

# 画像情報特論 (2)

## - TCP/IP (1)

- インターネットプロトコル (IP)
- インターネットQoS

情報ネットワーク専攻 甲藤二郎

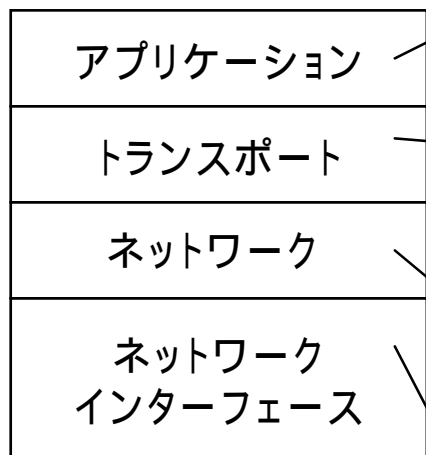
E-Mail: [katto@waseda.jp](mailto:katto@waseda.jp)

# インターネットプロトコル

IP (Internet Protocol)

# インターネットの基礎

## プロトコルスタック



### アプリケーション

HTTP, RTSP, FTP, Telnet, ...

← **RTP**: 実時間メディア用途

### 端末・端末間

**TCP**: 誤り訂正、順序制御、フロー制御 ... 信頼性重視

**UDP**: オーバーヘッド少 ... 低遅延、高速性重視

### 端末・ルータ間、ルータ・ルータ間

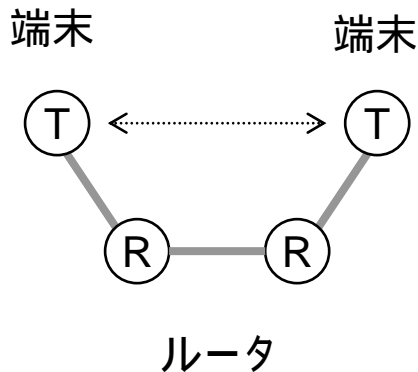
**IP**: 経路制御、フラグメンテーション

**ICMP**: エラー通知

**IGMP**: マルチキャスト (mbone)

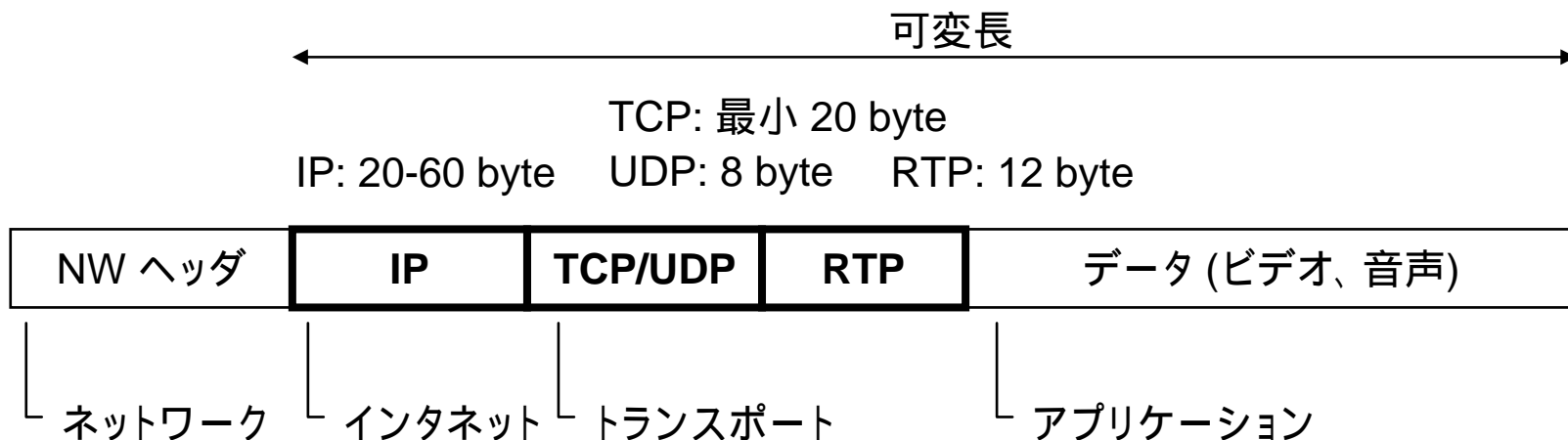
### 個別リンク

イーサネット, PPP, X.25, ATM, ...

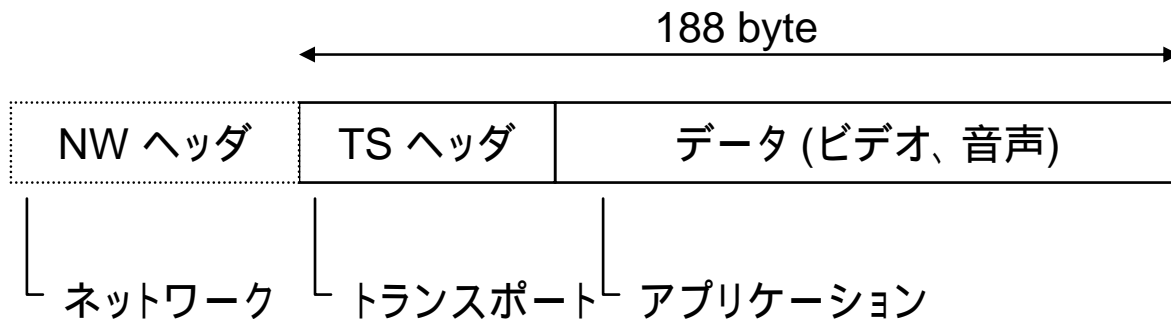


# IP データグラム

## IP データグラム



cf. MPEG-2 トランスポートストリーム (ITU-T H.222)



# IP ヘッダ

## IPヘッダ

4 byte

Version	ヘッダ長	サービスタイプ	パケット全長	
フラグメント識別値			フラグ	フラグメントオフセット
TTL (生存時間)	上位プロトコル		ヘッダチェックサム	
<b>送信元 IPアドレス</b>				
<b>受信先 IPアドレス</b>				
(オプション)			(パディング)	
データ				

パケット長:

データのフレーミング (可変長)

TTL:

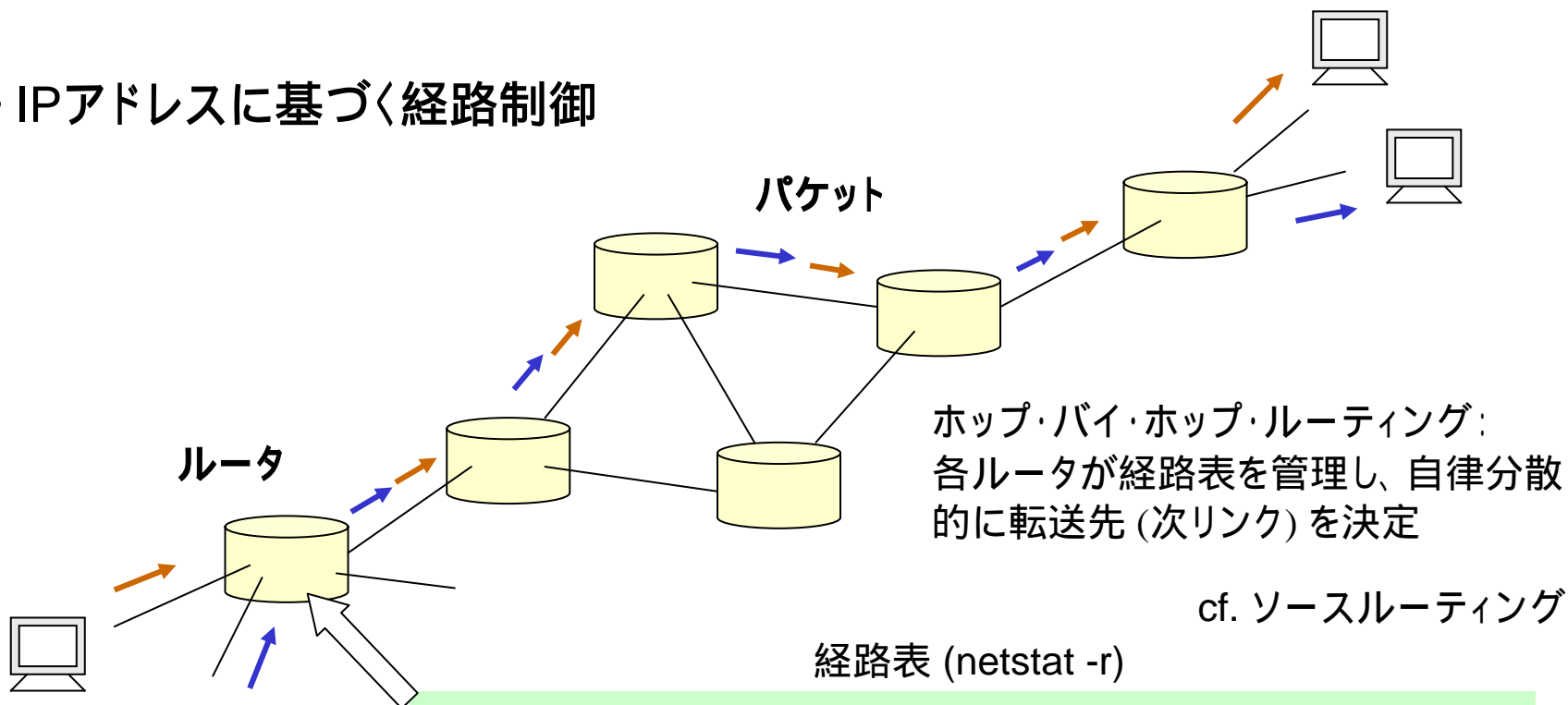
パケット生存時間 (ルータのホップ数)

