

# インターネットインフラ特論

## 11. QoS(遅延、帯域)の保証

太田昌孝

[mohta@necom830.hpcl.titech.ac.jp](mailto:mohta@necom830.hpcl.titech.ac.jp)

<ftp://chacha.hpcl.titech.ac.jp/infra11j.ppt>

# QoS (Quality of Service) の保証とは？

- 通信品質の保証
  - もっぱら帯域と遅延と遅延のゆらぎ(ジッタ)
- ネットワーク中に必用、エンドでは不可能
  - とどかないパケットをでっちあげられない
- 帯域と遅延は密接に関連
  - キューイング理論によれば、十分な帯域があれば遅延は小さい
  - 無限に遅延が許されれば、帯域など不要

# ネットワーク

- 物流網
  - 郵便、宅配便、コンビニ
- 情報通信網
  - 出版網（書籍、新聞、レコード（CD）、映画）
  - 金融網
  - 電話網
  - 放送網
  - インターネット

# 出版網

- 同じ情報を大量に配布
- 情報流通は遅くていい
- 著作権法による保護
- いまのインターネットの好餌
  - 壊滅寸前

# 金融網

- お金のやりとりを管理
- 物流網でもあるが、今や、情報通信網としての面がはるかに大きい
- セキュリティー！！！！

# 電話網

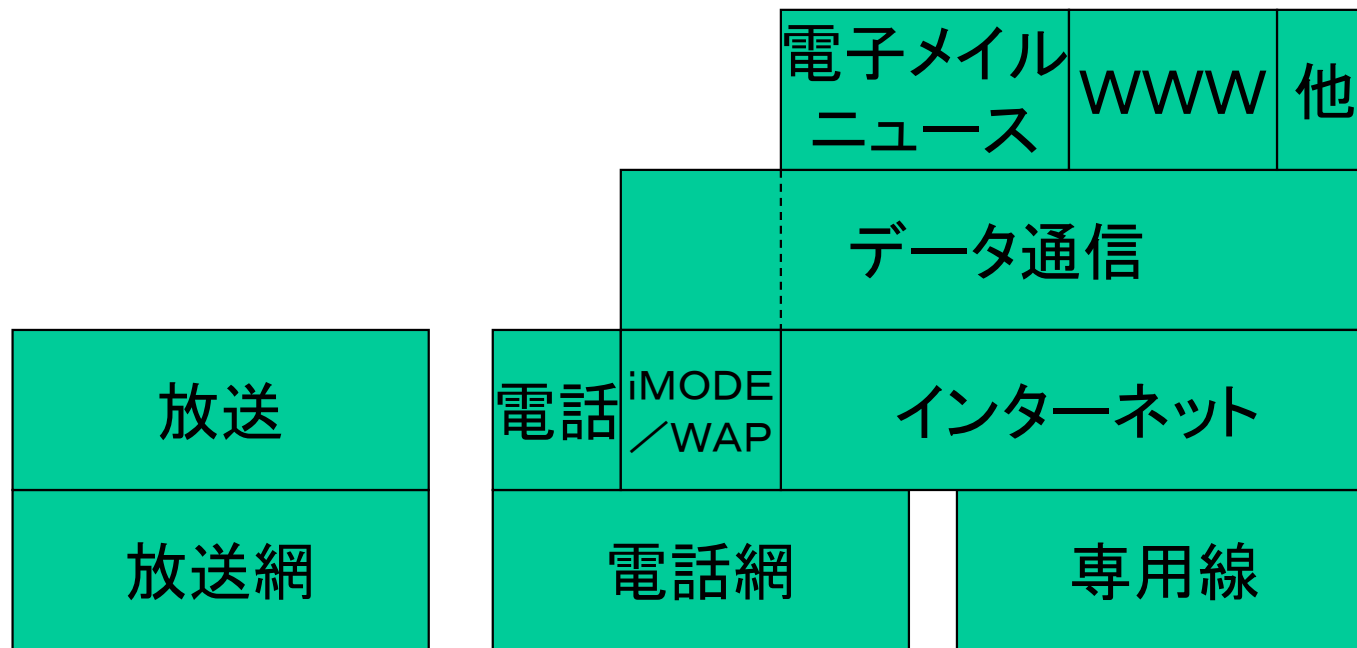
- 音声を実時間で伝送する網
  - 音声伝送の帯域を確保
  - 音声伝送の遅延を最小化(保証)
- 専用線事業も
  - あくまで音声伝送事業が主
- 遅くて高い
- 電電公社として保護、電気通信事業法で開放

# 放送網

- 音声、画像を実時間で多数に伝送する網
  - 伝送帯域を確保
  - 遅延を最小化
- 電波による広域一対多通信
  - ブロードキャスト／マルチキャスト
- 放送法による保護

|     |     |       |
|-----|-----|-------|
| 放送  | 電話  | データ通信 |
| 放送網 | 電話網 | 専用線   |

かつてのネットワーク



現在のネットワーク

|          |    |               |     |   |
|----------|----|---------------|-----|---|
| 放送       | 電話 | 電子メール<br>ニュース | WWW | 他 |
| ストリーミング  |    | データ通信(バッチ)    |     |   |
| インターネット  |    |               |     |   |
| 専用線(含無線) |    |               |     |   |

今後のネットワーク

# QoSの例

- 電話
  - 帯域64Kbps、遅延0.1秒以下
- CD再生
  - 帯域1.5Mbps、遅延1秒以下
- テレビ放送
  - 帯域6Mbps、遅延1秒以下

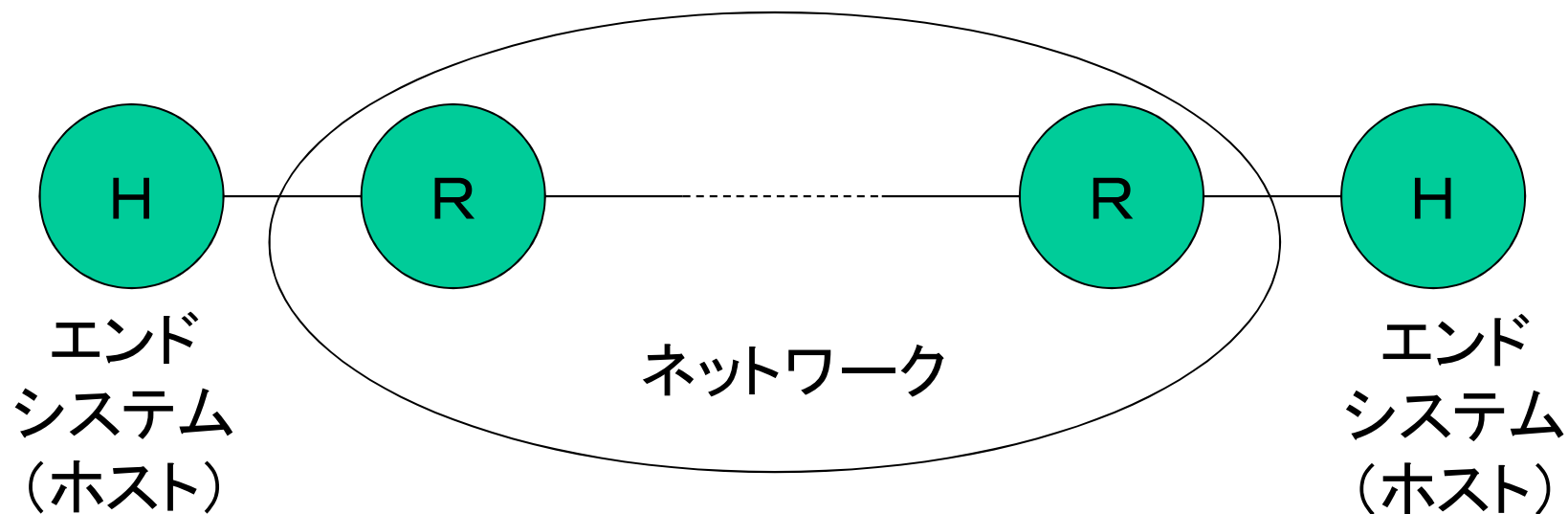
# QoS保証の単位

- 同一ホスト間でもQoS要求は違う
  - ファイル転送
    - ベストエフォート
  - 音声対話
    - 遅延を小さく
  - 画像伝送
    - 帯域を広く
- ポート番号により、通信を区別

