T2 TCP/IP再入門
～わすれちゃいけないUDP ICMP～

InternetWeek2014
JPNIC 高田 寛
JPNIC 岡田 雅之

一般社団法人 日本ネットワークインフォメーションセンター

忘れちゃいけないインターネットの概要

インターネット どう変わった？

- 世界最大のコンピュータネットワーク
  - 20年前：4400万ホスト　　今：10億ホスト
  - 20年前：1億5000万ユーザ　今：55億ユーザ

- 世界最大の情報ネットワーク
  - 20年前：メール　　今：チャット等
  - 20年前：WWW　　今：SNS/ゲーム等

- TCP/IP技術を用いたコンピュータネットワーク
  - 100Gイーサ、クラウドやSDNなどの変化があってもこの部分の基本は変化無し

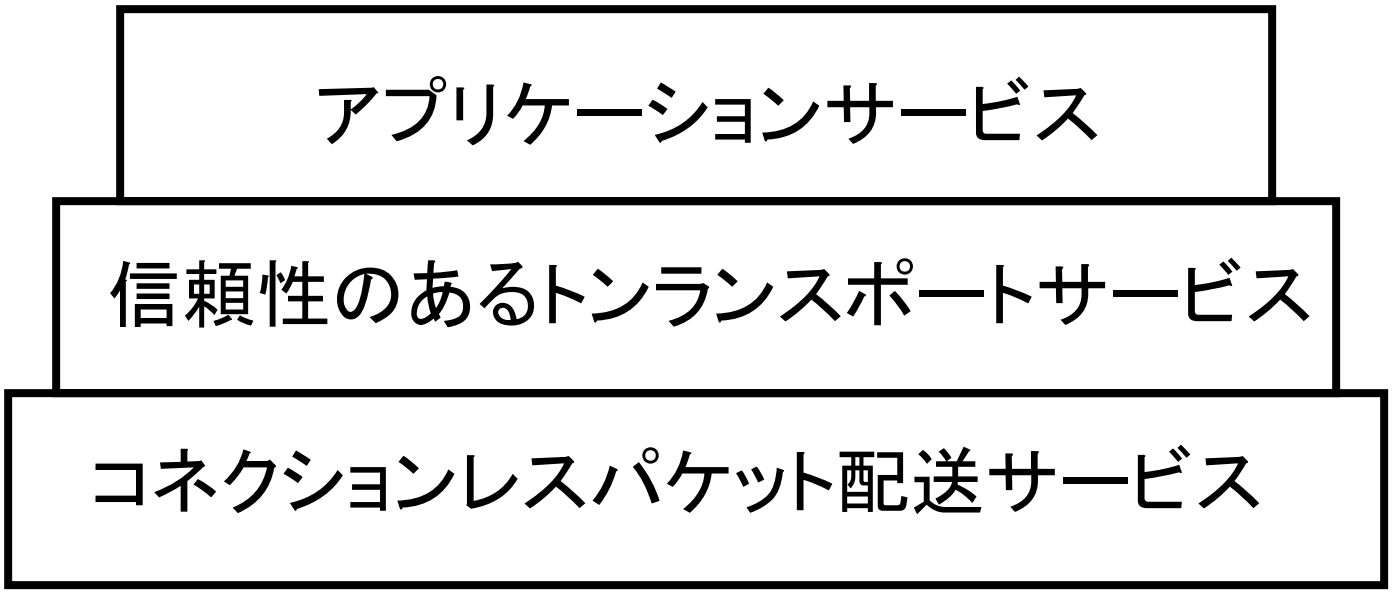
標準化

- 国際標準 (international standard)
  - 母体: 政府，通信事業者(=電話会社)
  - 標準化プロフェッショナル
    - ✓実装より標準に重き
    - ✓標準が技術水準(要求)に追いつかない
- 業界標準 (de fact standard，事実上の標準)
  - 母体: ユーザとメーカ
  - 実装プロフェッショナル
    - ✓標準より実装に重き

インターネットの標準

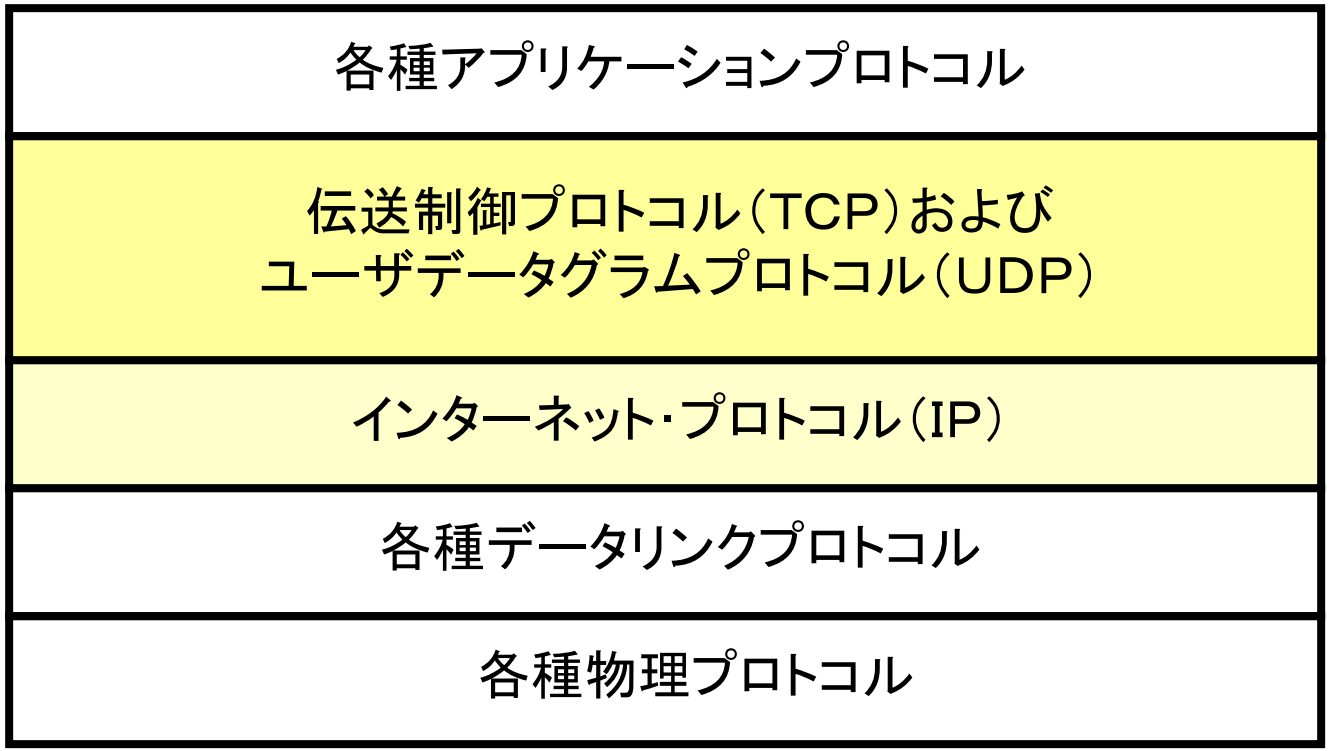
- 世界のインターネット技術者がオープンな場で検討
  - IETF – Internet Engineering Task Force
  - IAB – Internet Architecture Board
  - ISOC – Internet Society インターネット学会
- 技術資料はRFCとして公開
  - Internet Draft
  - RFC – Request for Comments
    - ✓もはやRFCは７０００番台
    - ✓しかしTCP/IPに関係するものは変化が小さい

インターネットの3階層



インターネットで使われるプロトコル

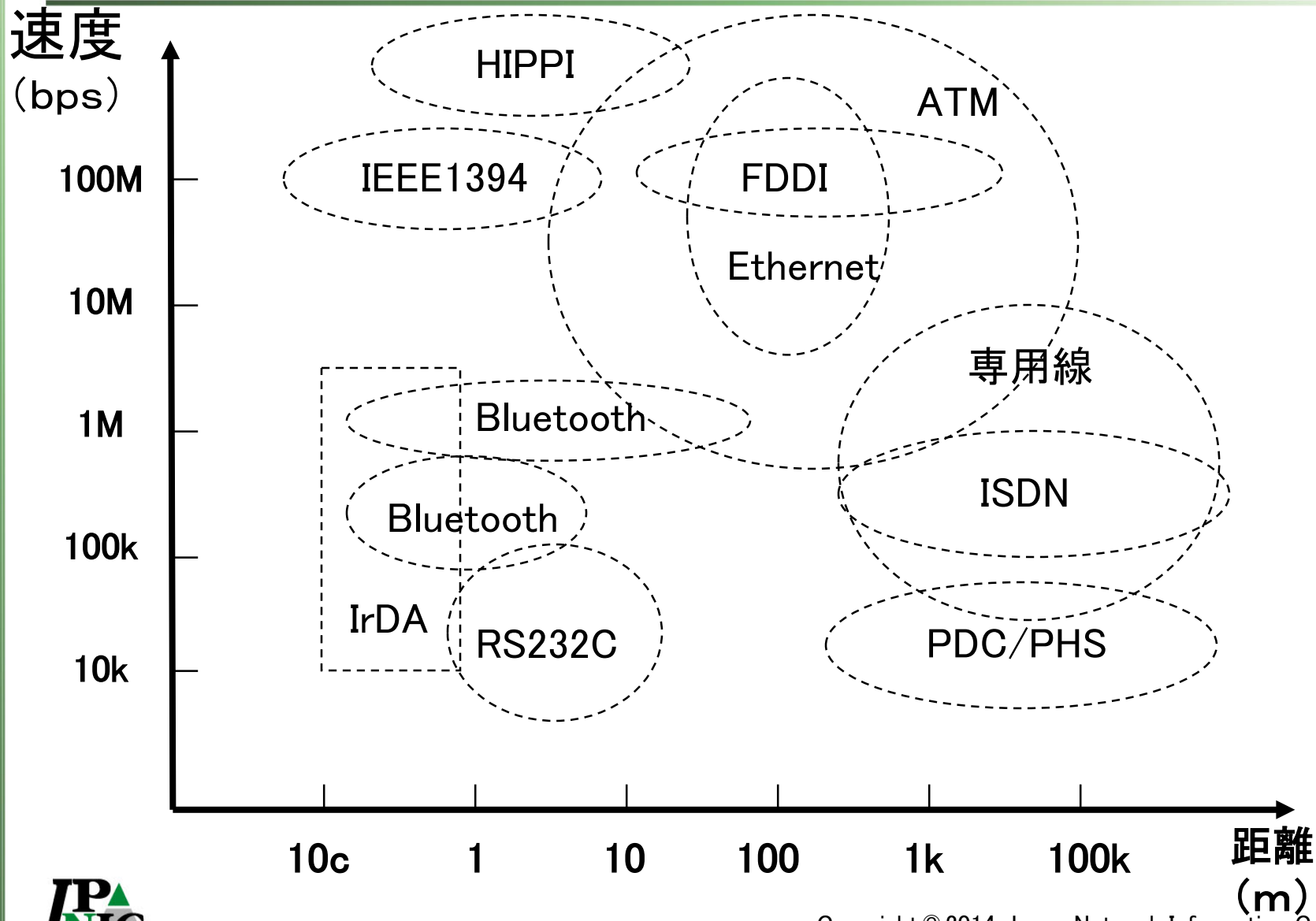
- TCPとIPを中心にしたプロトコルスタック



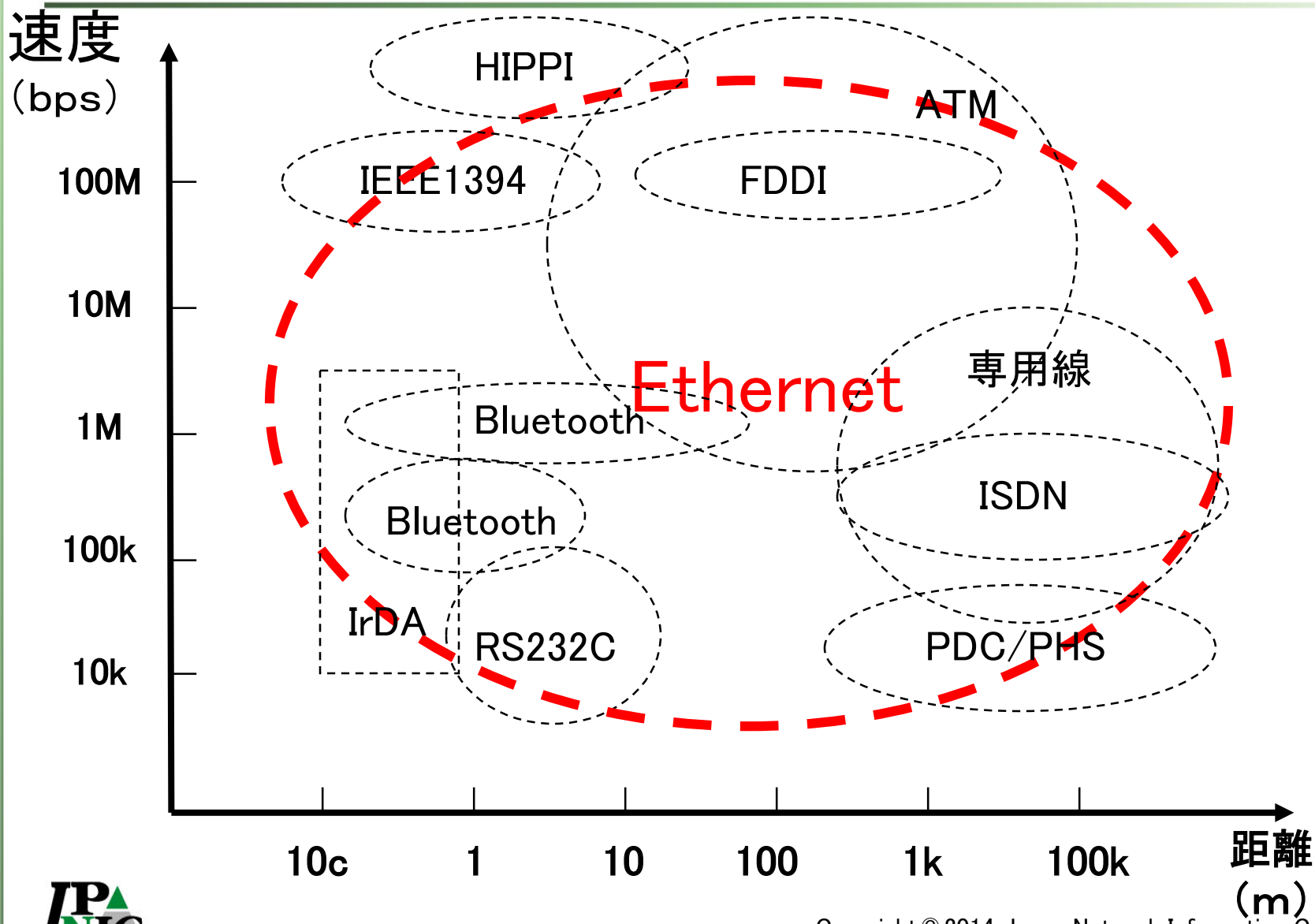
インターネットは

- 複数のネットワークを相互に接続
  - 論理的に一つのネットワークに見せる技術
  - 一つのネットワークアーキテクチャでは構成できないネットワークを構成
    - ✓地球規模での広域性
    - ✓低速から高速まで
    - ✓スケーラビリティ
- ポイントはＩＰとＩＰアドレス
  - ＩＰアドレスという論理アドレスによって，インターネット上のノードを識別

