

評価方法

- 中間レポートと、期末レポート
- 出席はとらないが、、、
- 質問やコメントを義務付ける
 - 学期中、講義に関する技術的な内容の質問やコメントを最低2回、授業中に行うこと
 - よい質問やコメントは、成績の加点対象
 - 質問者は、講義終了後に名前と学籍番号を申告のこと

インターネット応用特論

2. トランスポート層：TCP、輻輳制御、 ロングファットパイプ、マルチホーミング

太田昌孝

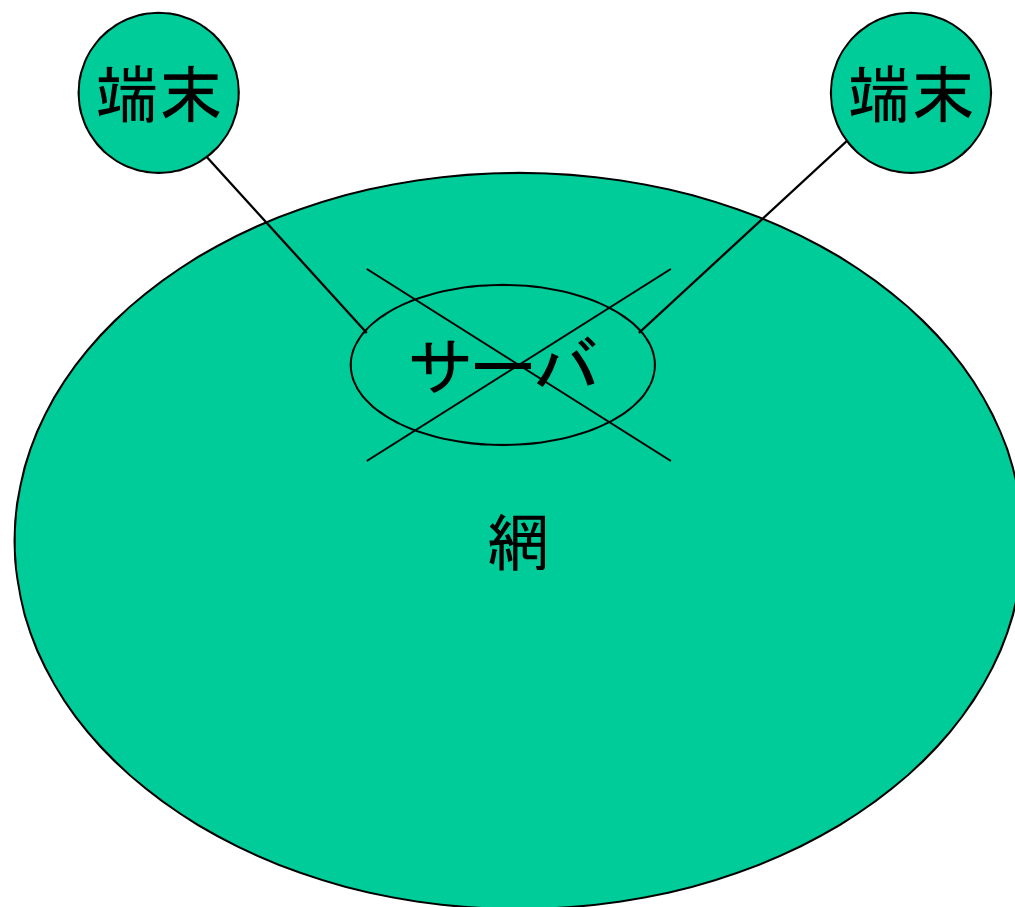
mohta@necom830.hpcl.titech.ac.jp

<ftp://chacha.hpcl.titech.ac.jp/appli2j.ppt>

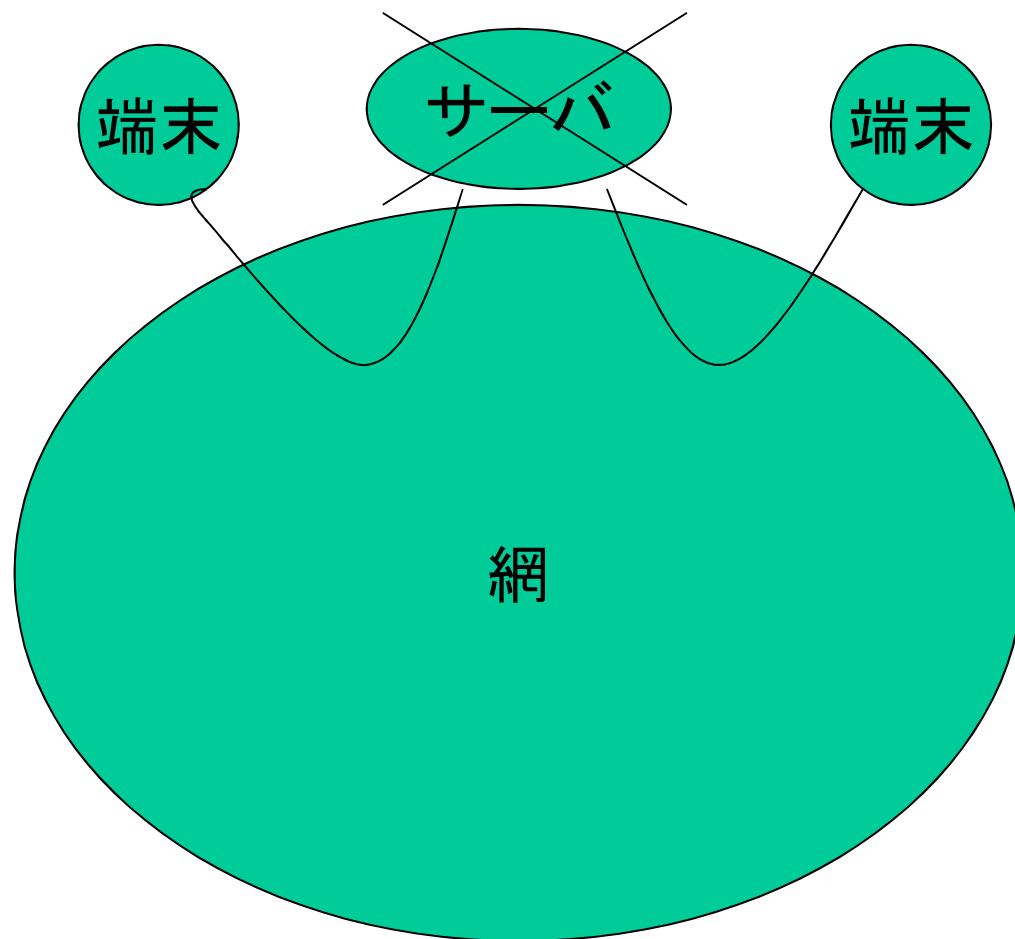
インターネットの基本原則

エンドツーエンド原則

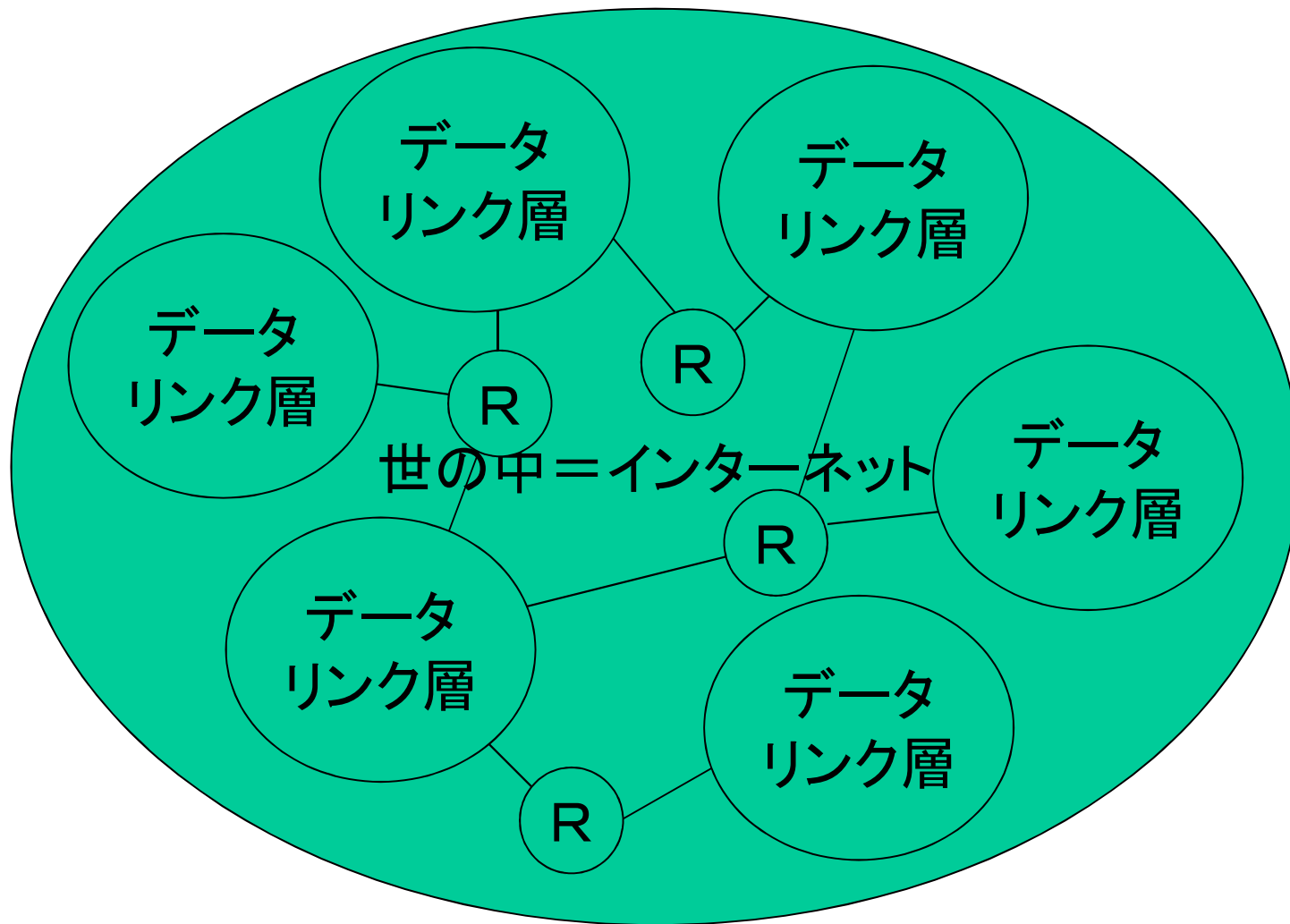
- 網は極力何もしない
 - 速度制御もしない
- 関係する端末が直接通信
 - 余計な中間サーバは置かない