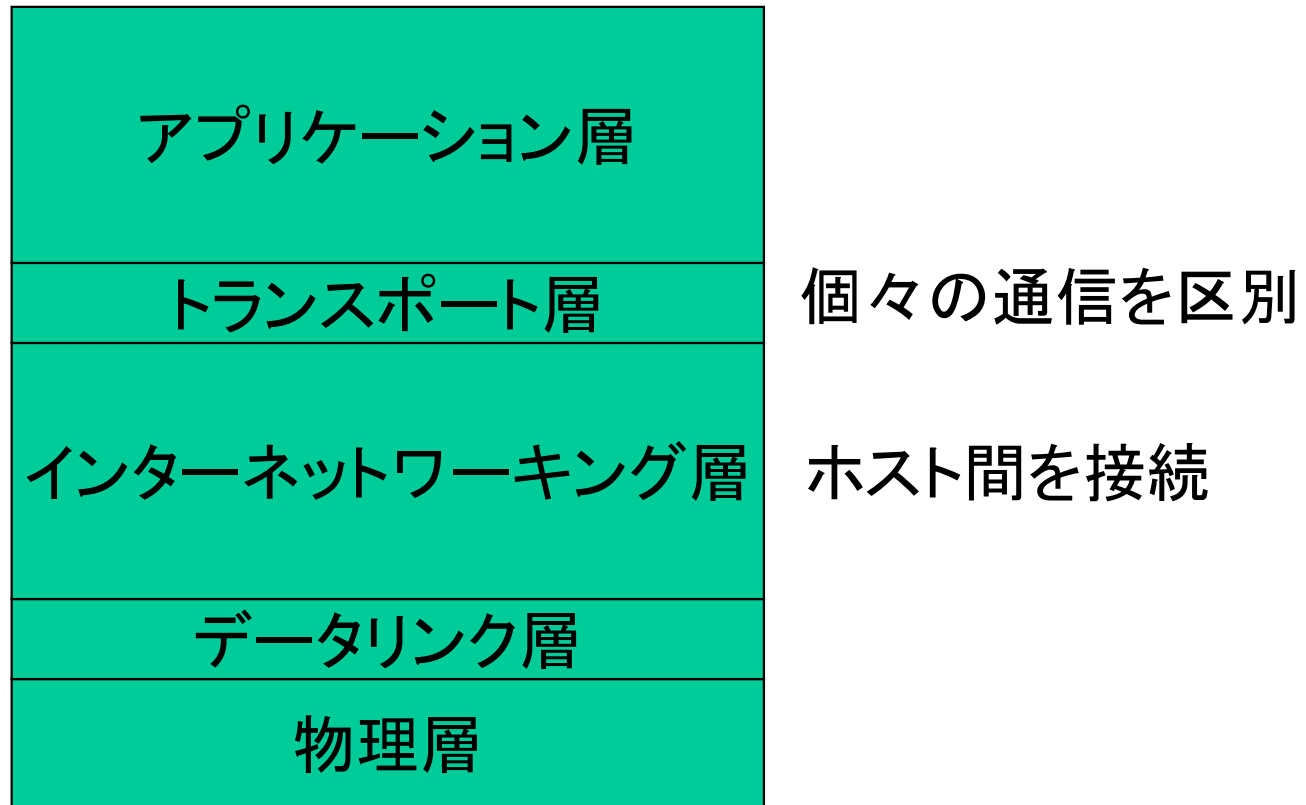
|           |
|-----------|
| アプリケーション層 |
| トランスポート層  |
| ネットワーク層   |
| データリンク層   |
| 物理層       |

|         |
|---------|
| ネットワーク層 |
| データリンク層 |
| 物理層     |

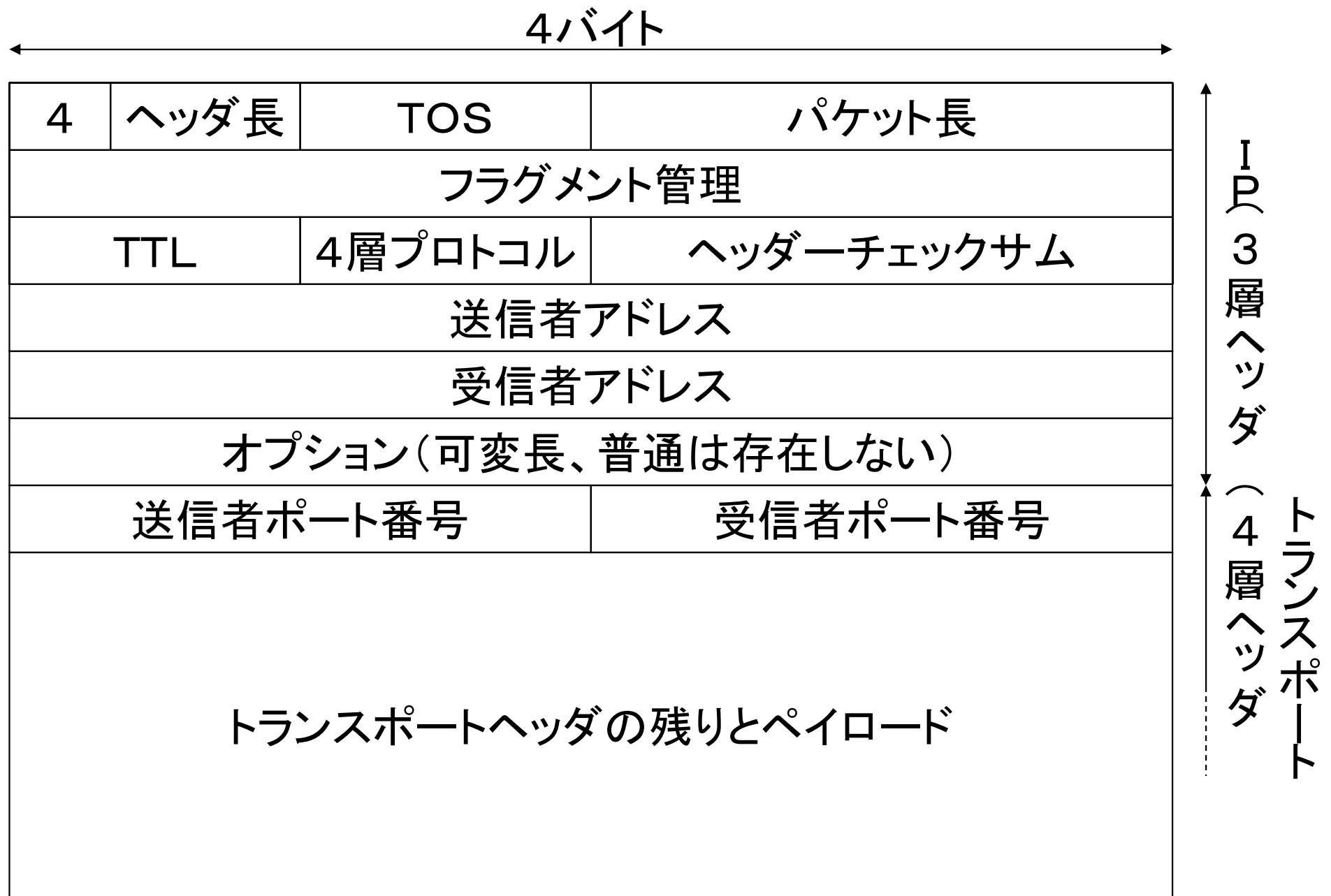
|           |
|-----------|
| アプリケーション層 |
| トランスポート層  |
| ネットワーク層   |
| データリンク層   |
| 物理層       |



ベストエフォートインターネット



インターネットのレイヤリング構造

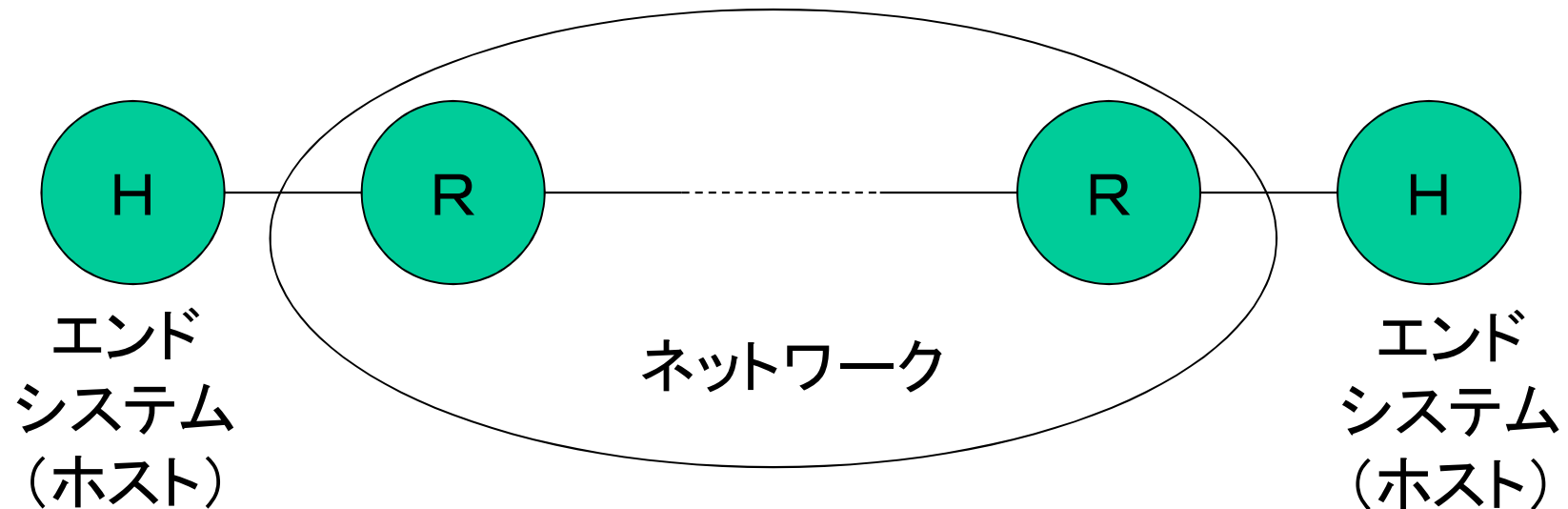


IPv4パケットフォーマット

|           |
|-----------|
| アプリケーション層 |
| トランスポート層  |
| ネットワーク層   |
| データリンク層   |
| 物理層       |