IPアドレス:

インターネット全体で固有のアドレス。ARP によって  
MACアドレスに変換される (Ethernet の場合)

# IP の機能

## • IPアドレスに基づく経路制御

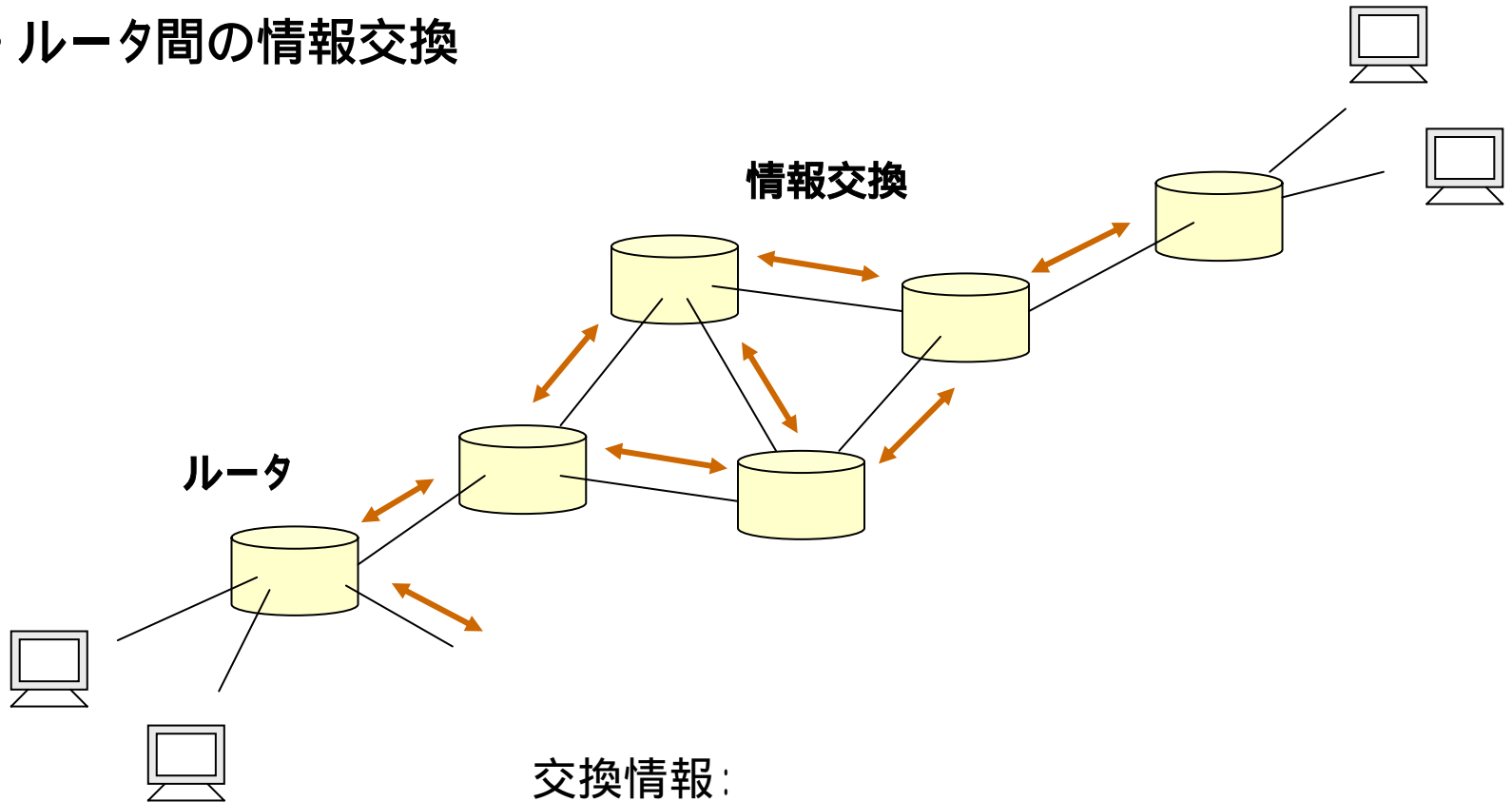


目的地	ネットマスク	次ルータ
133.9.2.x	255.255.255.192	133.9.1.a
133.9.3.x	255.255.255.192	133.9.1.b
133.9.4.x	255.255.255.192	133.9.1.c

default	0.0.0.0	133.9.1.d
---------	---------	-----------

# 動的経路制御

- ルータ間の情報交換

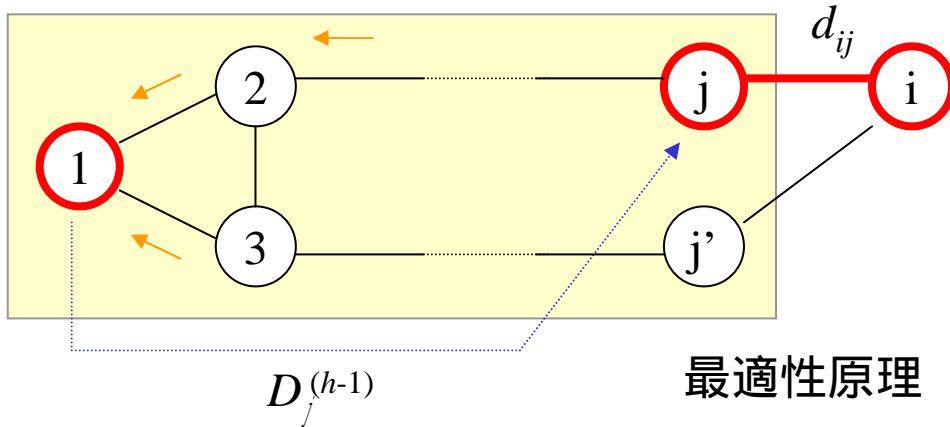


交換情報:

ホップ数、遅延、帯域幅、...

現状はホップ数のみ使っているのがほとんど  
インターネットQoS関係でその他も考慮 (QOSPF)

# Bellman-Ford アルゴリズム



$D_i^{(h)}$ : ルータ 1 からルータ  $i$  までの  
ホップ数  $h$  以下の最短経路

$d_{ij}$ : リンク  $(i, j)$  のコスト (交換情報)

初期条件:  $D_i^{(0)} = \infty$

1 回目の情報交換:  $D_i^{(1)} = \min_j [D_j^{(0)} + d_{ij}]$

2 回目の情報交換:  $D_i^{(2)} = \min_j [D_j^{(1)} + d_{ij}]$

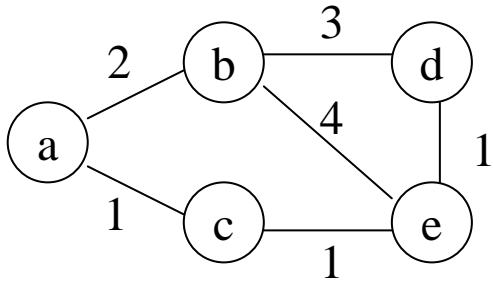
$h$  回目の情報交換:  $D_i^{(h)} = \min_j [D_j^{(h-1)} + d_{ij}]$

ネットワーク全体のノード  
数を  $N$  とすると、最大  $N-1$   
回の計算で収束

使用例:  
距離ベクトル制御 ... RIP



# RIP (Routing Information Protocol)

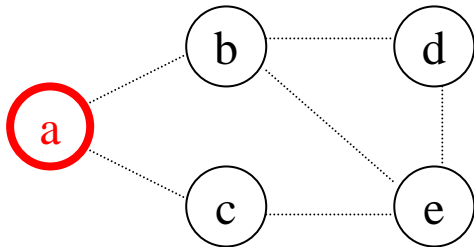


各ルータは隣接ルータへのコストのみ保有 (初期状態)

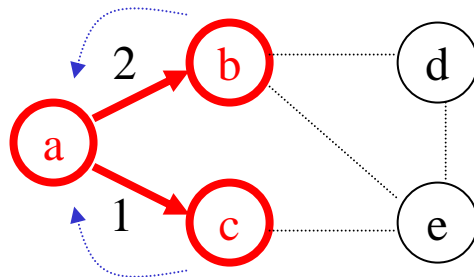


隣接ルータ間の情報交換の度に最短経路を更新

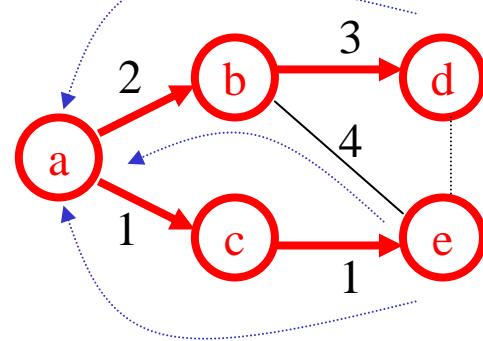
(1) 初期状態



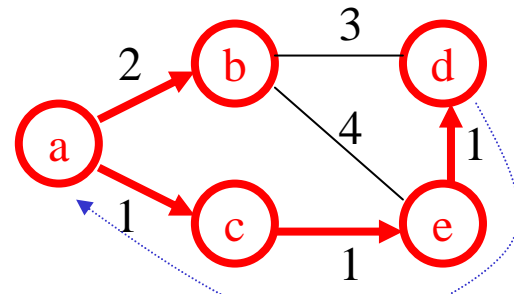
(2) 1回目 (ホップ数1まで)