網は何もしない

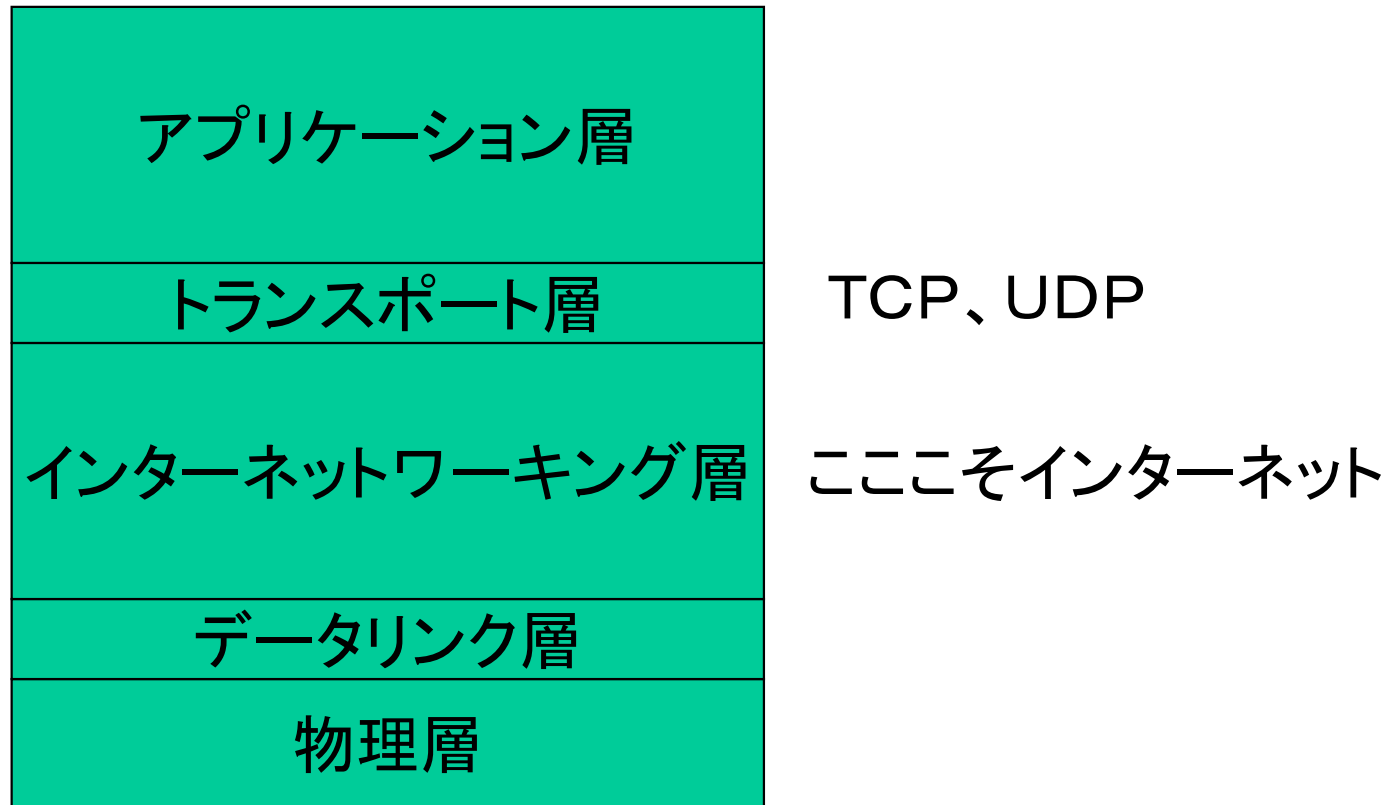


網外にも余計なサーバなし



R : ルータ

CATENETモデル

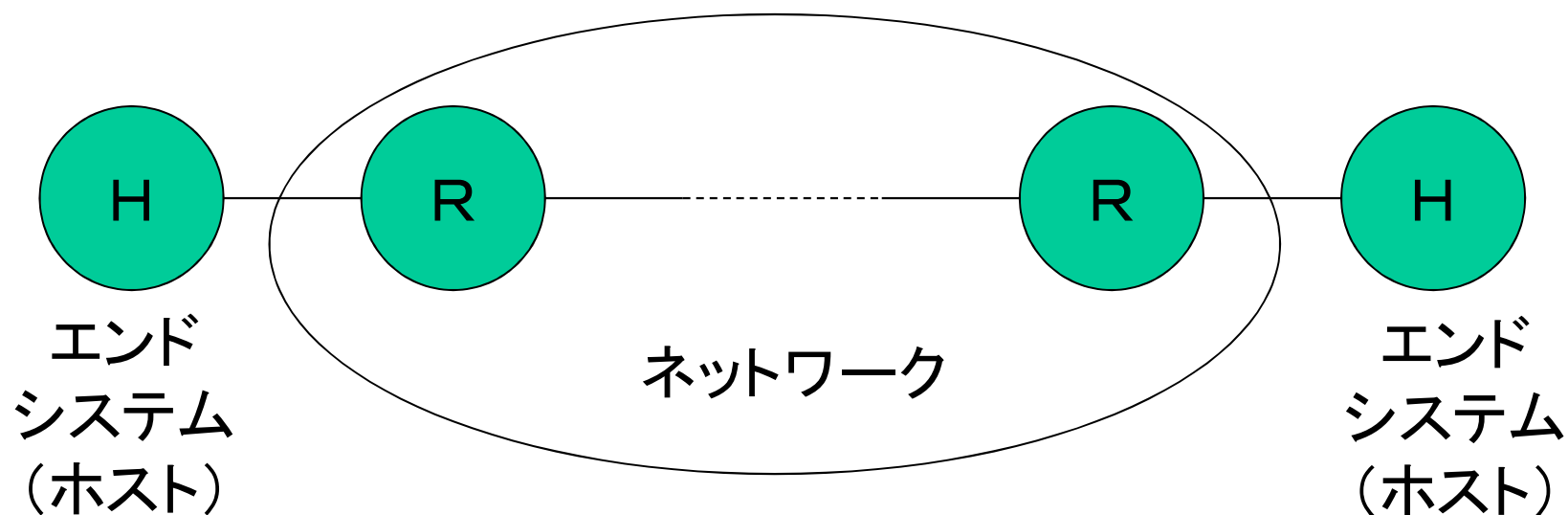


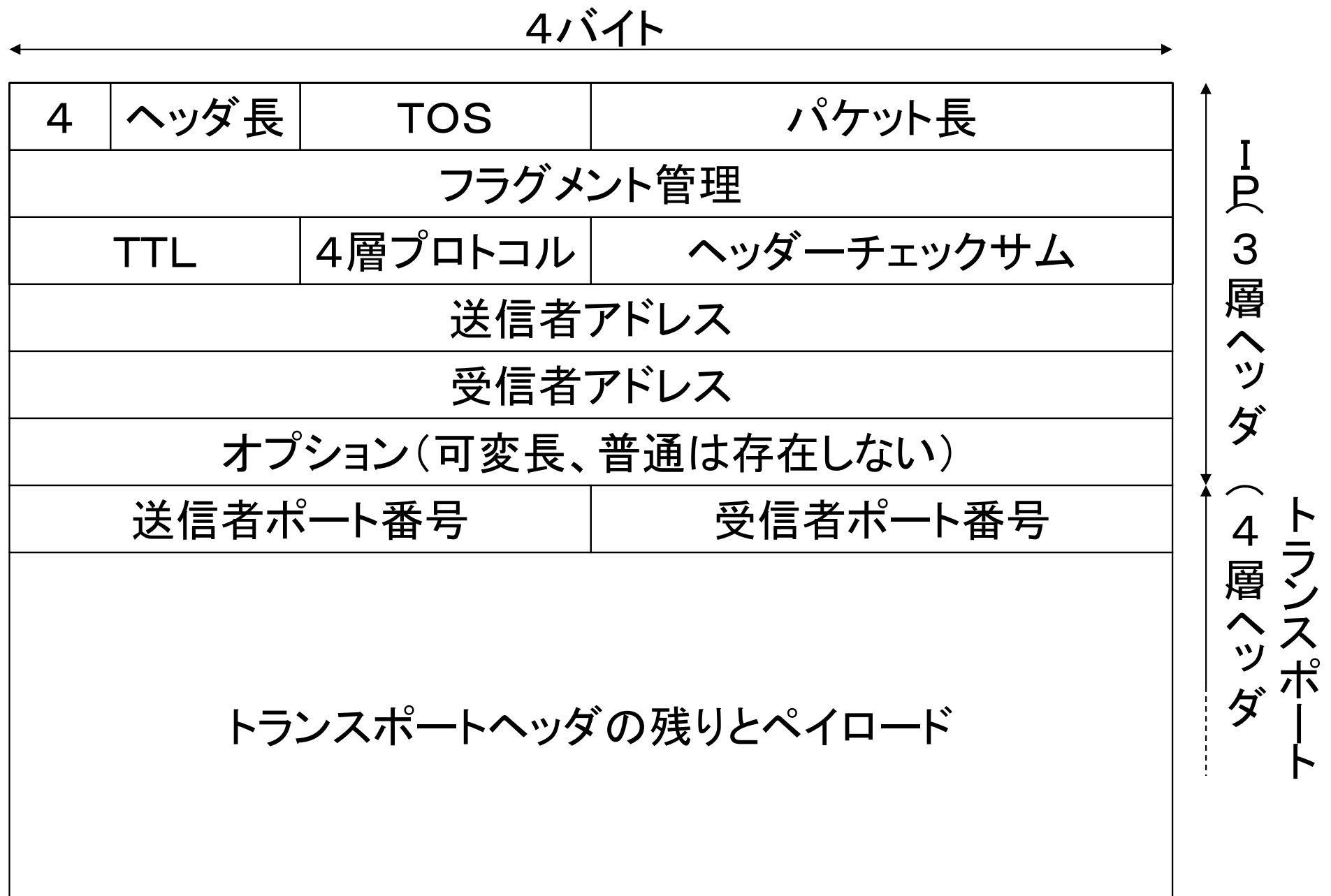
インターネットのレイヤリング構造

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





IPv4パケットフォーマット

IP

- 受信者アドレスによって、TTLを1以上減らしつつ、パケットを目的地にフォワード
- MTUが小さくなるとパケットを分割（フラグメンテーション）
- TOSはType of Service
- IPv4 (RFC791) は、アドレス空間が32ビット
 - IPv6への移行が望まれたが、、

TCPとUDP

- TCP (RFC793)
 - Transmission Control Protocol
 - データの誤りや欠けを検出して再送
 - 伝送レートの調節
- UDP (RFC768)
 - User Datagram Protocol
 - 何もしない(全てアプリケーションまかせ)

電話網とインターネット

- どちらがエラーが多い？

電話網

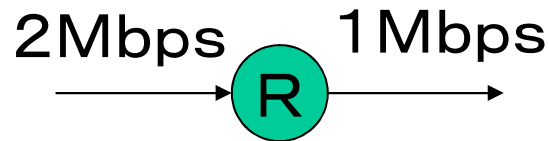
- 音声伝送(会話)のための網
- 通信の帯域を保証
- 通信の遅延を小さく
 - 会話には、0.1秒程度以下の遅延が望ましい
 - データが欠損しても再送の暇はない(そもそもアナログでは不可能に近い)
- データの誤りや欠けは雑音になる
 - 多少なら会話にはたいした問題ではない