20年前のネットワークアーキテクチャ

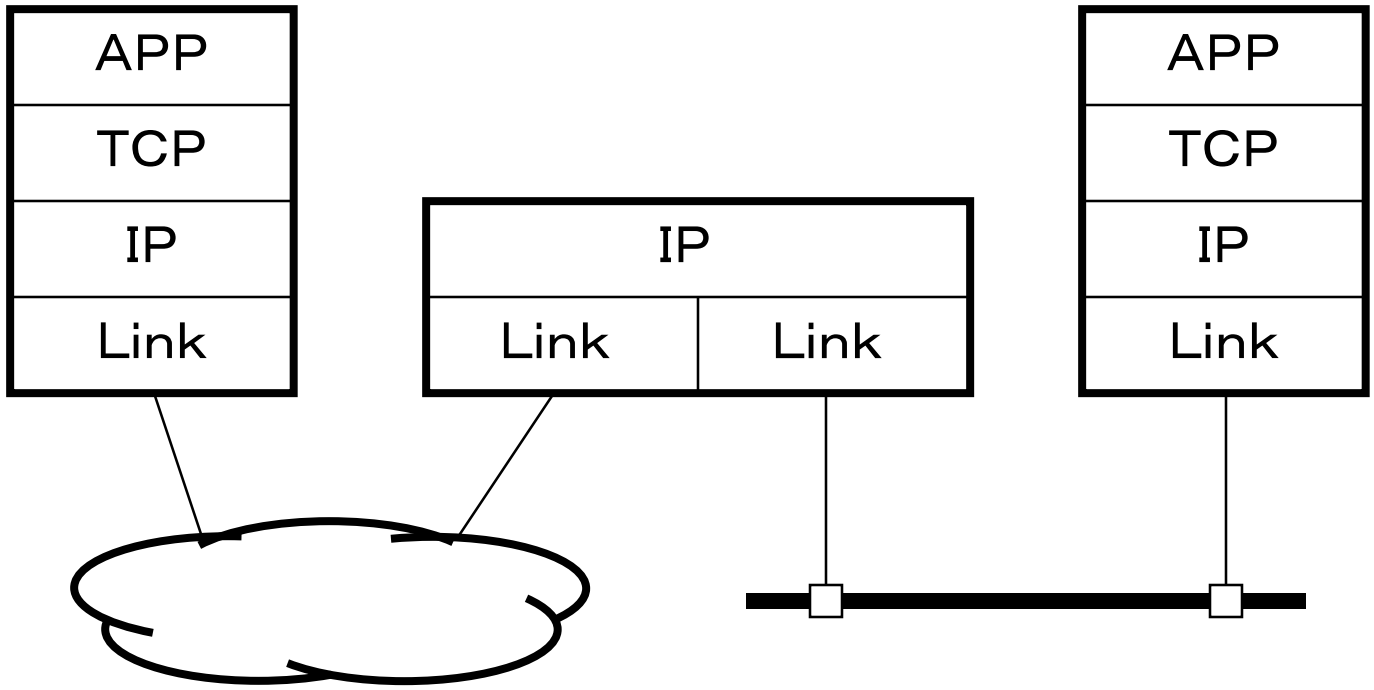


今のネットワークアーキテクチャ

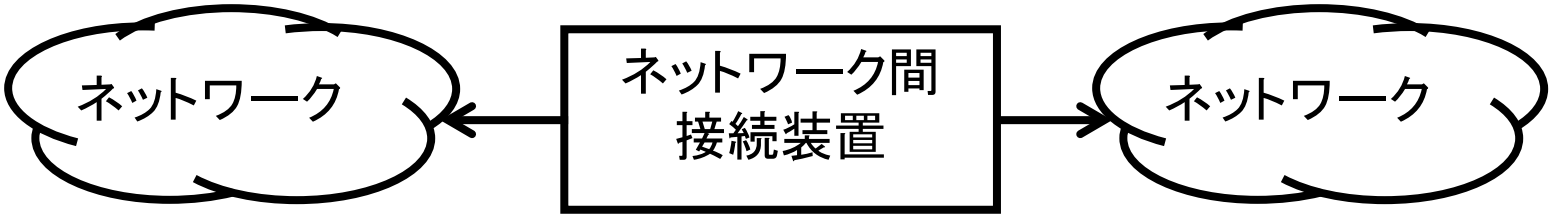


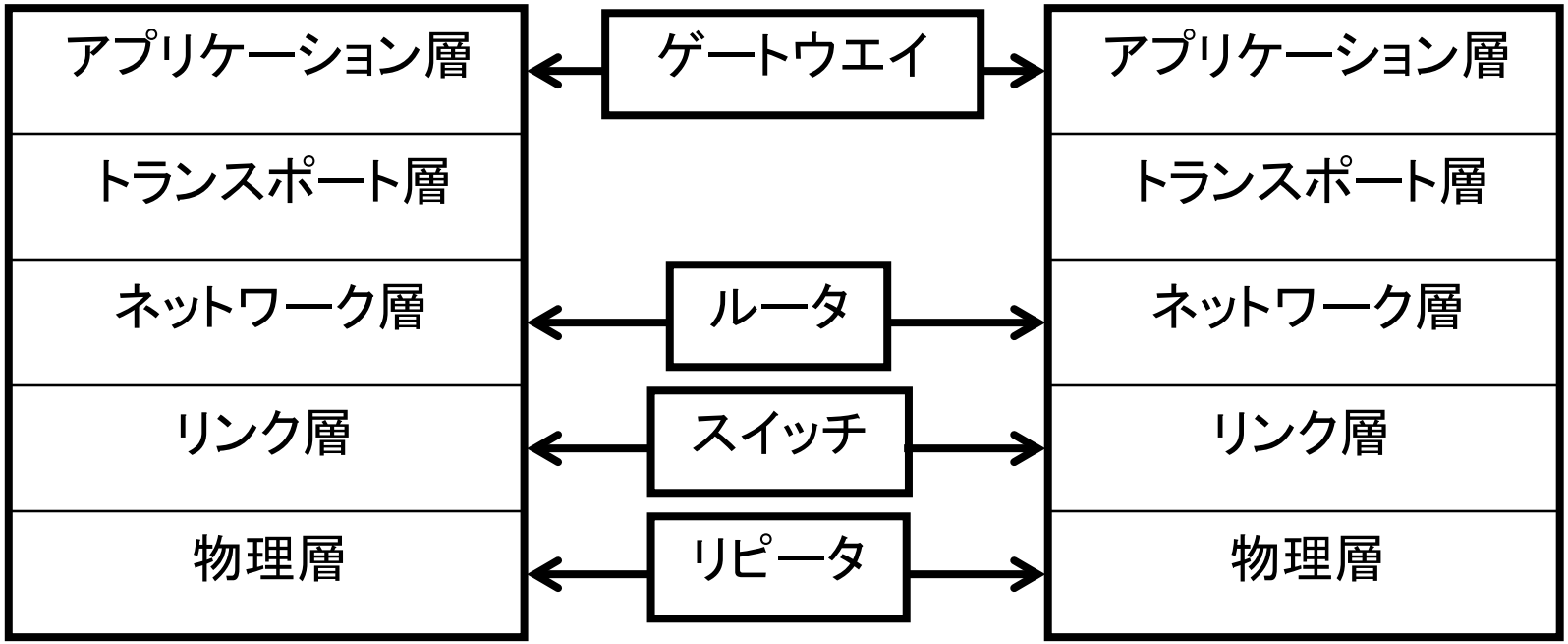
インターネットプロトコル（ＩＰ）

- 異なったメディアを統合して論理的なネットワークに



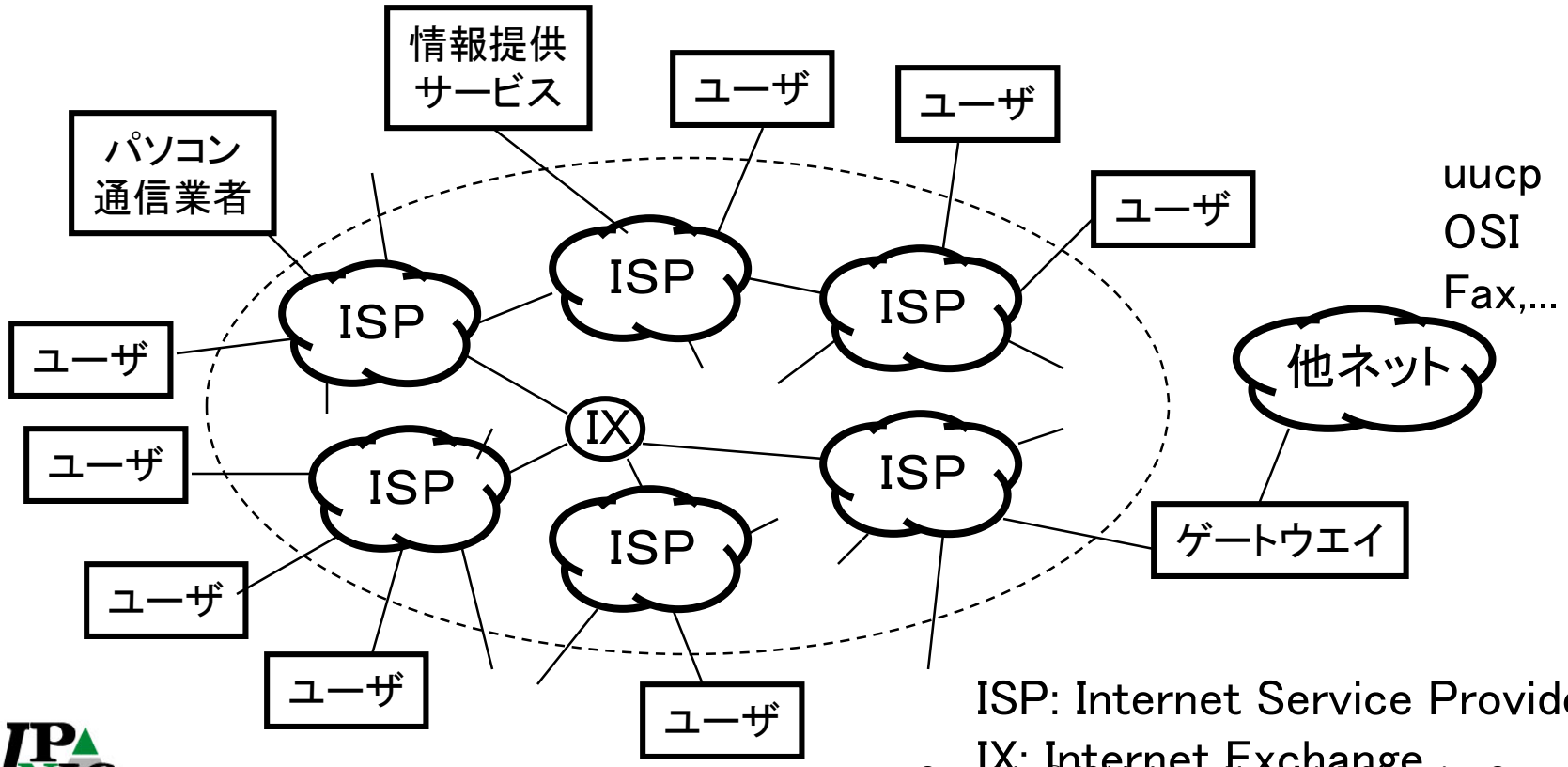
ネットワーク接続装置





インターネットの構成

- ＩＳＰ（Internet Service Provider)の相互接続
  - 商用プロバイダ
  - 学術プロバイダ



IP: Internet Protocol

機能

- RFC 791 で定義
  - RFCは更新されることが多いが現役でRFC791
- データグラムを他のホストに配送する
  - 信頼性を保障しない
    - ✓データグラムが紛失するかもしれない
    - ✓データグラムの順番が保証されない
    - ✓データグラムが重複するかもしれない
- フラグメンテーション
- 経路制御
  - 経路制御についてはルーティングのプログラムを！

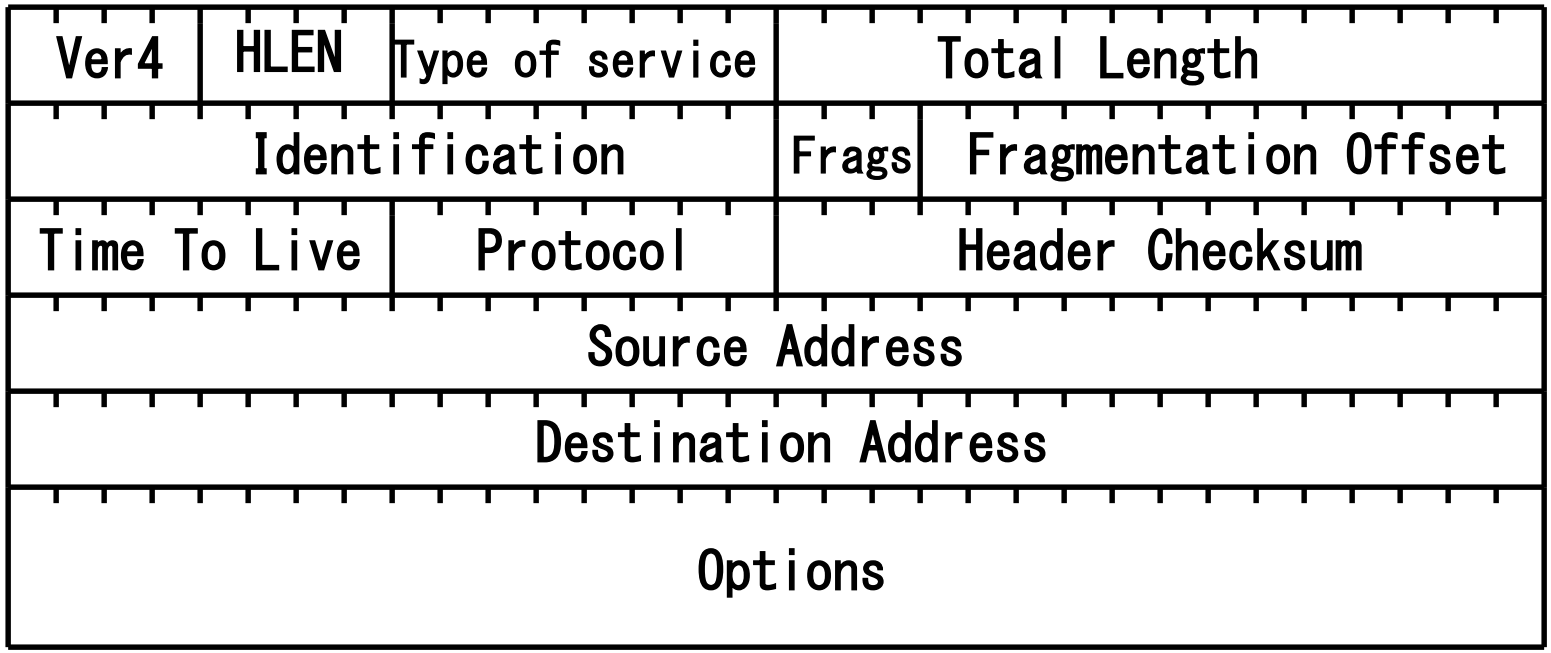
機能

- データ配送の基本単位
- ネットワークの抽象化
  - 物理層，リンク層の違いを隠蔽する
  - 仮想的に単一のネットワークに見せる

概要

- IPデータグラムを他のホストに配送
  - IPアドレスで相手を指定
- データをデータグラムに分割して配送
- コネクションレス
  - 誤り訂正，紛失，到着順序などの処理は上位層で行う
- 経路制御とアドレス
  - さまざまなネットワークを経由
  - ルータによる中継
- フラグメンテーション

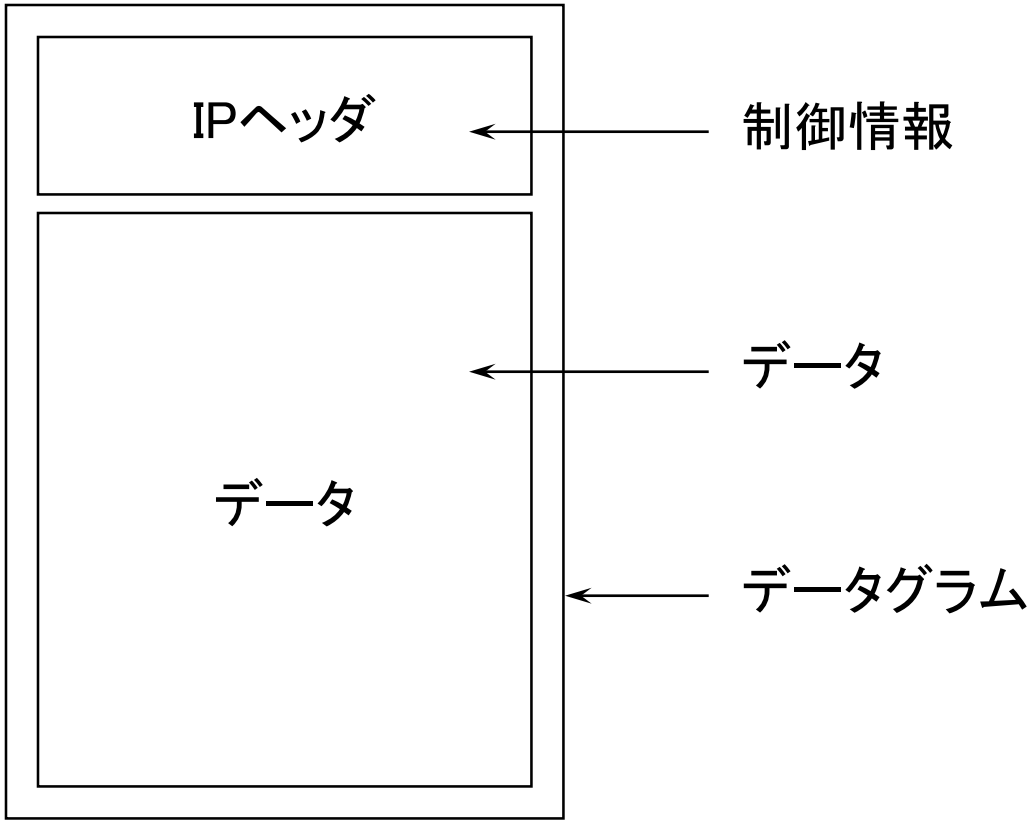
IPv4ヘッダフォーマット



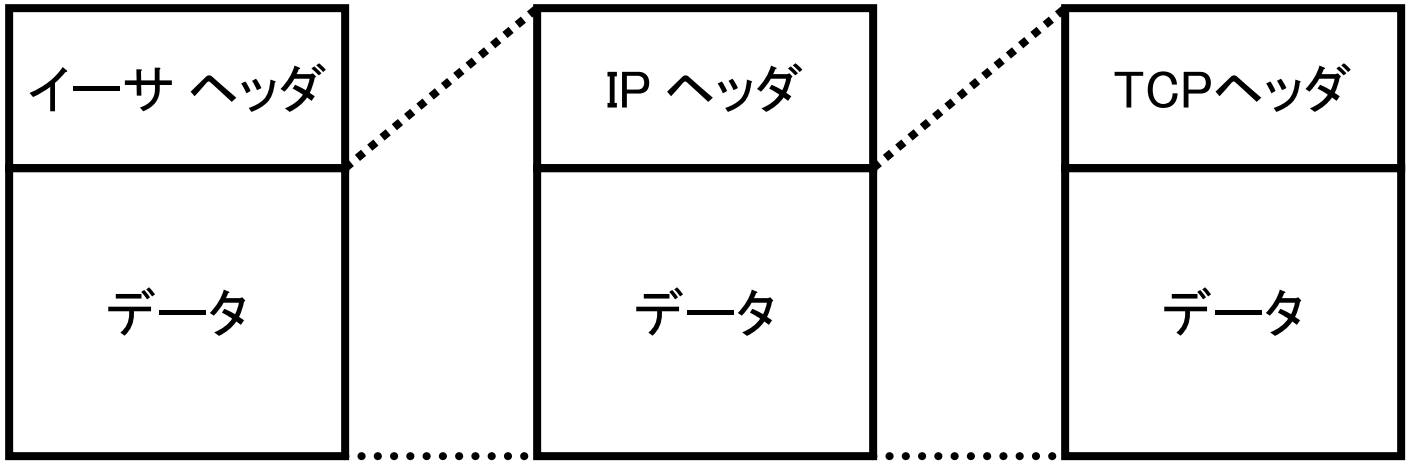
IPデータグラム

- 基本転送単位

  = ヘッダ + データ

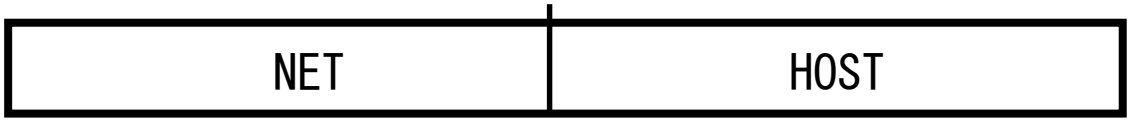


カプセル化



IPアドレス

- 固定長の論理アドレス
  - IP: 32ビット
  - ホストのネットワークインターフェースを識別
- 二つの部分
  - ネットワーク部 ー 経路指定，ネットワークを識別
  - ホスト部 ー ネットワーク内のホストを識別



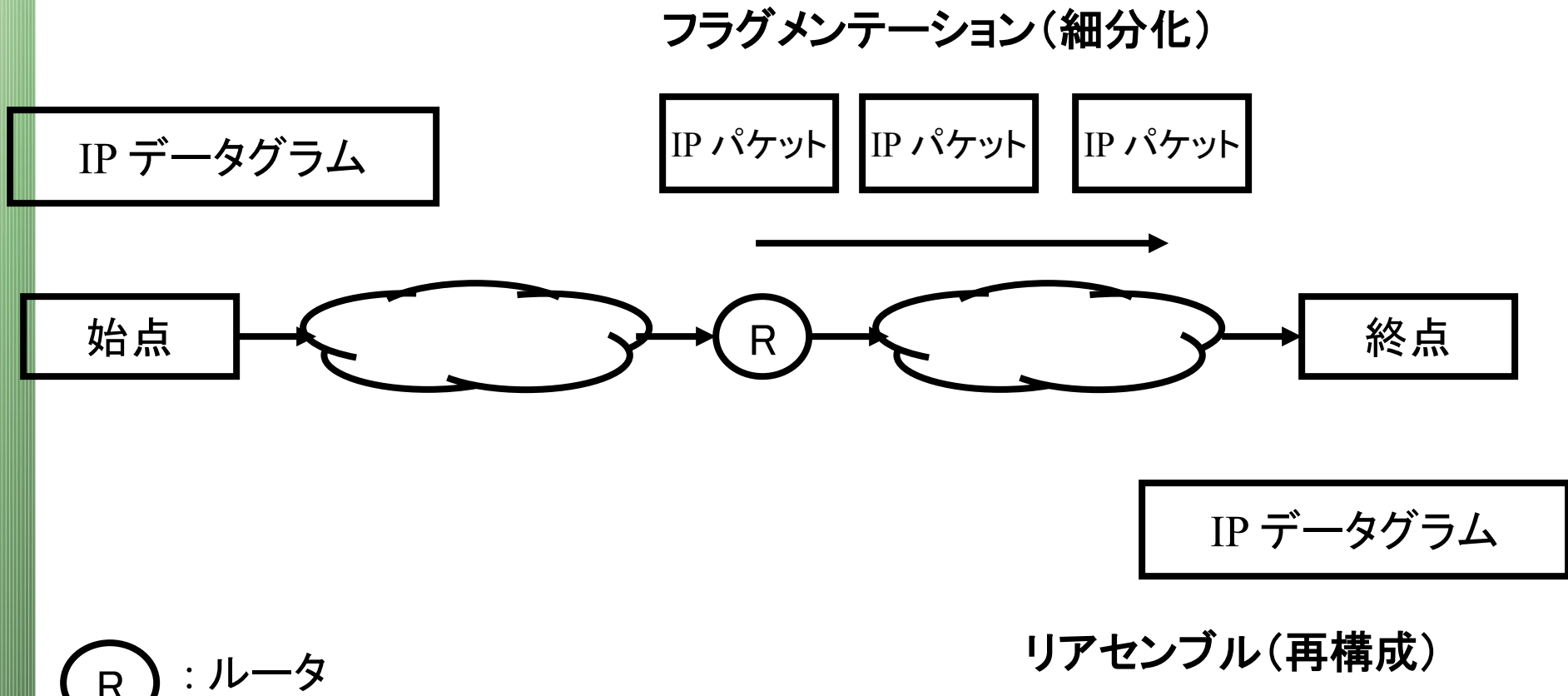
TTL

- Time To Live
  - IP パケットの生存時間(単位: 秒 ←仕様)
- (事故や設定ミスで)IP パケットがたらい回しにされても，そのうち消える
- ルータを経由すると TTL を 1 減らす（←実装）
- TTL が0 になったらルータは，
  - そのパケットを破棄する
  - 送信元にエラーメッセージを返す → ICMP

