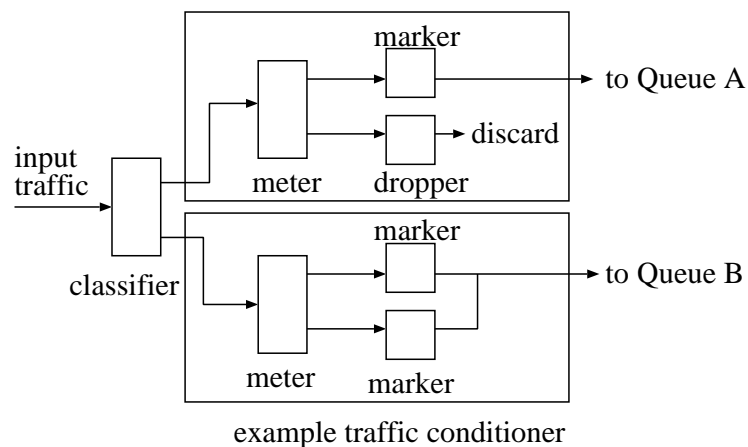
## Diffservを構成するコンポーネント

- 入力インターフェイス
  - ▶クラシファイア、トラフィック・コンディショナ
- 出力インターフェイス
  - ▶クラシファイア、キュー構造 (PHB)、シェーパ



## トラフィック・コンディショニング

- 入力インターフェイス
  - クラシファイア
  - トラフィック・コンディショナ
    - ▶メーター、アクション・エレメント(marker, dropper)



## パケット・クラシフィケーション

- パケットを対応するクラスにマップ
- パケット・フィルタ
  - マッチするパケットを検出するルール
- 2種類のクラシファイア
  - BA(Behavior Aggregate)クラシファイア
    - ▷ DSフィールドのみを参照
    - ▷ コアノードで使用される
  - MF(Multi-Field)クラシファイア
    - ▷ パケットヘッダのDSフィールド以外も参照
    - ▷ エッジノードで使用される

## トラフィック・プロファイル

- 契約に指定されたルール
  - 例 : token-bucket r, b
    - ▷ パケットごとの判定
      - in-profile: 契約値内
      - out-of-profile: 契約値外

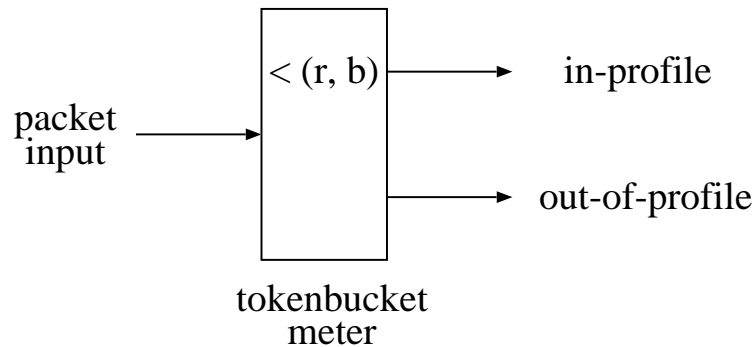


## メーター

- クラシファイアが選択したパケットが
  - トラフィック・プロファイルに適合しているか判定
  
- メータの種類
  - 平均レート
  - トークンバケット
  - color-aware/color-blind

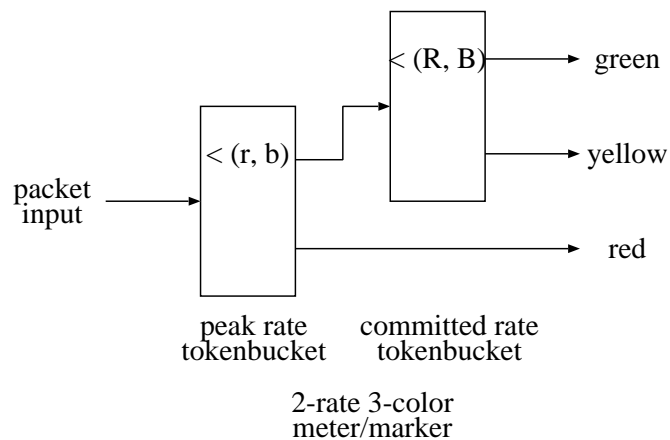
## トークンバケット・メーター

- profile: r:rate, b:depth
- output: in-profile or out-of-profile



## 2-rate 3-color meter/marker

- peak profile:  $r$ :rate,  $b$ :depth
- committed profile:  $R$ :rate,  $B$ :depth
- output: green, yellow or red



