講義日程

		授業計画		課題
05/12	第９回	ネットワーク層１ ルーティング・輻輳制御	５章	ルーティングの種類を理解し 輻輳制御手法を説明できる
05/16	第１０回	ネットワーク層２ インターネットとサービス品質	５章	インターネットの制御プロトコルを理解し ネットワーク間の接続について説明できる
05/19	第１１回	トランスポート層１ トランスポート・プロトコルの要素	６章	誤り制御とフロー制御を理解し 輻輳制御について説明できる
05/23	第１２回	トランスポート層２ UDP と TCP	６章	TCP の信頼性を理解し TCP のコネクション管理を説明できる
05/26	第１３回	アプリケーション層 DNS, 電子メール, www	７章	DNS, 電子メール, www のしくみを理解し ストリーミング，P2P について説明できる
05/30	第１４回	ネットワークセキュリティ１ 対称鍵暗号，公開鍵暗号	８章	暗号アルゴリズムを理解し SHA-1,2 と RSA について説明できる
06/02	第１５回	理解度確認総合演習 (期末試験) 第９回から第１４回までの内容の演習形式による確認		第９回から第１４回までの理解度確認と 到達度自己評価

Transmission Control Protocol (TCP)

Request for Comments (RFC): インターネット規格

歴史

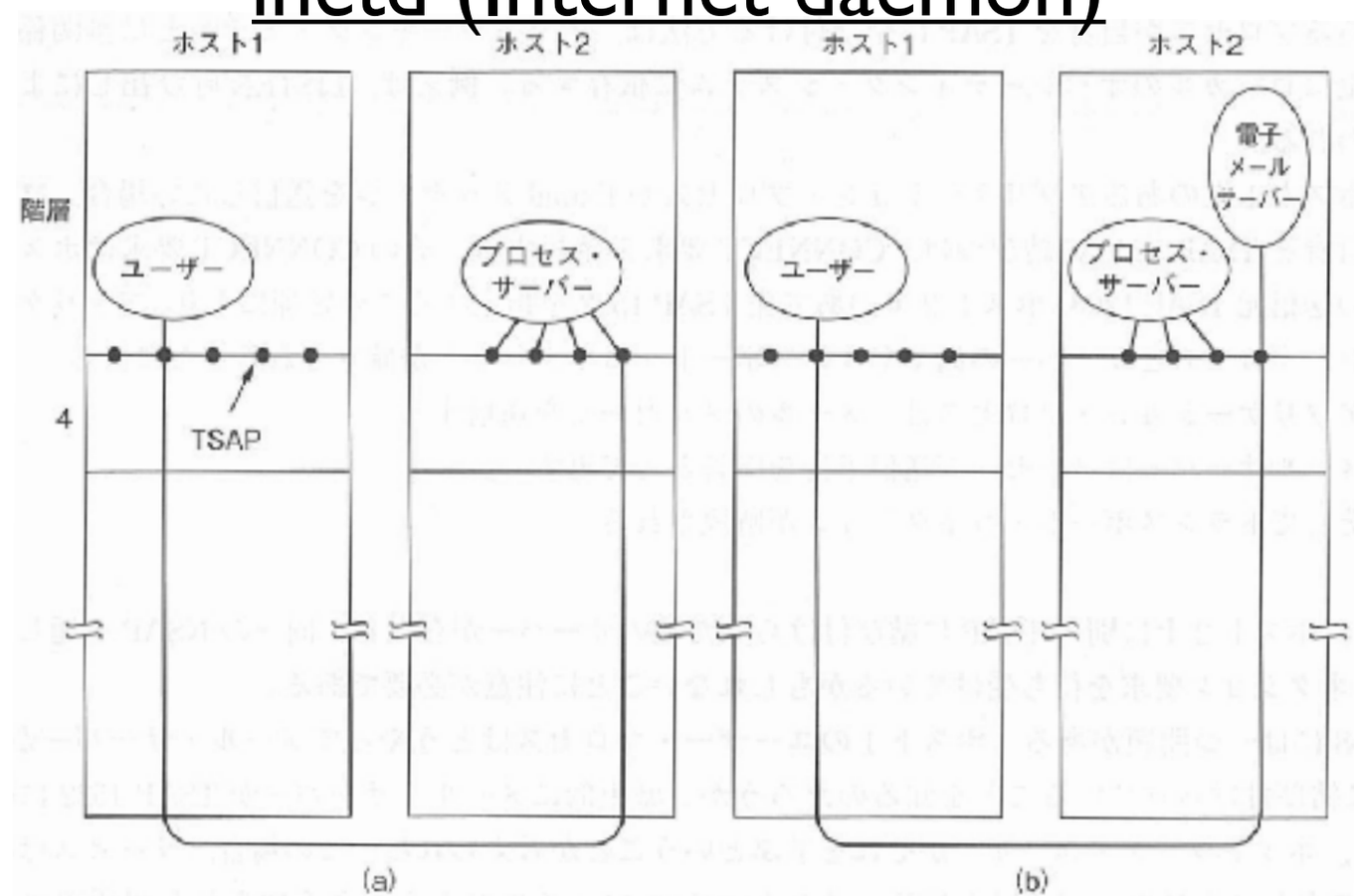
RFC 793: 初期の提案
RFC1122: バグの修正
RFC1323: 高速化のための拡張
RFC2018: 選択的確認通知
RFC 2581: 輻輳制御
RFC 2873: QoSのためのヘッダフィールド解釈変更
RFC 2988: 再送タイマーの改善
RFC 3168: 明示的な輻輳通知

(課題6)

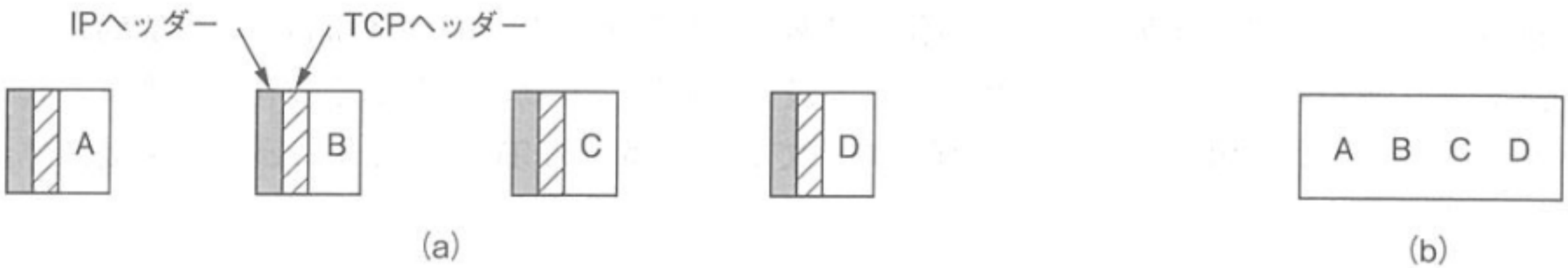
TCPのポート

ポート	プロトコル	利用
20, 21	FTP	ファイル転送
22	SSH	遠隔ログイン（Telnet のおきかえ）
.25	SMTP	電子メール
80	HTTP	ワールド・ワイド Web
110	POP-3	遠隔電子メール・アクセス
143	IMAP	遠隔電子メール・アクセス
443	HTTPS	セキュア Web（SSL/TLS 上の HTTP）
543	RTSP	メディア・プレイヤー・コントロール
631	IPP	プリンタ共有

inetd (Internet daemon)



Transmission Control Protocol (TCP)



TCP バイト・ストリームの特徴

受信プロセスは送信プロセスがどのような単位でデータを送ったか検知できない

TCP Push

転送を送らせずTCPにすぐ転送するように指示する

> TCP_NODELAY

緊急フラグ

TCPにデータを蓄積することをやめさせ、コネクションが持っている全データをすぐに送信するようにさせる

現在はほとんど使われていない

TCPプロトコル

順序番号

32-bit
各セグメントが固有の番号を持つ

セグメント長

65,535-20(ヘッダ)-20(オプション)=65,495

(課題7)