フラグメンテーション

- リンク層によってパケットの最大長が決まっている
- 最大転送単位: MTU (Maximum Transfer Unit)
- MTU より大きな IP パケットの中継
  - パケットを MTU に合わせて分割
- 最小の MTU は 576 バイト
- フラグメンテーションは，終点ノードで元のIPデータグラムに戻される

フラグメンテーション(2)



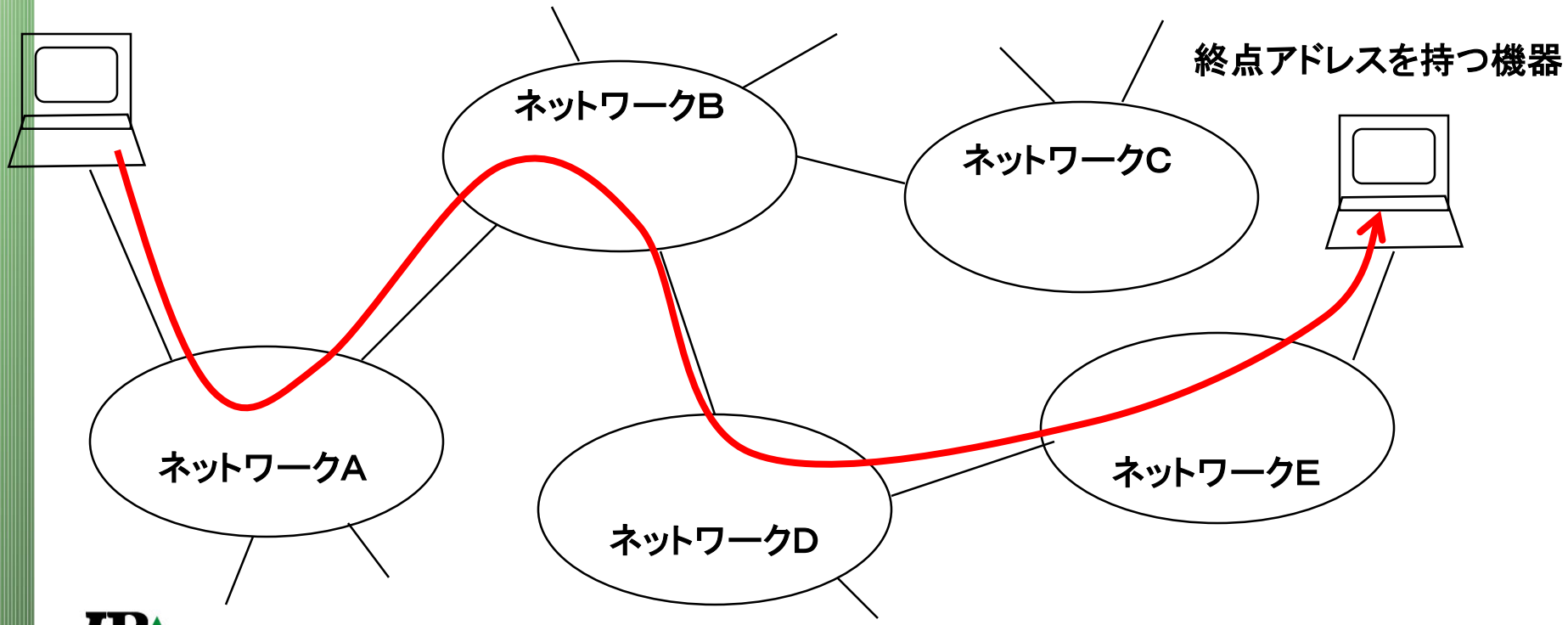
フラグメンテーション(3)

- 途中のルータで再構成しないのは?
  - 経路が一定でない
  - 遅延が発生する
- 途中のルータに負荷をかける
- 終点ノードでもリアセンブルの処理に時間がかかる
- IPv6では．．．
  - Path MTU Discoveryにより経路上の MTU をみつける
  - 初めから最小MTU以下のIPデータグラムを使う

ＩＰアドレスと経路制御

- ＩＰパケットに含まれるあて先アドレス（終点アドレス）によって経路が決まり，正しくあて先に配送される．

始点アドレス	終点アドレス	データ



経路表

- インターネット上のルーターはそれぞれ経路表を持ち，ＩＰデータグラム中の終点アドレスによって，どのインターフェイスに転送するかを決めている

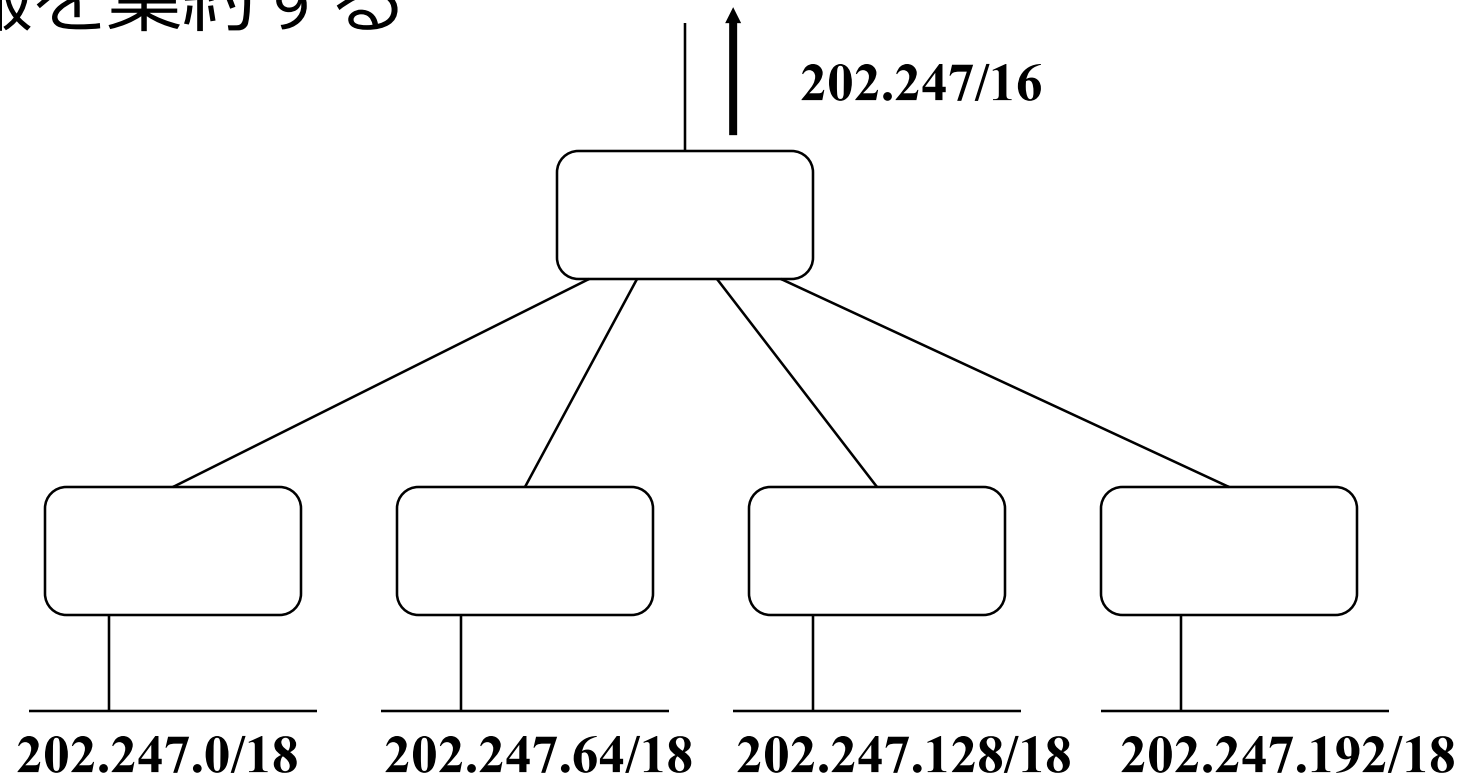
Destination	Gateway	Flags	Interface
127.0.0.1	127.0.0.1	UH	lo0
192.168.3.179	192.168.3.11	UGH	ed0
133.2.0.0	133.2.2.253	UG	ed1
202.2.8.16	202.2.8.26	UG	vx0
172.16.0.0	133.2.2.3	UG	ed1
202.2.8.24	202.2.8.27	U	vx0
131.41.0.0	133.2.2.253	UG	ed1
203.18.98.0	202.3.8.26	UG	vx0
133.2.74.0	133.2.2.253	UG	ed1
133.2.2.0	133.2.2.7	U	ed1
133.3.0.0	133.2.2.253	UG	ed1
192.168.3.0	192.168.3.2	U	ed0
202.2.6.0	202.3.8.26	UG	vx0

経路制御

- IPデータグラムの中継経路の制御
  - 終点アドレスによる配送経路決定
    - ✓ルータでの経路表の管理
    - ✓hop ｂｙ hop
  - 発信元が配送経路を指定
    - ✓source routing
    - ✓パケット中に配送経路を明示的に指定
- 経路はアドレスとマスク長により管理
- 経路表が大きくならないためのアドレス割り当て
  - ＣＩＤＲ（Classless InterDomain Routing)

アドレスの階層構造

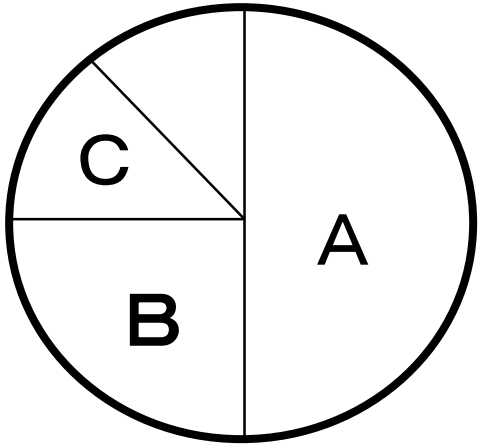
- アドレスを階層的に割り振ることによって経路情報を集約する

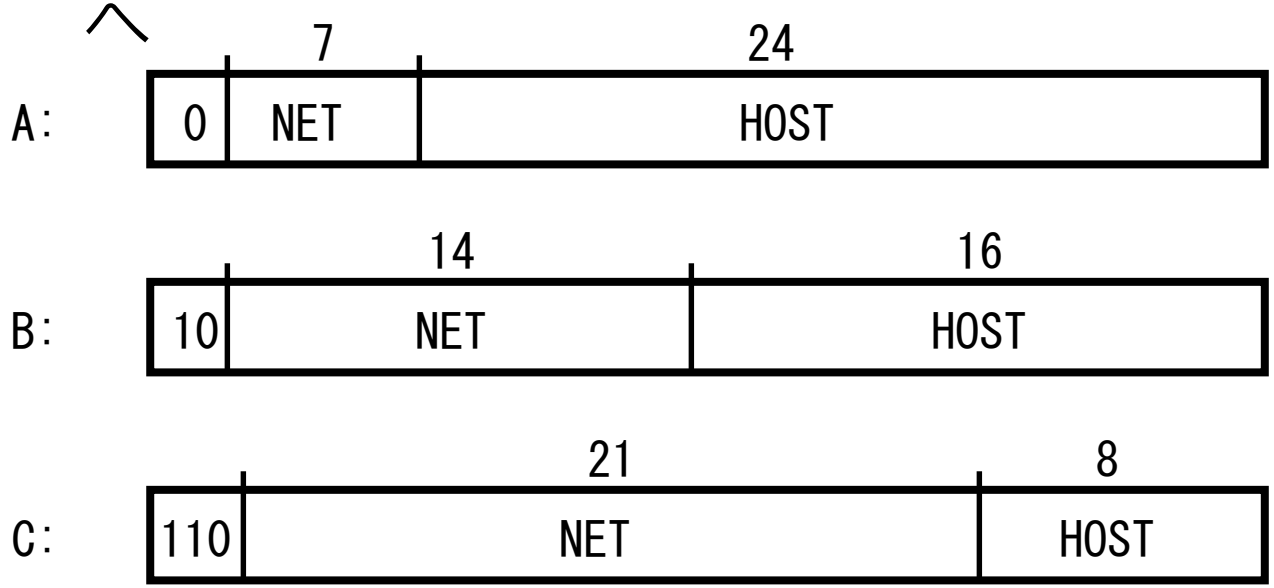


- 地域や国，プロバイダにアドレスブロックを割り振り，下位ネットワークの経路情報を１つにまとめる

アドレスのクラス

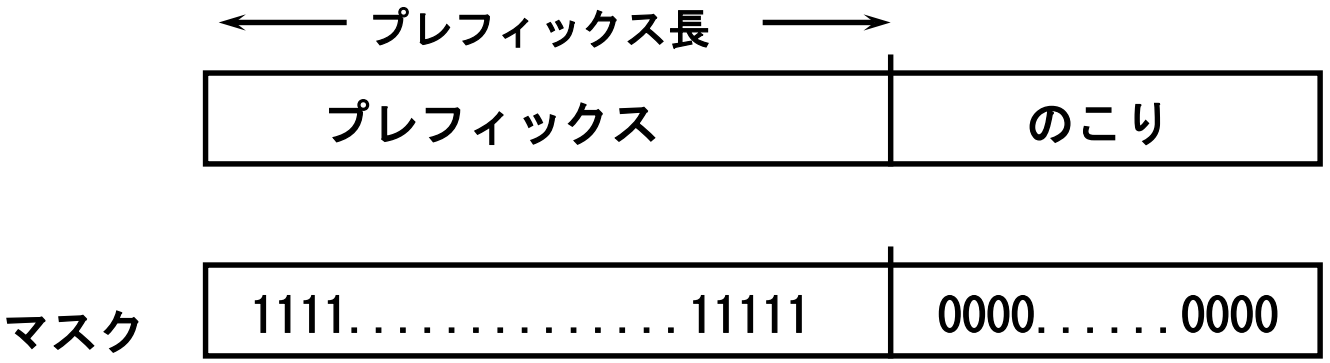
- 最初のビットでクラスを指示
- クラスフルアドレスとも
  - その非効率さからクラスレスへ





クラスレスのアドレス体系

- ネットワーク部は可変



- プレフィックス長によるネットワーク番号表記

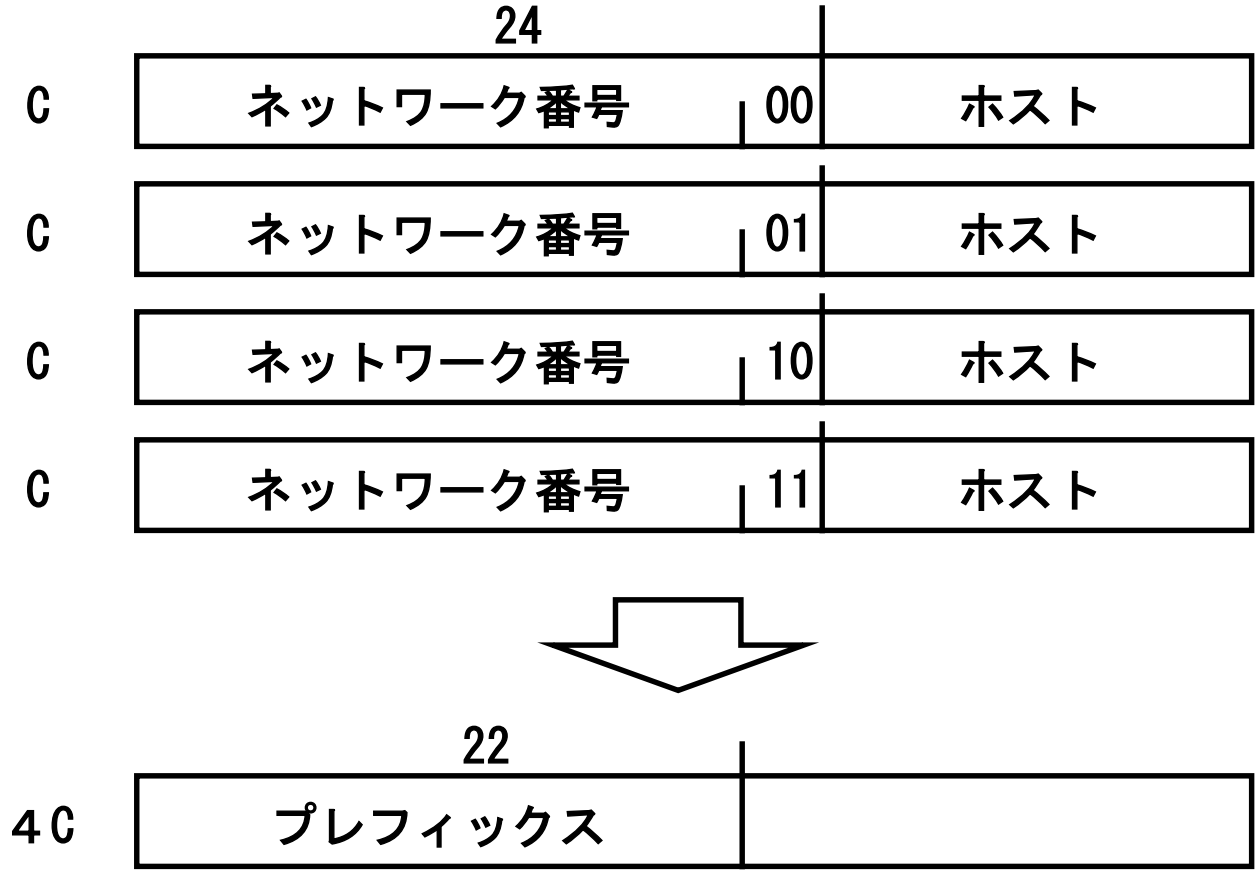
例）133.201.2/24

プレフィックス長

ネットワーク番号

アグリケーション(集約)

- 連続したネットワークのブロック化



アグリゲーションの特徴と問題点

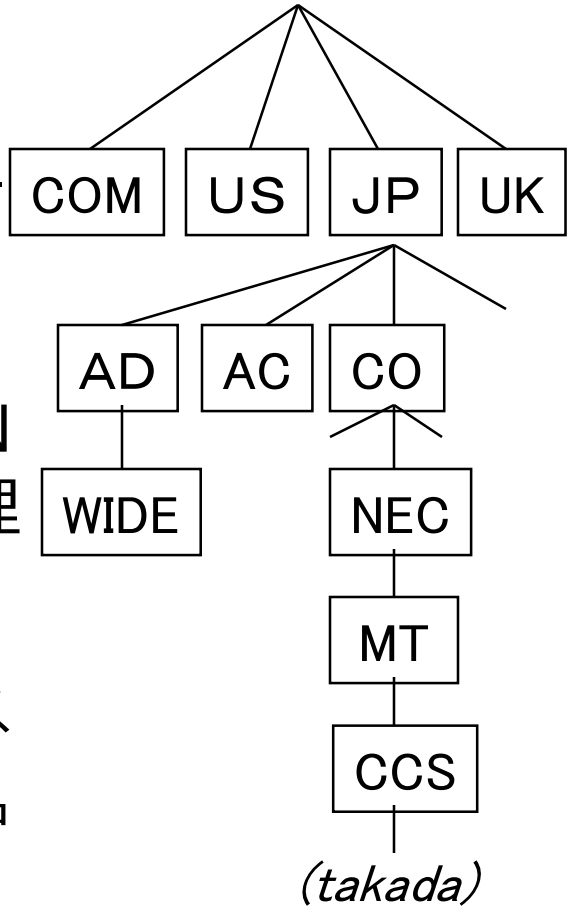
- アドレス空間を広くして，経路情報の増加を押える
  - 経路爆発問題：最近50万経路を超えました。
    - ✓ルータやスイッチは512,000にちょっとした問題が・・・
- ネットワークトポロジに対応して割り当て
  - 上流のトポロジの変更でアドレスを変更する必要
  - ＩＳＰの変更でも
  - マルチホームの場合は？