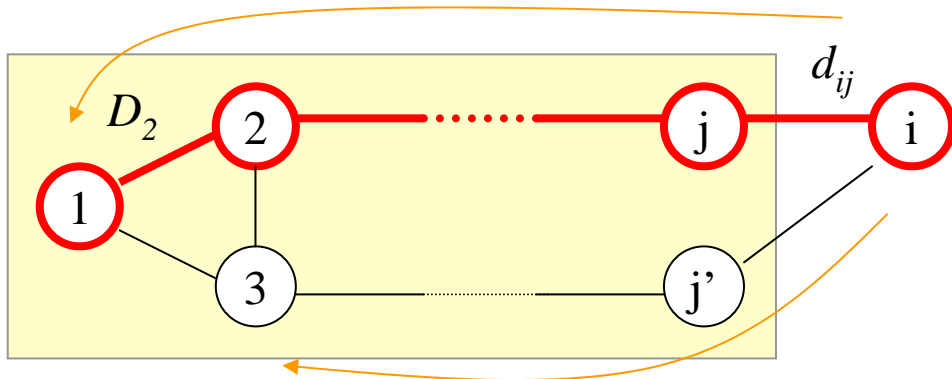
(3) 2回目 (ホップ数2まで)



(4) 3回目 (ホップ数3まで)



# Dijkstra's アルゴリズム



$D_i$ : ルータ 1 から  $i$  までの経路長

$P$ : ルータの集合

$d_{ij}$ : リンク  $(i, j)$  のコスト

初期条件:  $P = \{1\}$ ,  $D_1 = 0$ ,  $D_k = \infty$  ( $k \neq 1$ )

情報交換: すべてのルータ間でリンク状態の情報交換 (フラッディング)

ステップ1:  $D_i = \min_{k \notin P} D_k$  となるルータ  $i$  を探索 (Shortest Path)

$P = P \cup \{i\}$  と集合  $P$  を更新

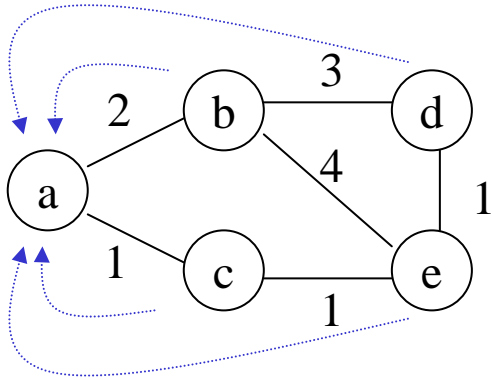
$P$  がすべてのルータを含んだら終了

ステップ2:  $j \notin P$  に対して  $D_j = \min[D_j, D_i + d_{ij}]$

ステップ 1 に戻る

使用例: リンク状態制御 ... OSPF

# OSPF (Open Shortest Path First)

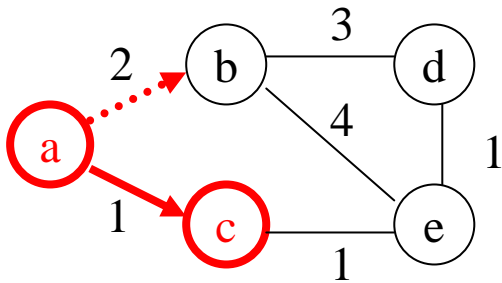


トポロジ (接続情報) とリンクコストを一斉にフラッディング

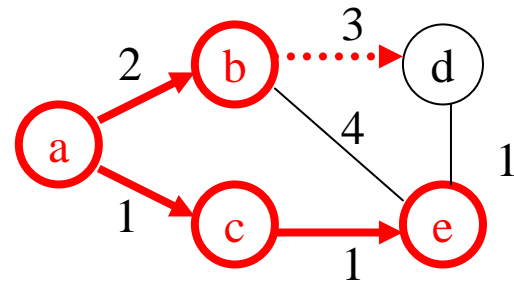


ローカルに Shortest Path を繰り返し探索

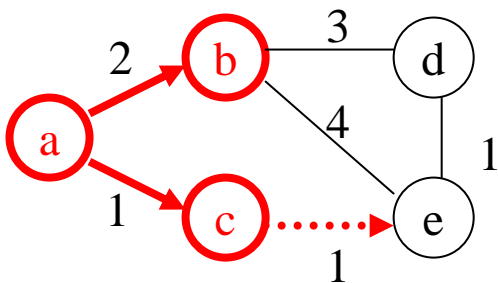
(1) フラッディング直後 ( $P = \{a, c\}$ )



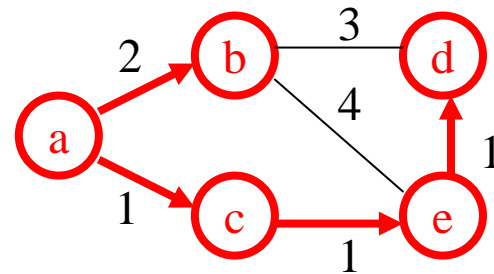
(3) 3回目 ( $P = \{a, b, c, e\}$ )



(2) 2回目 ( $P = \{a, b, c\}$ )



(4) 4回目 ( $P = \{a, b, c, d, e\}$ )

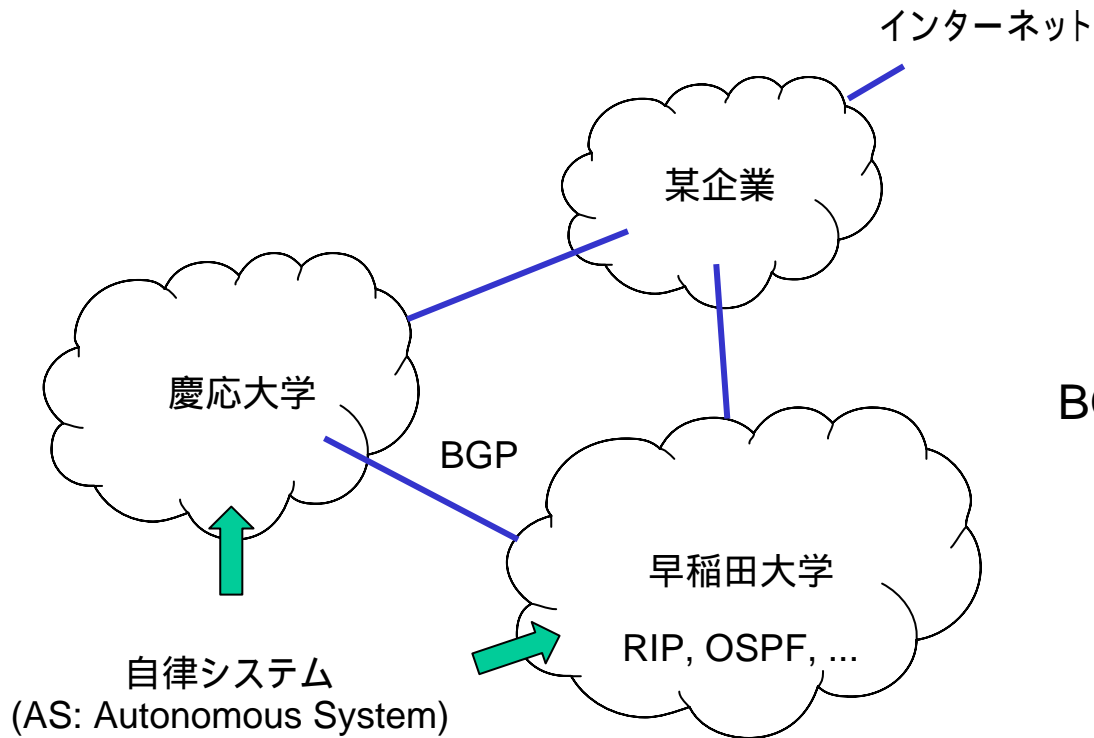


# IGP と EGP

- 経路制御プロトコルのスケーラビリティ

IGP: 自律システム内で使われる経路制御プロトコル (RIP, OSPF, ...)

EGP: 自律システム間で使われる経路制御プロトコル (BGP, ...)



BGP: パスベクトル経路制御

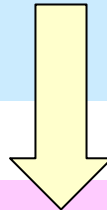
距離コスト +

経路上の自律システムのリスト

経路の到達可能性なども考慮

# インターネットプロトコルの欠点

- 蓄積交換 (store and forward) 故に、パケット転送時間の増大 (**delay**)、転送時間の揺らぎ (**jitter**)、パケット廃棄の発生 (**packet loss**) 等の問題は避けられない。
- パケットの到着順序が逆転することがある (順序制御)。\* ただし、実際には経路制御は静的であり、順序逆転はほとんど発生しない。