TCPはセグメントの分割、蓄積を行うことができる

パスMTU発見を実装している

Sliding Window Protocol

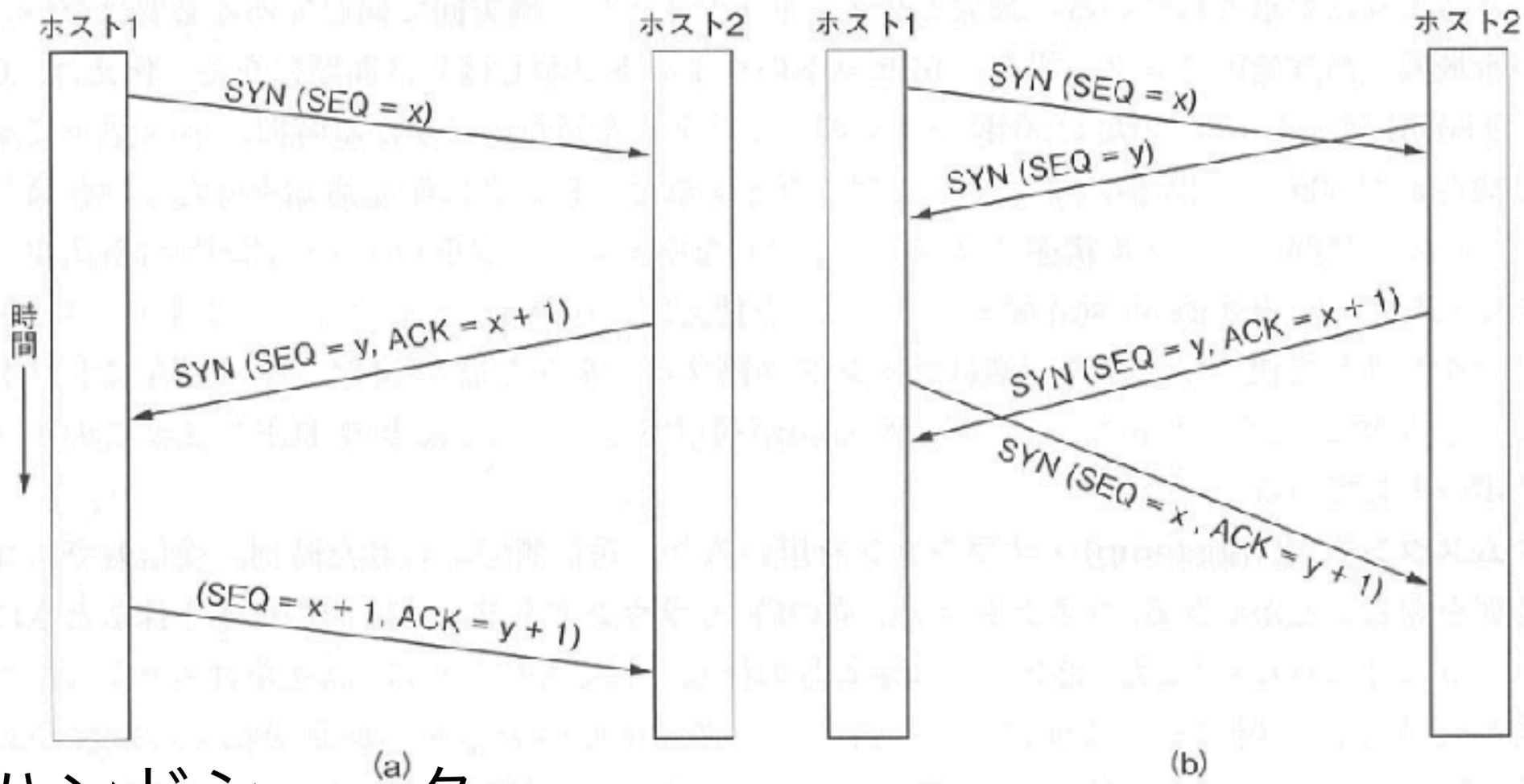
動的なウィンドウサイズ
セグメントは間違った順番で届くかもしれない
再送されたセグメントは分割のされ方が異なるかもしれない

TCP Segment Header

ECE: 送信元に輻輳通知
CWR: 輻輳ウィンドウ縮小
URG: 緊急ポインタ
ACK: 確認通知有効
PSH: PUSHかどうか
RST: リセット
SYN: コネクション要求
FIN: コネクション解放

送信元・あて先ポート: IPアドレスと組み合わせて48ビットの固有な番号となる
順序・確認通知番号:TCPストリームでは全てのバイトに番号が振られる
TCPヘッダ長: 32ビット単位でのTCPヘッダ長
ウィンドウ幅: 何バイト送れるか
チェックサム: ヘッダのチェックサム(必須)
緊急ポインタ: 現在の順序番号から緊急データまでのオフセット
オプション: 最大セグメント長, ウィンドウ幅, タイムスタンプ, SACK

TCP Connection Establishment



３方向ハンドシェーク

1. ホスト1のCONNECTプリミティブがSYN=1,ACK=0のセグメントを送信
2. ホスト2は受理するならSYN=1,ACK=1を返信
3. 双方が同時にSYNを送ると1つしか接続は確立されない

SYNフラッド

1990年代のwebサーバ攻撃

SYNクッキーで対策（暗号的関数で順序番号を生成）

TCP Connection Management Modeling

コネクション解放

1. 単方向がそれぞれ独立に解放される
2. FIN=1のセグメントを送る
3. 受理されればその方向は切断される
4. 双方向に切断されたらコネクションは解放される
5. タイマーを使って２軍隊問題を避ける

TCP接続の有限状態機械

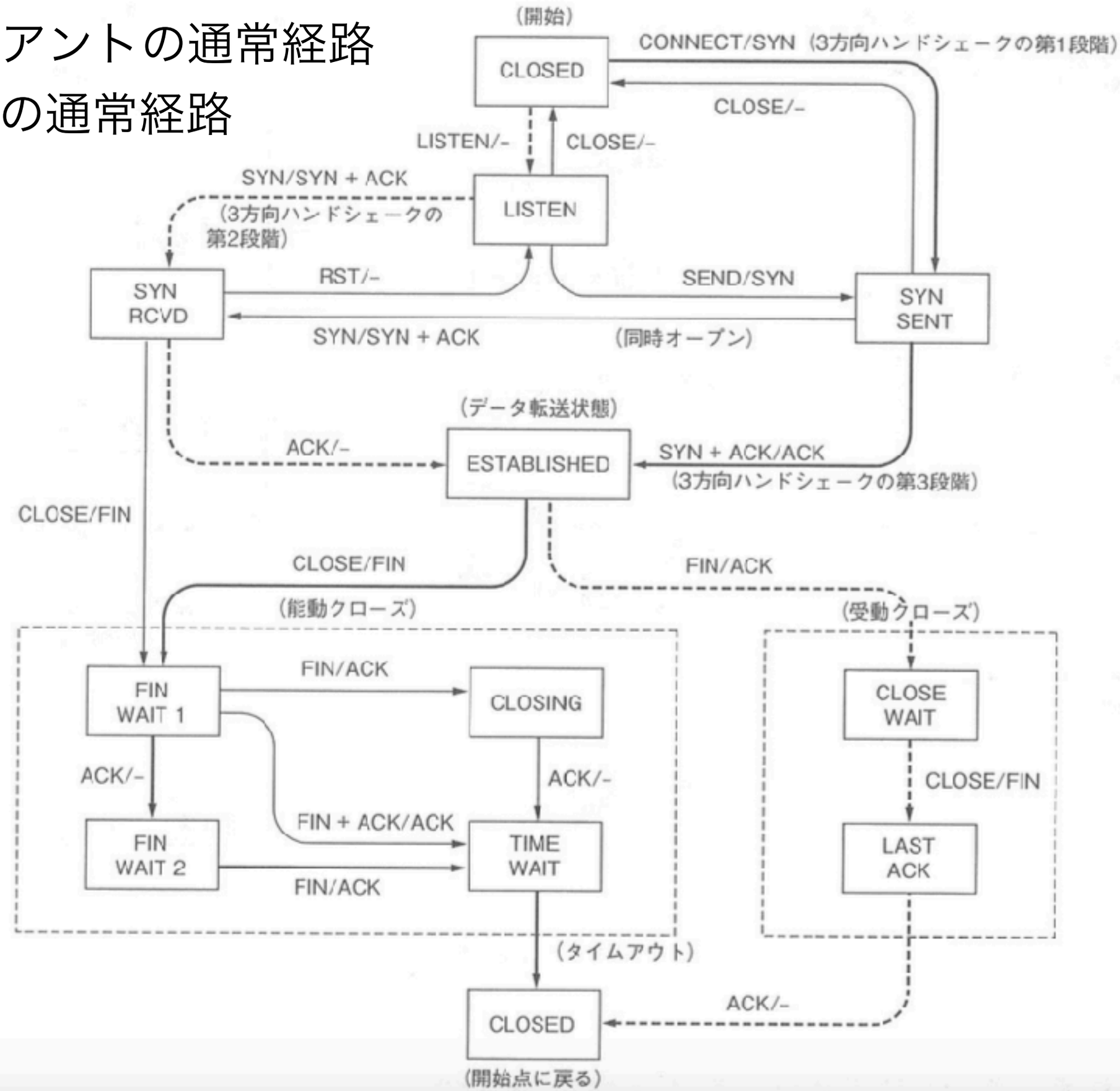
状態	説明
CLOSED	アクティブまたは待機中のコネクションはない状態である
LISTEN	サーバーは呼び出しを待っている
SYN RCVD	コネクション要求が到着した（ACK を持つ）
SYN SENT	アプリケーションがコネクション確立を開始する
ESTABLISHED	通常のデータ転送状態である
FIN WAIT 1	アプリケーションが自身の終了を主張する
FIN WAIT 2	もう一方がコネクション解放に合意する
TIME WAIT	すべてのパケットが消え去るのを待つ
CLOSING	両端が同時にコネクション閉鎖を試みる
CLOSE WAIT	もう一方が先にコネクション解放を開始する
LAST ACK	すべてのパケットが消え去るのを待つ