## アクション・エレメント

- マーカ
  - DSフィールドに特定のDSCPを書き込む
- シェーパ
  - パケットを遅延させてプロファイルに適合させる
  - (あまり使われない)
- ドロッパ
  - パケットを廃棄する

## PHB (Per-Hop Behavior)

- パケットフォワーディング動作の記述
  - 外部から観測できる記述（特定の実装を指さない）
  - 例
    - ▷ 最低帯域を保証する
    - ▷ ウェイトに比例した余剰帯域の分配を行なう
- PHBグループ
  - セットでひとつのクラスを構成する複数のPHB
    - ▷ AFのDrop precedence
  - 同一の属性を持つ独立した複数のPHB
    - ▷ AFのClass

## PHBの実装

- パケットスケジューリング
- バッファ管理
- 各集約フローは到着順序を守って送出する必要
  - トランスポートのパフォーマンス

## 標準PHB

- Default PHB
- Class Selector PHBグループ
- EF PHB
- AF PHBグループ

## コードポイントの割り当て

- コード空間
  - xxxxx0: Standard PHBs ( 3 2 個 )
  - xxxx11: Experimental/Local Use ( 1 6 個 )
  - xxxx01: Experimental/Local Use\* ( 1 6 個 )
- スタンダード空間
  - 000000: Default PHB ( 1 個 )
  - xxx000: Class Selector PHBs ( 7 個 )
  - ccdd0: Assured Forwarding PHBs ( 1 2 個 )
    - ▷ccc: class {1,2,3,4} dd: drop prec {1,2,3}
  - 101110: Expedited Forwarding PHB ( 1 個 )

## Default PHB

- DSCP = 000000
- ベストエフォート
  - スターブしない必要
    - ▷ 最低限の資源割り当てを保証する

## Class Selector PHBグループ

- IP Precedence互換の優先度指定
- Precedenceが大きい
  - 遅延が小さく、パケット損失も少ない
- 実装例
  - WFQ, WRR, CBQ

## EF (Expedited Forwarding) PHB

- 低損失、低遅延、低ジッタサービス
- 2つの構成要素
  - PHB
    - ▷ 最低送出レートが保証される
  - Conditioning
    - ▷ すべてのノードで最大流入量を
      - 保証される最低送出レート以下にする
    - ▷ 厳密なポリシングの必要
    - ▷ 規定値を越える流入パケットは捨てる
- 実装例
  - PQ, WFQ, WRR, CBQ

## AF (Assured Forwarding) PHBグループ

- 4つの独立したクラス
  - (例: ファース、ビジネス、エコノミー、マルチキャスト)
- 各クラスに3つのドロップPrecedence
  - 3レベルあればTCPをUDPから守れる
  - Precedenceに応じた確率的な廃棄
  - クラス内のパケット順序入れ換えの禁止

## PDB (Per-Domain Behavior)

### □ DSドメインのエッジ・エッジでの挙動の記述

#### ○標準化の第2フェーズ

▷PHBはノードの挙動を記述

▷PDBは抽象化をDSドメインのエッジ間に広げる

### □PDBパラメタの例

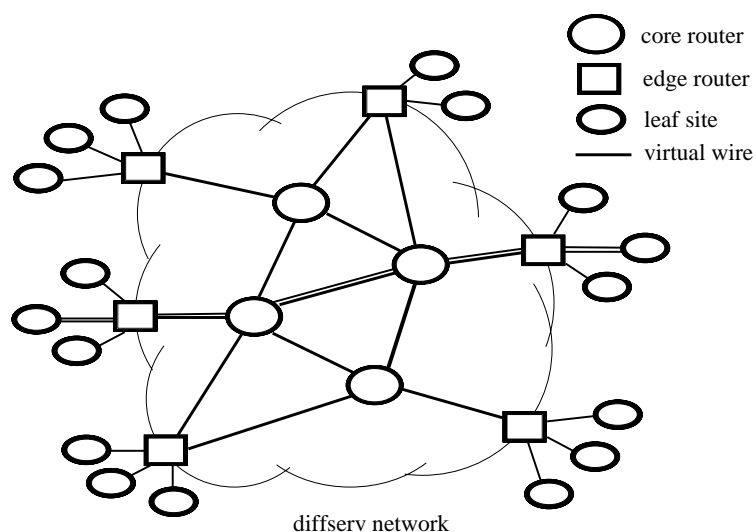
○最大ホップ数、エッジの数

○最低帯域、最大遅延、バッファ容量

### □Virtual Wire PDB

○EF PHBを使った仮想専用線サービスPDB

## Virtual Wire PDB



## サービス構築例

- EFを使った仮想専用線
- AFを使った最低帯域保証
  
- (あくまでも理解を助けるための例です)