インターネット

- コンピュータ間通信のための網
- 1ビットの誤りは、しばしば致命的
 - 文字コード、プログラム
 - データの誤りや欠けの検出は必須
 - TCPによる信頼性のある通信を多用
- 音声伝送などにはUDPを利用
- 軽いプロトコル(DNS、TFTP、、、)もUDP

パケット落ちの原因

- 伝送エラーがあると、パケットは失われる
 - そうそうあることではない
- ルータのバッファがいっぱいになると、パケットをおとすしかない
 - インターネットでのパケット落ちの主因



電話網での混雑制御

- 帯域予約してる以上、混雑は関係ない？
- 帯域予約で落ちる
 - 話中にみえる

TCP

- コネクションオリエンテッド
- 原理は単純
- 各種の異常事態への対処は複雑
- レートコントロールの理論や実装も複雑
 - すべてエンドシステムで対処

TCPとインターネット

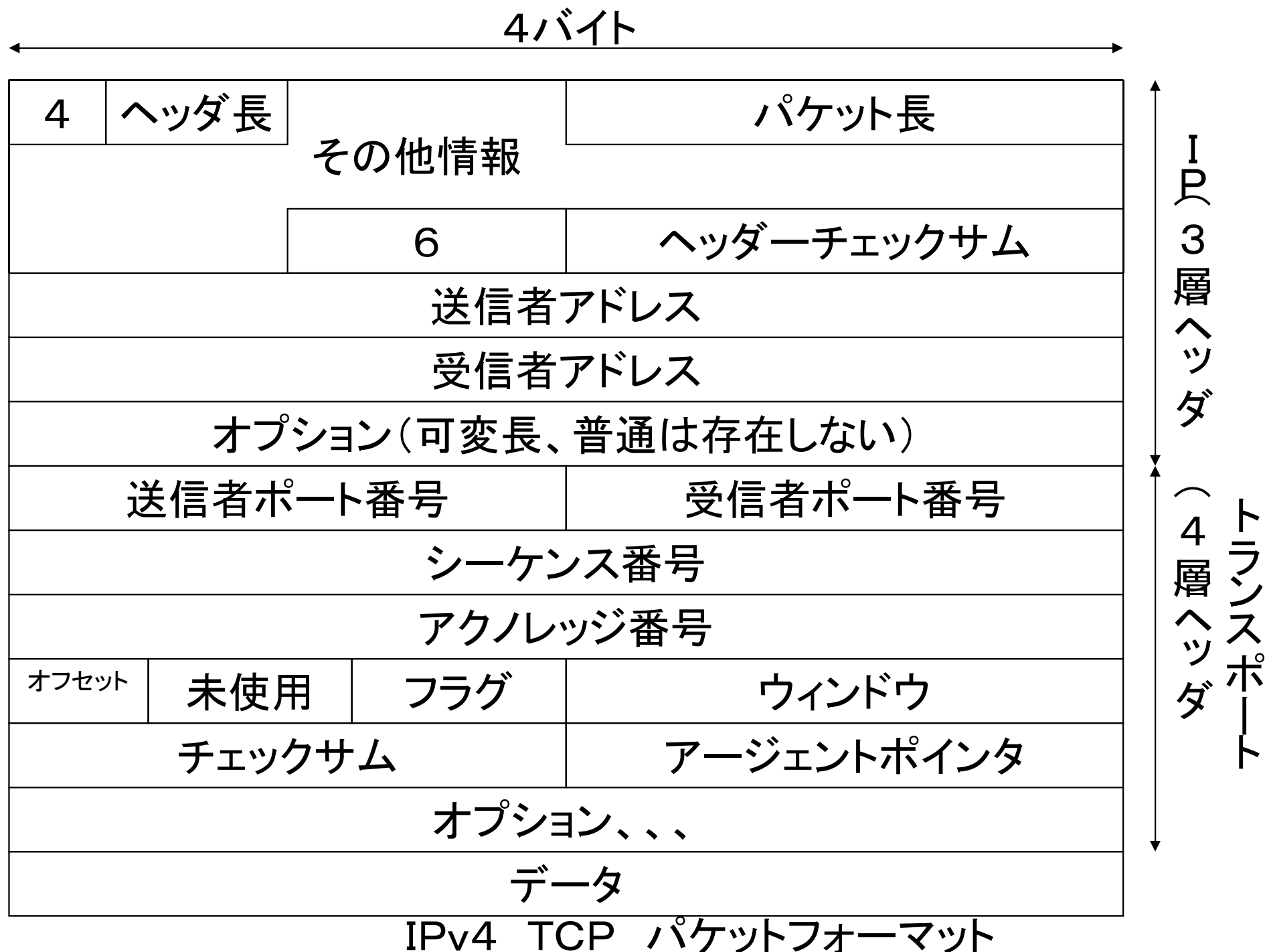
- TCPはインターネットで最も重要なトランスポートプロトコル
 - ほとんどのアプリケーションで利用される
 - TCPの混雑制御のおかげで、インターネットはかろうじて動作している？
 - いずれ破綻？
- TCP/IP
 - インターネットプロトコル群の別名

TCPの基本動作

- 接続の開始
 - シーケンス番号をあわせる
- データの送信
 - データをウィンドウの範囲で送る
- データの受信確認
 - 受け取ったデータはアクノレッジ
- データの再送
 - アクノレッジされないデータは再送

エンドツーエンド原理と信頼性

- エラーのないネットワークはない
 - 喪失情報の回復には再送信しかない
 - 再送信のためにネットワーク内でデータをバッファ(ネットワークの高機能化)しても
 - その部分でエラーするとそれまで
 - 経路変更にも対処できない(可用性の低下)
 - 極めてエラーの多いデータリンクでは有用かも
 - エンドでデータを保持することは必須
 - ネットワークのいたずらな高信頼化は無意味



送信者ポート番号、受信者ポート番号、チェックサム

- 意味はUDPに同じ
- コネクションは、(送信者アドレス、送信者ポート番号、受信者アドレス、受信者ポート番号)で区別
- チェックサムは省略不可

シーケンス番号

- 送信側がつけるデータの通し番号(バイト単位)
- ff. . . fの次は0にもどる

アクノレッジ番号

- 受信側が受け取りを確認したシーケンス番号

オフセット

- TCPヘッダの長さ(32ビット単位)

フラグ

- ・ U R G アージェントポイント利用
- ・ A C K アクノレッジ番号利用
- ・ P S H 受信側でバッファしない
- ・ R S T リセット
- ・ S Y N コネクション初期化
- ・ F I N コネクション終了