TCP接続の有限状態機械

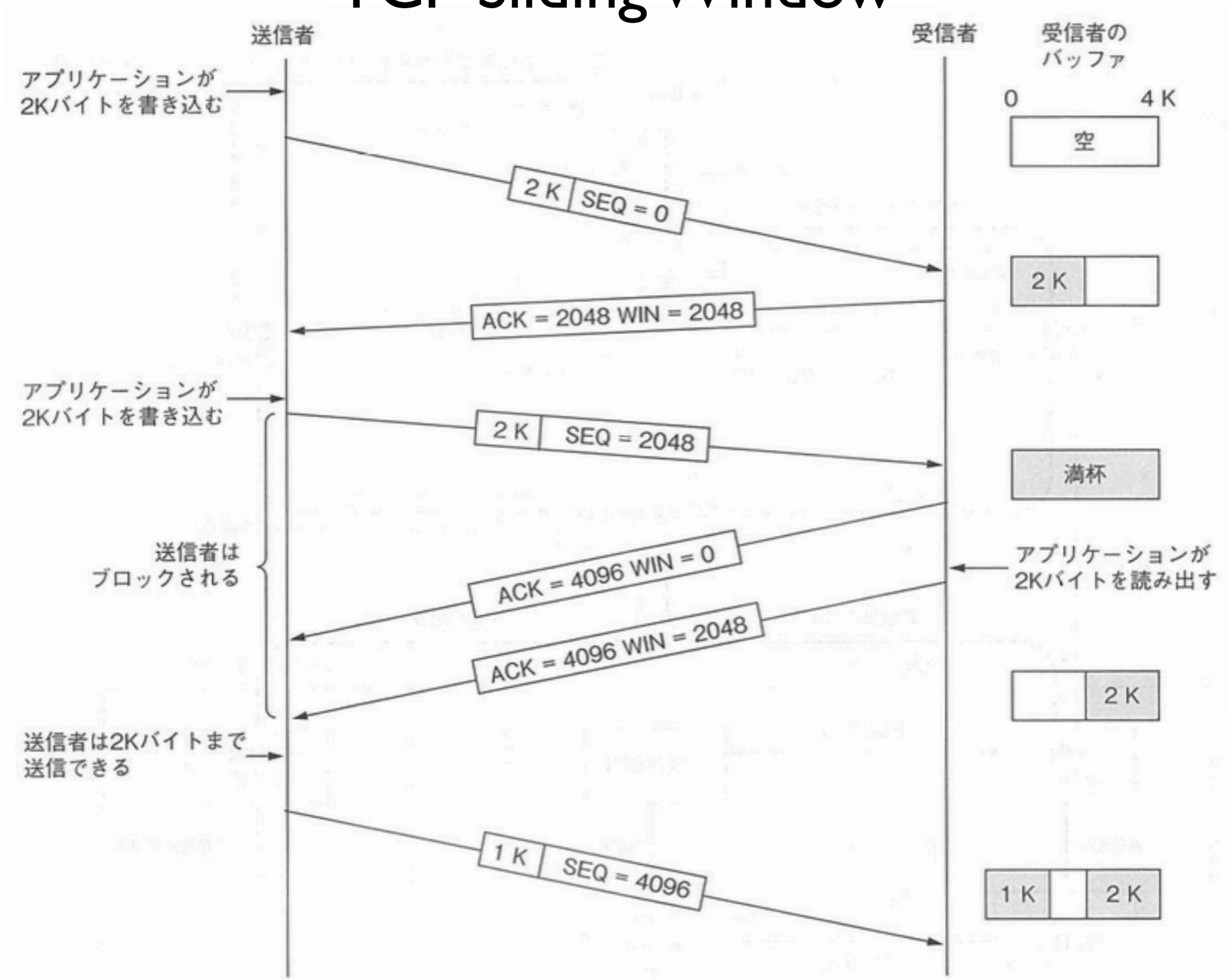
太い実線：クライアントの通常経路

太い点線：サーバの通常経路

(課題8)



TCP Sliding Window



ウィンドウ・プローブ

送信者が１バイトのセグメントを送信し受信者に期待される次のバイトとウィンドウ幅を通知するよう呼びかける

Nagle のアルゴリズム

遅延確認応答

確認通知とウィンドウ更新を最大500ms遅らせて、ピギーバックできるようにする

Nagle のアルゴリズム（まとめて送信）

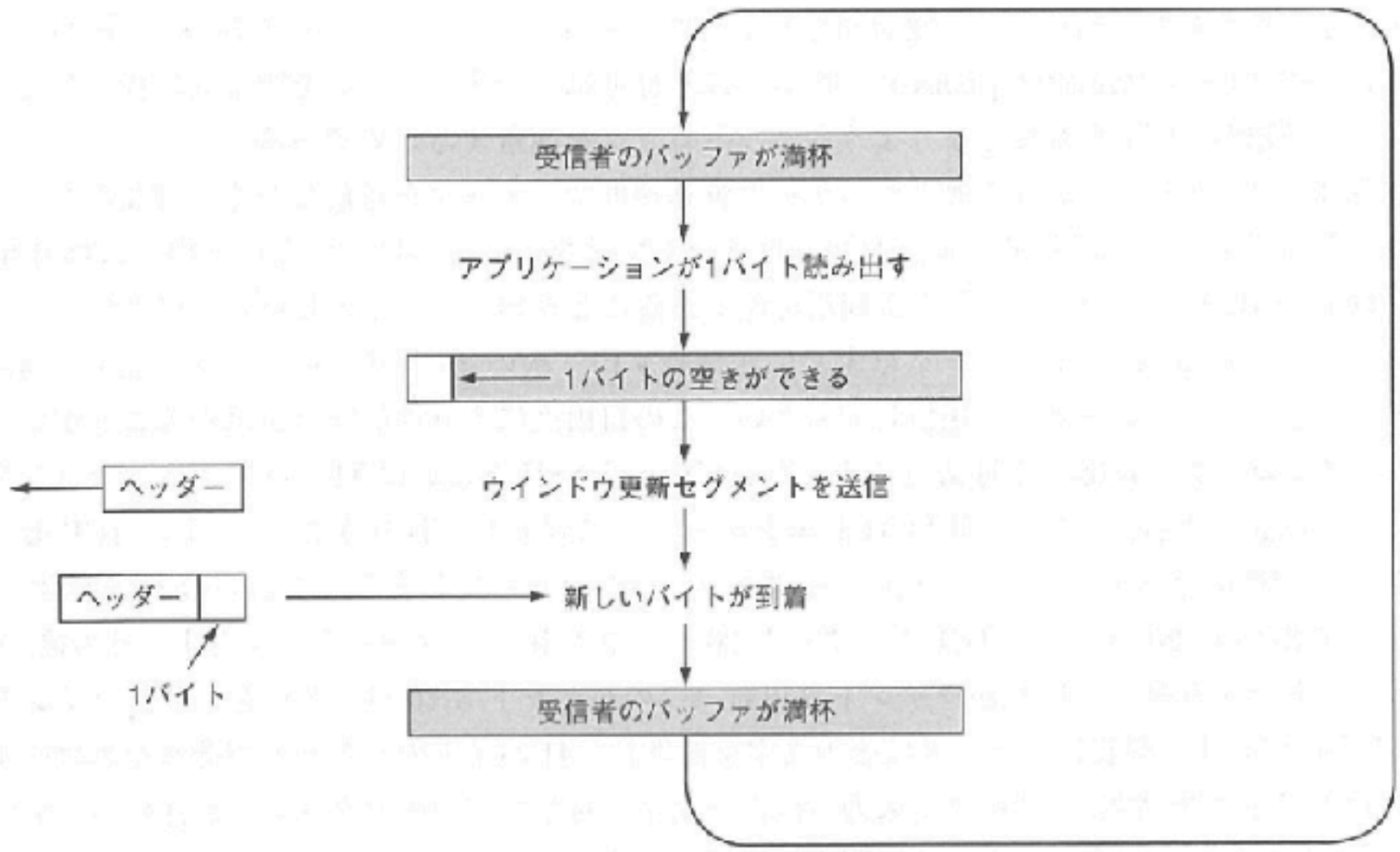
1. データがアプリケーション層から大量に送られてくる場合、一つ目だけ送信し残りをバッファリングしながら確認通知を待つ
2. バッファリングされたデータをまとめて一つのセグメントとして送り、その確認通知がくるまでまたバッファリングを続ける
3. バッファが上限に達した場合もセグメントは送信される