## EFを使った仮想専用線サービスの例 (1)

- 顧客契約
  - connection: from <src> to <dst>
  - profile: <r>:rate, <b>:tokenbucket depth
    - ▷ in-profile:
      - delay: less than <msec>
      - packet loss: less than <%>
    - ▷ out-of-profile:
      - discard
- エッジの設定
  - classifier: <src><dst> to customer's TC
  - TC (traffic conditioner):
    - ▷ tokenbucket meter: <r>,<b>
      - in-profile: mark <EF DSCP>
      - out-of-profile: drop



## EFを使った仮想専用線サービスの例 (2)

### □ Provisioningの例

- DSドメイン内のすべてのルータで
  - ▷ EF用に容量の10%をリザーブ
- サービスの販売
  - ▷ <src> to <dst>のパス上のすべてのノードで
    - EFの合計がリザーブ値を越えないように販売
- 保証できるパスの遅延を計算して顧客契約に反映

## AFを使った最低帯域保証サービスの例 (1)

### □ 顧客契約

- connection: from <src>
- committed profile: <R>:rate, <B>:tokenbucket depth
- peak profile: <r>:rate, <b>:tokenbucket depth
  - ▷ in-committed-profile: packet loss less than <%>
  - ▷ in-peak-profile: packet loss less than <%>
  - ▷ out-of-profile: best effort

### □ エッジの設定

- classifier: <src> to customer's TC
- TC (traffic conditioner):
  - ▷ trTCM: <R>,<B>,<r>,<b>
    - green: mark <AF11 DSCP>
    - yellow: mark <AF12 DSCP>
    - red: mark <AF13 DSCP>

## AFを使った最低帯域保証サービス (2)

### □ Provisioningの例

- DSドメイン内のすべてのルータで
  - ▷ AF1用に容量の50%を割り当てる
- サービスの販売
  - ▷ 実績ベースでgreenパケットが廃棄されないように販売
- トラフィック集中が発生する可能性
  - ▷ 実績ベース以外のルール必要性

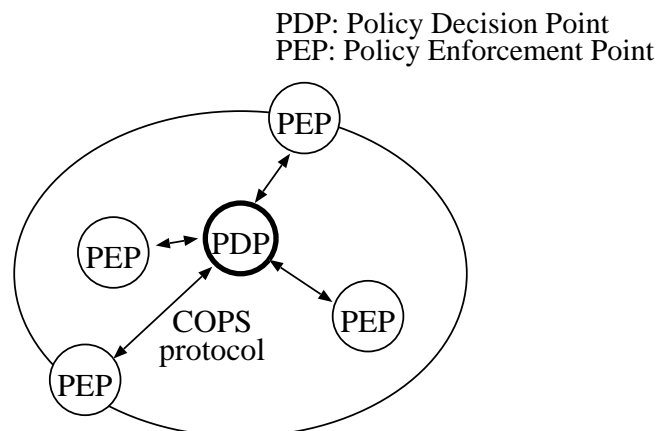
## ポリシーサーバ

### □ モデル

- PEP トラフィック制御をするルータ
- PDP ポリシー管理サーバー

### □ プロトコルの標準化 (COPS, PIB)

- まだ一般化された実装がない



## ポリシー制御プロトコル

- COPS (Common Open Policy Service)
  - もともとRSVPのために提案、後に一般化
  - クライアント/サーバ・モデルの簡単なプロトコル
    - ▷ アドミッション制御のためのRequest/Decision
    - ▷ 遠隔システム設定 (COPS-PR)
  - トランスポートにはTCPを利用
  - オープンなオブジェクト(ポリシールール)
    - ▷ PIB (Policy Information Base)
      - RSVP PIB, diffserv PIB
  - セキュリティサポート
    - ▷ Integrity object or IPSec