ＩＰアドレスと名前

- 機械，プログラムはアドレス(数字)を好む
- ユーザ(人間)は名前を好む
- 例
  - 社員名と社員番号
  - 住所と緯度経度

DNS

- Domain Name System, RFC1034. 1035
  - IPアドレス，ドメイン名，メールサーバなどを管理する分散データベース
  - ドメイン名階層にあわせたＤＮＳサーバの配置による分散管理
- 引きかた
  - 正引き　ドメイン名→ＩＰアドレス
  - 逆引き　ＩＰアドレス→ドメイン名



takada@ccs.mt.nec.co.jp

ICMP: Internet Control Message Protocol

機能

- IP データグラム配送のエラー報告と制御
- RFC 777, 792 で定義

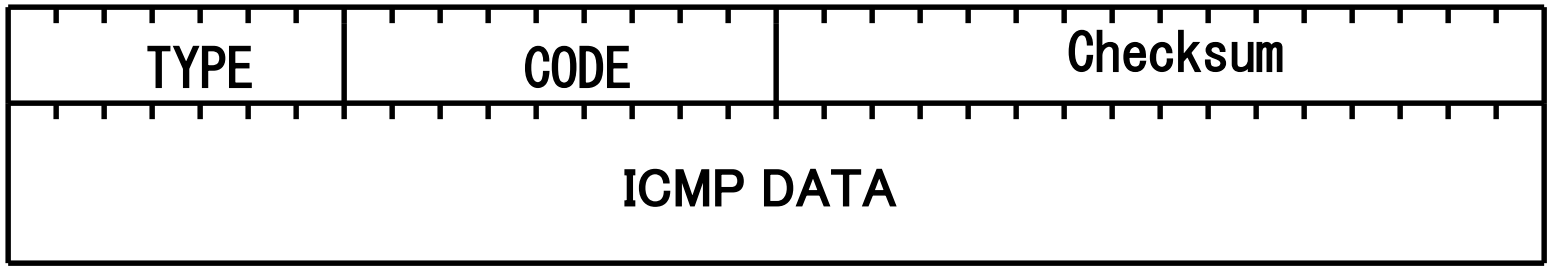
機能

- 配送時のエラーを発見
- 網状態の問い合わせ

概要

- 一般にIPと一緒に実装
- IPの機能を補う
- データグラム配送中に起きたエラーを通知
- 網・宛先の状況を問い合わせ

ICMPフォーマット



配送方式

- IPデータグラムのデータ
- データが紛失する可能性あり
- ICMP パケットがエラーの原因になったら?
  - エラーの通知はしない

主なメッセージの種類

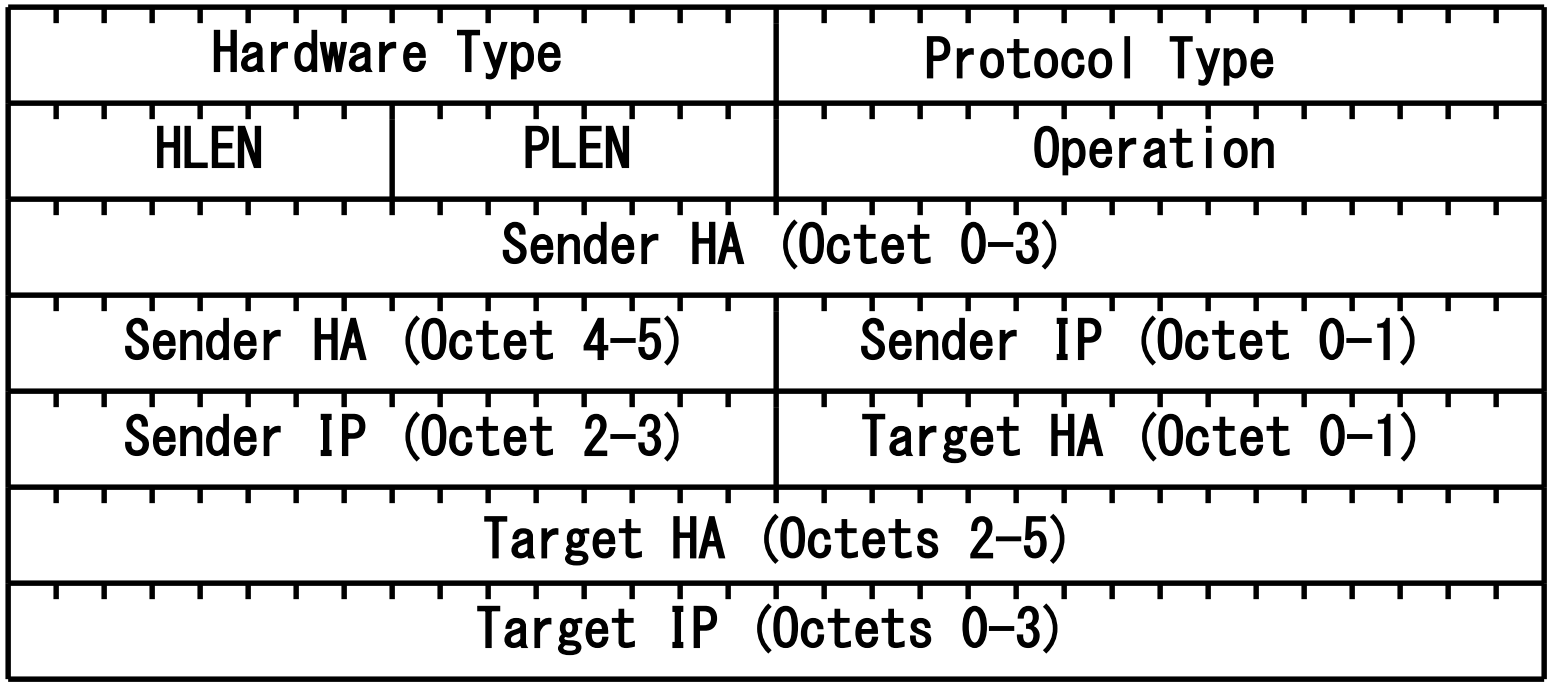
- エコー要求とエコー応答（ｅｃｈｏ，echo ｒｅｐｌｙ）
- 終点到達不可能報告 (host, net, port, protocol)
- 始点抑制 (source quench)
- 経路変更要求（ｒｅｄｉｒｅｃｔ）
- 全体時間経過済み（ｔｉｍｅ ｅｘｃｅｅｄ）

ARP: Address Resolution Protocol

機能

- IPアドレスから物理アドレスへの問い合わせを行うプロトコル
- RFC 826 で定義

パケット形式



アドレス解決問題

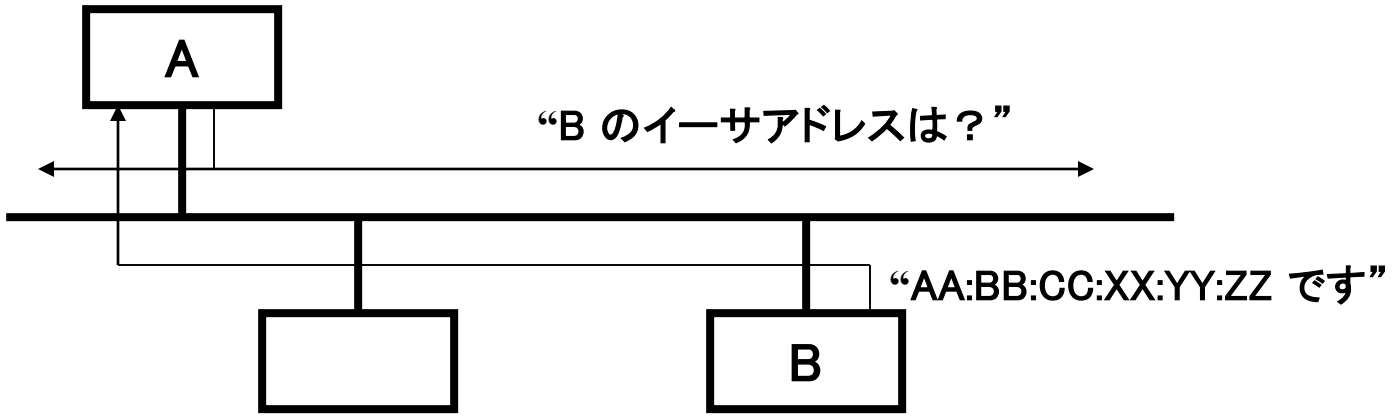
- 実際の通信では，リンク層の識別子を使う
  - イーサネットアドレス
  - ATM アドレス
  - FDDI アドレス
- IPアドレスからリンク層識別子への変換機構が必要
  - それぞれのインタフェイスモジュールが対応

静的な解決

- ホストに，通信相手の識別子とIP アドレスの組を設定しておく
- 長所
  - 簡単
  - 速い
- 短所
  - マシンの追加やボードの変更が発生すると，各ホストで再設定が必要になる

動的な解決

- ブロードキャスト型メディアでの解決方法
  - イーサネット
- 解決したいアドレスをブロードキャストで，全員に問い合わせる
- 対象のホストは，それに答える



動的な解決(2)

- 長所
  - ネットワーク構成やホストが変わっても大丈夫
- 欠点
  - 余計な通信が発生
  - ブロードキャスト型でないと使えない
  - 嘘つきがいるとセキュリティ上の問題が発生する
    - ✓通信を盗聴される

UDP: User Datagram Protocol

機能

- 信頼性を保証しないデータグラム配送
  = IPパケット+ポート番号+チェックサム
- RFC 768 で定義
- TCPと比較して高速のため、
  - 最近になってオンラインゲームなどに需要が高まる
  - でも信頼性がまったく確保されていない・・・

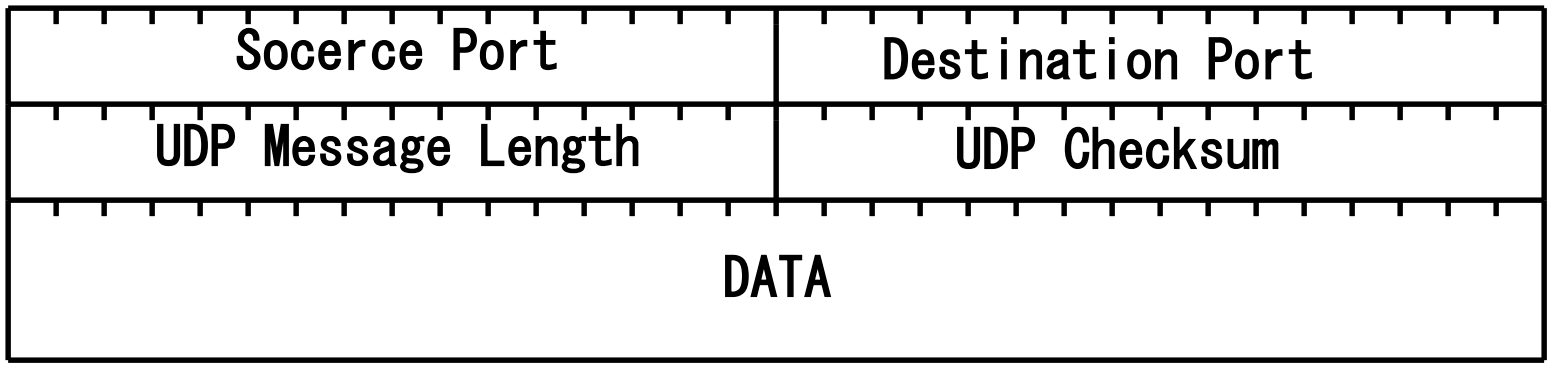
目的

- ユーザ(プログラマ) が利用できるデータグラム通信
  - NFS, TFTP, DHCP, DNS, ...

概要

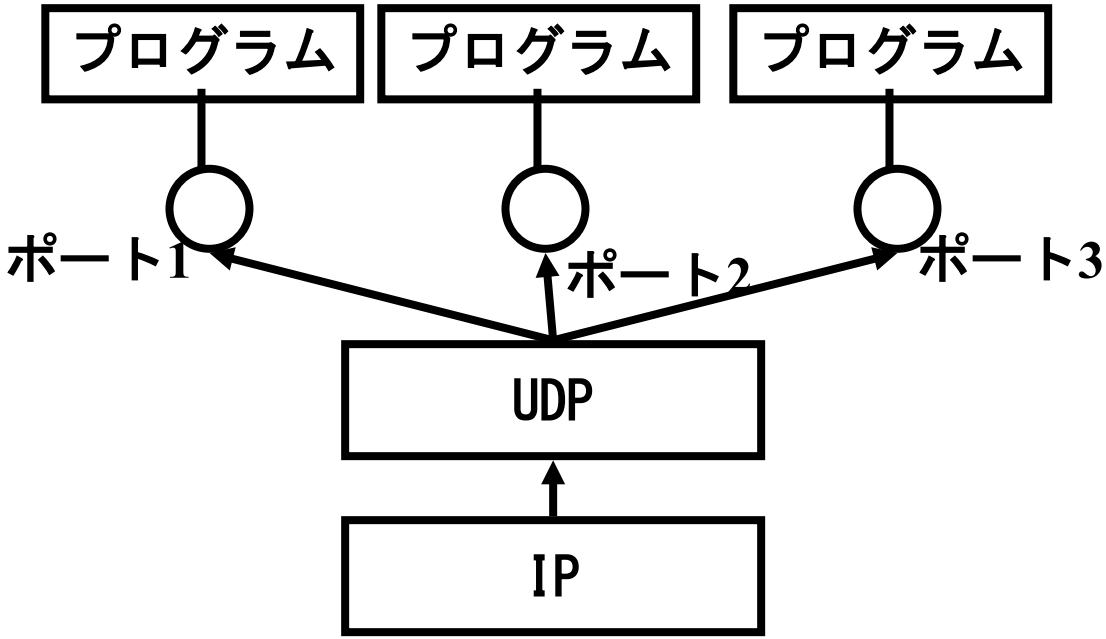
- コネクションレス型の通信形態
- ホスト間のオーバヘッドの少ないデータ転送を提供
  - コネクション開設・解放の手続きが不要
  - データの完全性（順序の保証、エラーパケットの再送など）を保証せず → 上位層で解決
- 状態の管理はアプリケーション任せ
- ポート番号によりサービスを選択

パケット形式



ポート

- 終点のサービスを識別するのに利用
  - IPアドレスは終点ホストだけを識別



代表的なサービス

- domain(DNS)　53/UDP
- DHCP　67, 68/UDP
- SNMP　161/UDP
- NFS　2049/UDP

TCP: Transmission Control Protocol

機能

- 信頼性のあるデータ配送
  - ストリーム型
  - バーチャルサーキット(コネクション型)
  - バッファ付き転送
  - 全二重

目的

- ユーザが下位プロトコルを意識せずに通信を行う

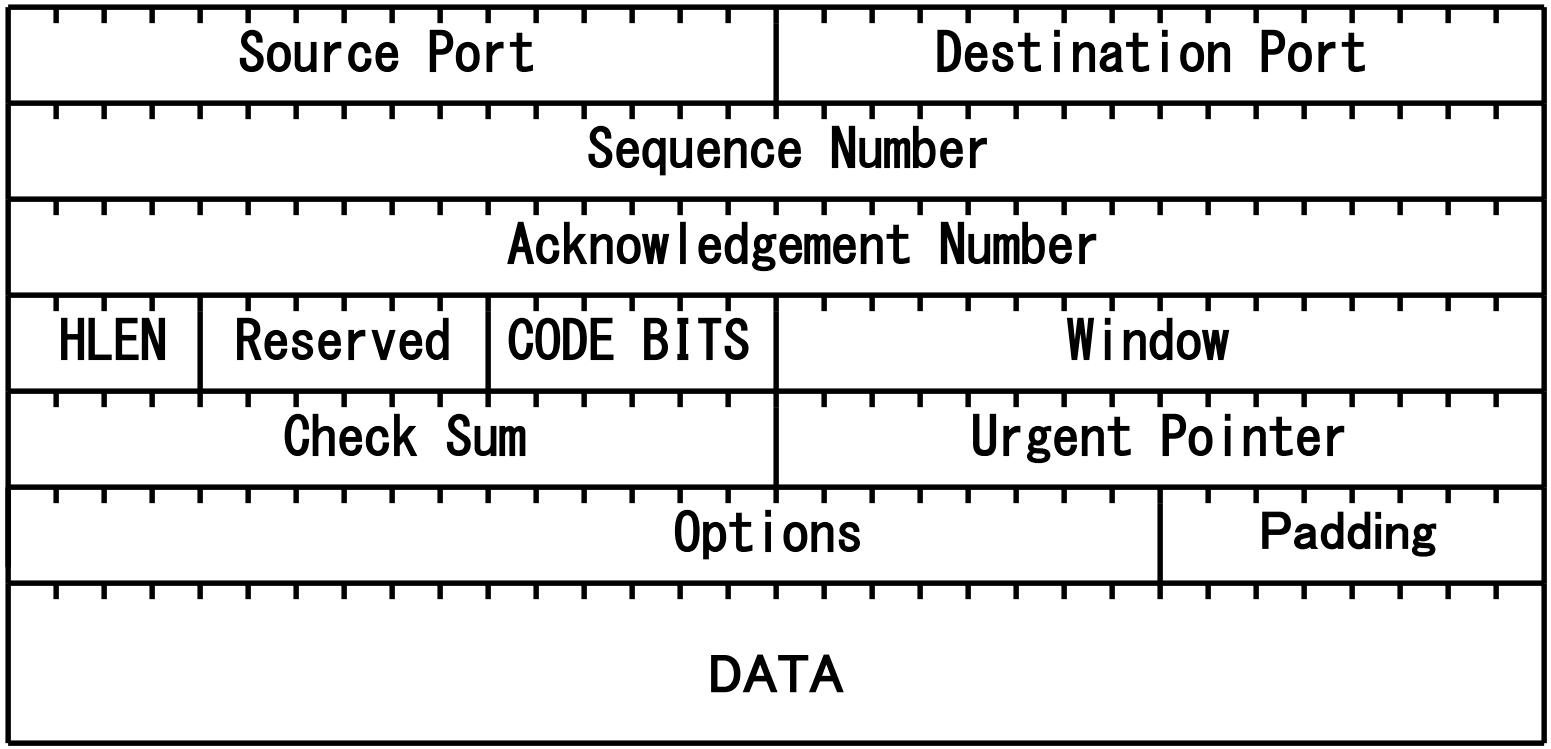
概要

- RFC 793 で定義
- 転送単位: セグメント
- ホスト間の信頼性の高い通信を提供
- 再転送付き肯定応答
  - 肯定応答: ACK
- スライディングウィンドウ
- バイト単位のウィンドウによるフロー制御
- 輻輳制御
- コネクションとポート

概要(2)

- コネクション型の通信形態
- ストリーム型のデータ転送
  - 単純バイト列
  - レコードの概念は無い

パケット形式



フィールド

- Source Port: 始点ポート
- Destination Port: 終点ポート
- Sequence Number: シーケンス番号
- Acknowledgement Number: ACK 番号
- HLEN: ヘッダー長
- CODE BITS: セグメントの内容を示すビット
  - SYN
  - FIN
  - RST
  - ACK
  - URG
  - PSH