ウィンドウ

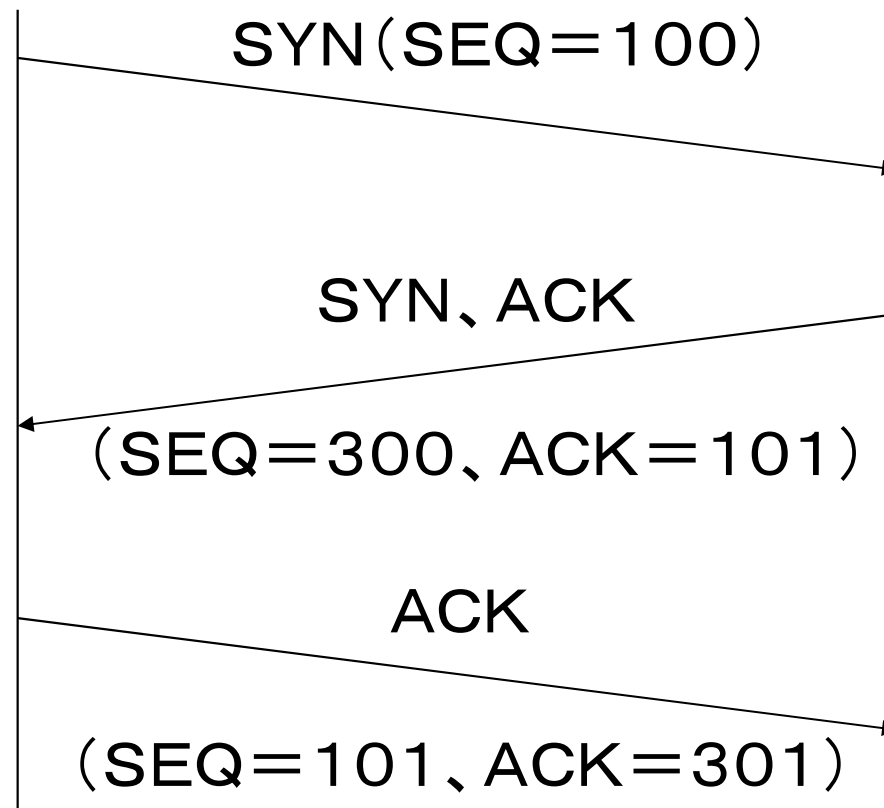
- 送信側がアクノレッジなしに送っていいデータの量
 - 受信側のバッファの量
- 高速でデータを送る場合、ウィンドウを大きくしないと、、、
 - 最大伝送速度 = (ウィンドウサイズ) / (RTT)
- 混雑回避のため、送信側も独自にウィンドウを管理して、自主規制

エージェントポイント

- 緊急データの場所を示す

TCPの接続の確立

3-wayハンドシェーク



混雑制御

- インターネット内では帯域を管理しない
- みんながパケットを好きに送ると、パケット落ちが大量に発生
- パケット落ちがおきるかおきないかぎりぎりの速度で送ると、みんなが幸せ
- 紳士協定でしかないが
 - TCPに組み込まれて広く普及
 - 両端で協力しないと協定は破れず

混雑制御の実際

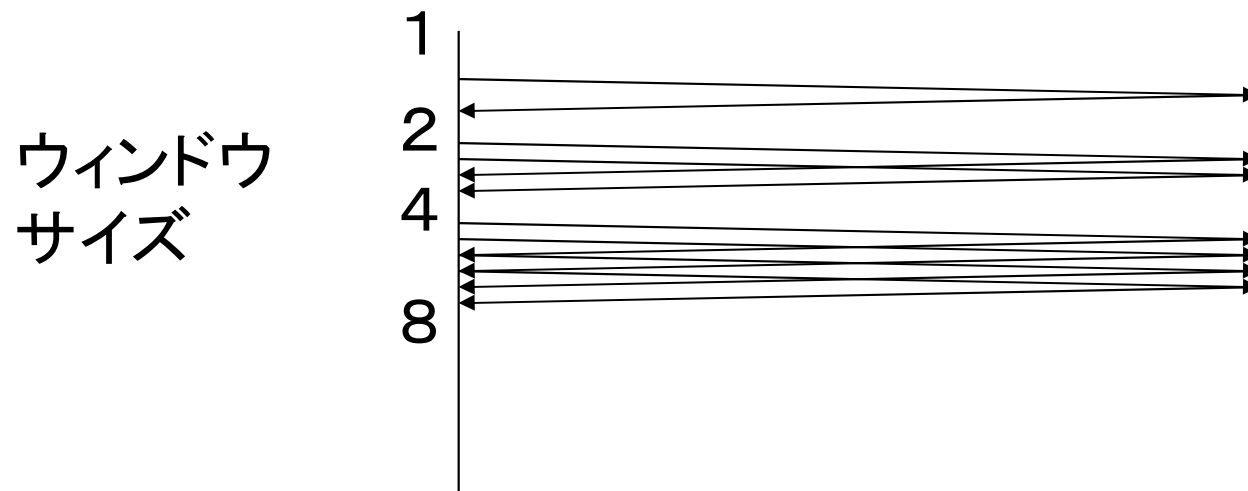
- インターネットの混雑状況に応じてTCPの速度を制御
 - 混んでいる場合、(送信側)ウィンドウを小さくする
- 混雑とは？
 - パケット落ち
- パケット落ちは、タイムアウトか、アクノレツジ番号が増加しないことにより検出

混雑制御とエンドツーエンド原理

- TCPの混雑への対応
 - ルータはパケットを落とすだけ
 - エンドシステムではネットワーク中の混雑状況を推測し、複雑な制御を行う

スロースタート

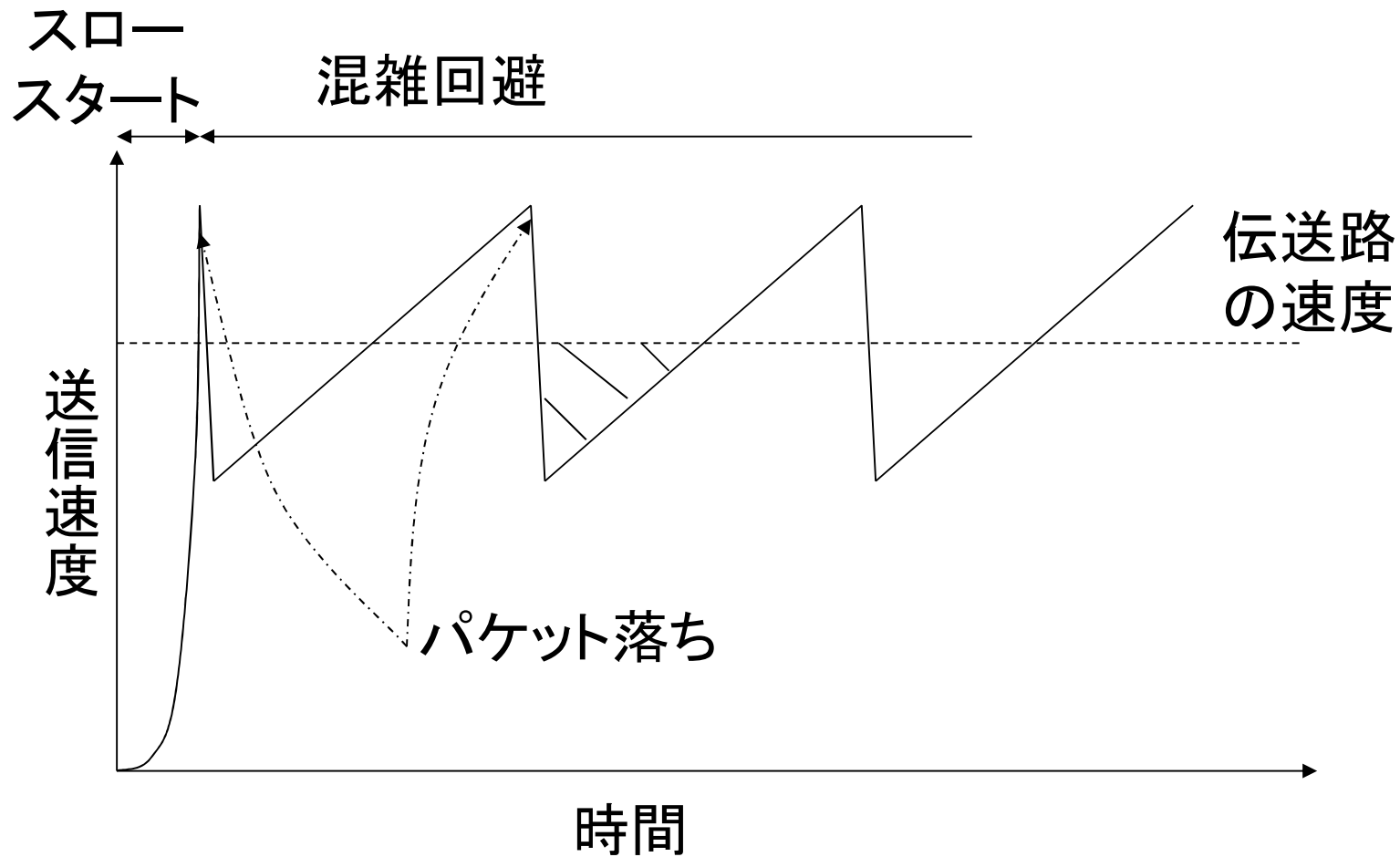
- 最初のウィンドウは最小に
- ACKが順調に帰れば、ウィンドウサイズを一定量だけ増やす
 - 伝送速度は指数的に増加