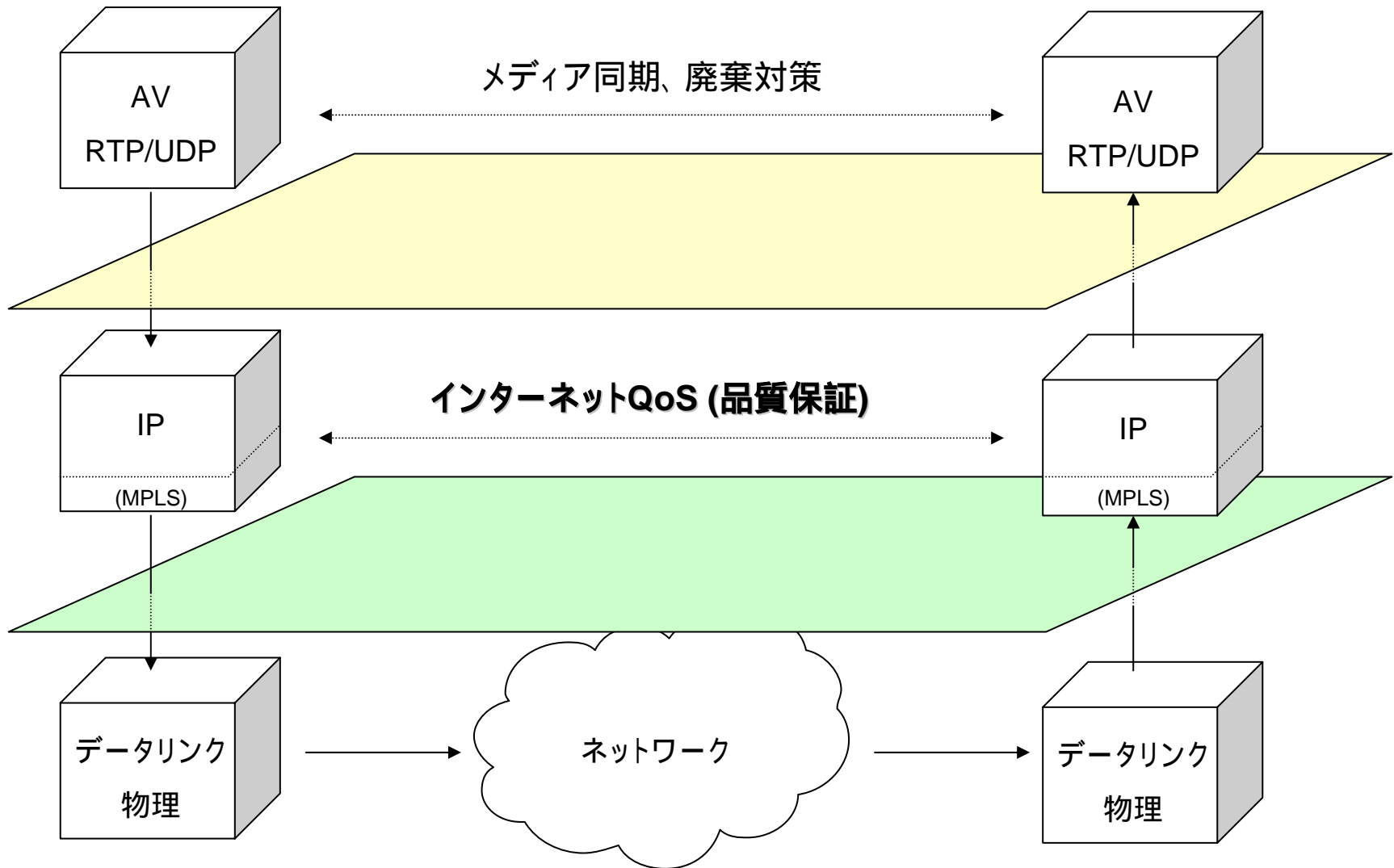
インターネットでもある程度の品質保証 (QoS 保証) を実現したい。

インターネット QoS

# インターネットQoS

MPLS、Diffserv、トラフィックシェイピング、(RSVP)

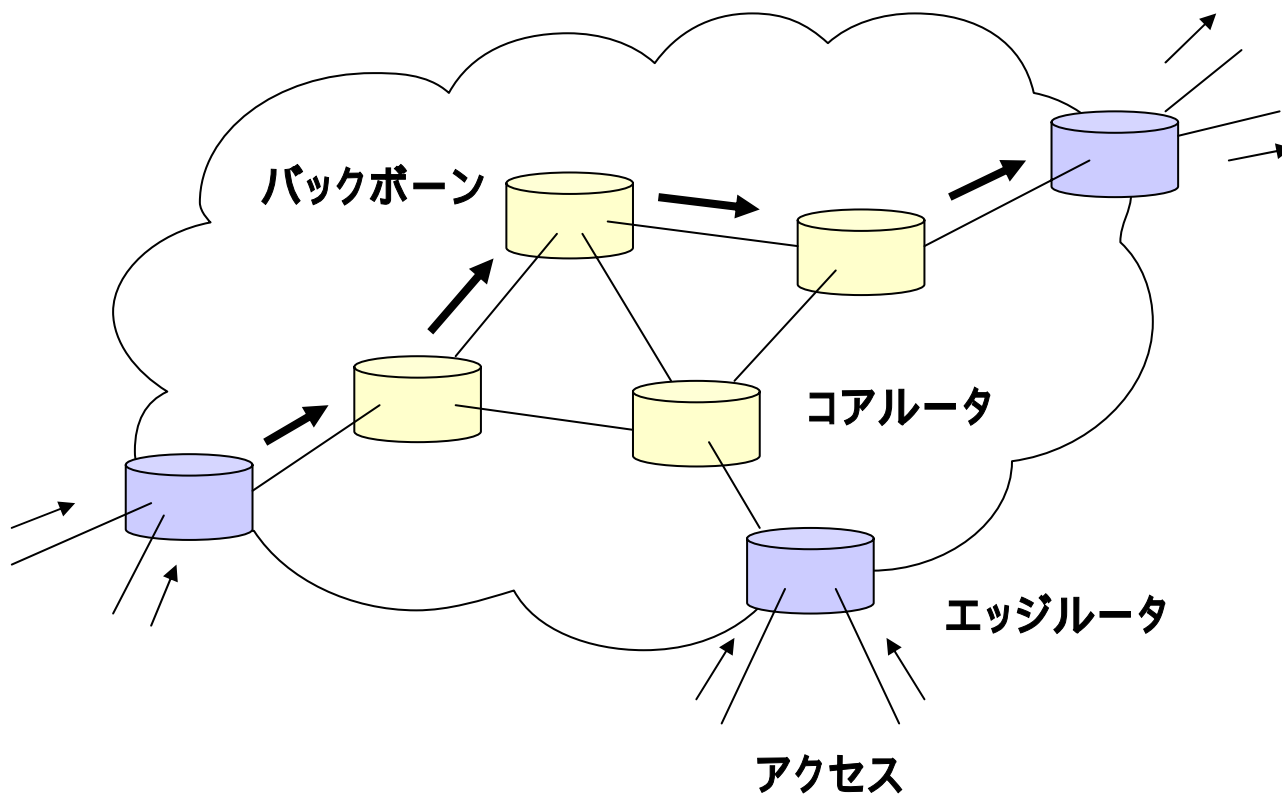
# インターネットQoS



# スケーラビリティ

## コンセプト

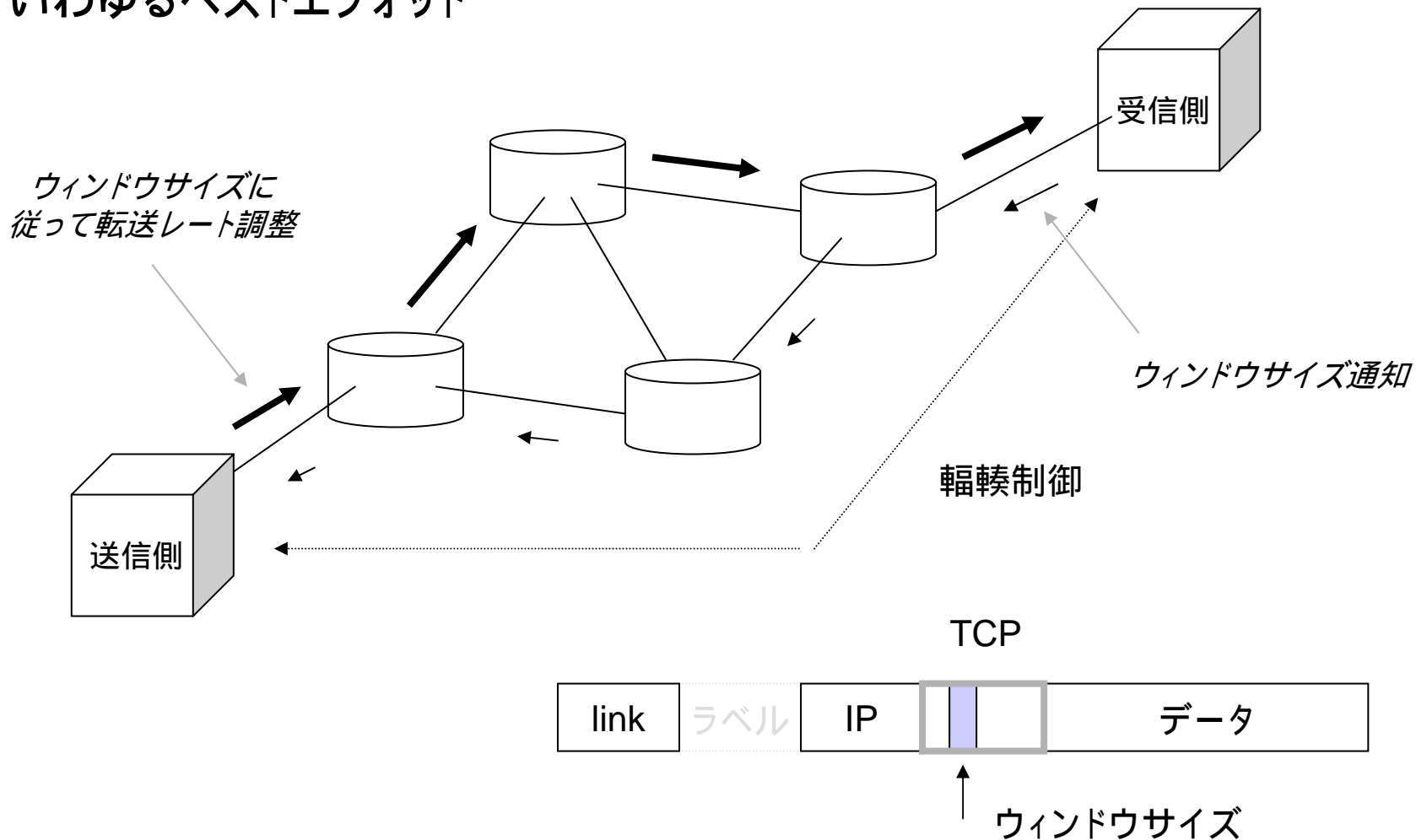
- エッジルータ： トラフィックシェーピング
- コアルータ： パケットの高速転送