Nagleのアルゴリズムを止める

TCP_NODELAYを用いる

遅延確認通知と併用するとデッドロックに陥る場合があるため

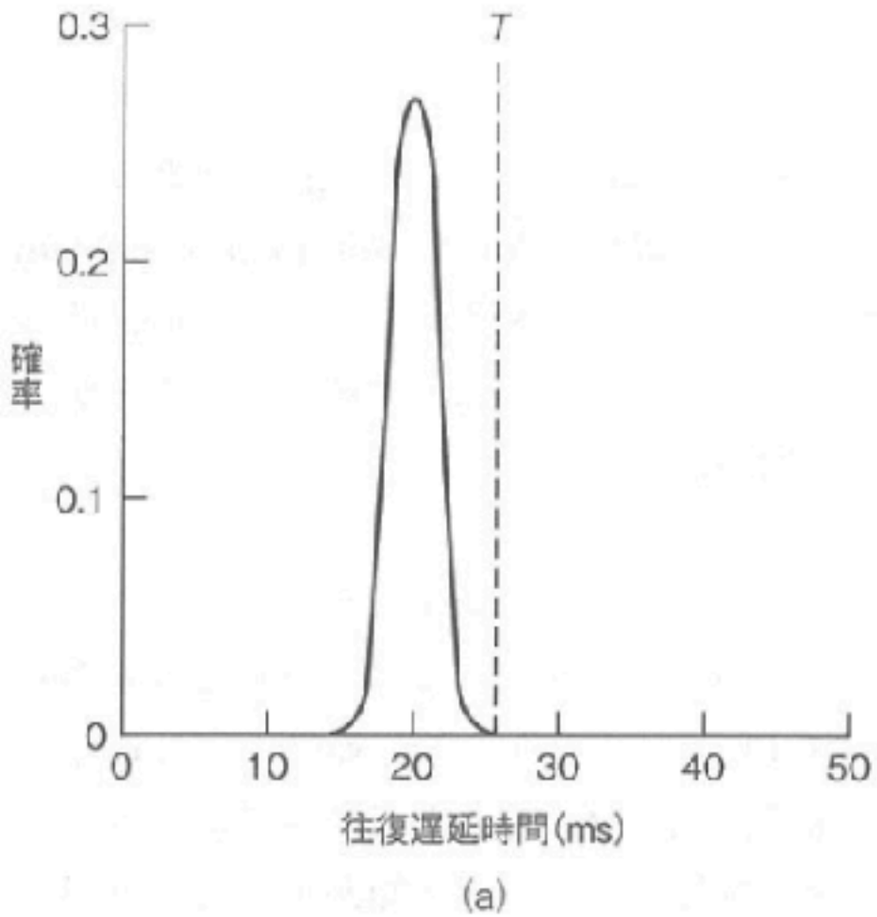
Silly window syndrome



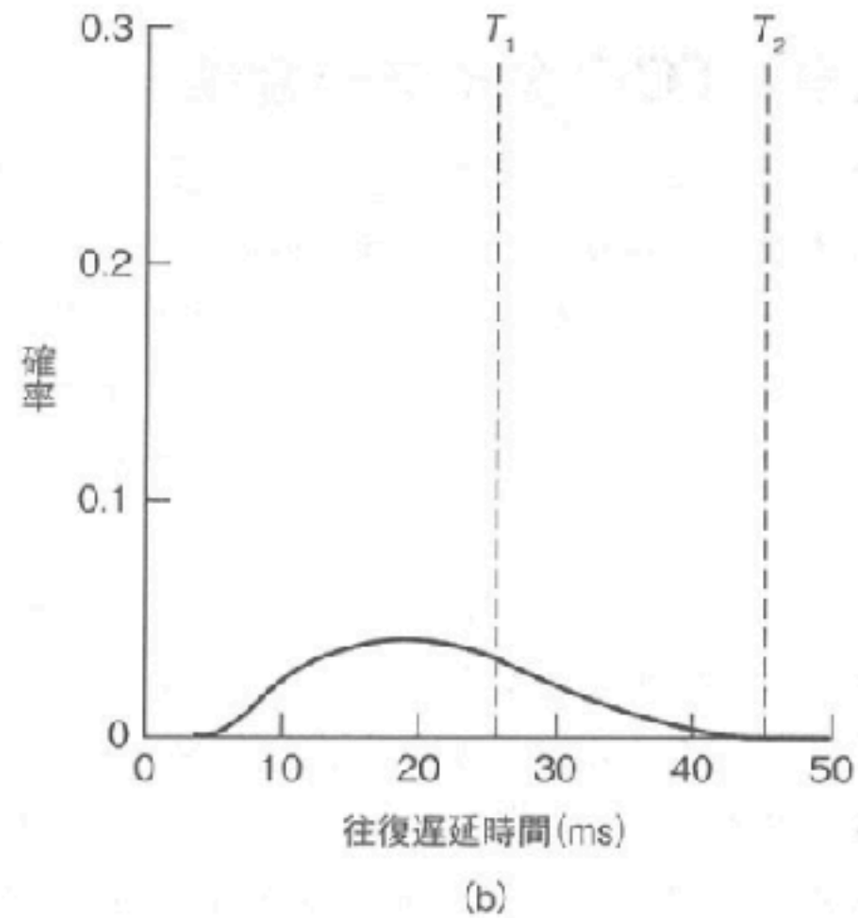
受信側の対話型アプリケーションがデータを１バイトずつしか読み取らないときに起こる

<u>Clarkeの解決法（まとめて受信）</u>
コネクション確立時に通知した最大セグメント長、あるいはバッファの半分の大きさの空きができるまではウィンドウ更新を送らない

TCP タイマー管理



データ・リンク層



トランスポート層

<u>タイムアウトの間隔はどれくらいにすべきか？</u>

データ・リンク層では輻輳がないため遅れたものは失われたと考える
トランスポート層では往復遅延時間を予測することは難しい

タイムアウトを短くすると不必要な再送が起こってしまう
長くしすぎるとパケット紛失が起こったときに再送遅延が大きくなる

TCP タイマー管理

指数荷重移動平均(EWMA: Exponentially Weighted Moving Average)
ネットワーク性能を継続的に計測し、それに基づいてタイムアウト時間を調節できるような動的アルゴリズムを利用

Jacobsonのアルゴリズム
平滑化往復遅延時間(SRTT: Smoothed Round-Trip Time)を求める

$$SRTT = \alpha\, SRTT + (1 - \alpha)\, R$$

(課題10)