|          |
|----------|
| トランスポート層 |
| ネットワーク層  |
| データリンク層  |
| 物理層      |

|           |
|-----------|
| アプリケーション層 |
| トランスポート層  |
| ネットワーク層   |
| データリンク層   |
| 物理層       |



QoS保証付きインターネット

# QoS保証のために

- ジッターの吸収
  - RTPのタイムスタンプによりエンドで吸収
- 各ルータ上での遅延、帯域の保証
  - ネットワークの機能として必要
  - 統計多重(100%の保証は必要ない)が重要
- QoSルーティング
  - 加法的制約が二つ以上だとNP完全問題
  - 経路の安定化が難しい

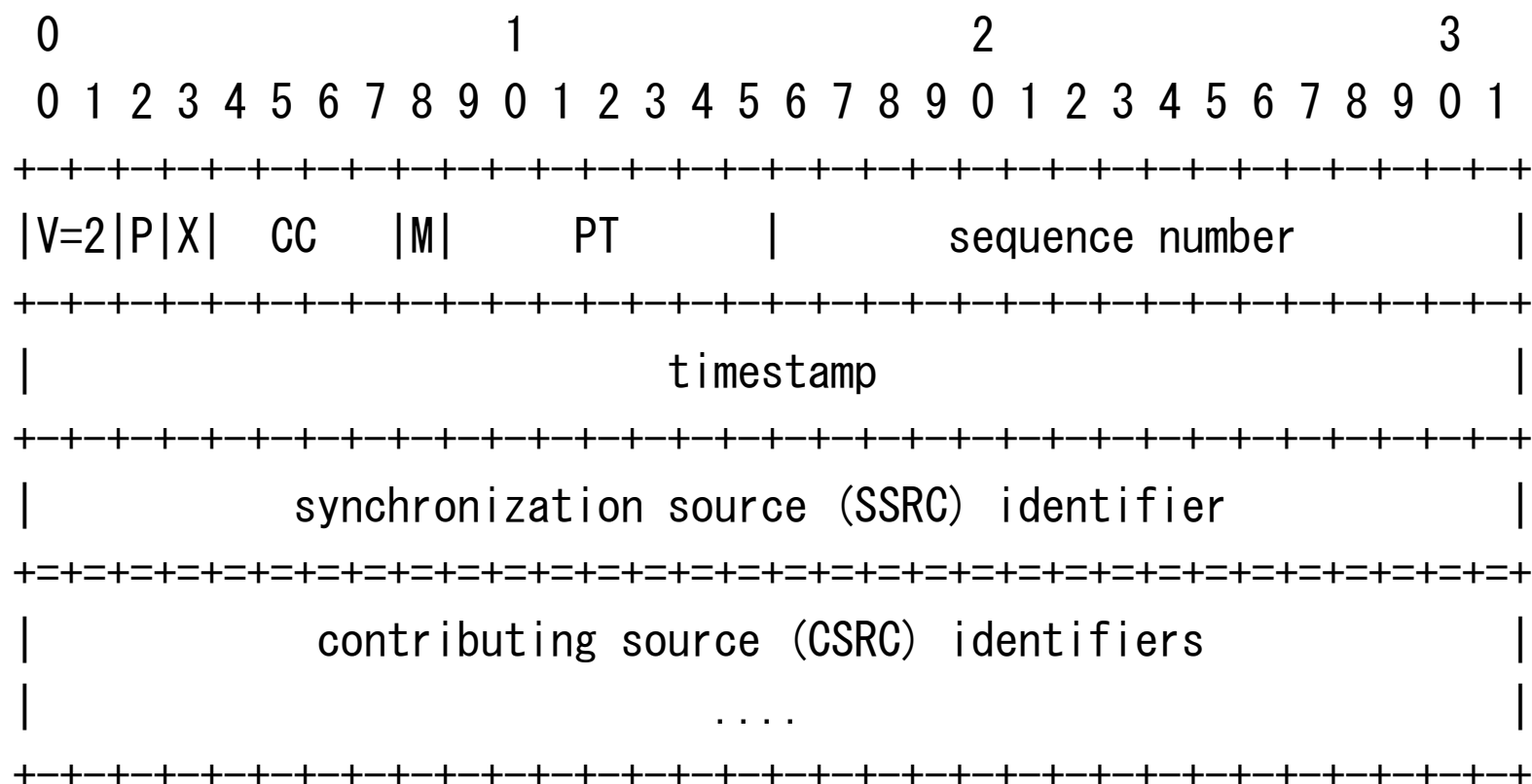
# 遅延とジッタ

- $\text{ジッタ} = (\text{最大遅延}) - (\text{最小遅延})$
- スムーズな再生には当初ジッタ分、最大その2倍のバッファが必用
  - 最小遅延の直後に最大遅延でデータが到着しても再生するものがない
  - 最大遅延の直後に最小遅延でデータが到着してもためておける
- RTPによるスムーズな再生

# RTP (Real-Time Transport Protocol) (RFC1899)

- ホスト間の同期のためのトランスポート／アプリケーション層のプロトコル
  - ネットワーク中の遅延やジッターが減るわけではない
  - ペイロードタイプ(PT)によるメディアの識別
  - シーケンス番号による順序の回復
  - タイムスタンプによるクロック同期
- 各種のメディアに対応(RFC1890他)

# RTPのヘッダー



# RTPとエンドツーエンド原理

- RTPはアプリケーションゲートウェイを仮定
  - トランスレータ
    - メディアを変換
  - ミキサー
    - 複数の音声(?)を混合、会議では当然？
  - エンドツーエンド原理は重要視されない
    - IPアドレスでなく、SSRCでソースを識別
- クロック同期はエンドツーエンド
  - ネットワークにグローバルクロックは不要

# バッファとクロックのずれ

- クロックのずれはスムーズな再生を妨害
  - 送信側のクロックが受信側より遅いと
    - 受信側のバッファはいずれ空に
  - 送信側のクロックが受信側より速いと
    - 受信側のバッファはいずれあふれる
- バッファをジッタの2倍用意して、再生速度を調整（送受クロックの相対誤差以上）
  - バッファが半分以下なら再生速度を遅めに
  - バッファが半分以上なら再生速度を速めに

# ジッタによる遅延

- 実遅延 = (ネットワーク最低遅延) + (バッファ遅延)
- バッファ遅延 =  $2 * (\text{ジッタ})$

# 最低遅延の原因

- 伝送経路での伝送遅延
  - 信号伝搬速度による限界
  - 経路一定なら一定、経路により変化
  - 動的経路変更があるなら最低遅延はほぼ0
    - 正確には(両端の距離)／(光速度)
- パケット入出力時間
  - パケット長／伝送速度
  - 各ルータ／ブリッジ等で発生

# ジッタの原因

- 動的経路変更により発生
  - $(道のり) / (信号伝搬速度)$ 
    - ほとんど無視できる程度
- キューイングにより発生
  - キューイングを細かく制御することによりジッタを制御し、QoS(最大遅延)を保証
  - ジッターを小さくするキューからは、まめにデータを取りだす

# QoS保証の困難

- キューイング
  - よく知られたQoS保証のための方式であるフェアキューイングは
    - スケールしない
    - 遅延が大きい(パケット長／通信速度に比例)
- 経路制御
  - 経路が不安定に
  - 階層的経路情報の縮約？

# ATM・PNNIの対応

- キューイング
  - フェアキューイングは
    - スケールしない→無視
    - 保証できる遅延が大きい→小さなセル(53B)
- 経路制御
  - 経路が不安定に→経路設定やり直しなし
  - 階層的経路情報の縮約？→実装の自由、不正確ならクランクバックで