フィールド(2)

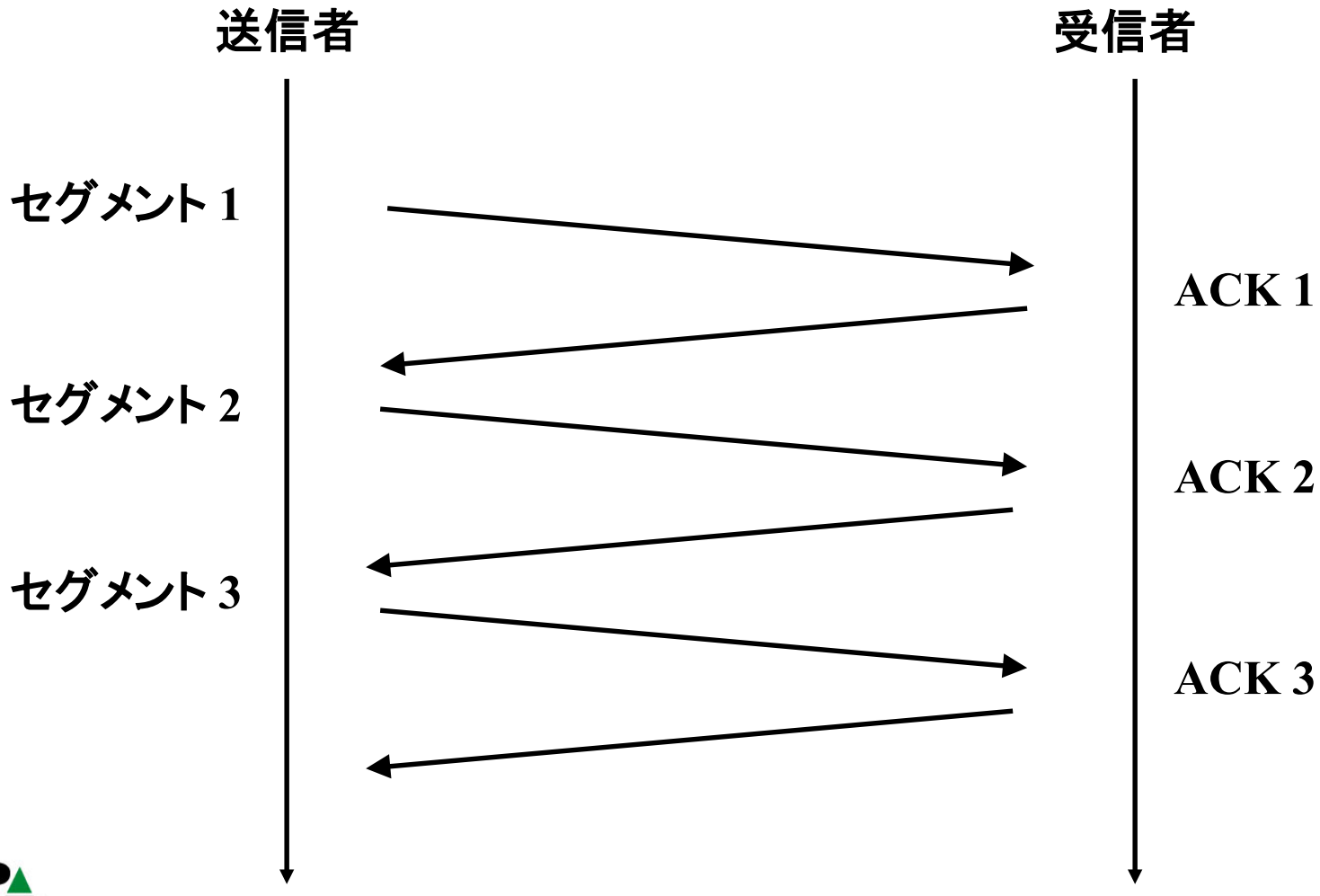
- Window:　　ウィンドウサイズ
- Check Sum:　チェックサム
- Urgent Pointer:　　緊急データポインタ
- Options:　　オプション

セグメント

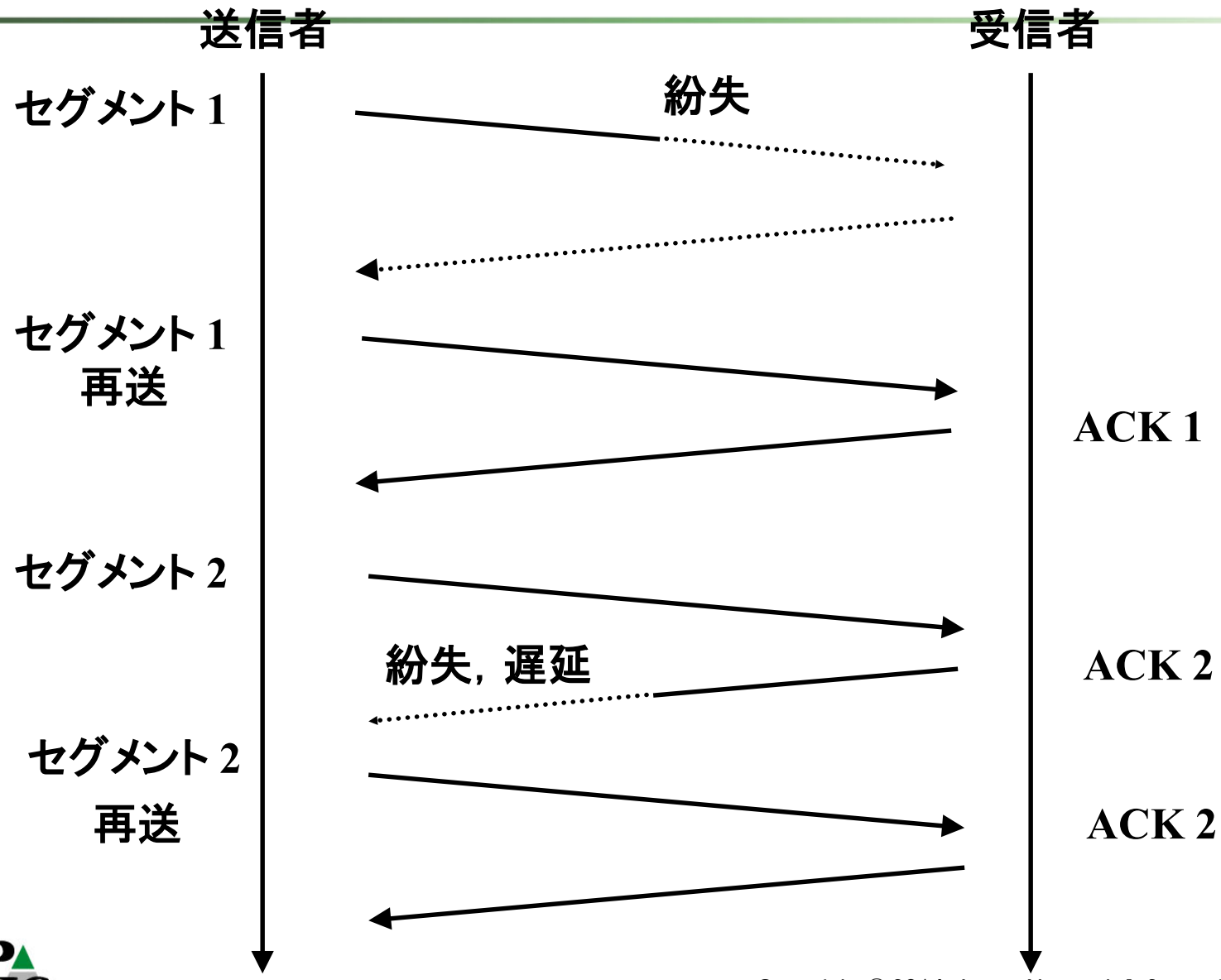
- TCP における転送単位
- シーケンス番号: データ(オクテット)に割り当てられた順序数
  - 32ビットの整数
  - 2^{32}-1 の次は 0 (循環する)
- ネットワークの状況で大きさが変化

再転送付き ACK

- セグメントごとに ACK が必要



再転送付き ACK(2)

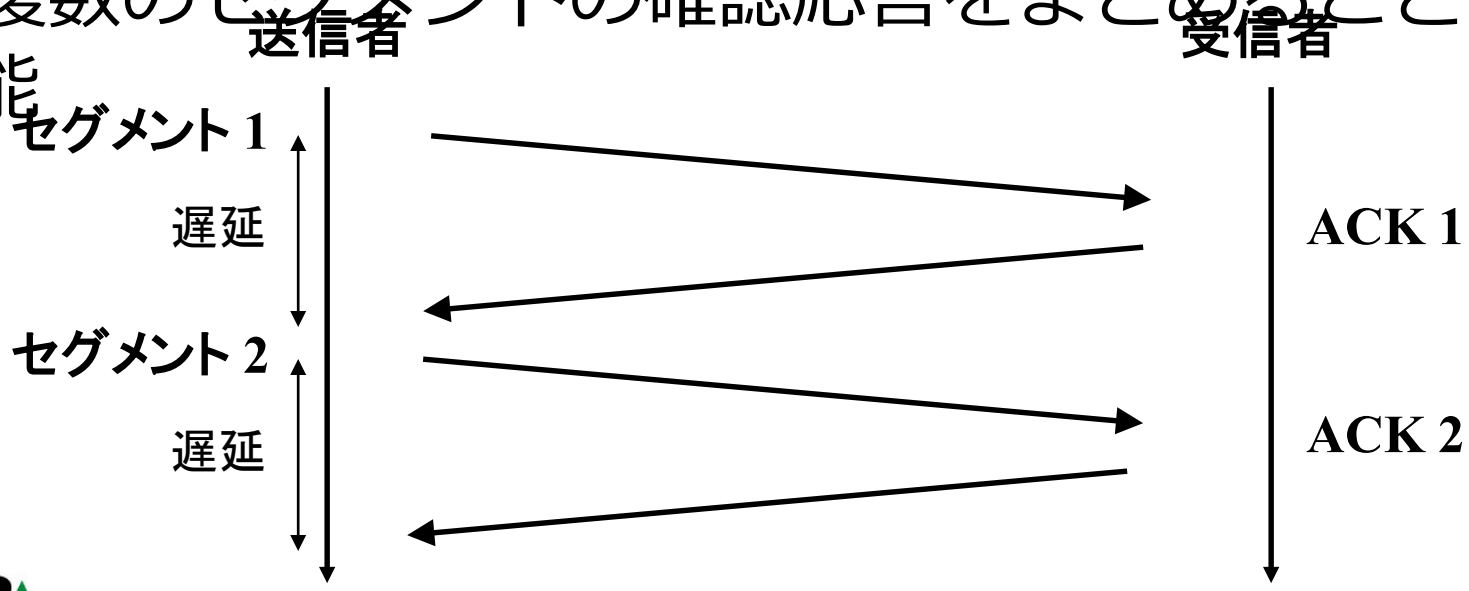


再転送付き ACK(3)

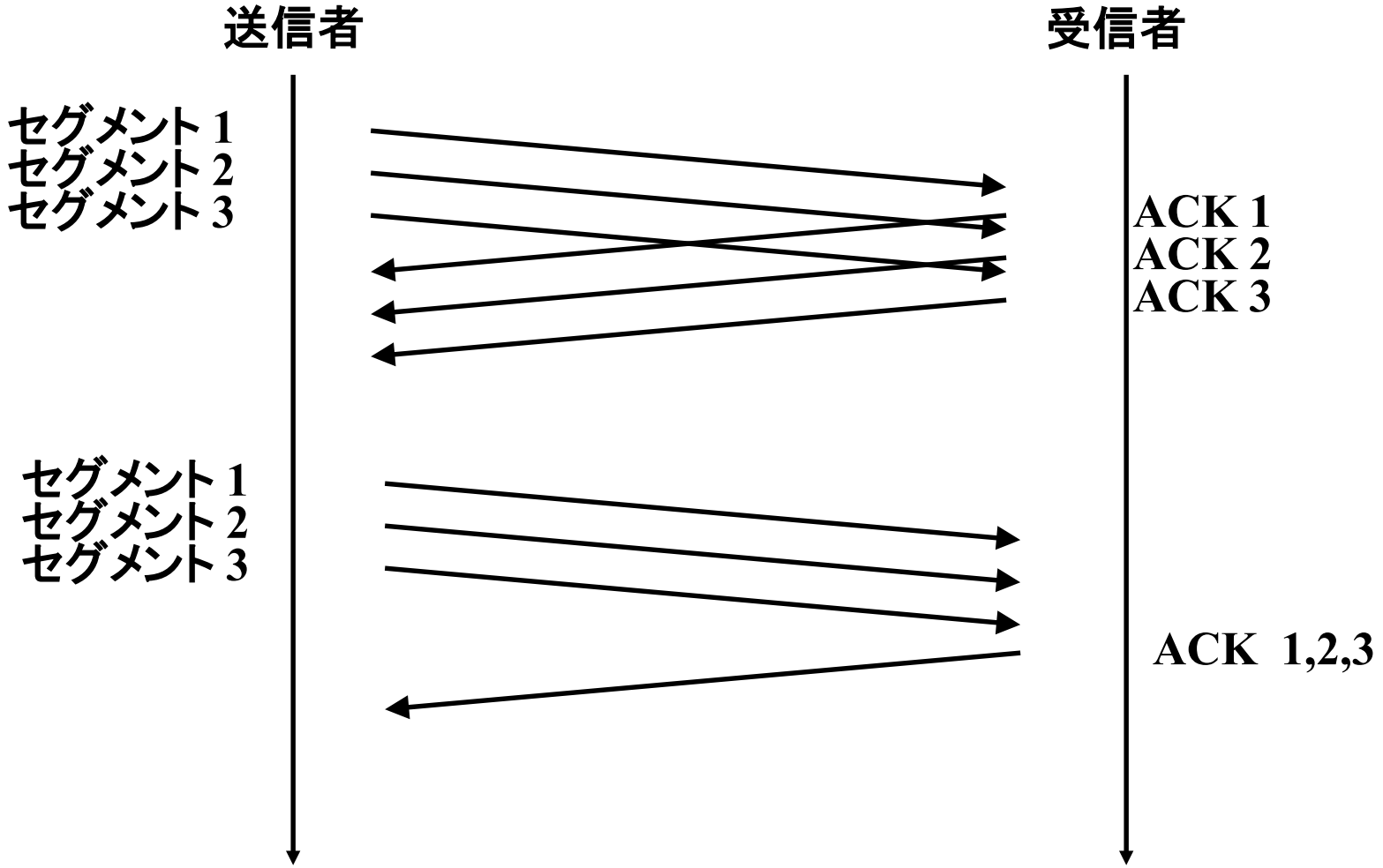
- ある時間以内に ACK が無ければ(タイムアウト)再転送する
- タイムアウト値はネットワークの状況で変化

スライディングウィンドウ

- 基本は「セグメント一つ送信して ACK を待つ」
  - でも効率が悪い
- ネットワーク内ををセグメントで埋める
- 確認応答を待たずに，送信して良いバイト列の集合
- 複数のセグメントの確認応答をまとめることも可能

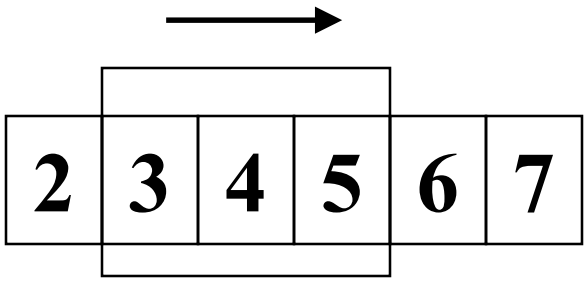


スライディングウィンドウ(2)



スライディングウィンドウ(3)

- ウィンドウの左側
  - ACK 受信済み
- ウィンドウの右側
  - 未送信
- ウィンドウ中
  - 送信済み
  - ウィンドウの一番左のセグメントの ACK を受信したら，ウィンドウが左に移動



ウィンドウ通知

- ウィンドウは，可変長
- 大きさは受信側が決定権をもつ
  - 受信側のバッファサイズ
- TCP ヘッダの Window フィールドで指定
  - 単位：オクテット
  - フィールドは16 ビット = 最大64 Kオクテット

帯域外通信(緊急データ)

- 既に送信したストリームを飛び越した通信
- 割り込み/シグナル(例: コントロール C)
- URG コードビットと URGENT POINTER フィールド

強制配送

- バッファリングはネットワークの利用効率を向上
- 会話的なアプリケーションにはバッファリングは不要
- PUSH オペレーションは，セグメントの強制的な配送
- PSH ビットをセットする
  - 受信側はバッファリングせずにアプリケーションに渡す

コネクションとポート

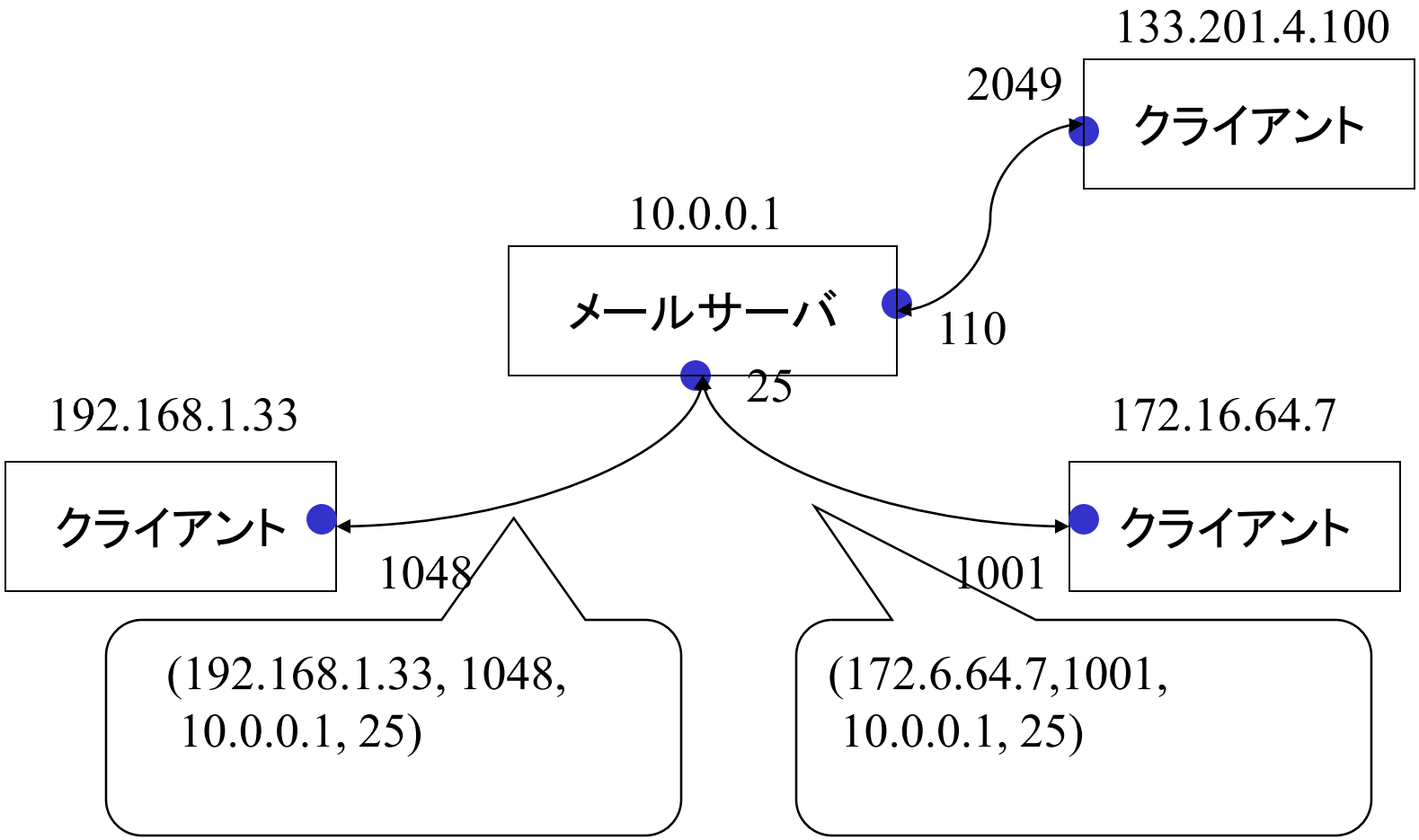
- ポート： サービスを指定
- 終点ホストのサービスを識別するのに利用
  - IPアドレスは終端ホストを識別
- コネクションで複数の同一サービスを識別
  - 同じポートを持つプロセスが存在
- コネクション
  - (始点アドレス，始点ポート，終点アドレス，終点ポート，プロトコル)