混雑回避

- パケット落ちがあれば
 - ウィンドウを半分に
- 以後、ウィンドウの増加量は、現在のウィンドウサイズに反比例させる
 - 伝送速度は線形に増加

TCPのトラフィック量の変動



パスMTUディスカバリー

- MTU
 - Maximum Transfer Unit
 - 最大パケットサイズ(データリンクごとに違う)
- パスMTU
 - エンド間の経路のMTUの最小値
- パスMTUディスカバリー
 - パスMTUの推測

パスMTUディスカバリーの必要性

- IPv4では、MTUを越えるパケットは、途中のルータが分割（フラグメンテーション）
 - 処理が重い
 - IPv6では、そもそも禁止
- パケットサイズが大きいほど、処理は軽い
- パスMTUぎりぎりを送るのが効率的

パスMTUディスカバリーの実際

- Don't Fragment Bitをたてる
- MTUが大きすぎると、ICMPエラーがかかる
- 定期的に大きめのパケットを送ってみる
- 実効性は疑問
 - ICMPパケットは、よくフィルターされる
 - マルチキャストでは使えない
 - 定期的にICMPエラーを送る処理は重い

混雑制御の拡張

- エンドシステムでできることはやりつくした
- ルータの機能を拡張
- RED
 - Random Eearly Drop
- ECN
 - Explicite Congenstion Notification

RED

- TAIL Drop
 - バッファがいっぱいになれば、パケットをおとす
- Random Early Drop
 - バッファがある程度ふさがると、低い確率でパケットを落とす
- TCPの混雑制御機構を早めに起動できる

ECN

- 混雑でパケットが落ちそうなときに、そのパケットにマーク
 - ToSビットのうち2ビットを利用
- 効果は疑問
 - 結局、送信元への通知には、RTTだけの遅延は生じる
- とはいえ、パケットを落とさずに済むといえ
ば済む

インターネットの破綻 (ゲーム理論的不安定性)

- パケット落ちは誤り訂正符合で回復可能
 - FEC(Forward Error Correction)
- 混雑が激しいと
 - 強力なFECを利用すれば、自分は幸せ
 - しかし、強力なFECは通信量を増やす
 - みんなが通信量を増やすと、混雑はよりひどく
 - あまり混雑がひどいと、FECにも限界が
- ネットワーク内での帯域保証は必須

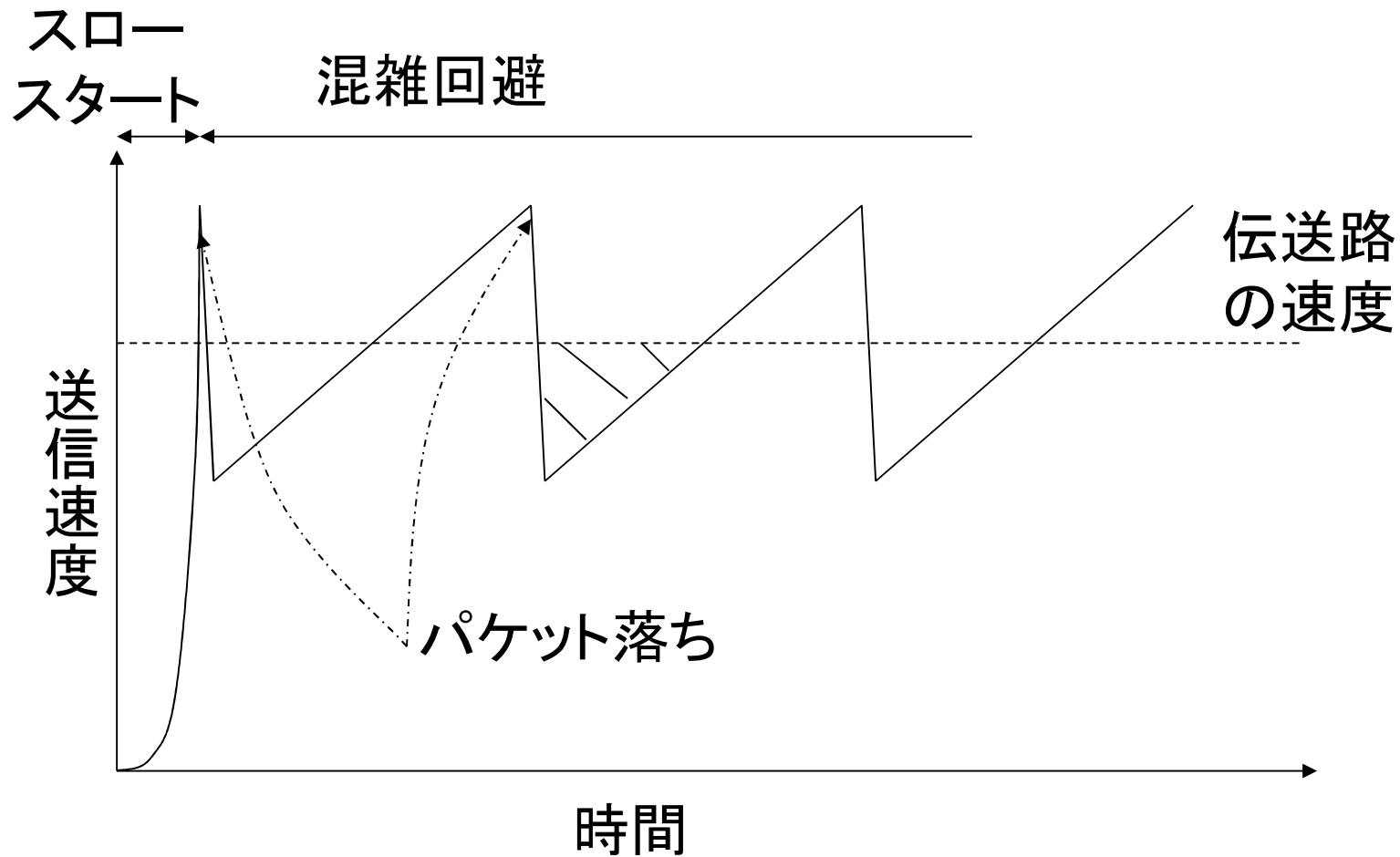
ロングファットパイプ

- TCPの性能はせいぜい
 - 最大伝送速度 = (ウィンドウサイズ) / (RTT)
- 長距離(太平洋越し等)ではRTTが大
 - 大きな伝送速度が必要だと使えない
 - RTT0. 1秒でウィンドウサイズ64KBだと5Mbps
- パケット落ちがおきると悲惨
- 各種の工夫が提案されているが、、、、
 - 帯域保証が筋か？