# IETF／RSVPの対応

- キューイング
  - フェアキューイングは
    - スケールしない→CoSであきらめる
    - 保証できる遅延が大きい→無視
- 経路制御→経路制御無し？
  - 経路が不安定に→PCEによる集中計算
  - 階層的経路情報の縮約？

# 本講義での対応

- キューイング
  - フェアキューイングは→使わず統計性導入
    - スケールする
    - 保証できる遅延が小さい(パケット長／リンク速度に比例)
- 経路制御→PATHメッセージの活用
  - 経路が安定に
  - 階層的経路情報の縮約も可能に

# 帯域制御

- 通信によりキューをわけける
- 広い帯域が必用な通信をもつキューからは、まめにパケットをとりだす

# ポリシング (policing)

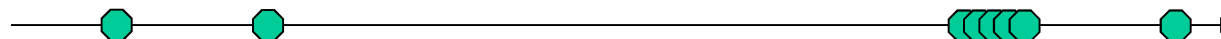
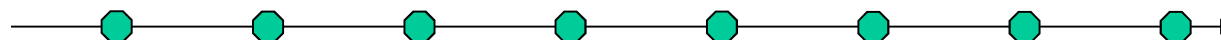
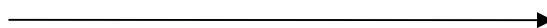
- キューからのとりだし頻度は、事前に予約した帯域による
- 実際にはより多くの帯域を利用されると
  - キューがあふれて(そのキューを共有している他の通信の)データがおちる
- 個々の通信の帯域を監視
  - 違反したらパケットを落とす(優先度を下げる)
  - ある程度のジッタを見込んだ監視が必用
    - トラフィックモデル

# トラフィックモデル

- トークンバケットモデル
  - 完全に周期的なパケット列をもとに
  - それぞれの位置から一定以下のずれを許容
  - ポリシングのアルゴリズムが存在
- ポワソンモデル
  - ポワソンもしくはポワソンよりバースト性小
  - ある意味、無限のずれを許容
  - 統計的なポリシング

# トークンバケットモデルでのずれ

各点の前方へのずれの許容範囲



# キューの制御

- 複数のキューをもつ
- 通信の種類（アドレスとポート番号で区別）により、キューをわける
- 次にデータを取りだすキューを、適切なアルゴリズムで選択
  - フェアキューイング？
  - PPQ！

# フェアキューイング

- 通信毎にキューをもち、優先度制御を行う
  - 優先度制御にはソートが必用
    - 通信の数が増えると、極めて遅い
      - 各パケットの処理時間は $\log(\text{QoS保証通信の数})$
- トークンバケットモデルに基づく通信に対して、ネットワークでの遅延のオーダーは
$$(\text{最大パケット長}) / (\text{通信帯域}) * (\text{段数}) + (\text{最大パケット長}) / (\text{リンク速度}) * (\text{段数})^2$$
- 遅延を100%保証、帯域を100%予約

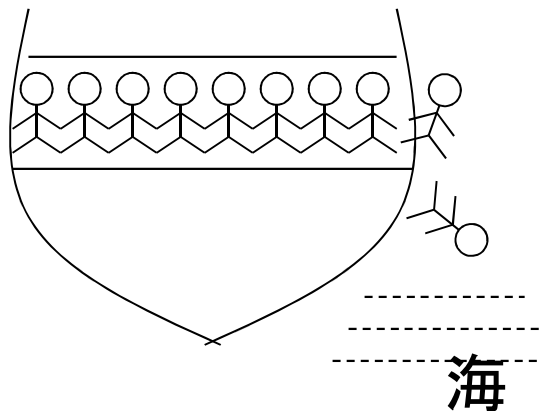
# フェアキューイングの問題点

- 保証できる遅延が大きすぎる
  - 最大パケット長1500Bで64Kbpsの通信で
    - 1段あたり0.2秒の遅延
- 処理が重すぎ、幹線では使えない
  - 幹線では、QoS保証通信が多く、処理が遅い
  - 各段でのバッファメモリも多量に必用
- パケットが100%到着するわけではない
  - 到着すれば遅延は100%保証されているが

# フェアキューイングを 利用するために？

- 最大パケット長を短くする
  - ATM？
  - 所詮気休め
- 中間でのシェイピング
  - (段数)<sup>2</sup>の項がなくせるが、処理は重い
- 複数の通信をまとめてQoS保証する
  - トラフィッククラスに対する保証 (DiffServe)
    - Class of Service (CoS)保証

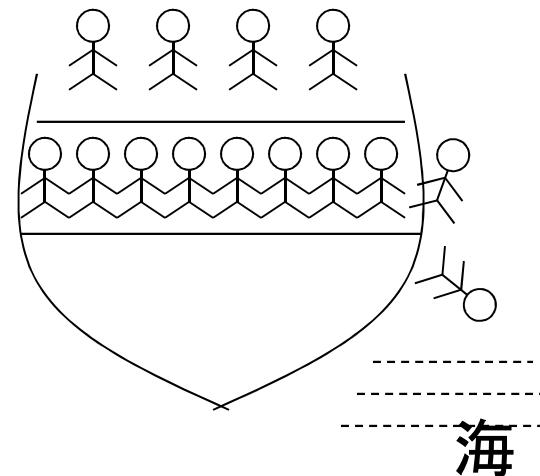
二等船室



a) これまでのインターネット

二等船室

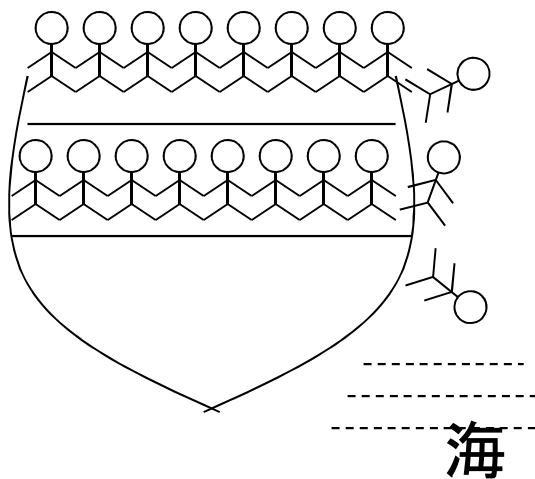
三等船室



b) DiffServeとインターネット

二等船室

三等船室



c) DiffServeとインターネット(混雑時)

DiffServeのクラスの意味するもの

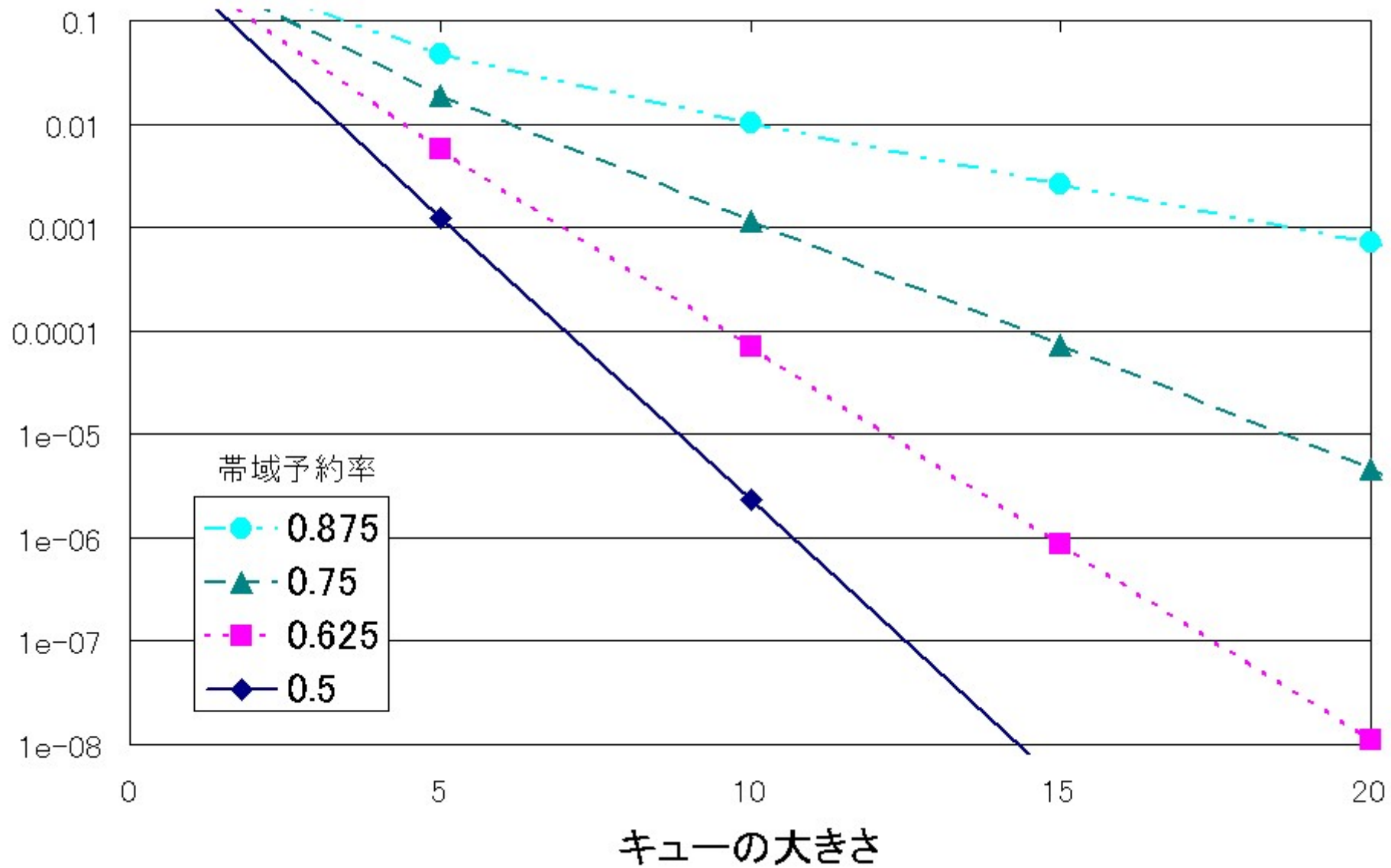
# PPQ (Policed Priority Queueing)