R: 確認通知を受け取るまでにかかった時間
α: 古い情報を取っておく割合

→遅延の分散が大きくなったときに対応できない

往復遅延時間の平均偏差 (RTTVAR : Round- Trip Time VARiation)

$$RTTVAR = \beta\, RTTVAR + (1 - \beta)\, |SRTT - R|$$

再送タイムアウト(RTO : retransmission timeout)

$$RTO = SRTT + 4 \times RTTVAR$$

TCP タイマー管理

タイムアウトによって再送されたセグメントをRTTの計算に用いるか？

Karnのアルゴリズム

再送されたセグメントはRTTの計算に用いない
送信が失敗するごとにタイムアウトを2倍にする

永続タイマー

永続タイマーが切れると送信者は受信者にウィンドウ幅を訪ねる

キープアライブ・タイマー

コネクションが長い間アイドルのとき、相手がまだ応答するか検査する
応答がない場合コネクションを終了する

TCP 輻輳制御

ネットワーク層から輻輳のフィードバックを受け取り、トラフィック・レートを低くするのはトランスポート層の仕事である

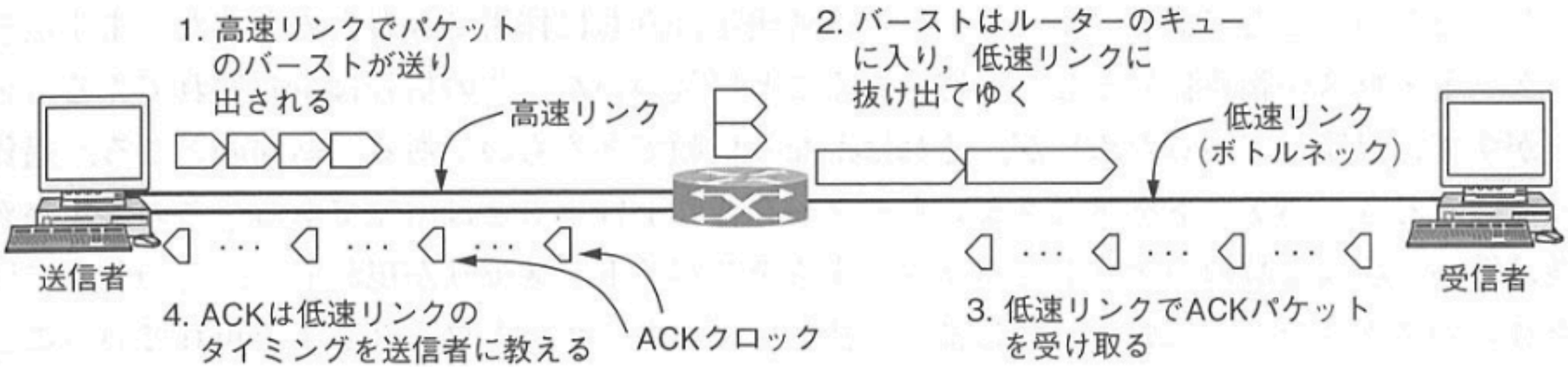
TCPの輻輳制御はパケット・ロスを信号としてAIMD制御ルールを使用する