TCPとルータのバッファ

- CAによりTCPの速度は鋸歯状に変動
- バッファしないと回線速度を使い切れない
 - $(\text{伝送遅延}) * (\text{伝送速度})$ だけのバッファが必要
- 一部幹線では巨大なバッファが必要？
 - 幹線は速い
 - 幹線は長い

TCPのトラフィック量の変動



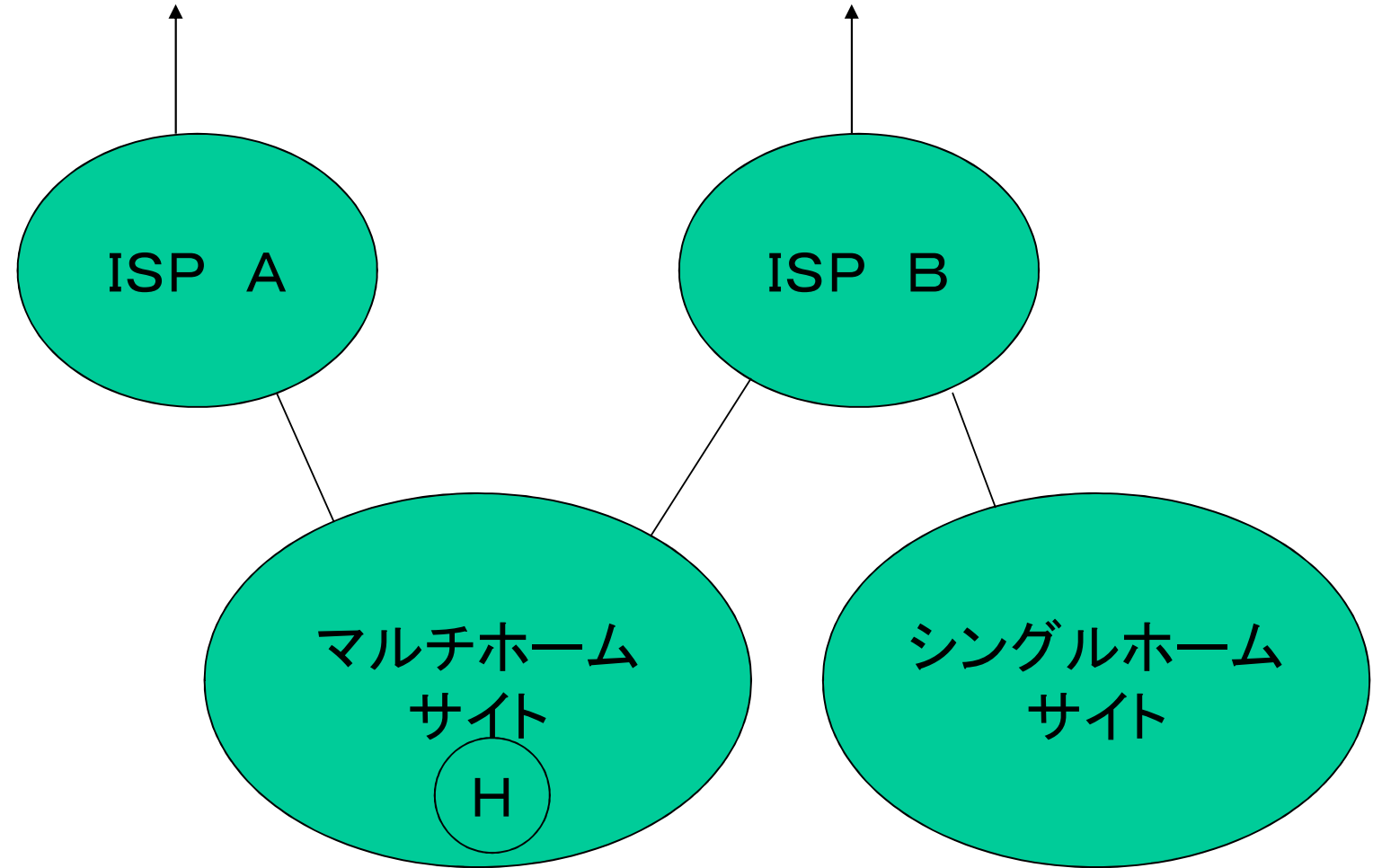
TCPと幹線ルータのバッファ

- 幹線では巨大なバッファが必要？
 - 幹線では多数(N)のTCPの変動が平均されるので(各TCPは独立)
 - 変動は $1/\sqrt{N}$ に
 - バッファは $1/\sqrt{N}$ に？
 - 回線速度を $1/\sqrt{N}$ の数倍犠牲にすれば
 - 総送信速度が回線速度を上回することは、まずない
 - バッファは短時間変動を吸収する数十パケット分で十分
 - » 光ルータが実用的に