- 統計的モデル(ポワソン分布)を利用
  - 各通信は統計的に独立と仮定
- 各段で遅延を99.9999%程度保証
  - $(\text{最大パケット長}) / (\text{リンク速度}) * 20$ 程度
- キューは2つ(BE用、QoS用)だけ
  - QoS用キューが絶対的に優先
- ポリシングは各通信個別に

# PPQの利害得失

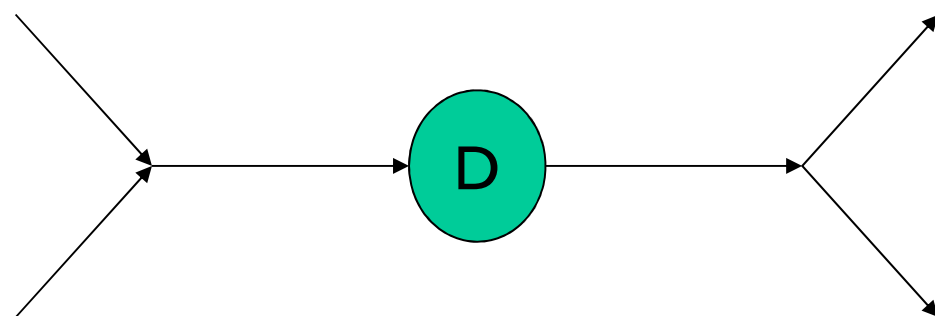
- 通信速度によらず遅延が極めて小さい
  - 最大パケット長1500B、100Mbpsのリンクで3ミリ秒程度
- パケット処理は高速、バッファも20パケット程度
- 帯域は80%程度まで予約(残りはBEで利用)
- たまにパケットが遅れる(落ちる)
  - エラーで落ちたものと思えばよい
- 複数の通信が完全に同期すると破綻
  - わざとずらせばよい

パケット落ちの確率



# 「バースト性はポワソンかそれ以下」の定義は？？？

- 望まれる性質
  - 以下の場合に「バースト性はポワソンかそれ以下」の性質が保存されること
    - 独立なフローの合流
    - $G$ が「バースト性はポワソンかそれ以下」の時の $G/D/1$ （ルータ通過）の出力
    - 独立なフローの合流後の $G/D/1$ の出力を元のフローに分離した場合



合流

ルータを通過

分岐

# 「バースト性はポワソンかそれ以下」の定義例

- 「発生率を $\lambda$ とするとき、長さ $L$ の区間の発生数の分散が $L\lambda$ 以下」と定義すると
  - 独立なフローが合流した場合「バースト性はポワソンかそれ以下」の性質は保存される
    - 合流した後の平均と分散は、各フローの平均と分散の和なので、分散の和は平均の和以下
- 平均のまわりの高次のモーメントでも
  - 5次までは性質は保存されることを確認

# 「バースト性はポワソンかそれ以下」の例

- ポワソン過程
  - 長さ $L$ の区間の発生数の分散は $L\lambda$
- 間隔1の周期事象
  - 長さ $L$ の区間の発生数は、先頭位置により
    - 確率 $(L - \text{floor}(L))$ で $\text{floor}(L) + 1$
    - それ以外は $\text{floor}(L)$
  - 長さ $L$ の区間の発生数の平均は $L$ 、分散は
    - $L - \text{floor}(L) - (L - \text{floor}(L))^2 \leq L$
    - $= L - L^2 \quad (L < 1)$

# G/D/1での安定性？

- 未解決
- 関連する課題
  - M/M/1の出力はポワソン
  - G/M/1の出力は「バースト性はポワソンかそれ以下」か？

# フローの分離後？

- 未解決

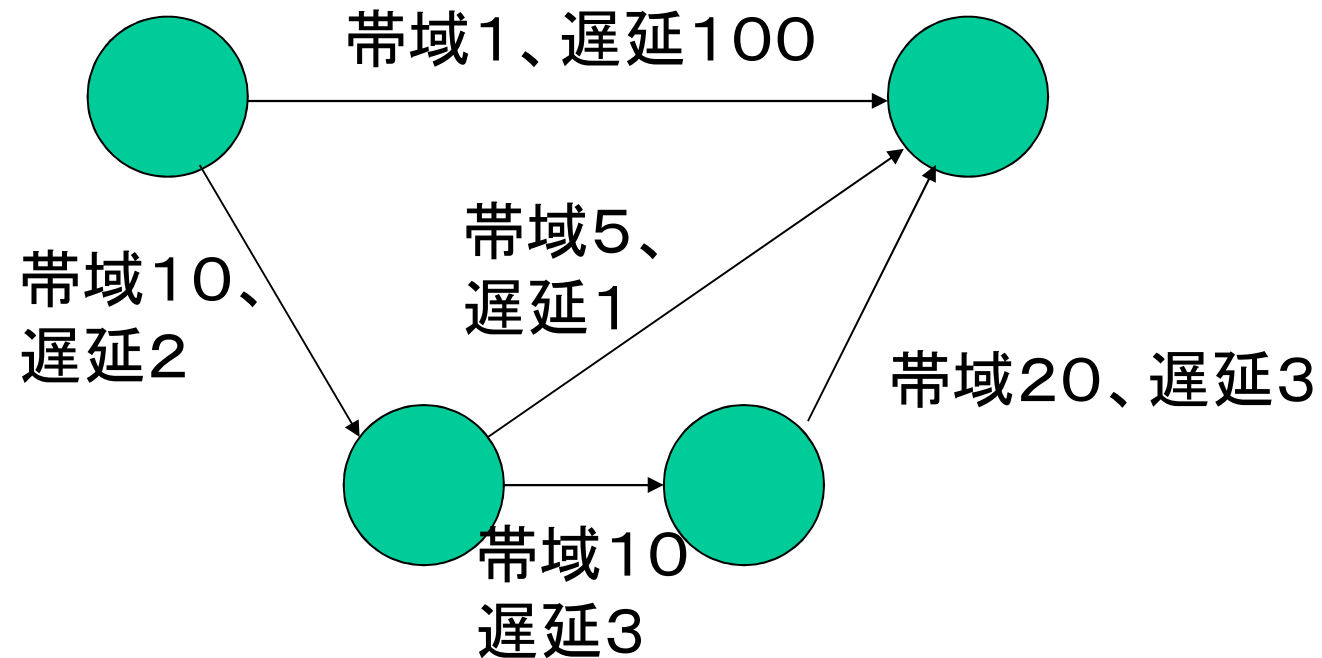
# 資源予約プロトコル RSVP

(Resource ReSerVation Protocol、RFC2205)

- マルチキャスト対応
  - 送信側からPATHメッセージを送る
  - 受信側からPATHの向きにRESVメッセージを送る
- ベストエフォート経路しか考えない
- 問題は多い
  - グループ管理者不在のマルチキャストは無理
  - 課金モデルがない