輻輳ウィンドウ (ネットワークの容量)
送信側がネットワーク上に滞在させることができるバイト数の上限

フロー制御ウィンドウ (受信側の容量)
受信側がバッファできるバイト数の上限

インターネットのあらゆるTCPアルゴリズムは損失パケットを輻輳に起因するとみなす

ACK クロック

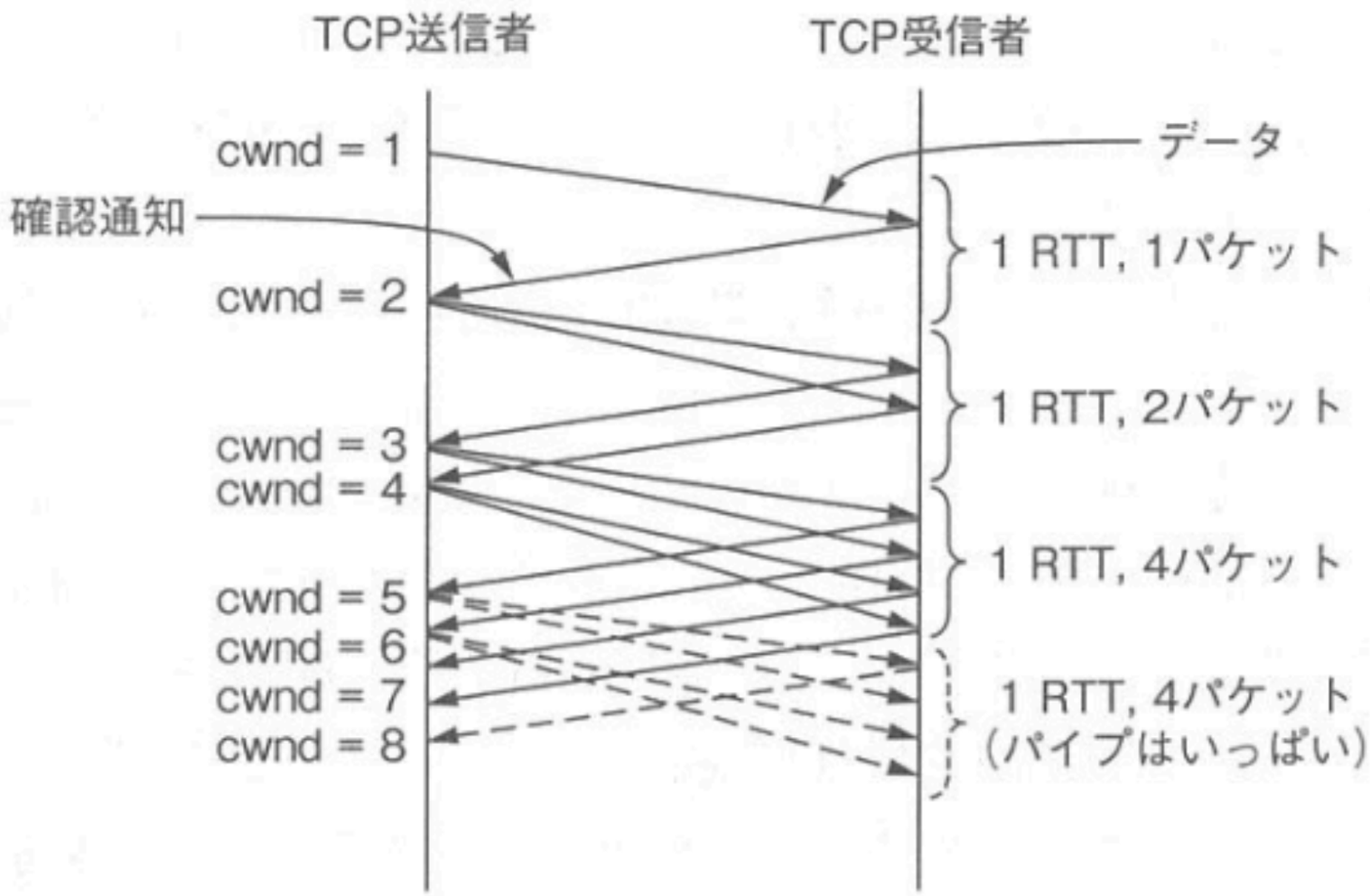


確認通知は経路上の最も遅いリンクを伝送するのとほぼ同じ間隔で戻ってくる

新しいパケットがこの速度で送り出されたら、最も遅いリンクでも送信可能なため輻輳は起きない

スロー・スタート

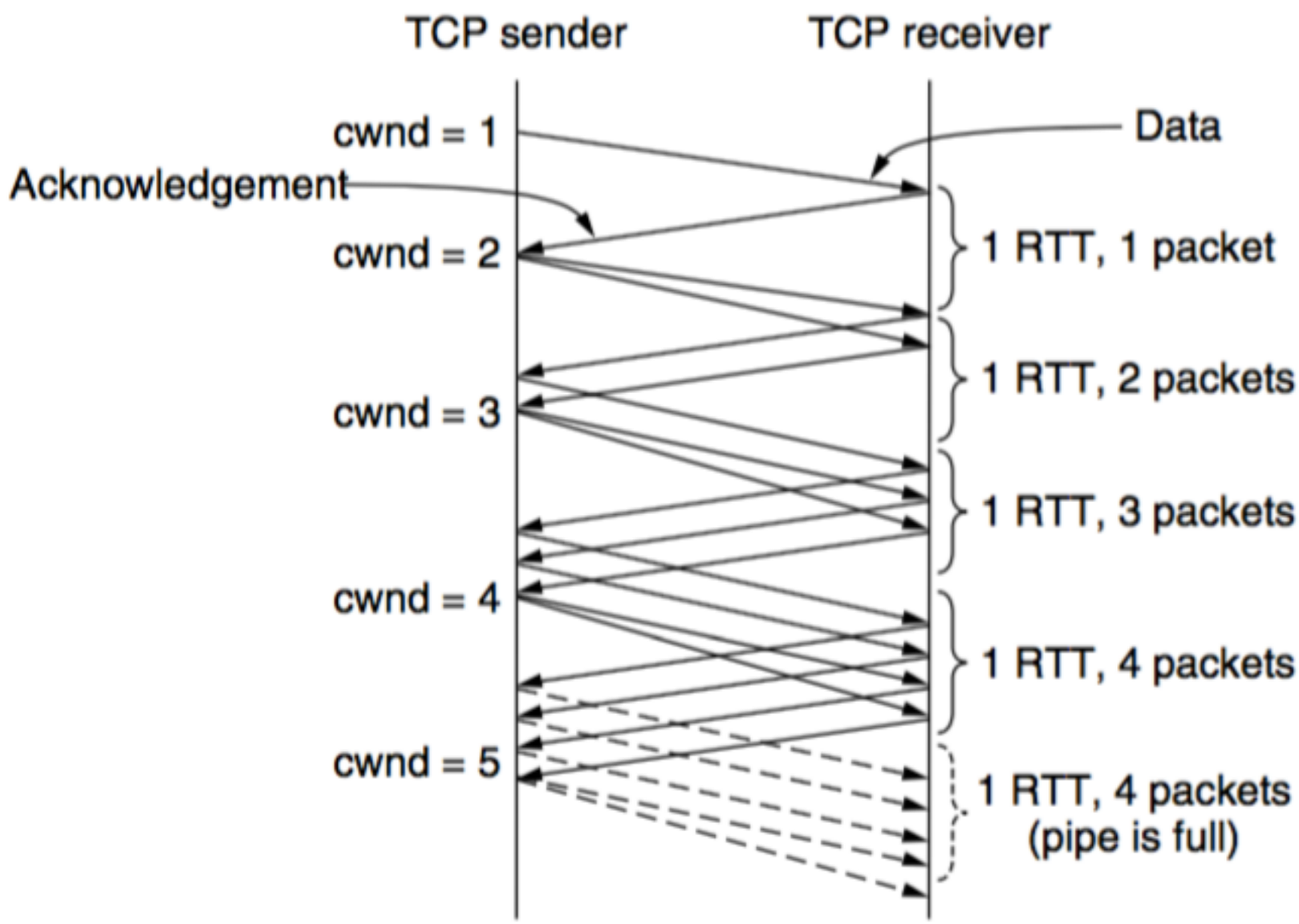
輻輳ウィンドウが小さいサイズから始まった場合AIMDルールが最適な状態を実現するまでに時間がかかってしまう



スロー・スタートでは、輻輳ウィンドウが指数関数的に増加するため、多くのパケットを早い段階でネットワークに送信してしまう

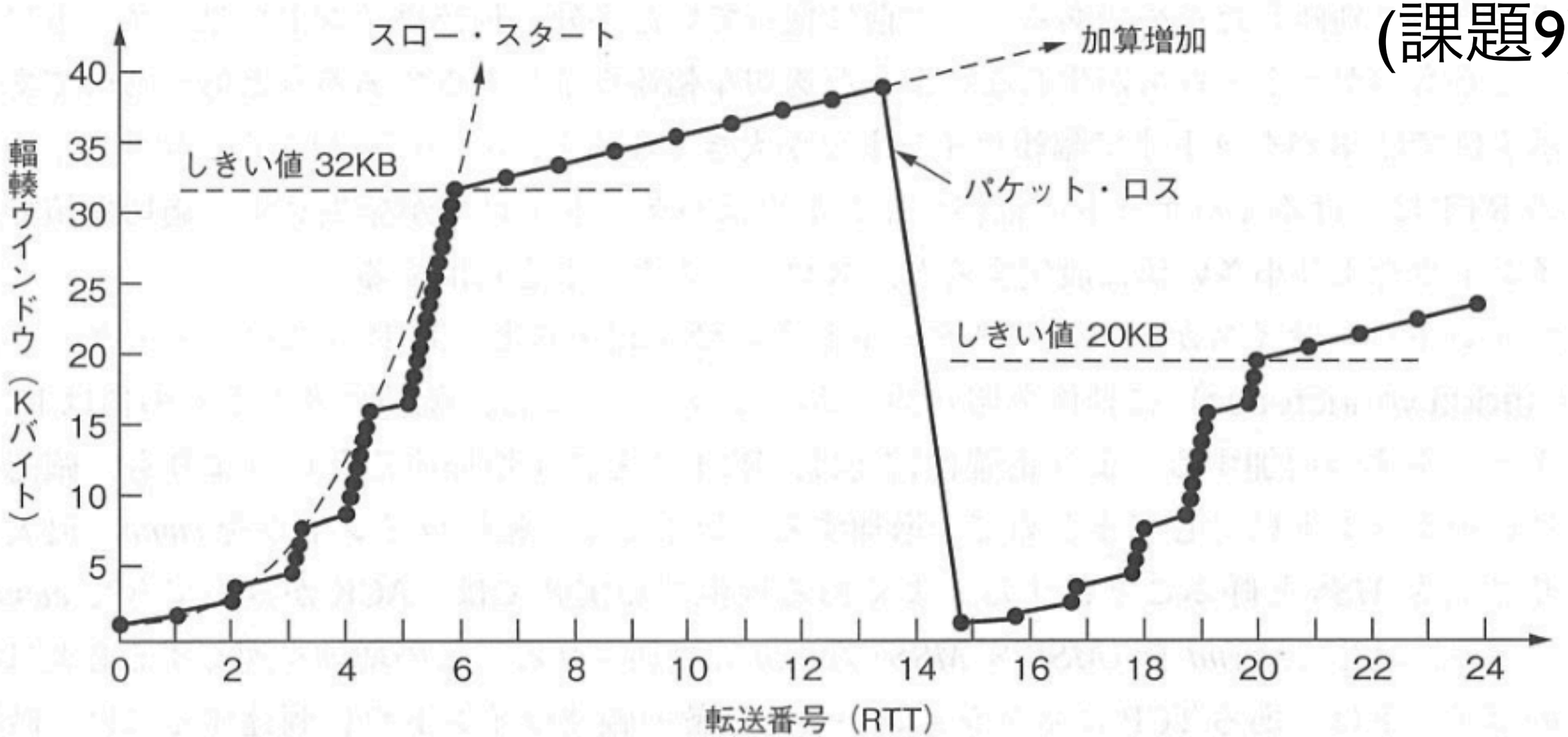
スロー・スタートしきい値

1. しきい値の初期値にはフロー制御ウィンドウ幅より大きな任意の値が設定される
2. パケット損失が検出されるとしきい値を輻輳ウィンドウの半分に設定
3. スロー・スタートしきい値を超えると加算増加に切り替わる