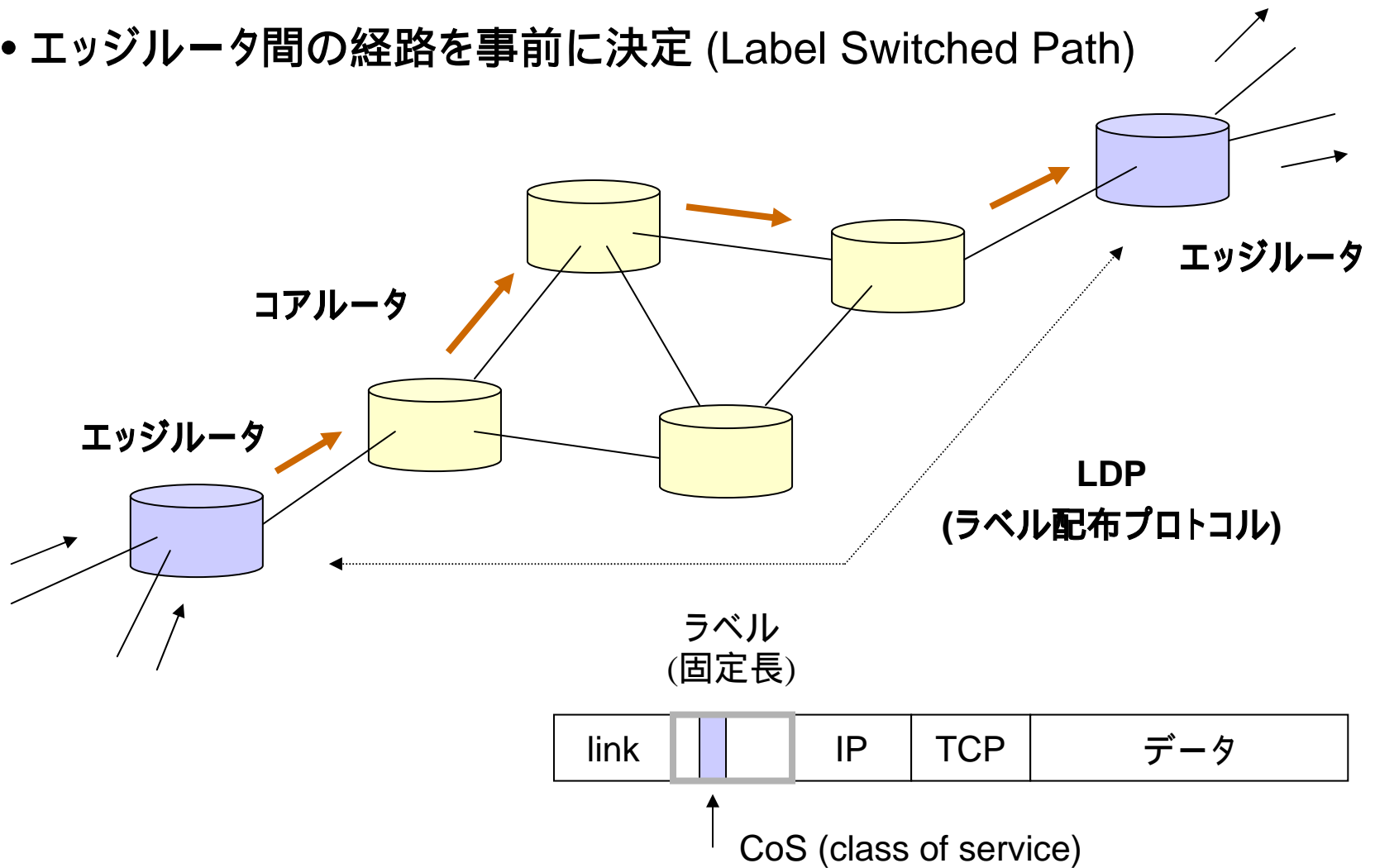
# End-to-End 制御 (従来: トランスポート層)

- TCP 輻輳制御 (ウィンドウ制御)
- いわゆるベストエフォット



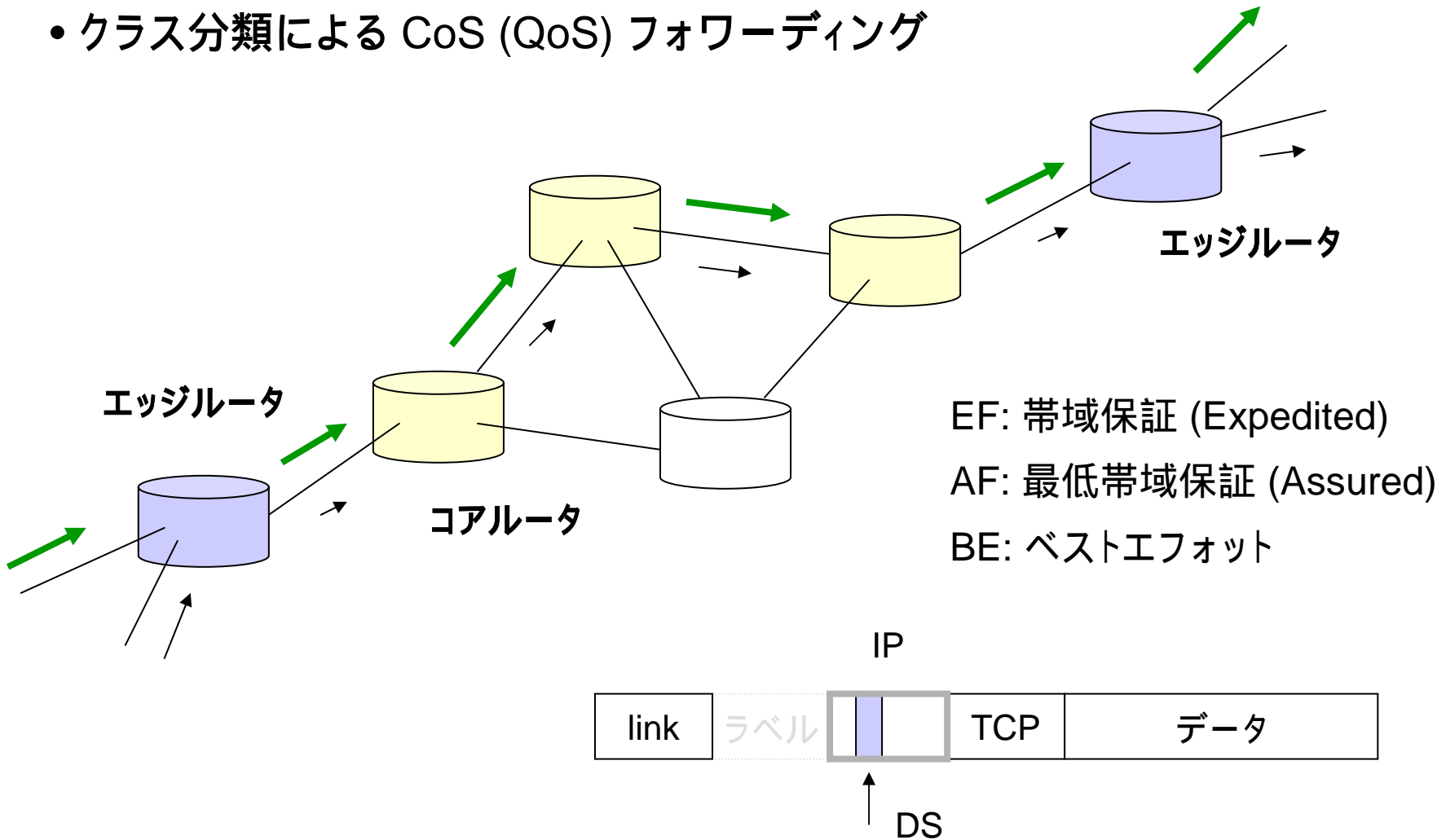
# (1) MPLS (ラベル・スイッチング)

- 固定長ラベルによるハードウェアスイッチング
- エッジルータ間の経路を事前に決定 (Label Switched Path)