コネクションとポート(2)

- 代表的なサービス

- SMTP (mail)　25/TCP
- NNTP (news)　119/TCP
- TELNET　23/TCP
- FTP　20, 21/TCP
- HTTP (www)　80/TCP
- POP　110/TCP

コネクションとポート(3)



再送

- ある時間内に確認応答が無ければ(=タイムアウト)再送
- 「ある時間」は，どのように決定?
  - リンク層の性能や現在のネットワークの状況に応じて
- 各セグメントの送信時刻と，それに対する確認応答の受信時刻の差
  - ラウンドトリップ時間 (RTT)

再送(2)

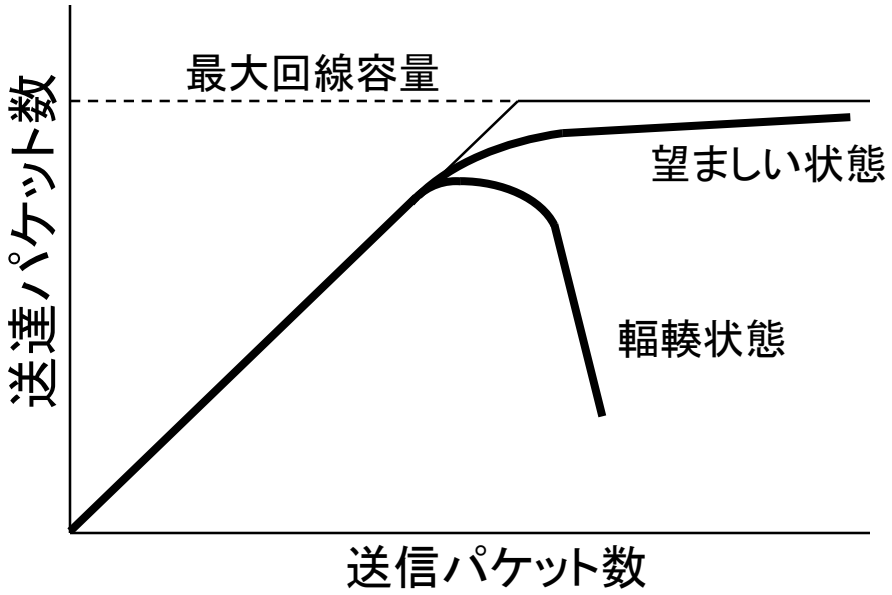
- RTT を使ってタイムアウト値を計算
- 加重平均
- RTT = (α×今までのRTT)+((1 - α)×新しいRTT)
  - ただし 0 < α < 1
  - αが1に近いほど RTT の変化に鈍い
  - αが0に近いほど RTT の変化に素早く応答
- タイムアウト = β × RTT
  - ただし β > 1
- 実装例: 4.4BSD, α=0.875，β=2

タイムアウト値の計算

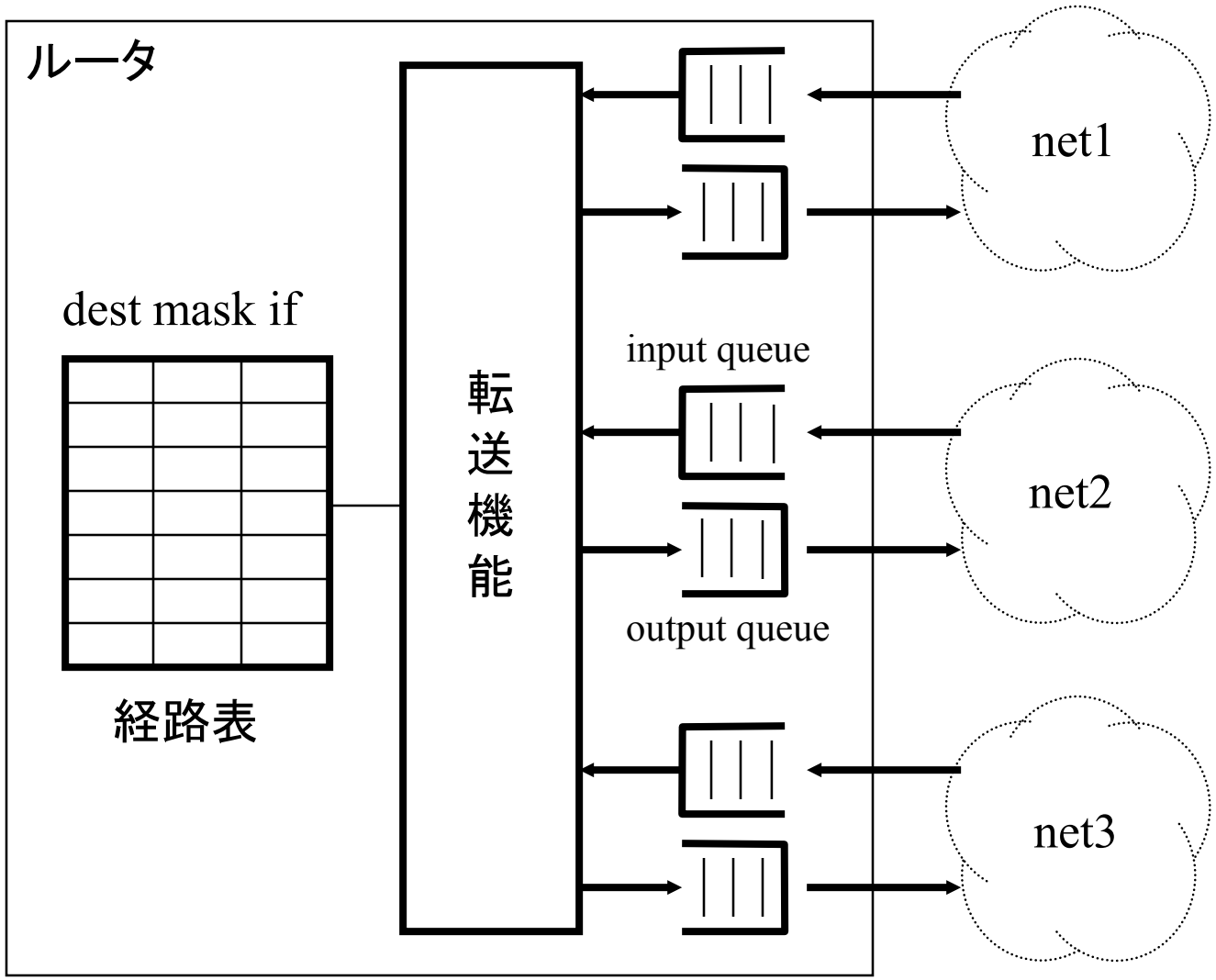
- 再転送の場合，受信した確認応答は…
  - 最初に送信したセグメントのものか，再送セグメントのものか区別できない
- Karn のアルゴリズム
  - RTTの計算に「再転送セグメント」の時刻は使わない
- バックオフの戦略
  - 再転送が起こるとタイムアウト値を再計算
  - 新タイムアウト値 = γ × 前のタイムアウト値

輻輳

- 輻輳（congestion）
  - ネットワーク上に過度のトラフィックが存在している状態
  - パケット喪失，性能の急激な劣化が発生

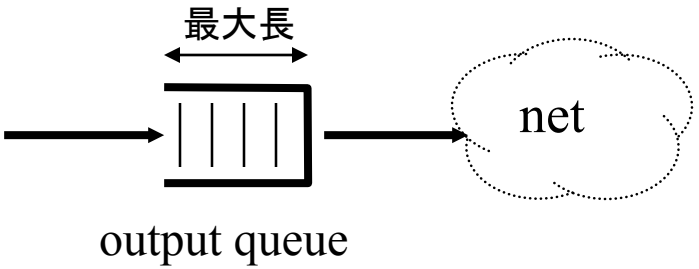


ルータの構成



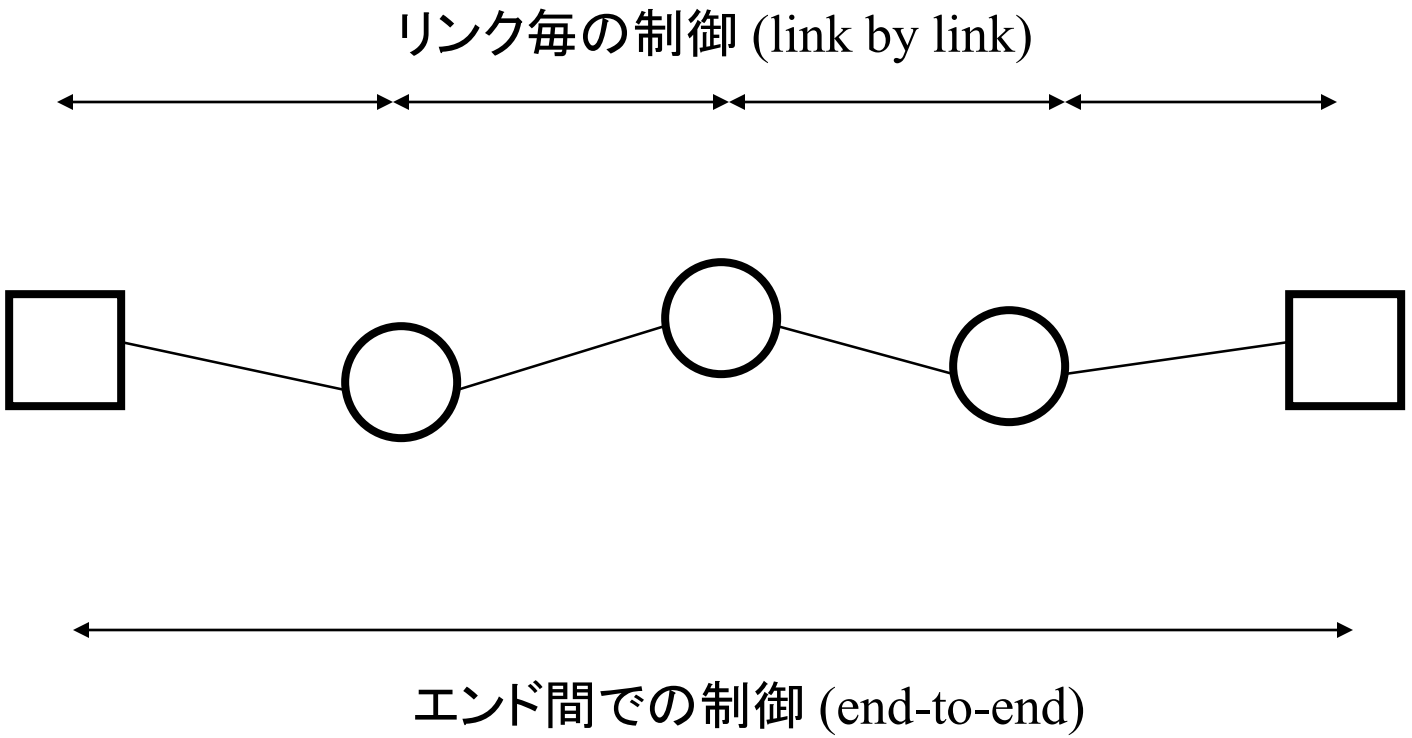
出力キューと溢れ

- 輻輳のほとんどは出力キューが溢れること



  - キューの大きさをむやみに大きくしてはいけない
    - ✓ルータのリソースは有限
    - ✓大きな遅延が発生
  - どのように溢れさせるか(捨てるか）がポイント

リンク毎の制御，エンド間での制御



輻輳回避

- TCP はエンド-エンドで輻輳を回避する
  - リンク間では行わない
  - 自律分散
  - ネットワーク全体の性能を上げる
- 輻輳が起こったら TCP は転送量を減らす必要がある
- 輻輳を避ける二つの技術
  - 倍数減少輻輳回避
  - スロースタート
- 転送量を一旦大きく減らして，少しずつ増やす

倍数減少輻輳回避

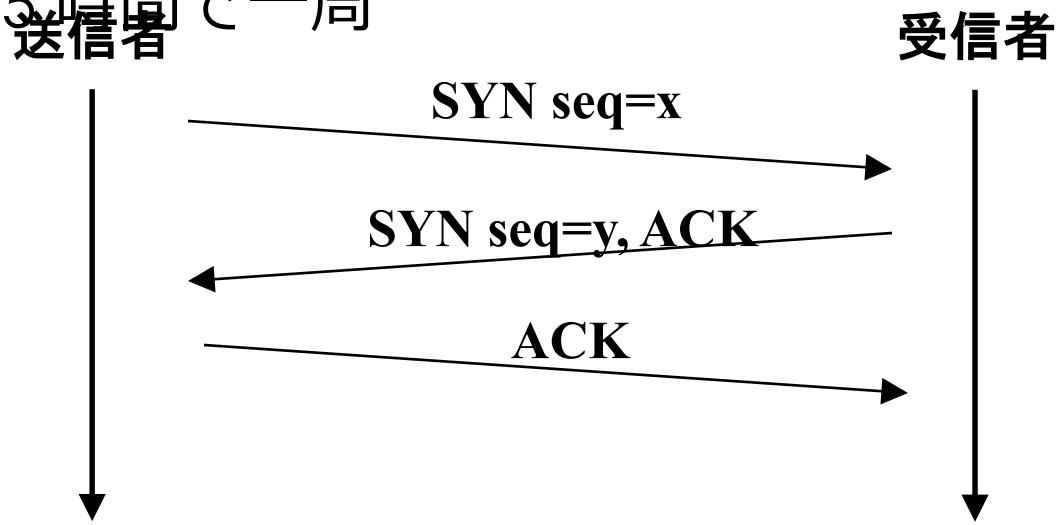
- 輻輳ウィンドウを使った制限
- 送信ウィンドウ = min(輻輳ウィンドウ，受信者ウインドウ)
- 安定状態では受信者のウィンドウと同じ
- 遅延(パケットの喪失)が起こると
  - 輻輳ウィンドウを半分にする(最小1セグメントまで)
  - タイムアウト値を大きくする
- 増やす時はスロースタート

スロースタート

- 初めからウィンドウサイズいっぱいの送信をしない
  - 1セグメントから始めてACKが返る毎に大きくする

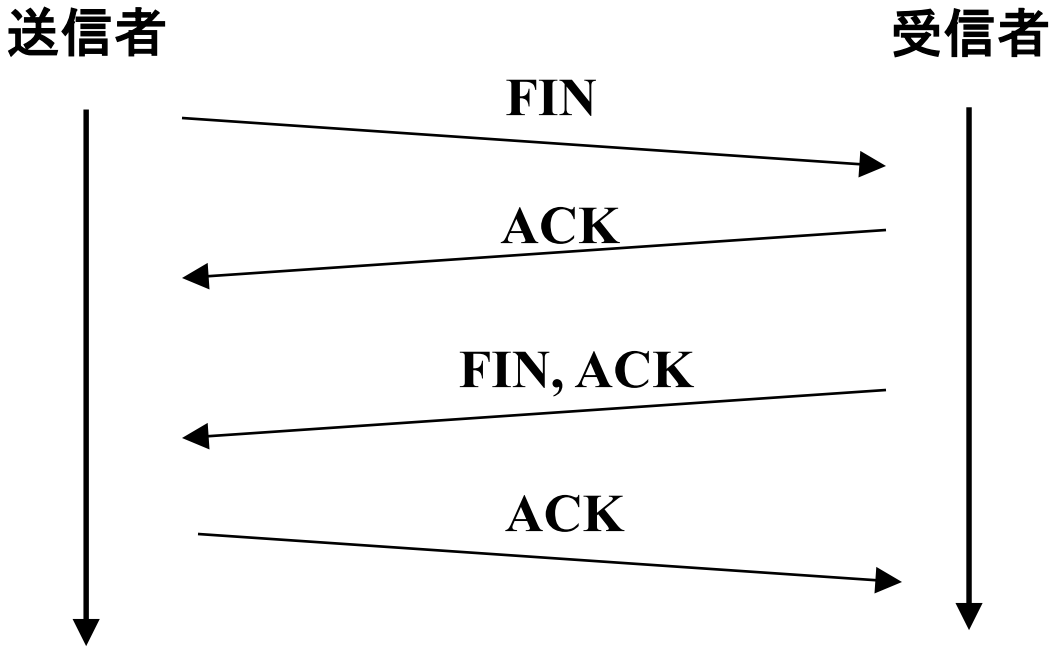
ＴＣＰコネクションの確立

- 3 ウェイハンドシェイク
  - SYN パケットの送信
- 初期シーケンス番号の一致
  - シーエケンス番号はランダムに決定
- 規格では，「4 μ秒に1つ加算されるカウンタ」を使う
  - 4.55 時間で一周



コネクションの終了方法

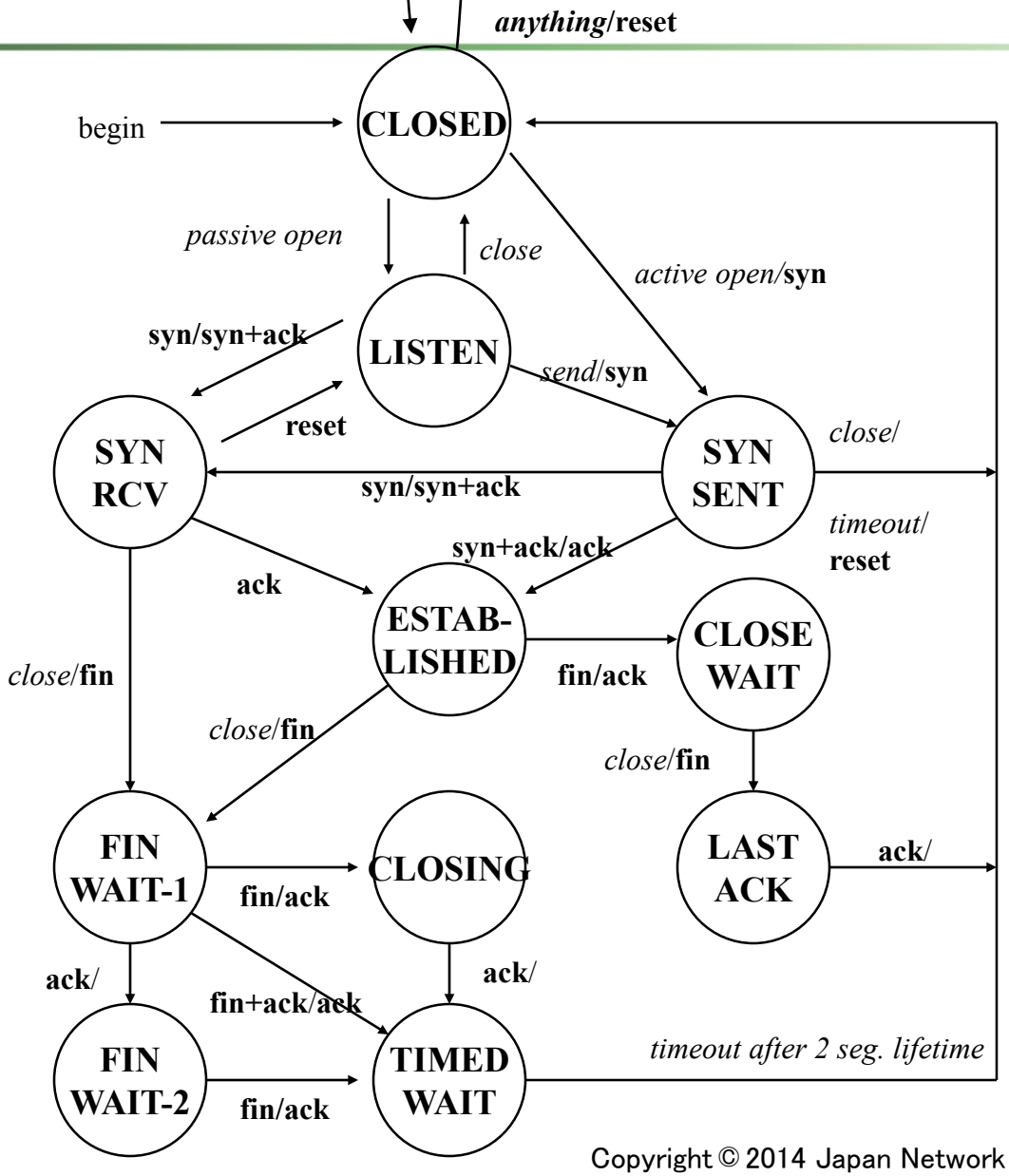
- FIN パケットを送信する
  - TCP は全二重なので片方向のみ閉じられる
    - ✓half-open
  - 相手が異常終了した場合も half-open
- 両方向とも閉じられたら終了