# QoSルーティング

- あたえられたリンク特性から、希望のQoSを満たす最短経路を求める



# QoSルーティングの困難(1)

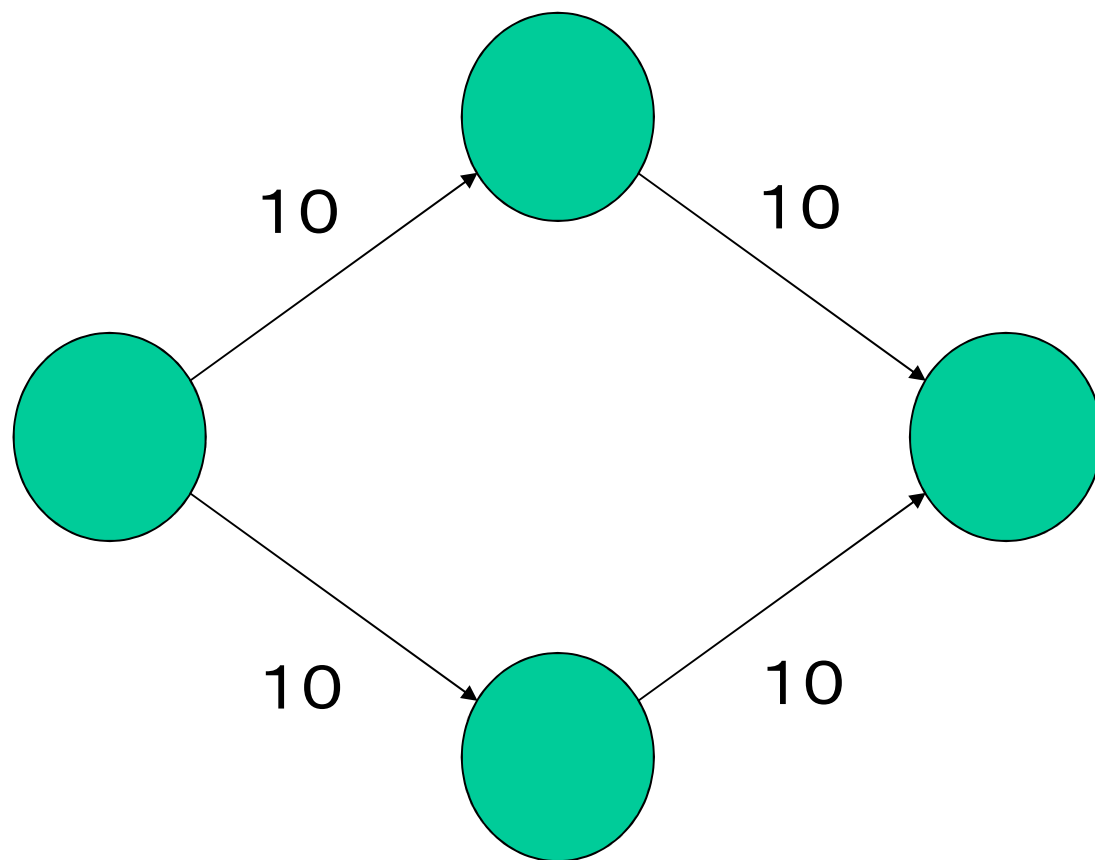
## NP完全性

- 帯域を満たすのは簡単
  - 帯域が足りないリンクを無視
- 遅延やジッタは加法性があるので
  - 単独ならダイクストラ等
  - 制約が複数だと、NP完全問題
- 幸い、動的経路選択を考える場合、最小遅延は考えなくていいので、遅延＝ジッタ
  - 課金がからむと妥協が必用

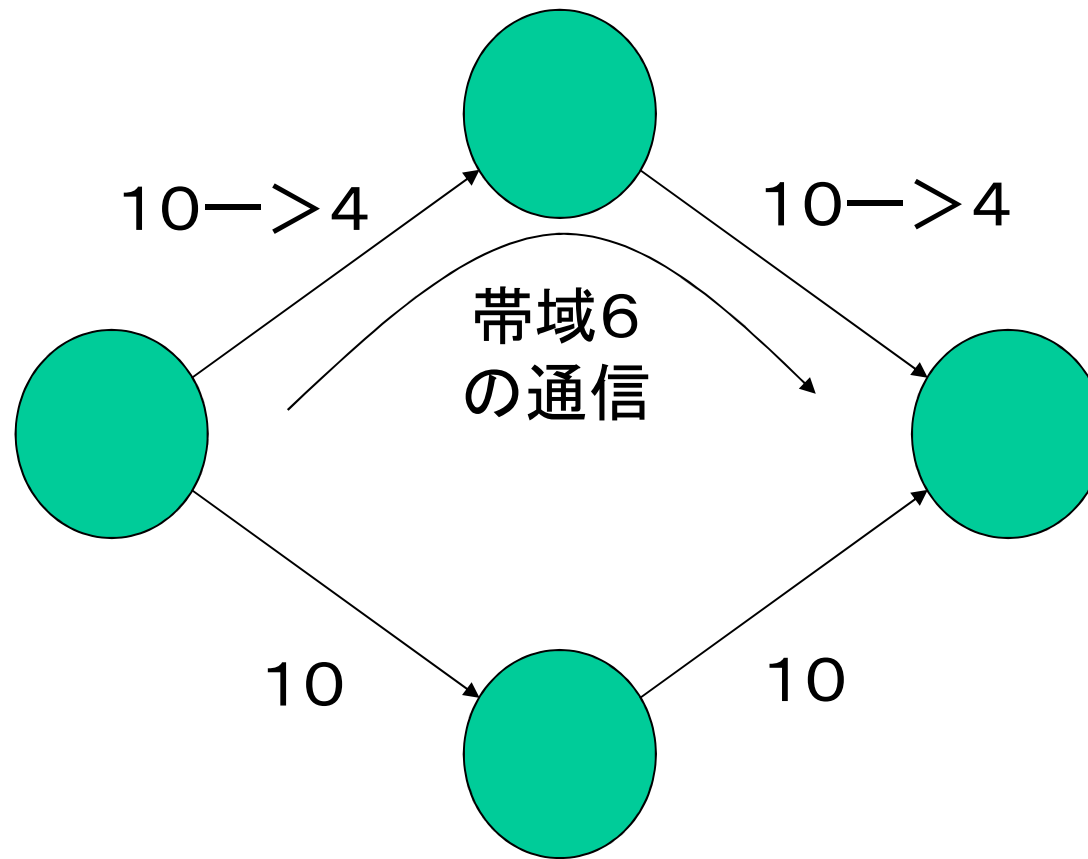
# QoSルーティングの困難(2)

## 経路の不安定性

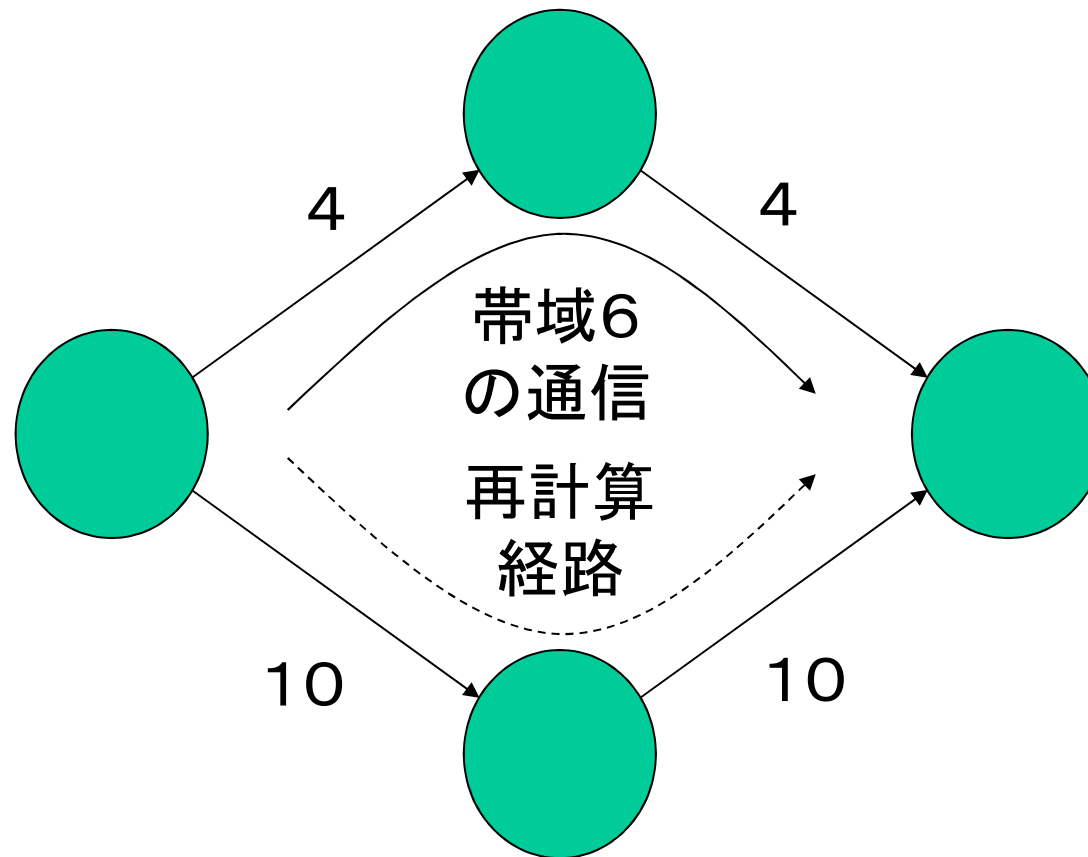
- ある通信が帯域を消費すると
  - リンクの利用可能帯域は減る
- 経路を再計算すると
  - その通信はもとのリンクを使えないかも
- 動的経路選択は不可能？
  - マルチキャストに動的にメンバーが増えると？