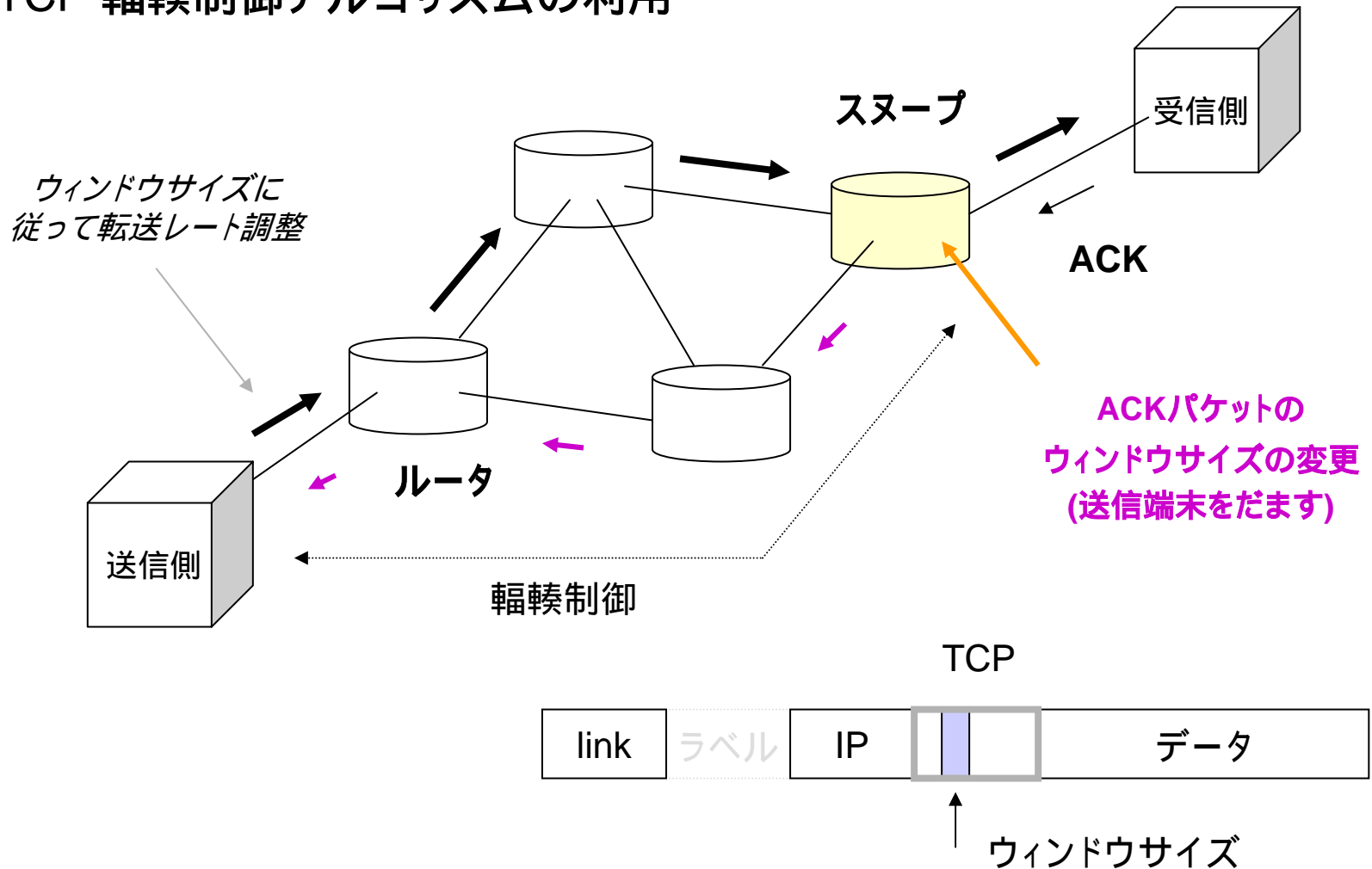
## (2) Diffserv (differentiated services)

- IP ヘッダの TOS フィールドの再定義 DS フィールド
- クラス分類による CoS (QoS) フォワーディング



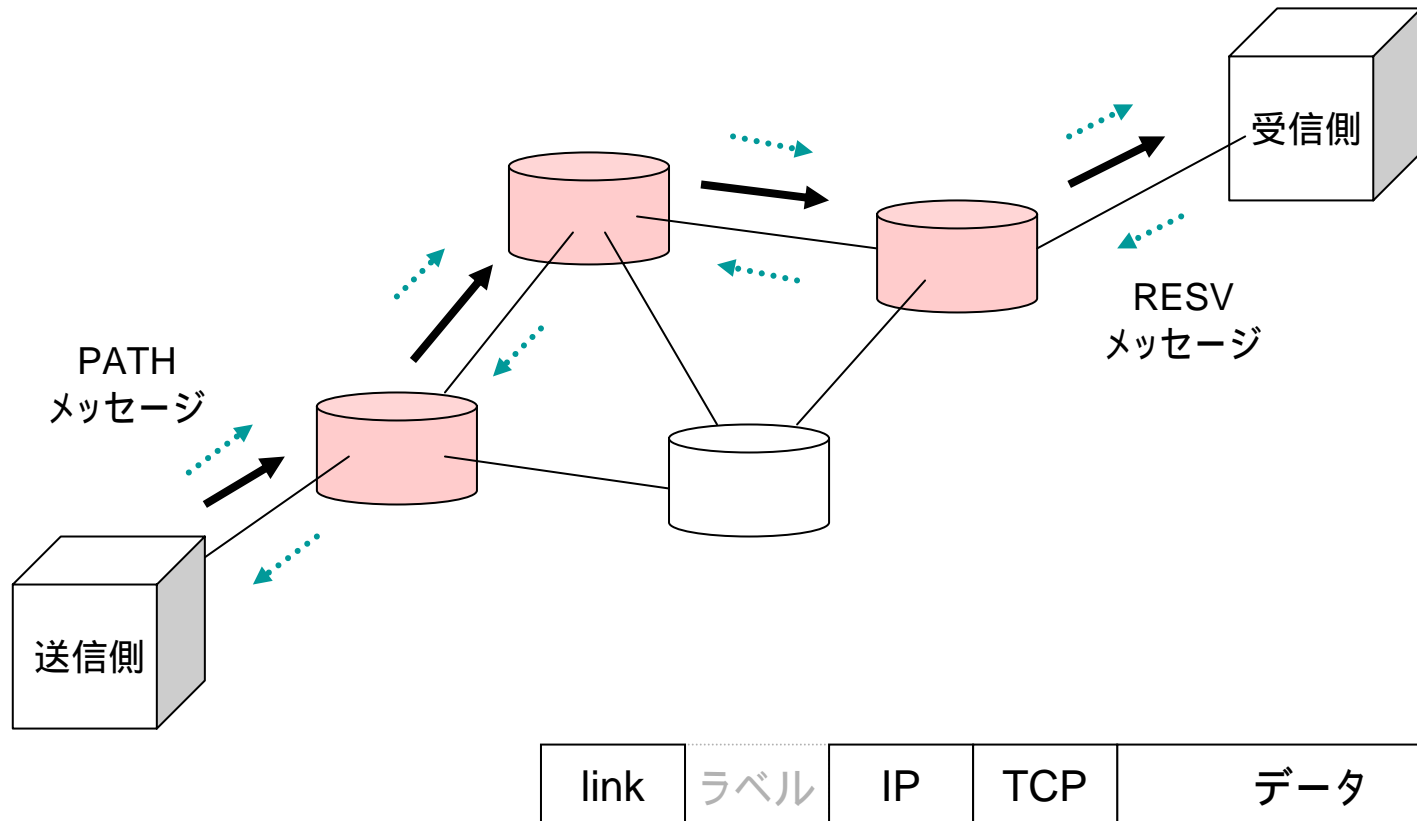
# TCP スヌーピング (L4)

- ルータによる TCP ヘッダのスヌーピング (L4-Switch)
- TCP 輻輳制御アルゴリズムの利用

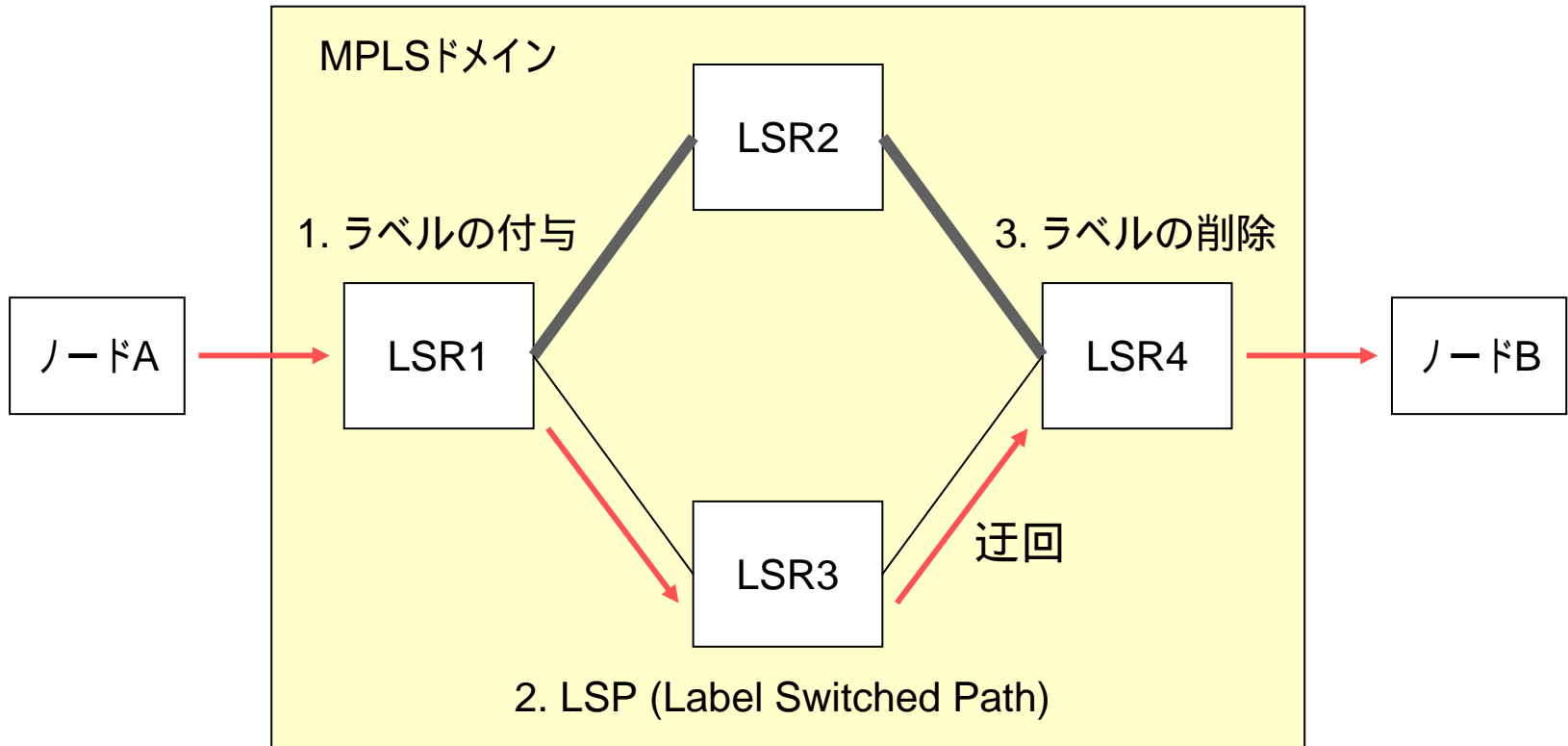


# RSVP (参考: intserv)

- ルータ間のメッセージ交換による帯域確保
- スケーラビリティに問題 (欠点)