## COPS-PR

- COPS-PR (COPS Usage for Policy Provisioning)
  - PEPの起動時
    - ▷ COPSコネクションをオープン
    - ▷ Configuration Request (PEP to PDP)
      - ハードウェア/ソフトウェア、パラメータのレポート
    - ▷ Decision (PDP to PEP)
      - Policyのダウンロード
      - PDPはポリシーをローカルメカニズムにマップして設定
  - PDPからの変更
    - ▷ unsolicited decision (install/update/delete)
  - PEPからの変更
    - ▷ unsolicited request

## DiffServの課題と動向

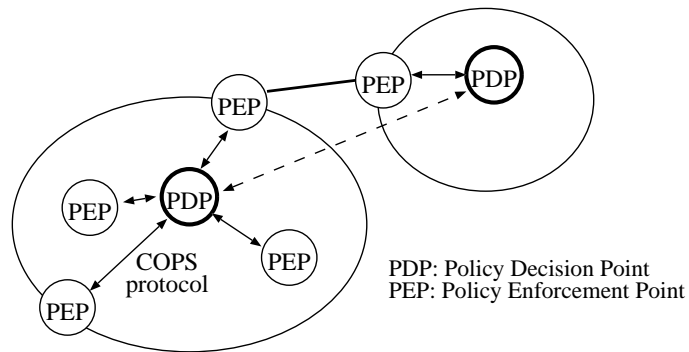
- DSドメイン境界での再マーキング
- 受信者ベースのサービス
- マルチキャスト
- Bandwidth Broker
- RSVP over DiffServ
- DiffServ over MPLS

## DSドメイン境界での再マーキング

- 等価なPHBへのマッピングが可能か？
- 契約量を越えた時
  - 別のマッピングが可能か？

## Bandwidth Broker

- 動的な資源配分のモデル
- 異なるDSドメインのPDP同士がピアリング契約を交渉



## 受信者ベースのサービス

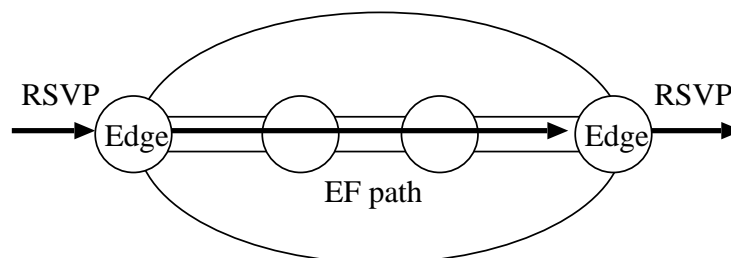
- DiffServは送信者の契約に応じた扱い
  - 受信者の契約が反映されない
  - 一般ユーザに利益が少ない
- 上位層で受信者契約を反映する仕組みが必要
  - 一種のシグナリング?
  - アイデアだけで実体はない

## マルチキャスト

- ユニキャストと混在させる問題
  - より多くの資源を消費する可能性
- グループはダイナミックなのでProvisioningが困難
- 境界問題
  - ピアドメインとのマッピング

## RSVP over DiffServ

- コア・ネットワークをDiffServでバイパス
  - エッジでのみRSVPの処理
  - RSVPのスケラビリティの問題を解決



## まとめ

### □ ネットワーク・サービス

- 従来は機器ベンダーが機器機能として提供
- 今後はISPの運用技術として実現される

### □ DiffServ

- 枠組は固まってきた
- 対応製品の登場

### □ DiffServの運用

- ネットワーク管理ツールの必要
- 現時点ではスタティックな設定
  - ▷小規模ネットワークなら十分利用可能
  - ▷大規模ネットワーク
    - 運用については分かっていない
    - 実証実験をとおした経験の蓄積が必要

### <関連リンク>

IETF: <http://www.ietf.org/>

IETF diffserv WG: <http://www.ietf.org/html.charters/diffserv-charter.html>

IETF issll WG: <http://www.ietf.org/html.charters/issll-charter.html>

IETF rap WG: <http://www.ietf.org/html.charters/rap-charter.html>

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

### <関連書籍>

Internet Performance Survival Guide. G. Huston.  
Wiley. ISBN 0-471-37808-9. 2000.

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.  
Addison-Wesley, ISBN 0-201-56333-9. 1993.