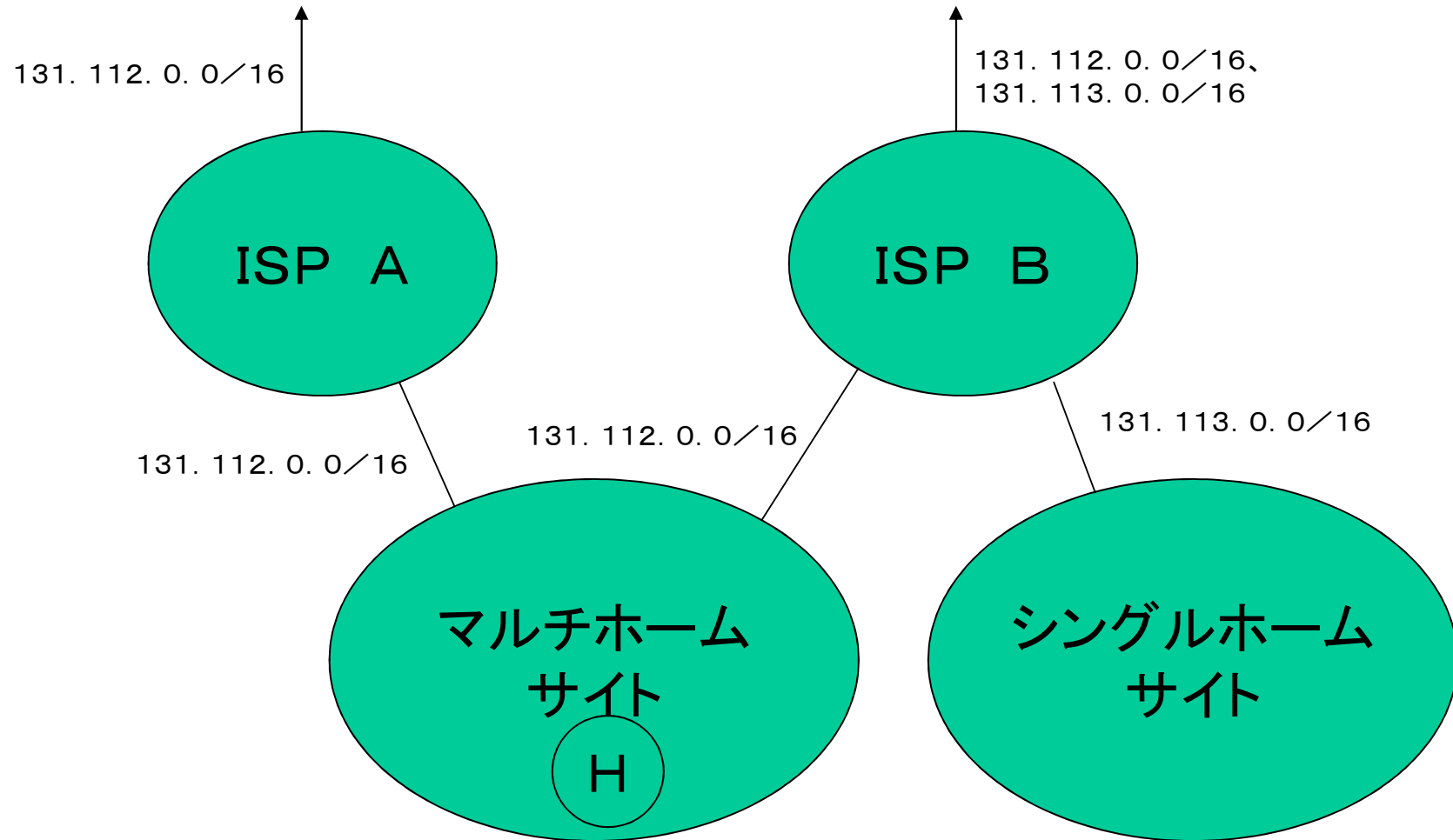
マルチホーミング

- 一般的なIPv4マルチホーミング
 - 1つのアドレス範囲を複数の経路で広告
 - 別のアドレス範囲は、別に広告
 - 他のアドレス範囲とは縮約できない
 - アドレス範囲ごとに経路表が必要
 - 経路表の爆発、破綻
- IPv6では、ホストが複数のアドレスをもつことを積極的に推奨
 - エンドツーエンドマルチホーミングの可能性

残りのインターネットへ



残りのインターネットへ



ルーティングによるマルチホーミング

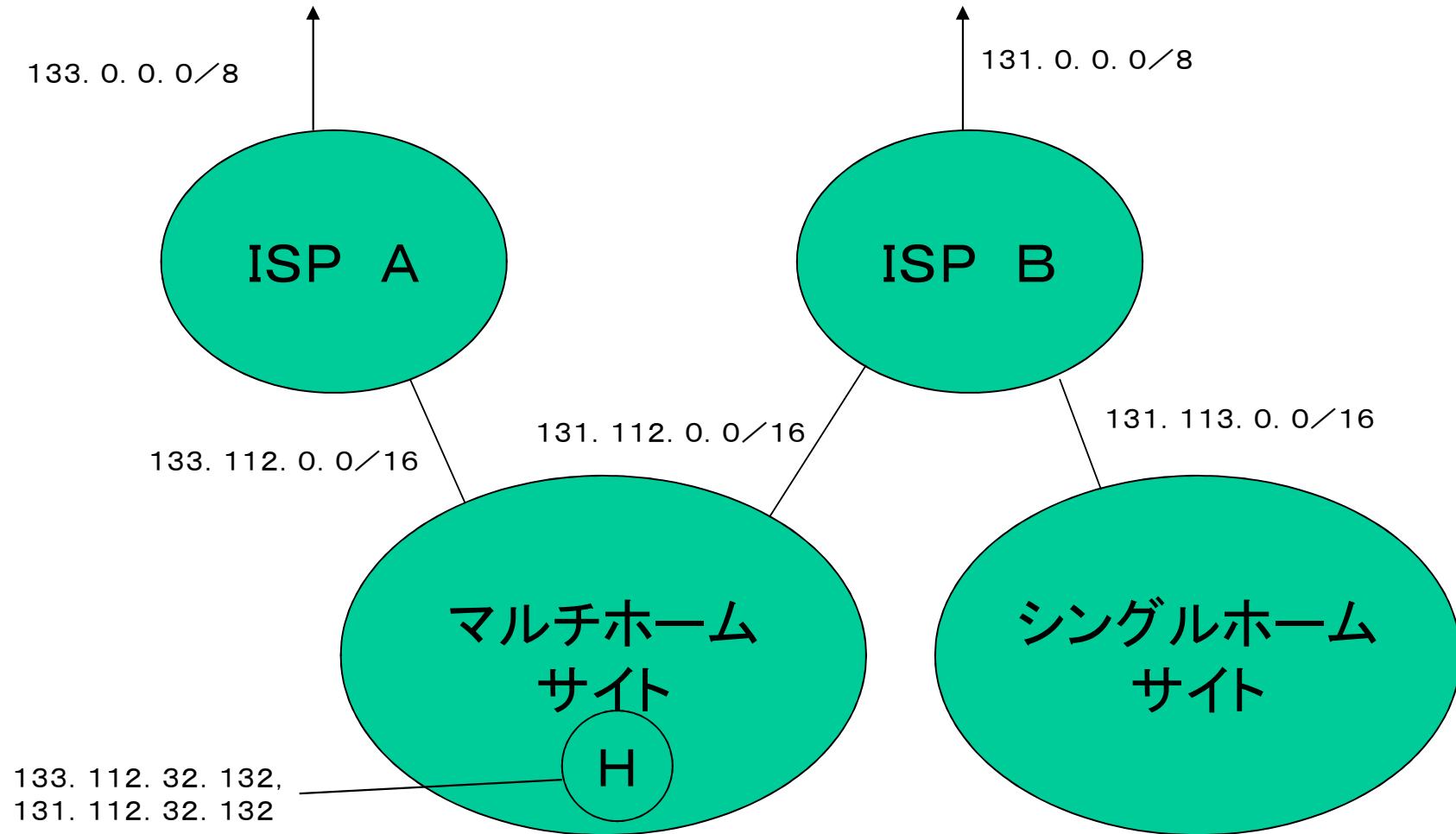
エンドツーエンド マルチホーミング(1)

- ホストは複数のアドレスをもつ
- アプリケーション／トランスポートが各アドレスを試す
 - 個々のアドレスは縮約可能
 - 経路表は爆発しない
 - 今後は必須

エンドツーエンド マルチホーミング(2)

- いつ次を試すかはアプリケーション・トランスポート次第
 - TCPでは標準的なタイムアウトを設定可能
- 縮約した経路表は参考にするだけ
 - 一部の経路がおちても大丈夫

残りのインターネットへ



エンドツーエンドマルチホーミング

電子メールとマルチホーミング

- 電子メール(SMTP+DNS(RFC974))はエンドツーエンドマルチホーミング
 - MXには複数のサーバを指定可能
 - サーバが複数のアドレスを持つ場合
 - 全部のアドレスを試してみる
 - インターネットでもっとも重要なアプリケーションとしては、当然
 - 他にはDNSも

まとめ

- インターネットは電話網なみに信頼できる
- インターネットでは網では速度管理もせず
 - 混めばパケットを落すだけ
- 落ちたデータはエンドがTCPで再送
 - 速度制御もTCPによりエンドで
- マルチホーミングもエンド(アプリケーション・トランスポート層)でやるべき