# MPLSのLabel Switched Path

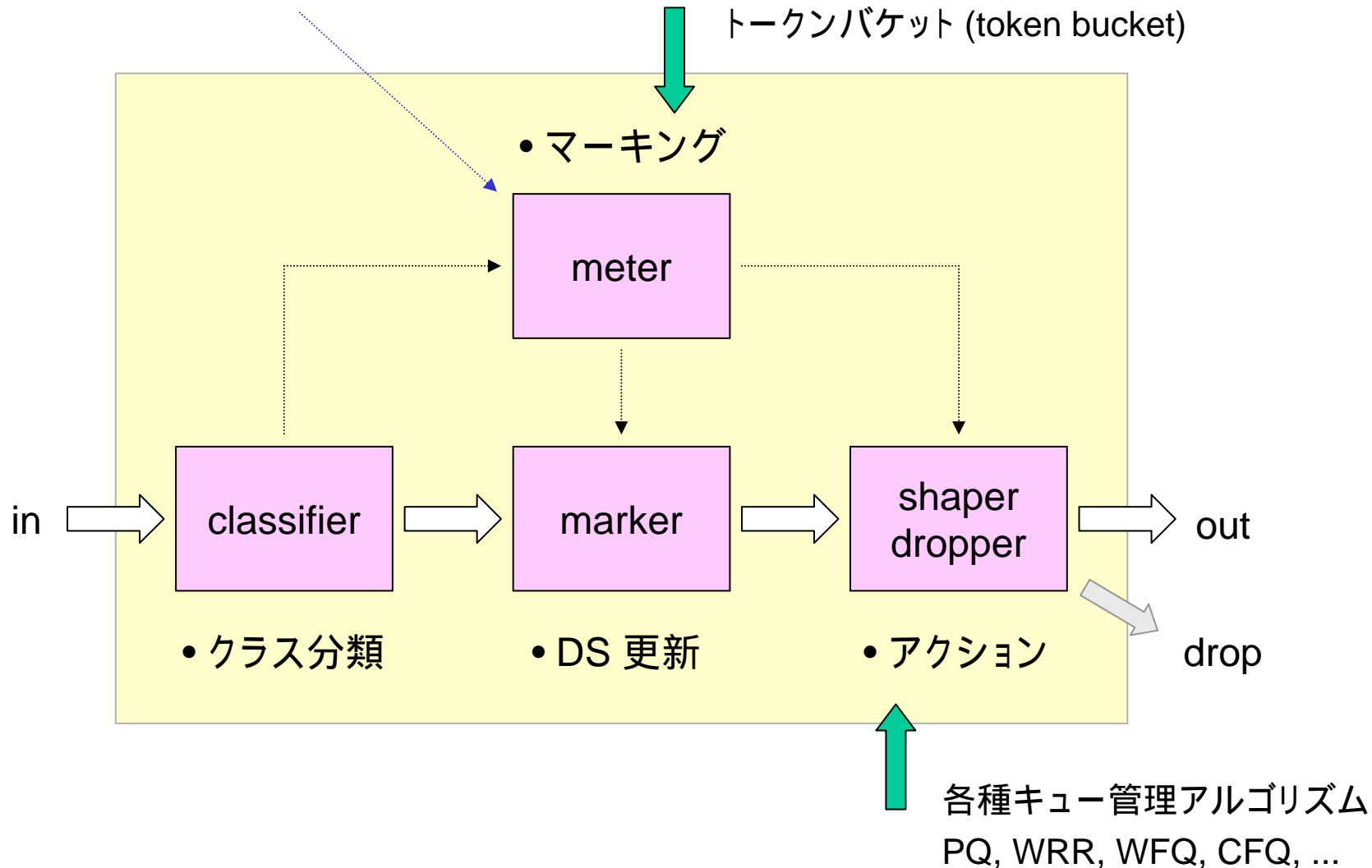


# Diffserv ルータの構成例

SLA 設定 (帯域ブローカ)

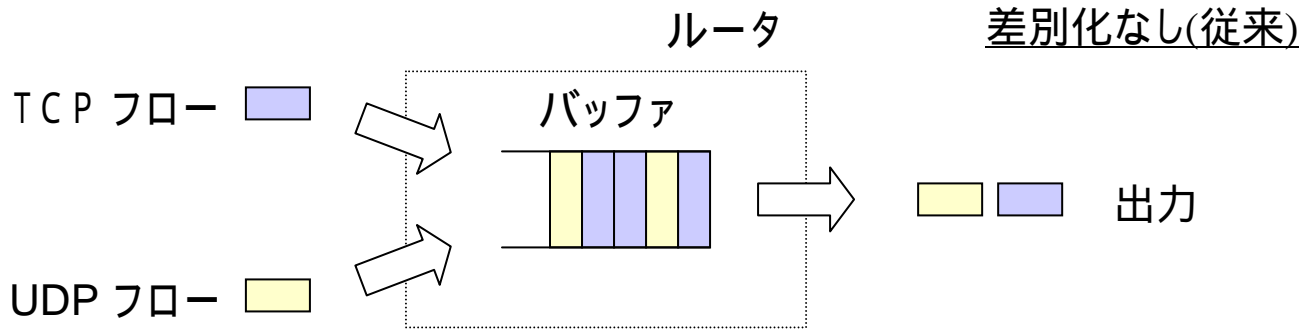
TCM (Three Color Marker)

トークンバケット (token bucket)

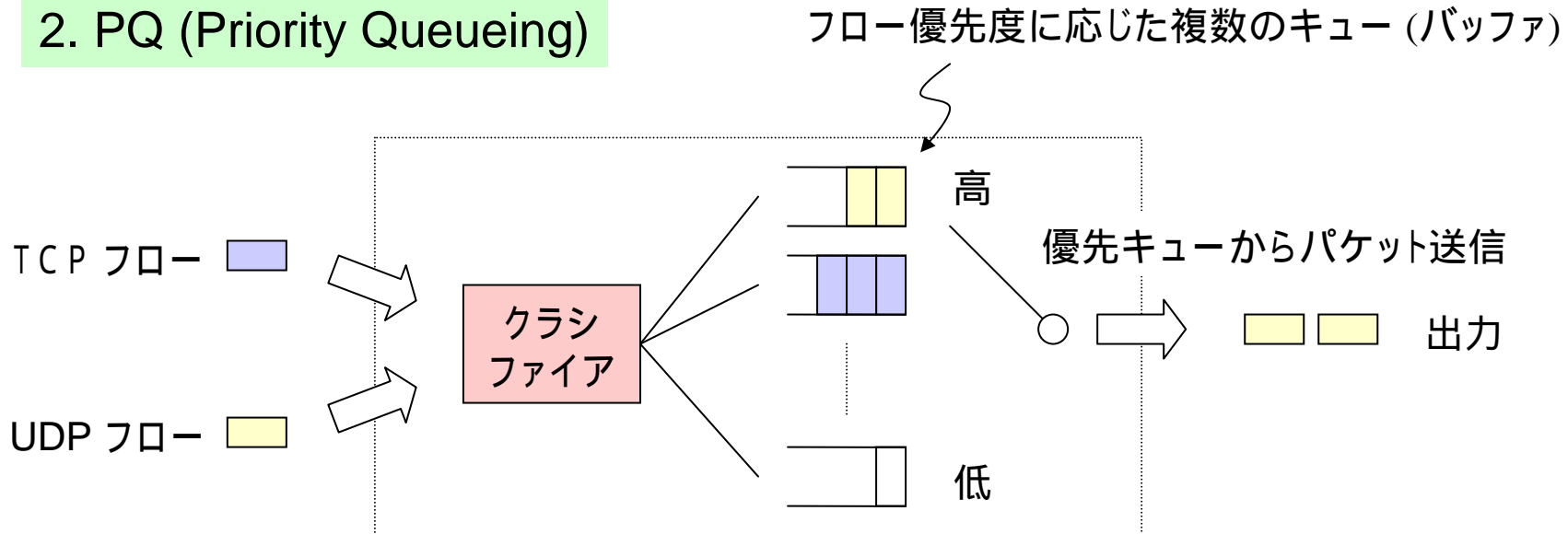


# フローの差別化 (1)

## 1. FIFO (First-In First-Out)



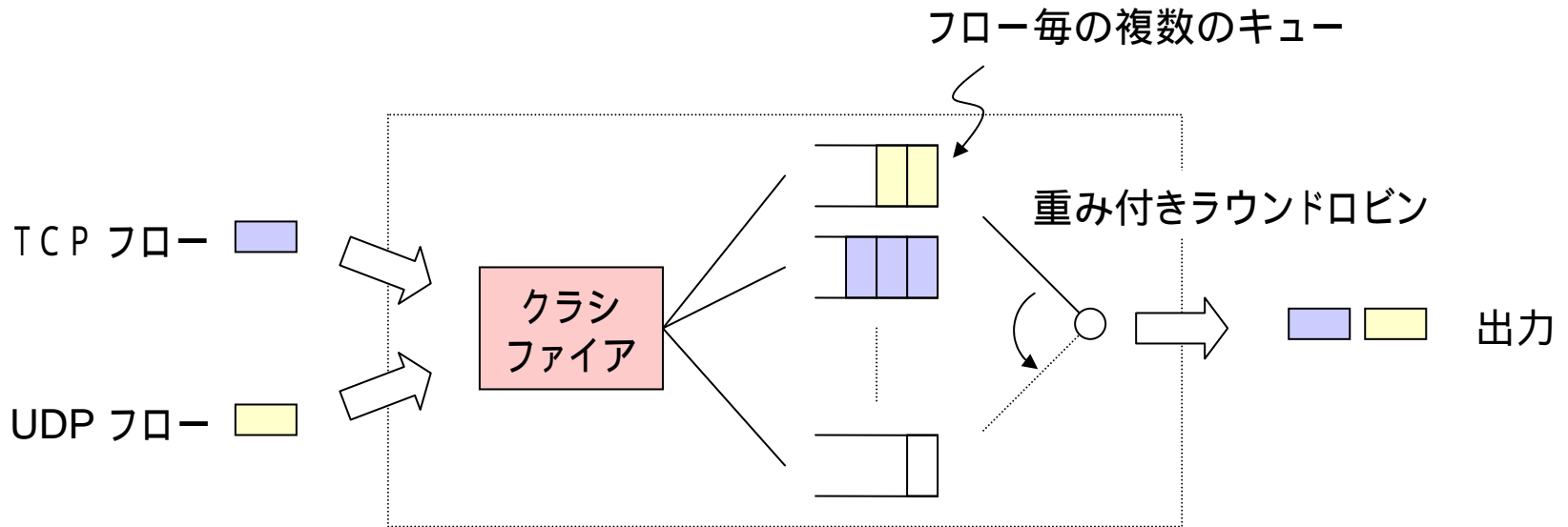
## 2. PQ (Priority Queueing)