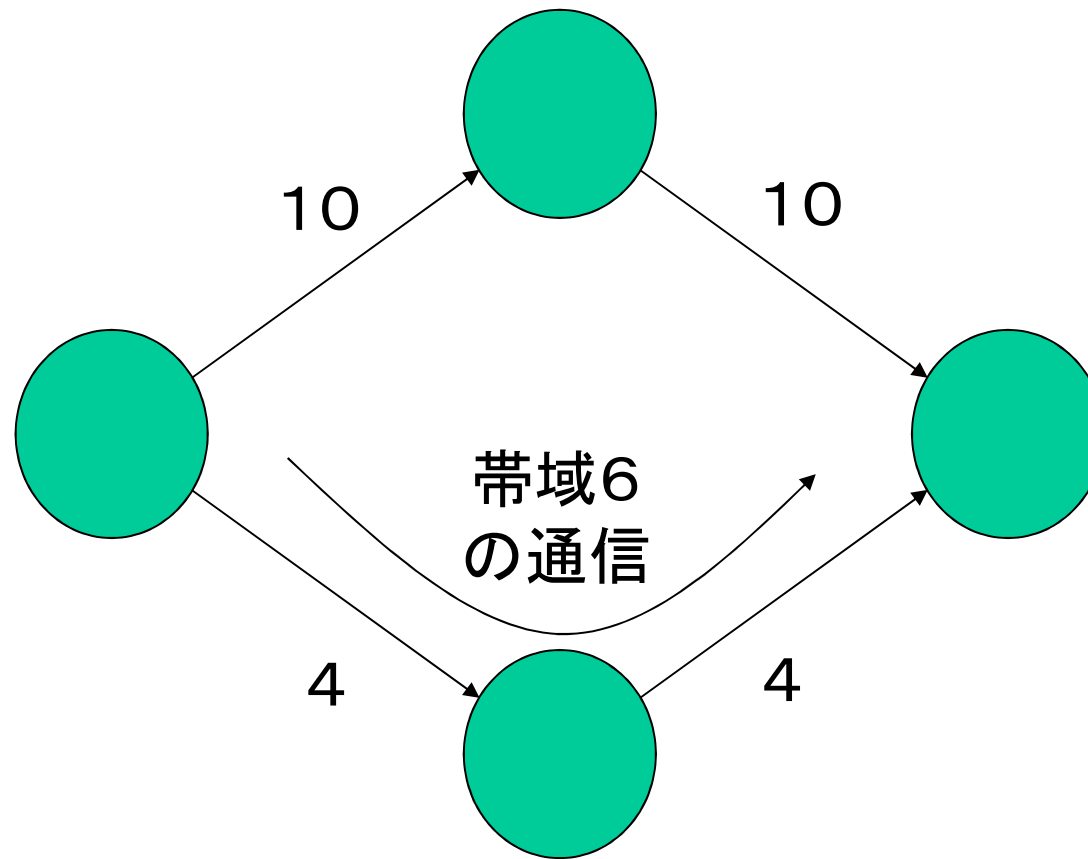
リンク帯域と通信による消費(初期状態)



リンク帯域と通信による消費(帯域6の通信の出現)



リンク帯域と通信による消費(経路の再計算)



リンク帯域と通信による消費(経路の変動)

# 経路の安定化のために

- 全リンクの個別通信の消費帯域を広告
  - 広告量が膨大になり破綻
- 動的経路再計算を行わない
- 送信端(もしくは受信端)で全経路を計算
  - 受信者多数のマルチキャストで破綻
- PATH QoS Collection
  - 資源予約メッセージ(RSVPのPATH)で
    - 実際にその通信が利用可能な帯域を運ぶ

# 経路安定化の別の方法

## Path Computation Element

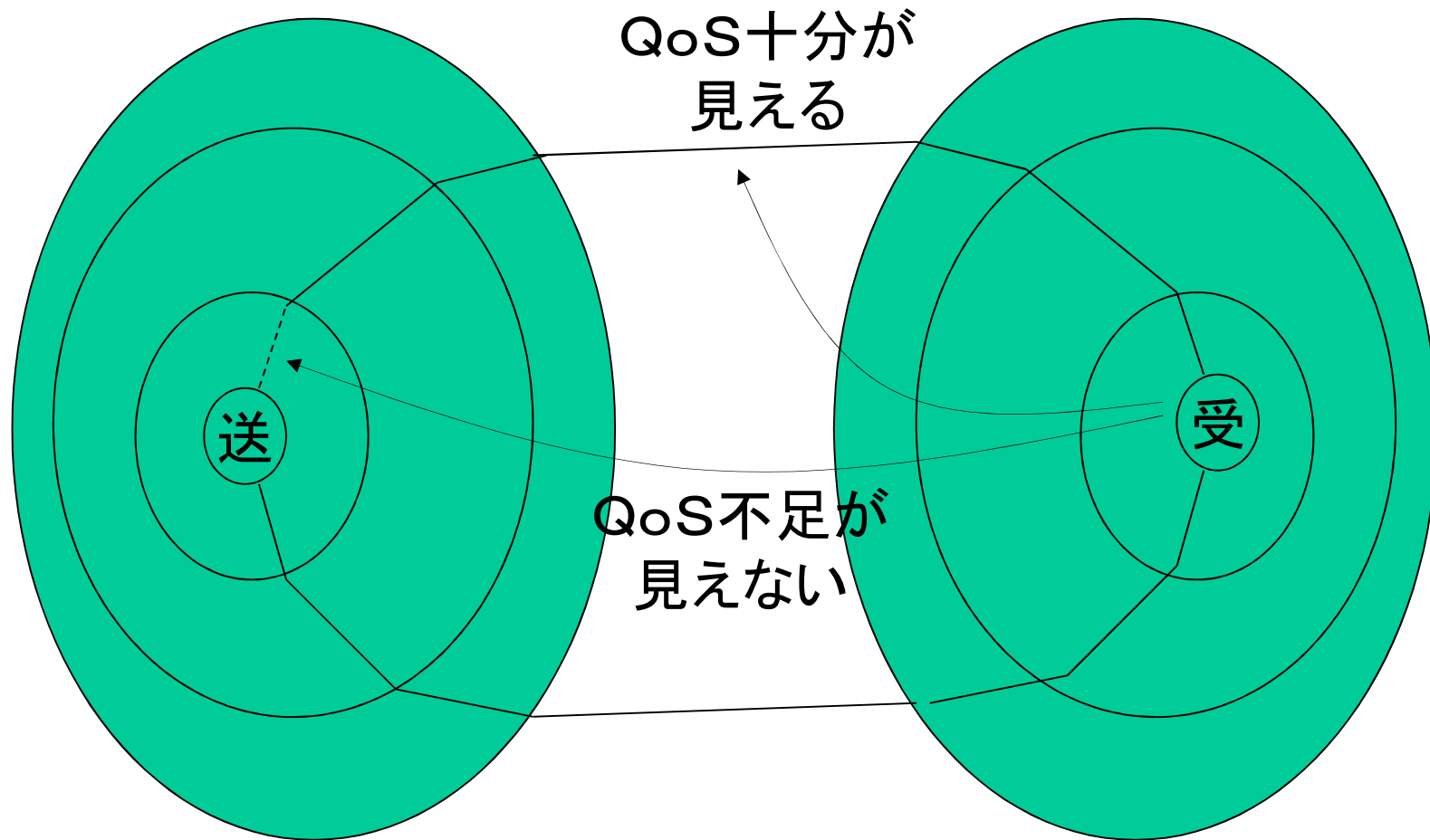
- 経路を一箇所で集中して計算
  - ルータにはたいした計算能力がないので(?)、専用の機械にやらせる
    - 分散したほうが、いいに決まっている
  - 一箇所で計算し全情報を把握するので(?)、不安定性はない
    - ISPはそれぞれ独立にPCEを持つので、PCE間の連携が必要となり、不安定性が生じる
      - PCEが連携できるなら全ルータも連携できるはず

## QoSルーティングの困難(3)

### 階層化とクランクバック

- 大規模化には階層ルーティングが必要
  - 経路情報の階層的詳細化
- 経路選択には全階層の経路情報が必要
  - 自分の周囲の詳細経路情報は手に入るが
  - 相手の周囲の詳細経路情報は手に入らない
    - トップレベルでの入り口によっては内部で相手にQoSを満たしつつ到達できない
- クランクバック(別経路選択)は複雑怪奇