スロー・スタートしきい値

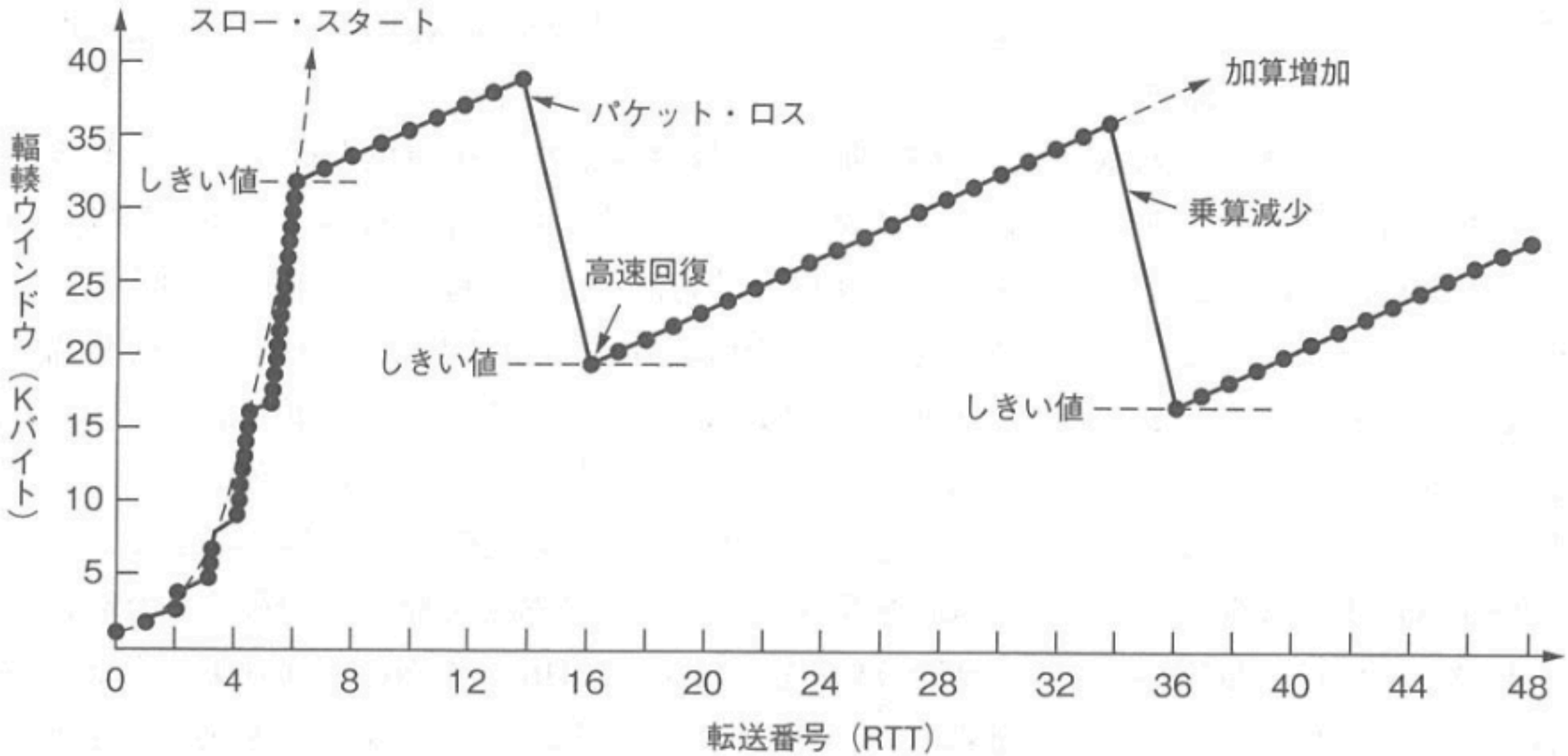
(課題9)



<u>TCP Tahoe</u>

スロー・スタートから加算増加に切り替える

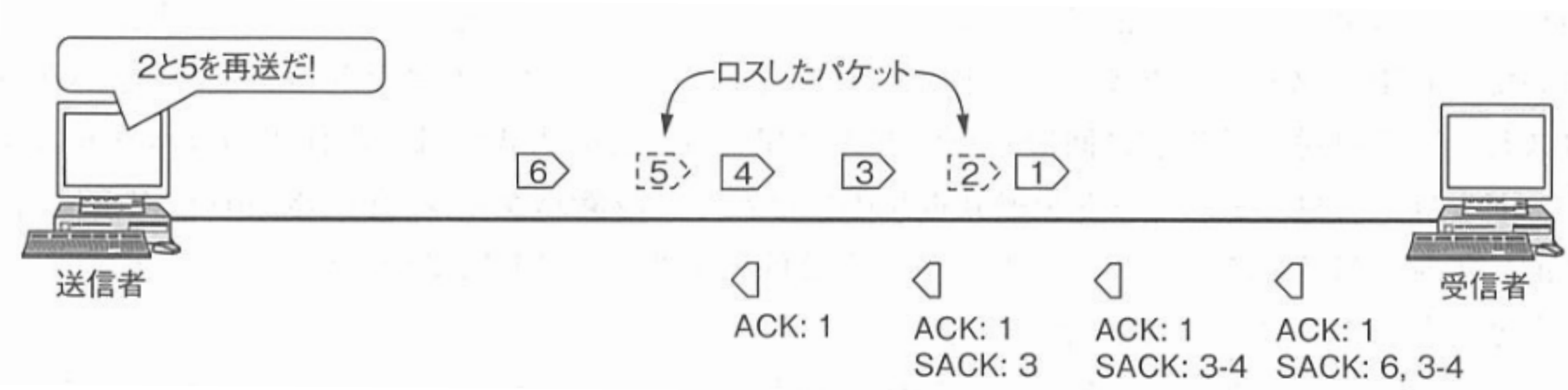
スロー・スタートしきい値



<u>TCP Reno</u>

スロー・スタートから加算増加に切り替える＋高速回復（乗算的減少）

選択確認応答(SACK: Selective ACKnowledgements)



受信したパケットの範囲を三つまでリストアップする。送信側はこの情報からどのパケットを再送すれば良いかを正確に分かる

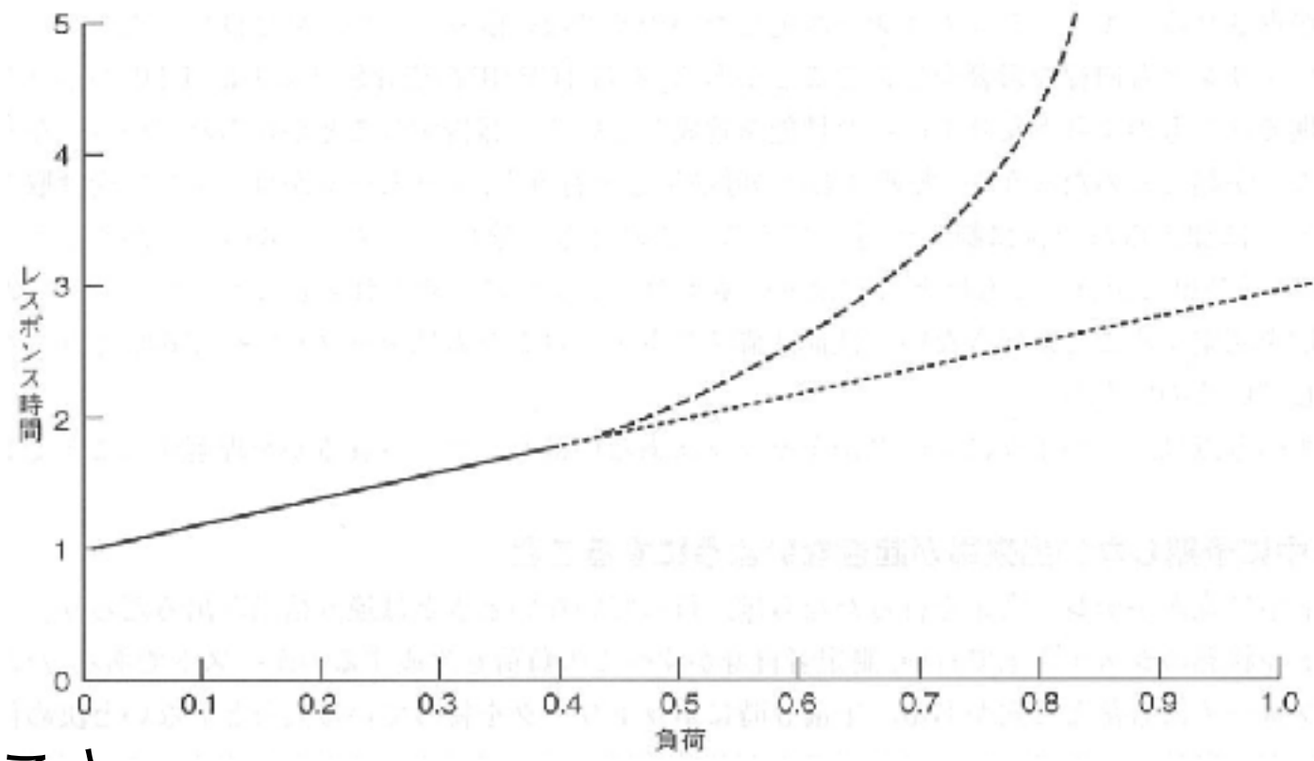
明示的な輻輳通知(ECN: Explicit Congestion Notification)
TCPコネクションでECNを利用するには、送信側と受信側の両方がECNエコー(ECE)と輻輳ウィンドウ縮小(CWR)フラグを設定する必要がある