コネクションのリセット

- 異常なコネクション(相手が異常状態など)の強制終了
- RST パケットを送信する

状態遷移



高速通信

- ギガビットイーサ，ATM などのメディアでは…
  - シーケンス番号をすぐに消費してしまう
- シーケンス番号は循環(2^{32}-1 の次は0)する
- TTL時間内に同一シーケンス番号のデータグラムが回ってくるため順序関係が分からなくなる

１０００ｋｍ，１Ｇｂｐｓ

- ファイバの伝送速度 ２．１ｘ１０^{8}m/ｓ
- 片道 ５ｍ秒の遅延，往復１０ｍ秒の遅延
- 単純計算で ６．４Ｍｂｐｓ のスループット

- １Ｇｂｐｓ をだすためには
  - １００Ｍバイトのウインドウサイズ
    - ✓１００Ｍバイトのバッファを用意
    - ✓ＴＣＰの拡張が必要
      - ✧ＲＦＣ１３２３ ＴＣＰ Extensions for Ｈｉｇｈ Ｐｅｒｆｏｒｍａｎｃｅ

経路制御プロトコル

終点アドレスによる経路制御ー静的経路制御

- 静的経路制御（Ｓｔａｔｉｃ Ｒｏｕｔｉｎｇ）
  - 終点アドレスに応じて転送先を静的に登録
  - ネットワーク単位のルーティング
  - ホスト単位のルーティング
  - defaultルーティング
    - ✓経路表に指定されていない終点アドレスのパケットの転送先を指定
- 外部との接続が単一の経路の場合
- 低速の回線による接続や、パケット交換による接続で、ルーティングプロトコルのトラヒックが無視できない場合
- 大きなネットワークでは管理しきれなくなる

終点アドレスによる経路制御ー動的経路制御

- 動的経路制御（Dynamic Routing)
  - ルータの経路表を動的に更新
  - 静的経路制御では、ネットワークトポロジの変更や障害時の経路変更が煩雑
    - ✓動的経路制御により、自動化
- 外部との接続に複数の経路がある場合
- トポロジが変化する場合
- 耐障害性の向上

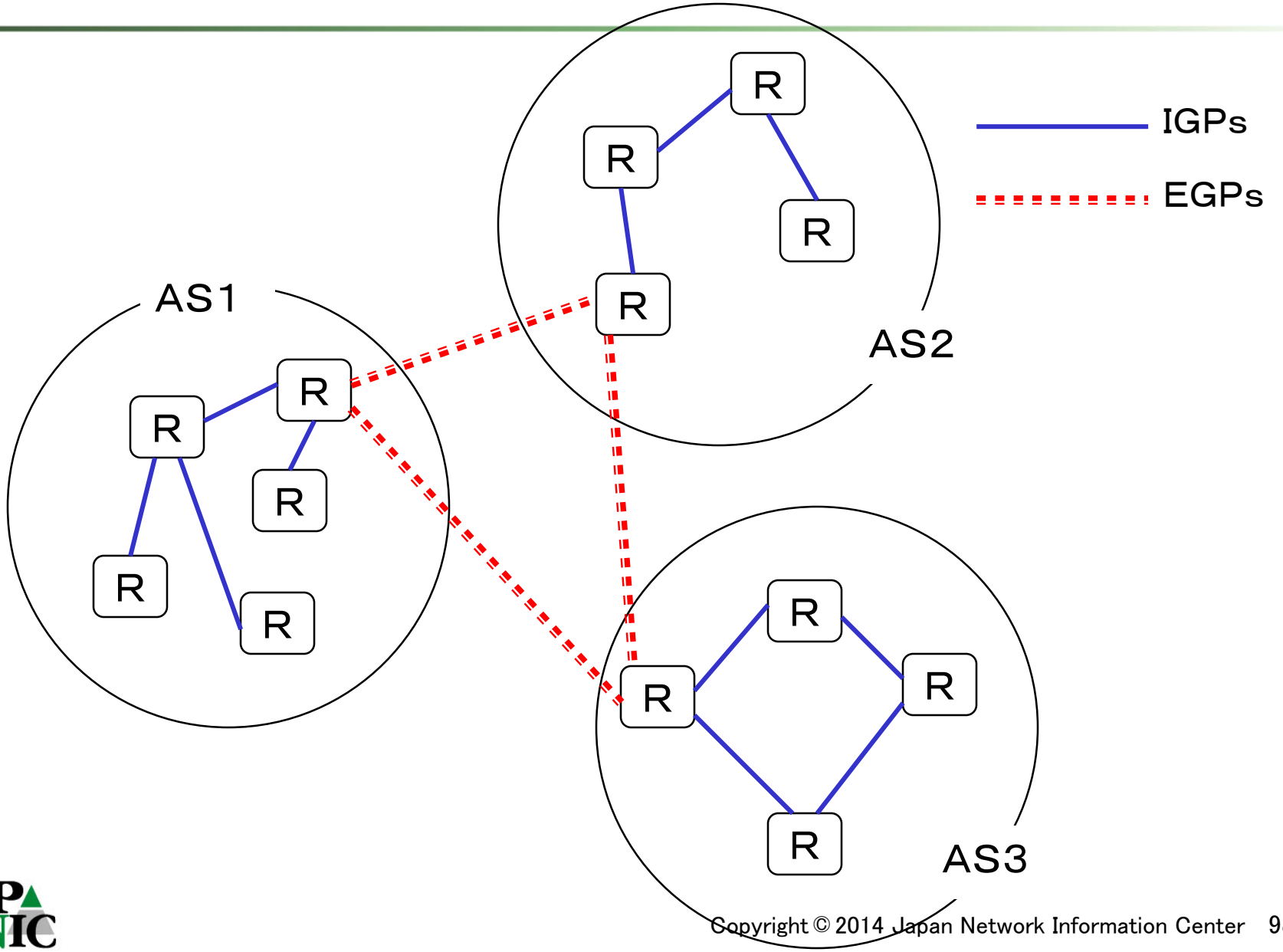
始点での経路制御

- ソースルーティング（Source routing）
- IPデータグラム中に配送経路を明示的に指定
- ほとんど使用されない
- セキュリティ上の問題が発生する可能性があるためルータでソースルートを禁止している場合が多い
- ソースアドレスで配送経路を分ける行為は"ポリシールーティング"といった言い方でVPNへ曲げたりするときに活用されている

IGPｓ と EGPｓ

- IGPｓ（Interior Gateway Protocols）
  - 単一の管理組織に属するネットワーク（AS：Autonomous System＝自立システム）内部のルーティングのためのプロトコル
    - ✓RIP（Routing Information Protocol）
    - ✓RIP 2
    - ✓OSPF（Open Shortest Path First）
- EGPｓ（Exterior Gateway Protocols）
  - 複数のASを結びつける
    - ✓EGP（Exterior Gateway Protocol）
    - ✓BGP（Border Gateway Protocol）

ＩＧＰｓとＥＧＰｓ（つづき）



動的経路制御の方式 - Distance Vector型

- Distance Vector型
  - 宛先ネットワークに対する距離（Metric）に基づいた経路選択
  - ルータが互いの経路表を交換する
  - 各々のルータが知っているのは最終的な宛先ごとの次の中継ルータ（ｎｅｘｔ ｈｏｐ）
  - 選択的経路情報の交換が可能
  - 代表的なプロトコルは RIP （RFC１０５８）

動的経路制御の方式 - Link State型

- Link State型
  - ネットワーク全体のトポロジ情報データベースを管理
  - 同一管理組織内のすべてのルータが同一のデータベースを保持
  - Dijkstraのアルゴリズムによる経路計算
    - ✓Shortest Path First
  - 経路が変化した場合の経路表の伝播が高速
  - 選択的な経路情報の交換は困難
  - 代表的なプロトコルはＯＳＰＦ（ＲＦＣ１５８３）

経路交換プロトコル　RIP

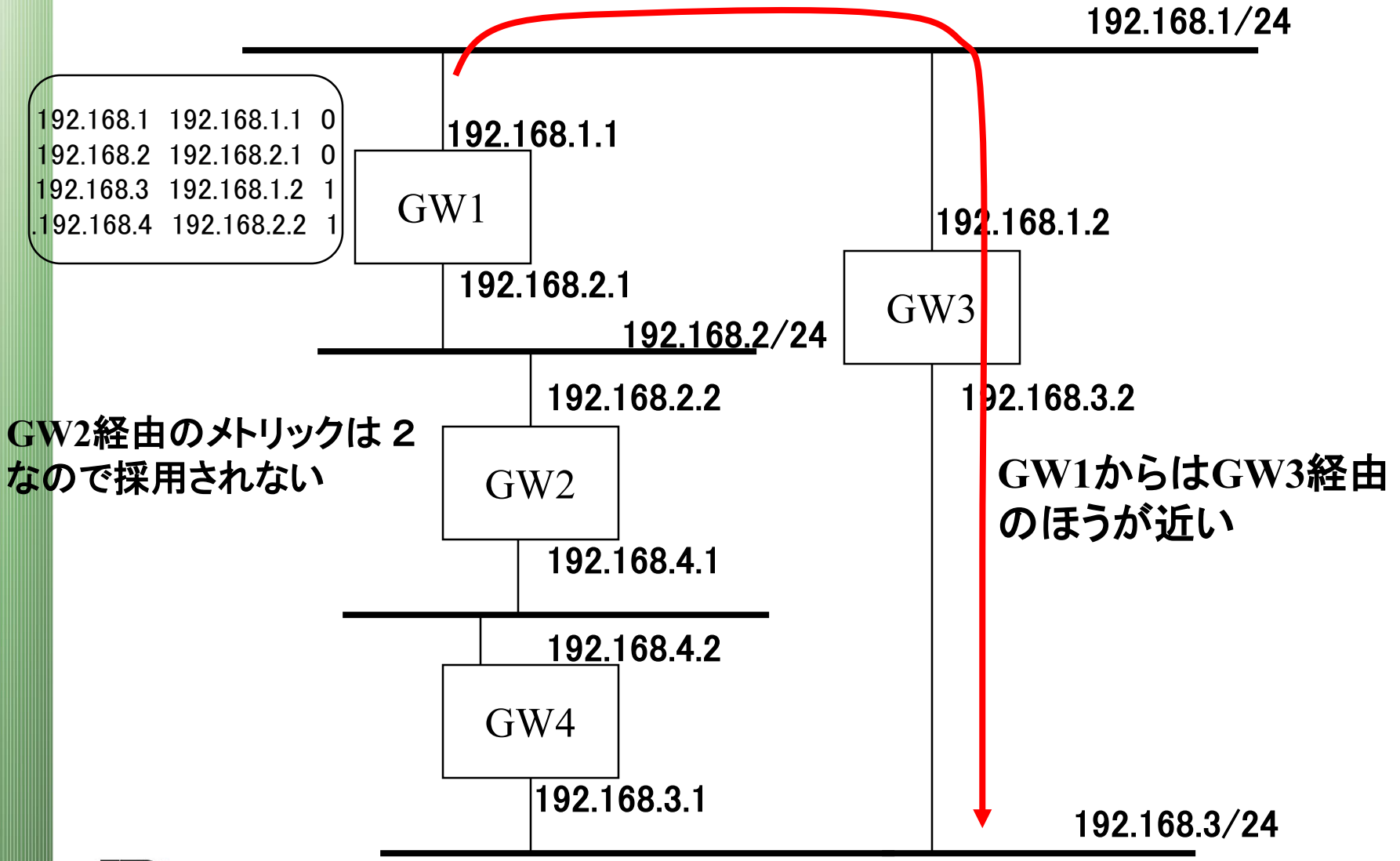
経路交換プロトコル - RIP

- IGP（内部ゲートウェイプロトコル）の一つ
- 標準的に利用されている（RFC１０５８）
- Distance Vector型アルゴリズムを使用
  - 宛先までに通過するルータの数（Metric）が最小になる経路を選択する
- 各ルータは30秒毎にルーティングテーブルの内容を広告

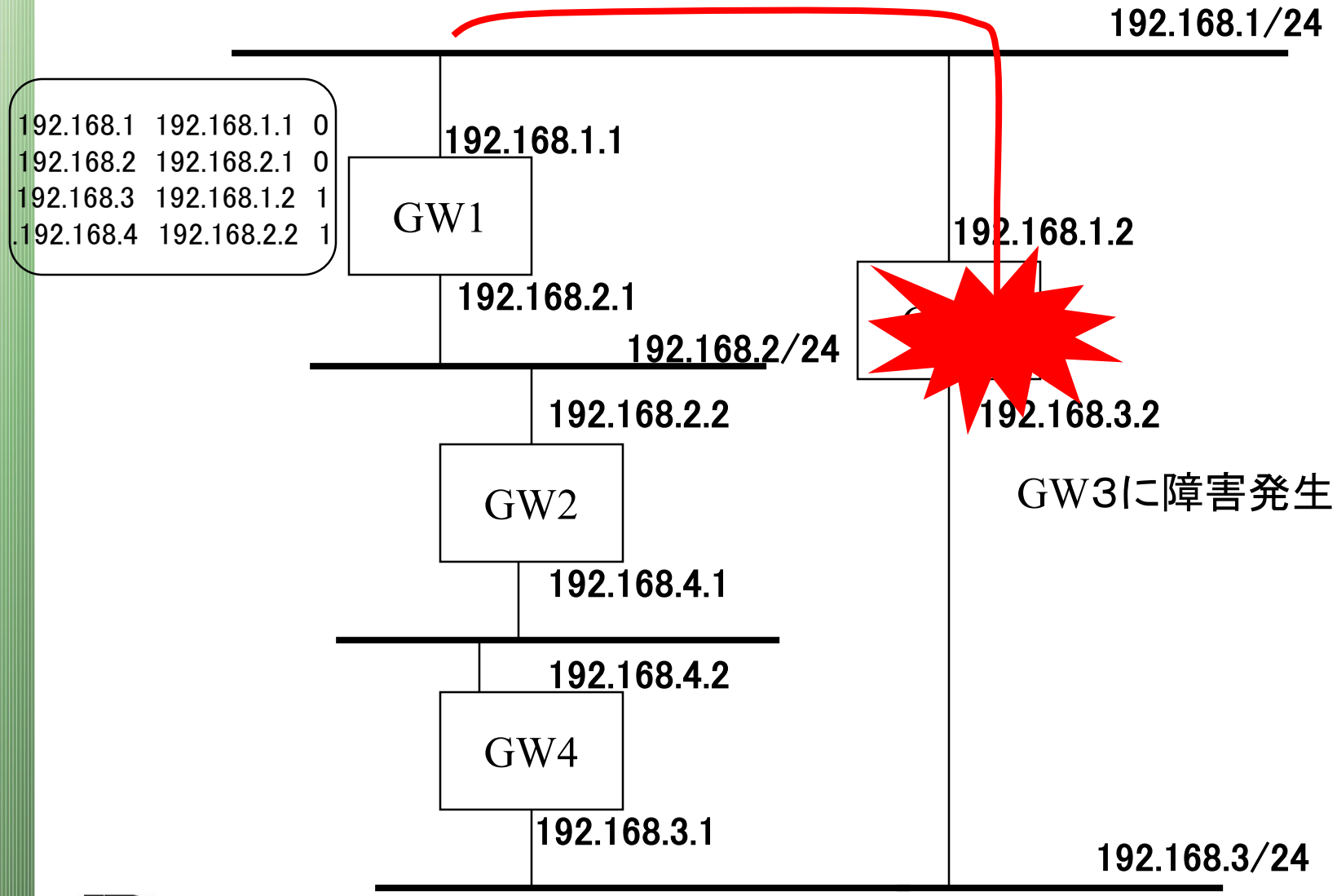
RIPの特徴

- 小規模のネットワークに適している
- １６ホップ以上のルーティングはできない
- ネットワークへの負荷が大きい
  - 30秒毎にすべてのルータがルーティングテーブルの内容をすべて広告
- ネットワークトポロジの変化や障害時のルーティングテーブルの収束に時間がかかる
- 回線の帯域幅をルーティングに反映できない
  - 高速と低速の複数の経路がある場合、低速な経路のほうがホップカウントが小さいと、より高速な経路が自動的には選択されない
- 可変長サブネットマスクに対応できない