# フローの差別化 (2)

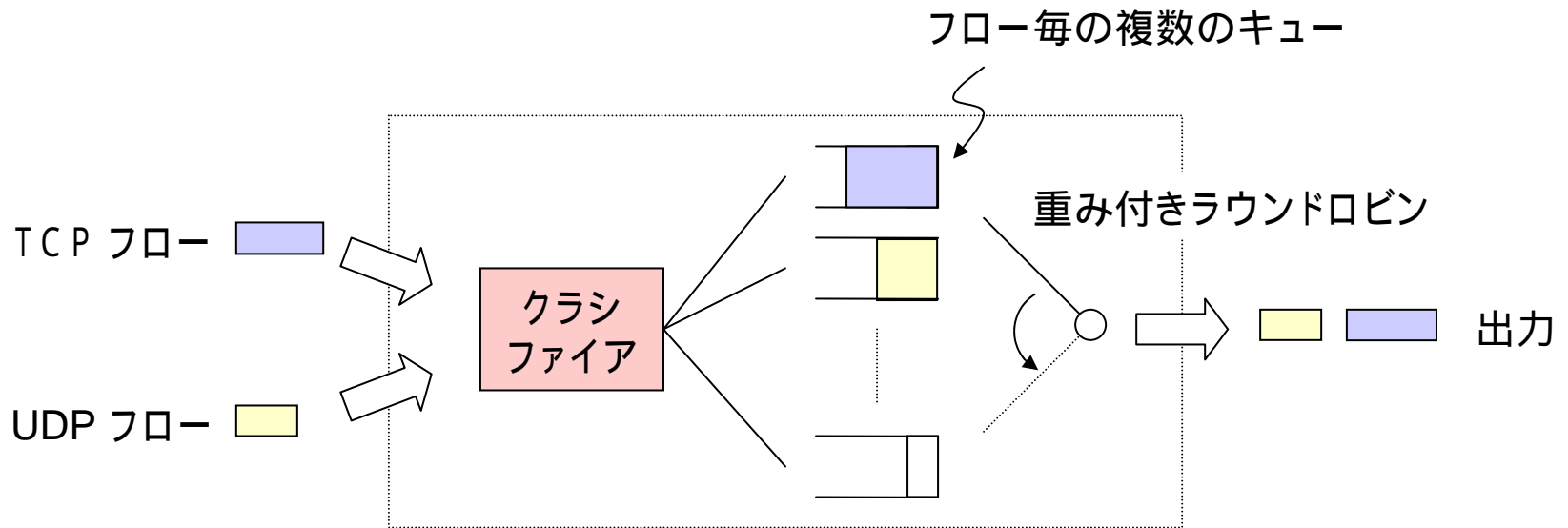
## 3. WRR (Weighted Round Robin)



- PQ: 優先キューが空になるまで非優先パケットは送出されない (欠点)
- WRR: 既定の個数のパケットを送出すると非優先パケットを送出する (改善)

# フローの差別化 (3)

## 4. DRR (Deficit Round Robin)

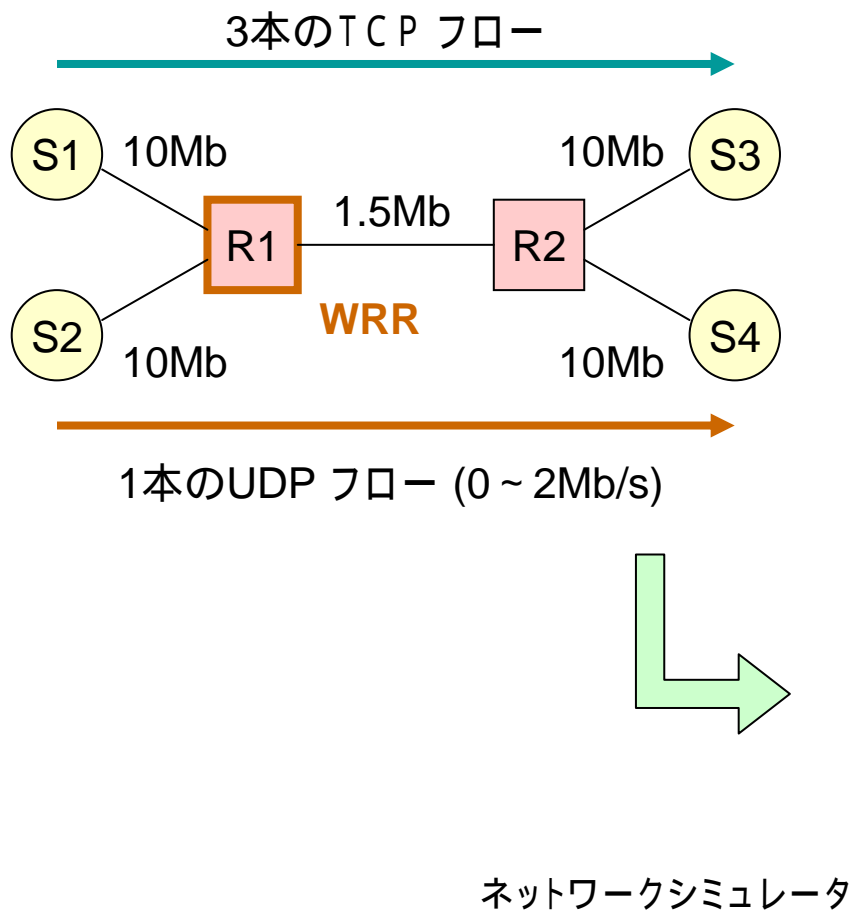


- WRR: 可変長パケットが来ると、長いパケットが優先 (欠点)
- DRR: パケットの「個数」ではなく、「バイト数」で重み付け (改善)

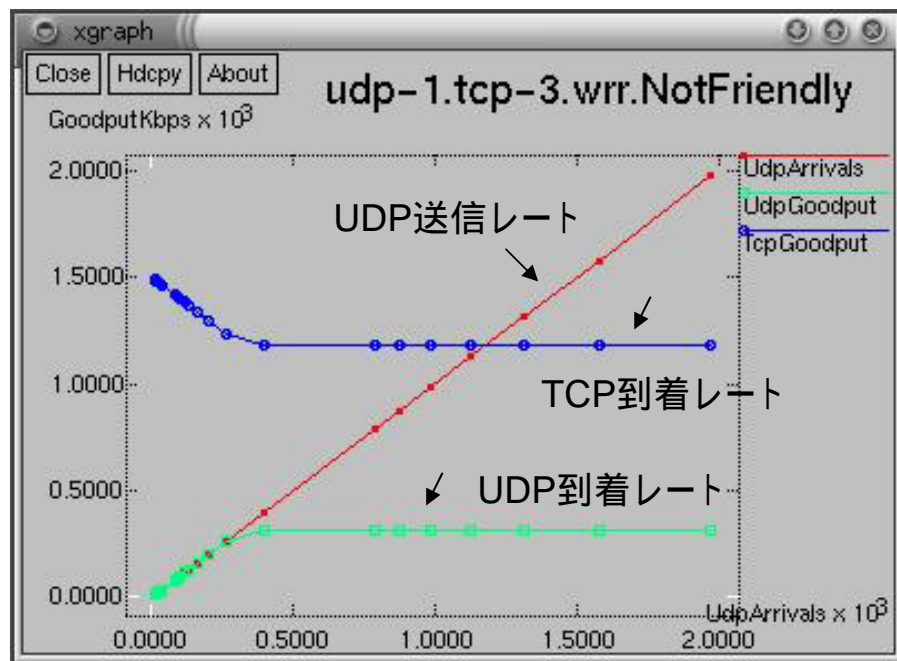


# フローの差別化 (5)

## • WRR の効果



3本のTCPフローと1本のUDPフローがほぼ均等に帯域をシェアしている。クラス許容量を超えたUDPは廃棄。



# これで QoS 課題は解決か？

- インターネット電話、インターネット放送にとって、より望ましい通信環境が提供されるのは明らか (大きな改善)
- ユーザ数の増加に伴う帯域ブローカ (ポリシーサーバ) の負荷の増大  
ポリシーサーバの階層化 (電話網・ATMに近づく...)
- ユーザ数の増加に伴う制御トラヒックの増大  
EF / MPLS は制御トラヒック収容のため？
- SLA に従わないユーザを正しく排除できるか？ インターネットの共有アクセスの利点が失われないか？  
適切なアドミッション制御、クラス分類、メータリング
- マルチキャストの大規模化に対応できるか？

他いろいろ ...