# 階層化環境で受信者から見た 送信者の周囲のQoS

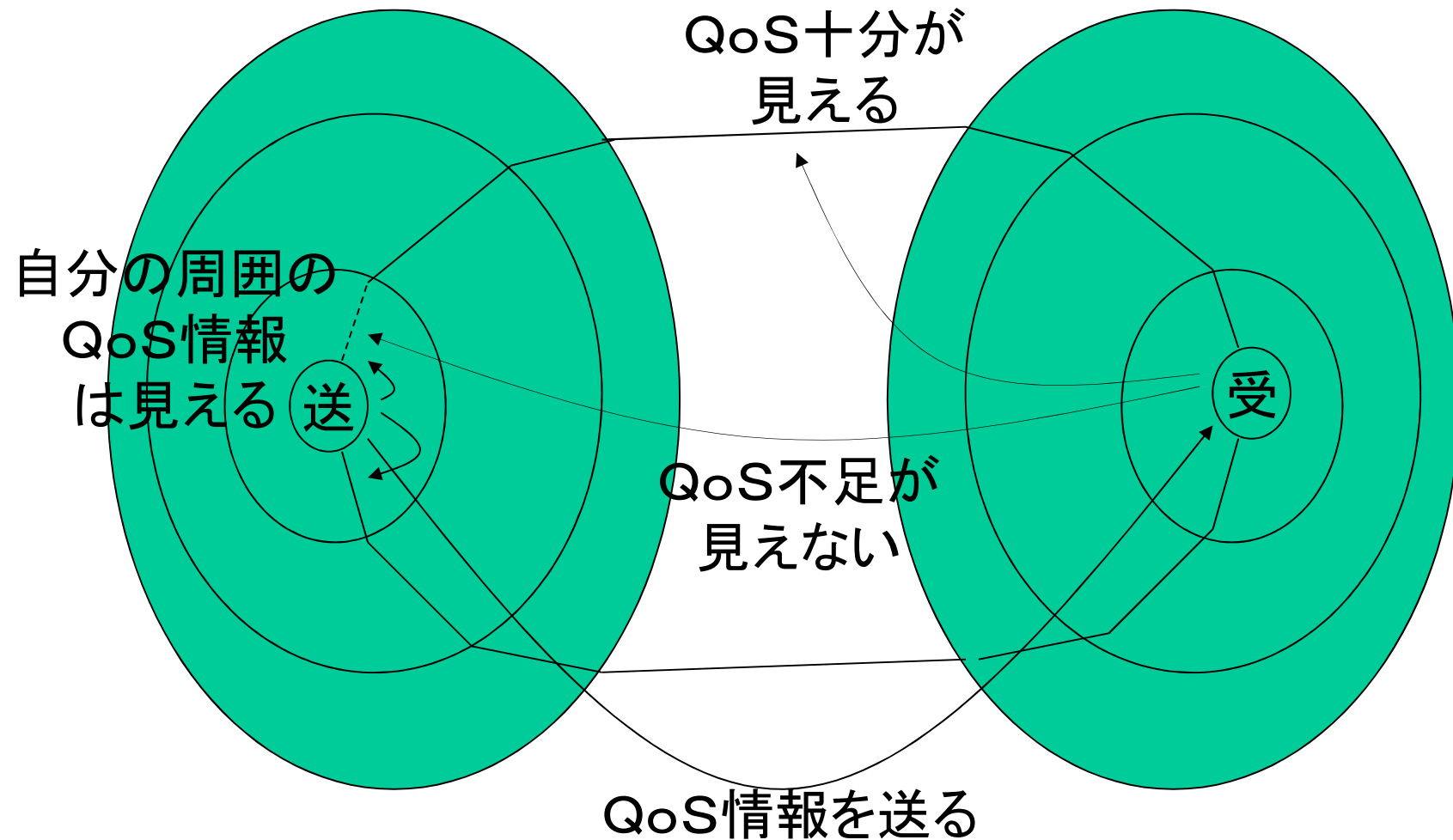


# 階層化のために

## First Aggregated QoS

- 送信側の経路情報(FAQ)をPATHメッセージで送る
  - 送信者は自分の周りの経路情報を添付
  - エリアをまたぐ際にその階層の入り口から送信者への上位階層の経路情報を追加
- 各ホップでは
  - 自分周辺の経路情報は経路制御プロトコルで
  - 送信者周辺の経路情報はFAQで

# 階層化環境で受信者から見た 送信者の周囲のQoS

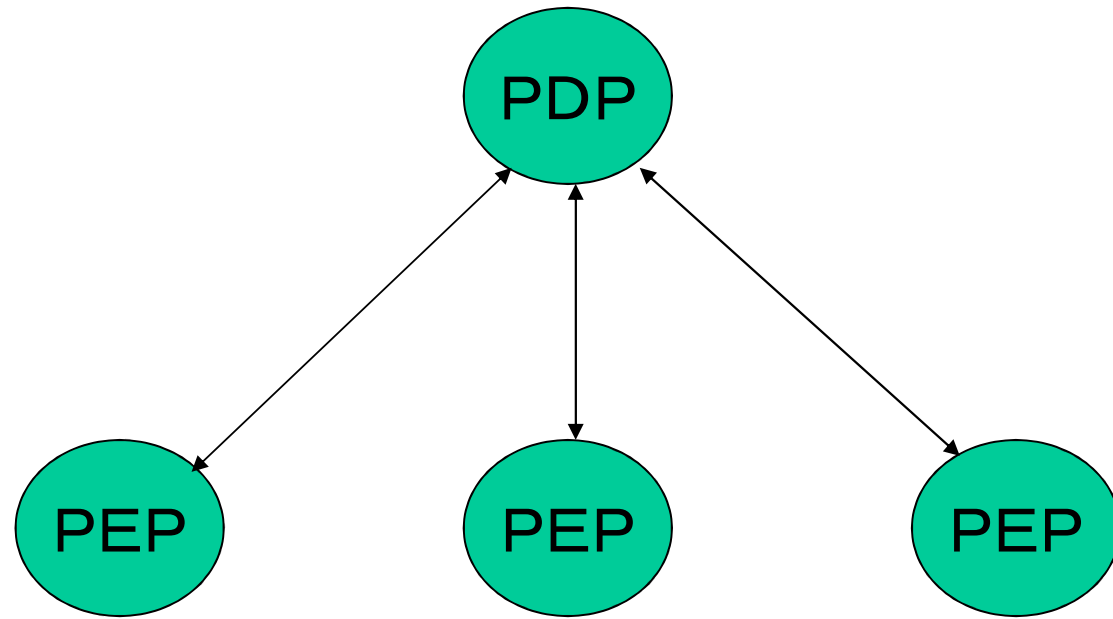


# QoS保証とポリシー

- QoS保証された通信は他に優先して扱う
  - みなにQoS保証を認めると無意味
- どの通信を優先すべきかを判断？
  - 個別にポリシーを設定？
  - LANならなんとかなるが、公衆網では破綻
- 結局、お金で解決
  - 優先度、時間、帯域などに応じた従量制課金
  - 消防、警察等はべつ

# PEPとPDP

- ポリシー制御の枠組み？
- PEP(Policy Enforcement Point)
  - ポリシーを施行する場所、ルータ等
- PDP(Policy Decision Point)
  - PCE(PATH Computaiton Element)が集中してポリシーを計算する場所
- エンドツーエンド原理違反
  - 負荷の集中、信頼性の欠如、判断材料の欠如、、、



# ポリシーと課金

- ベストエフォート通信は定額制課金
  - ISPは利用者のトラフィックを押しつけあう
  - ポリシーはISP間で決定
- QoS保証通信は従量制課金
  - ISPは利用者のトラフィックを奪い合う
  - ポリシーは利用者が決定
    - 利用者には十分な情報(各経路ごとの課金額)を
      - ISPは課金情報を隠せない

# マルチキャストと課金

- ポリシーは利用者が決定
  - 経路は利用者が決定
- 受信者のポリシーで経路が異なる？
  - マルチキャスト木を共有できない
- ポリシーはグループ管理者が決定
  - 受信者はそれに従う

# SRSVP

## (Simple RSVP)

- マルチキャストと資源予約を統合
  - マルチキャストはPIM-SM
    - 余計な拡張もなし
  - 利用者は遅延優先か料金優先のポリシーで経路を選択
    - ソースルーティングも可能
  - リンクステートルーティングプロトコルも必要

# まとめ

- QoS保証はネットワークの機能
  - エンドではできない
- キューイングはPPQ
- 経路の安定にはPQC、階層化にはFAQ
- QoS保証とマルチキャストとの統合が必須
- QoS保証には従量制課金が必要
  - BGPのポリシーでは対応できない