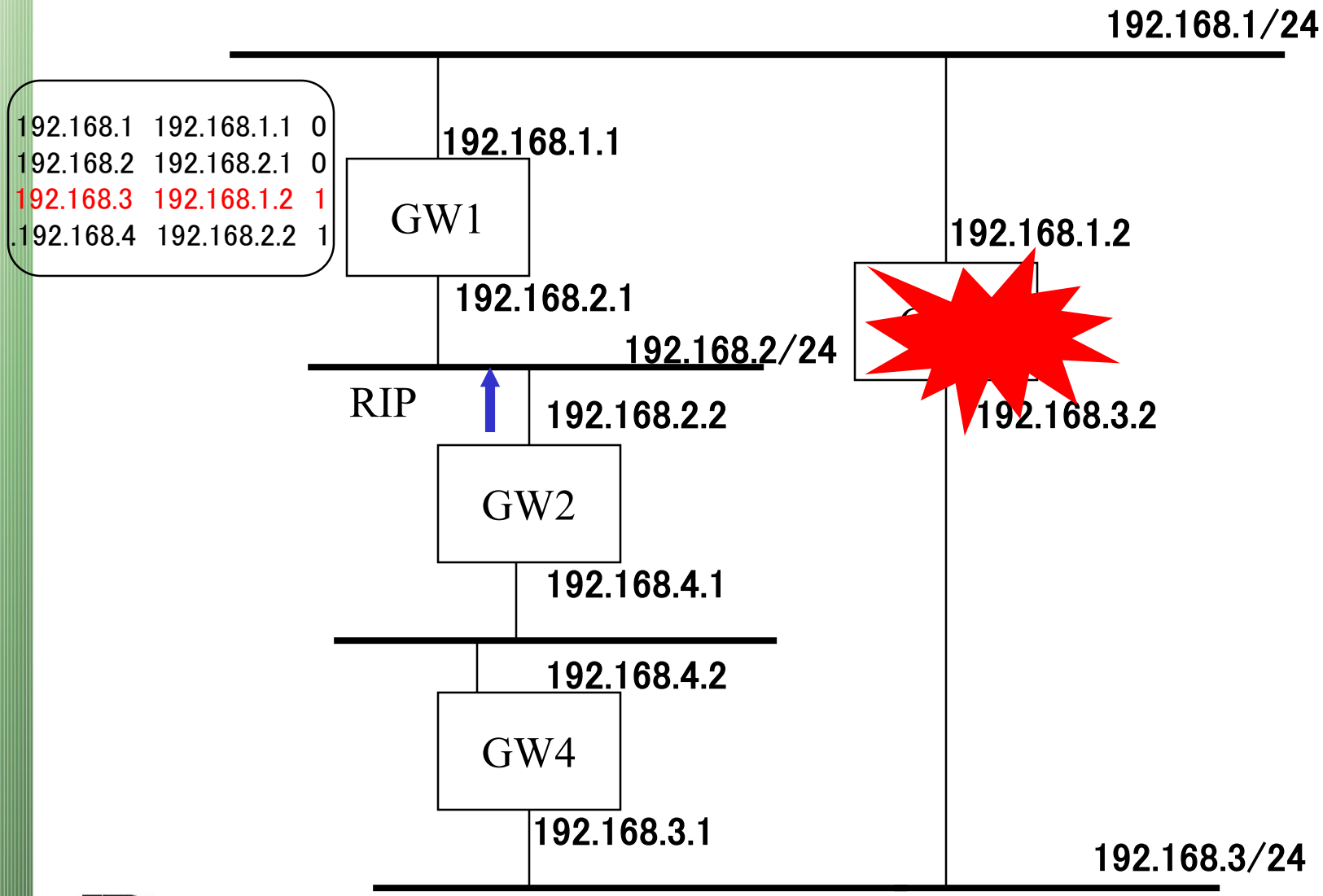
障害時のルーティング変更１



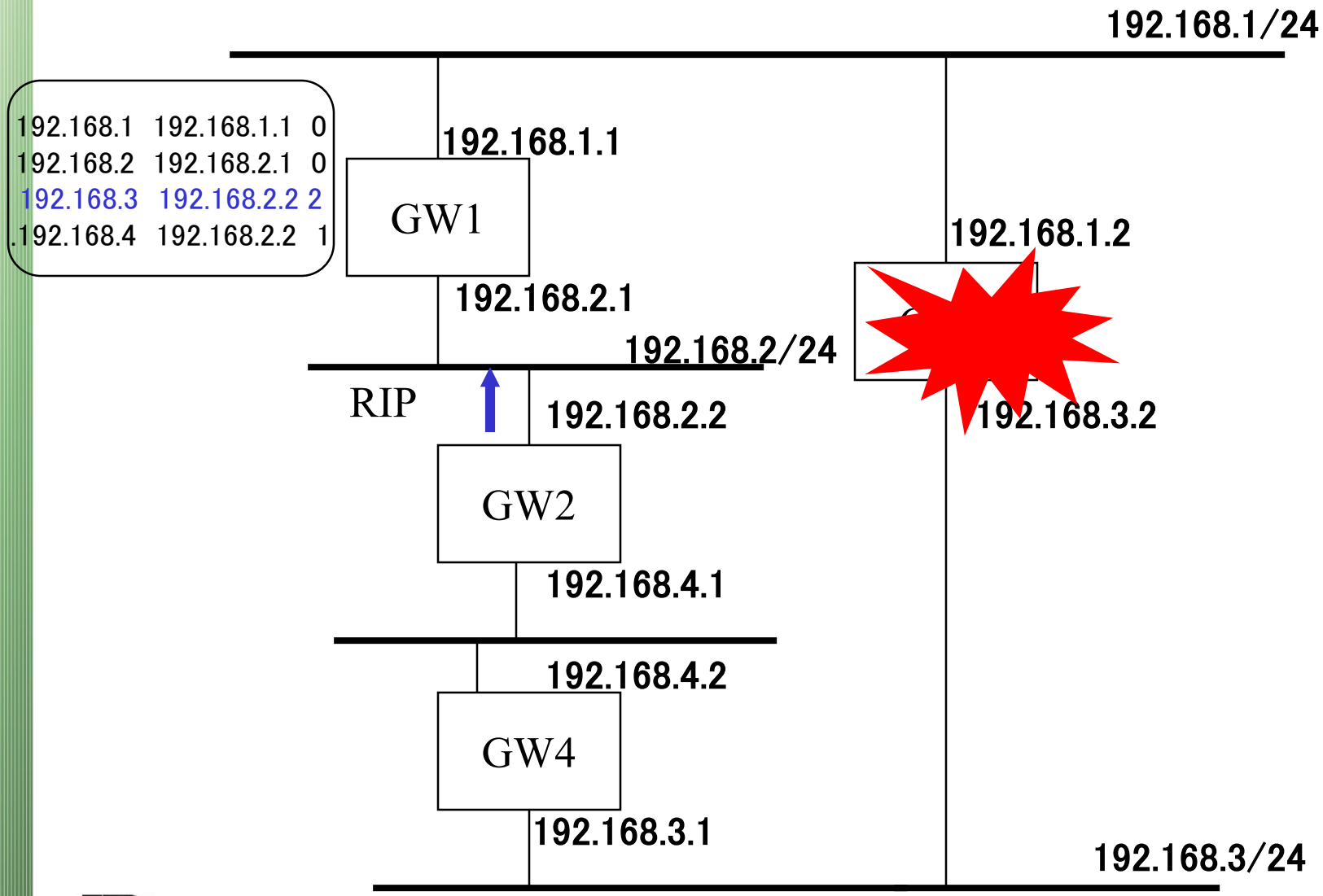
障害時のルーティング変更２



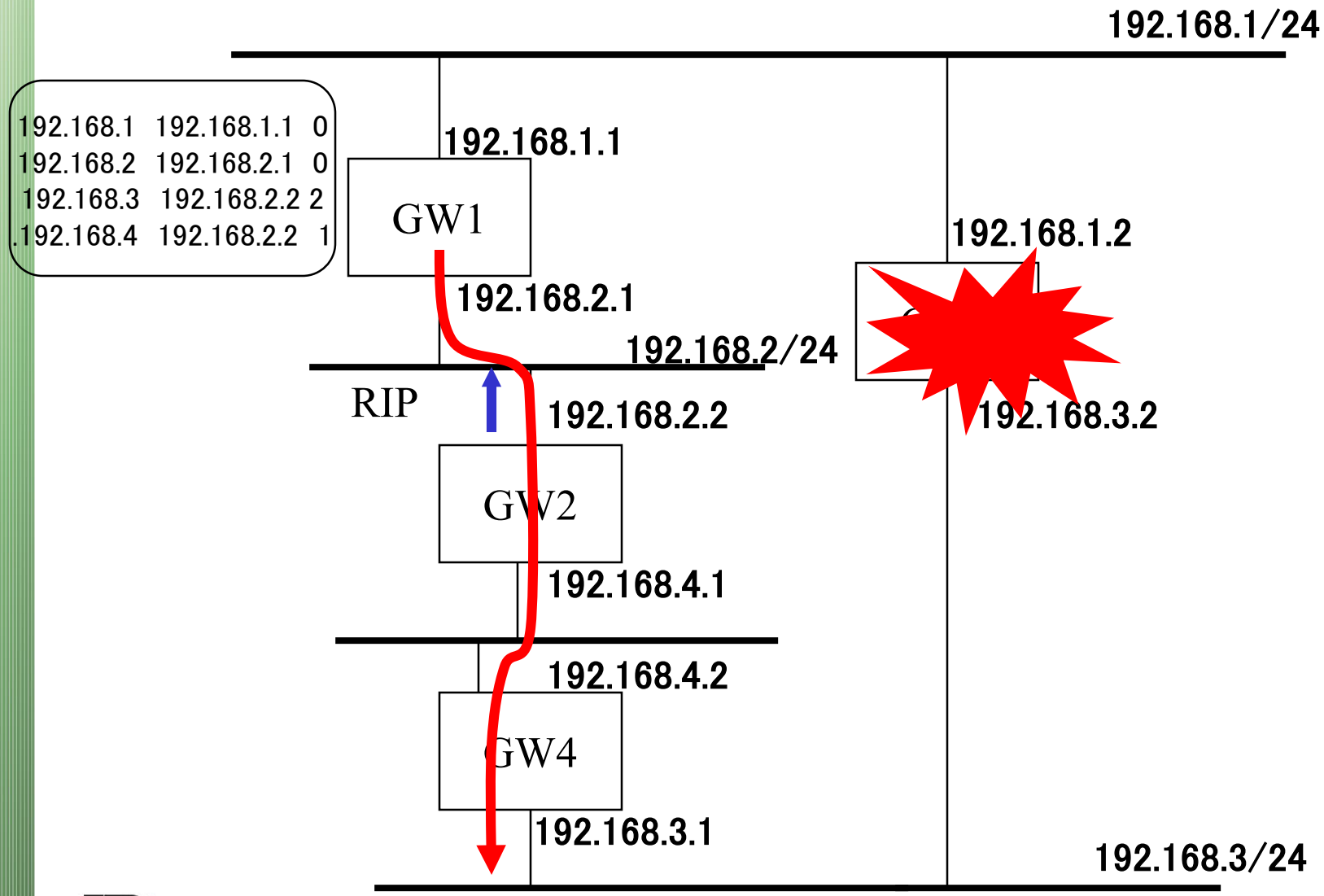
障害時のルーティング変更３



障害時のルーティング変更４



障害時のルーティング変更５



RIPの問題点

- 最大ホップ数が16ホップ以上のネットワークでは利用できない
- ルーティングループが発生したり収束に時間がかかる
  - スプリットホライズンによる回避
  - ポイズンリバースによる回避
- リンクの速度をルーティングに反映できない
  - 遅いリンクのメトリックを大きくする方法もあるが、16ホップの制限があるためむやみに大きくできない
- 可変長サブネットマスクに対応できない
  - AS内のサブネットマスク長は同じである必要がある

RIPの利点

- 簡単なアルゴリズム
  - 実装が容易
  - 堅牢
- 古くから利用されている
  - よく理解されている
  - 利用できる機器が多い

経路交換プロトコル　OSPF

経路交換プロトコル - OSPF

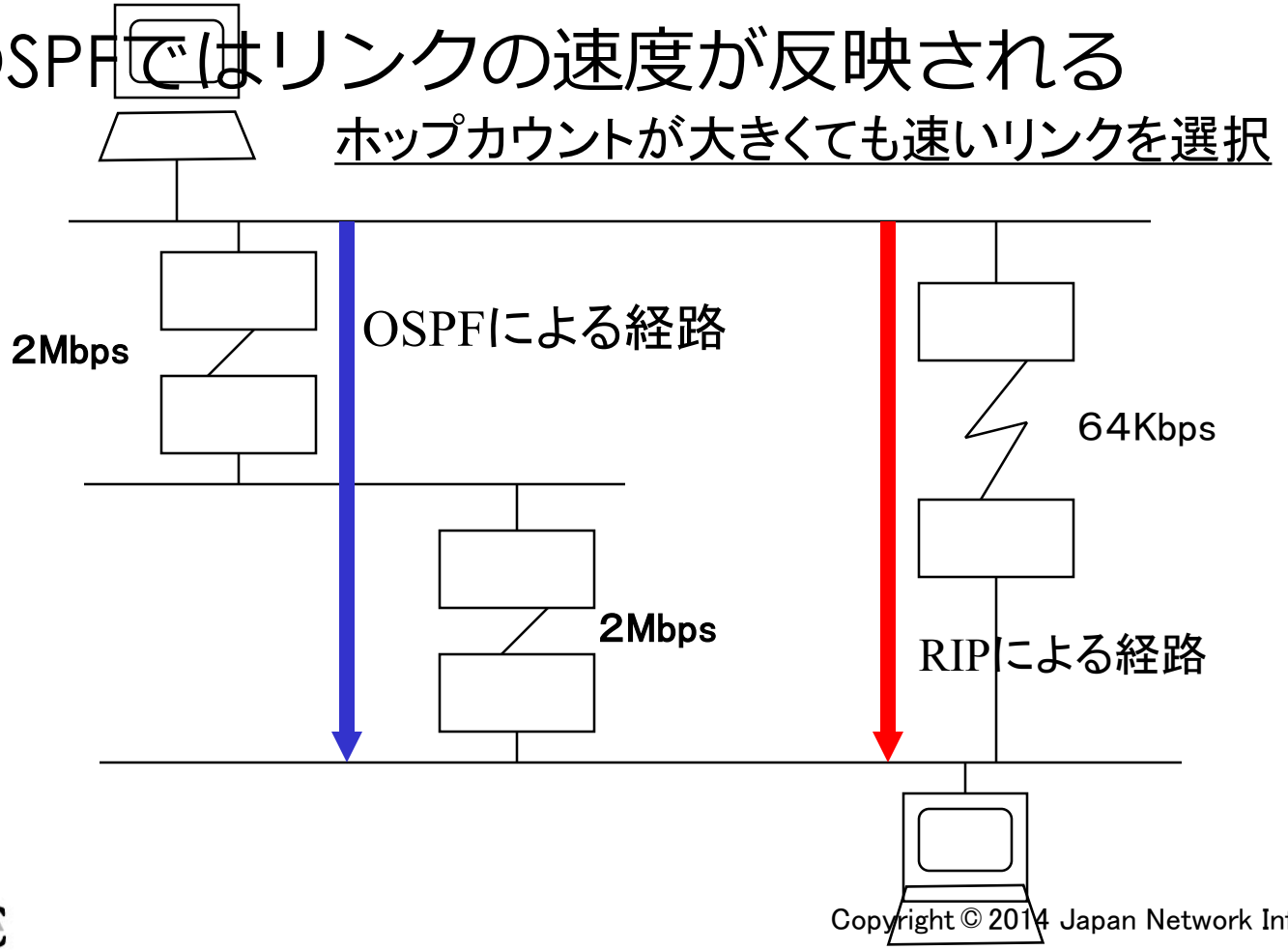
- IGP（Interior Gateway Protocol）の一つ
- ネットワークの規模が大きくなるにつれて、最大ホップカウントに制限のあるRIPに代わって使われるようになった
- Link State型アルゴリズム
  - AS内部のトポロジをAS内の全ルータが持つ
  - どのルータも同じデータベースを保持する
  - 各ルータは最短パスツリー（Shortest Path Tree)を構築
    - ✓最短パスツリー： 自分から見たAS内のトポロジ

ＯＳＰＦの特徴

- 大規模なネットワークに適している
- RFC１５８３
- トポロジの変更や障害に対してのルーティングテーブルの収束が速い
- 管理領域を複数のASに分割することが可能
  - –ルーティングのためのトラヒックの減少
  - –収束時間の短縮
- リンクの速度をルーティングに反映できる
- 同一コストの複数リンクに対応できる
- マルチキャストによる経路情報の交換
- RIPと比較すると複雑

リンクコスト

- RIPでの最適な経路はホップカウントが最小の経路
- OSPFではリンクの速度が反映される



TOSルーティング

- TOS（Type Of Service）ルーティングが可能
  - IPヘッダのサービスタイプフィールドを利用
  - サービスタイプによって異なるコストをリンクに与えることができる

例： 低速な地上リンクと高速な衛星リンク

  - 遅延の小さい地上リンクを対話型アプリケーションで利用
  - スループットの高い衛星リンクをファイル転送で利用

ＯＳＰＦのプロトコルフォーマット

- 直接ＩＰデータグラムを利用する
  - プロトコルＩＤ ＝ ８９
- ユニキャストもマルチキャストも利用できる
- ＯＳＰＦパケットには以下の共通ヘッダが使用される

経路交換プロトコル　BGP

JPNIC 一般社団法人 日本ネットワークインフォメーションセンター

経路交換プロトコル - BGP

- EGP(外部ゲートウェイプロトコル）の一つ
- RFC１７７１　BGP Version ４
- リンクステート型アルゴリズム
- 複数のAS間の経路情報の交換を行う
- 内部隣接装置と外部隣接装置
- TCPを利用する（PORT １７９）

BGPの動作

- それぞれのＢＧＰルータはルーティング情報ベース（ＲＩＢ：Ｒｏｕｔｉｎｇ ＩＮｆｏｒｍａｔｉｏｎ Ｂａｓｅ）を持つ
- 隣接ルータはあらかじめコネクションを確立し，Ｋｅｅｐ Ａｌｉｖｅ メッセージを定期的に交換することによってルータ間の到達性を確認する
- ルータ間での情報交換は，更新メッセージを利用して行う

ICMPによる経路の変更

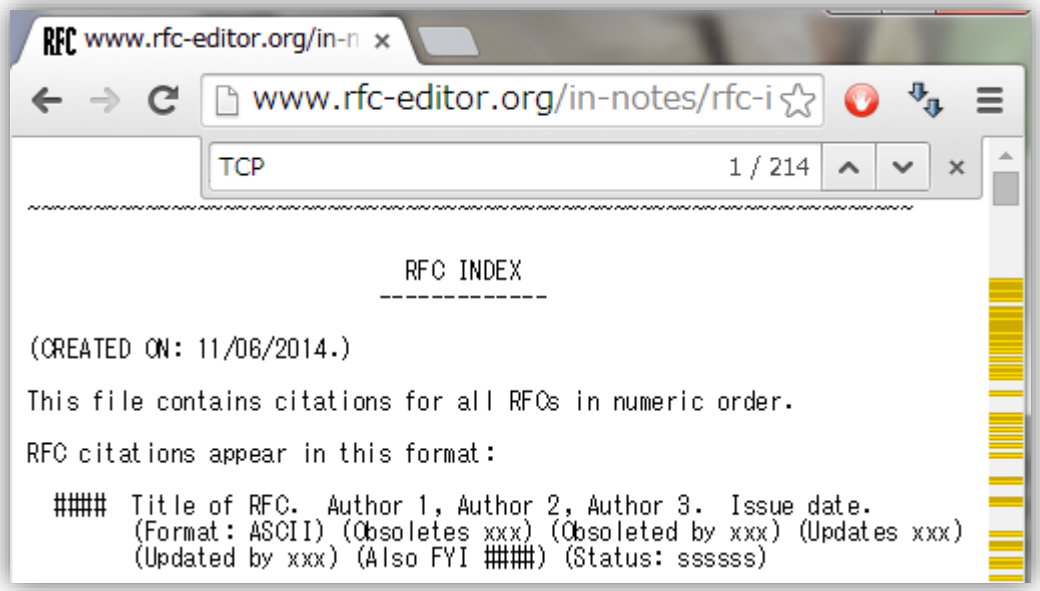
ICMP redirect

- ルーティングプロトコルではないが、経路情報の変更に重要な役割を果たす
- ICMP（Internet Control Message Protocol：RFC７９２）は、IPの処理中に発生したエラーを送信元に通知するプロトコル
- 誤った経路に対して配送されたIPデータグラムに対しゲートウェイ(ルータ）はICMP経路変更要求メッセージを送信元に対して送る
- ICMP経路変更要求メッセージを受け取ったゲートウェイは経路表を変更するため、次回からは正しい経路で配送が行われる

現代的なTCP・UDP

TCPやUDPの拡張

- RFCではさまざまな拡張が提案
  - すべて活用されているわけではないのが現状
- よく使われている、使われそうな技術を紹介



TCPの拡張

- TCP Syn Cookie
  - TCPへのDoS攻撃対策として考案された
  - TCP Cookie Transaction
    - ✓TCPのヘッダにトランザクションIDを加える提案
    - ✓あまり普及しなかったがその後、後述のTFOへ
- Multi Path TCP
  - たくさんの通信ポートを持つ機械の対応
  - WIFIとLTEを同時に活用しつつ、一つのTCPセッションに見せる
    - ✓帯域増＋冗長化
- TCP Fast Open（TFO）
  - 3 Way Hand Shakeの省力化 最近Linuxに実装

UDPの応用

- UDP Hall Puncing
  - 顧客間での通信にてIPマスカレードを超えて通信する仕組み
  - ゲームやP2Pで多様
- Reliable UDP
  - 信頼性のあるUDP
  - ゲームのパケットなどで、高信頼性を提供する仕組み
  - 標準的な実装が存在しない
  - ゲーム会社やOS会社が独自に提供されている
  - MSDNにはMS独自のものが存在