1. ECNをサポートするルータは輻輳が起きたら輻輳信号を設定する
2. 受信側は輻輳信号を見てECEフラグを設定して送り返す
3. 送信側はECEフラグを見てCWRフラグを設定して送り返す
4. これは重複ACKを通して輻輳を検知した場合に比べて圧倒的に良い

コンピュータ・ネットワークの性能問題

性能低下の原因

1. 過負荷
2. 資源の不均衡
3. ブロードキャスト・ストーム (誤り通知によるもの)
4. 電力障害後の一斉復旧
5. システム・チューニングの不足
6. タイムアウトの設定
7. ジッタ(リアルタイム・アプリ)



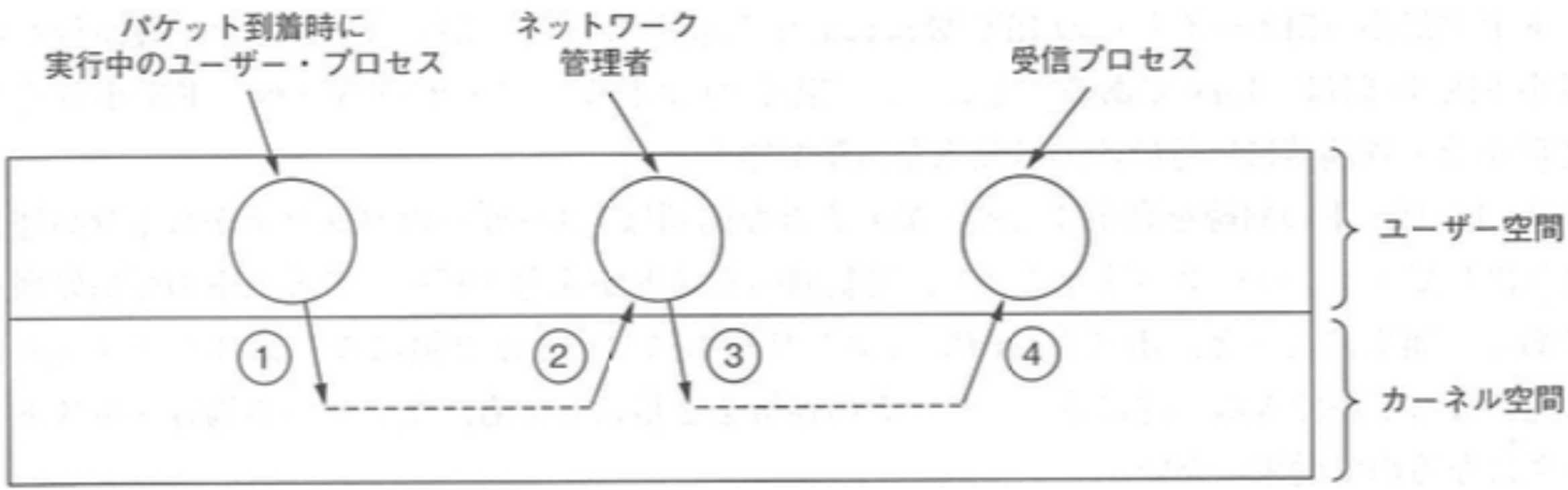
ネットワーク性能の計測

1. サンプル数が十分に大きいこと
2. サンプル数が代表的なものであること
3. キャッシュは測定に悪影響を与えることがある
4. テスト中に予期しない出来事が起きないようにすること
5. 精度の低いクロックに注意すること
6. 結果の推定には注意すること

コンピュータ・ネットワークの性能問題

高速ネットワークのためのホスト設計

1. ホストのスピードはネットワークのスピードよりも重要である
2. パケットの個数を減らし、ソフトウェアのオーバーヘッドを削減せよ
3. データ操作を最小にせよ
4. コンテキスト・スイッチの回数を最小にせよ
5. 輻輳からの回復を考えるよりも輻輳を回避せよ
6. タイムアウトを回避せよ

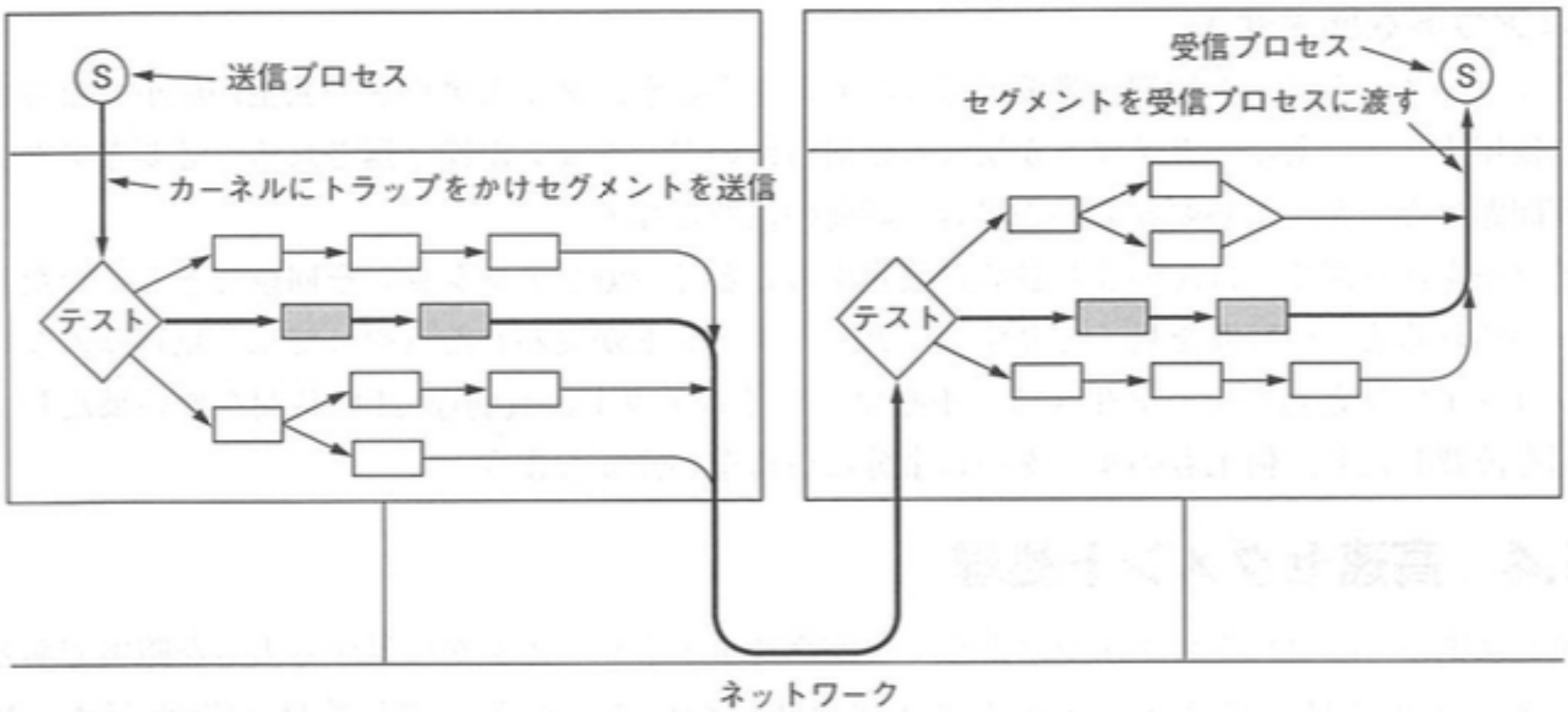


高速セグメント処理

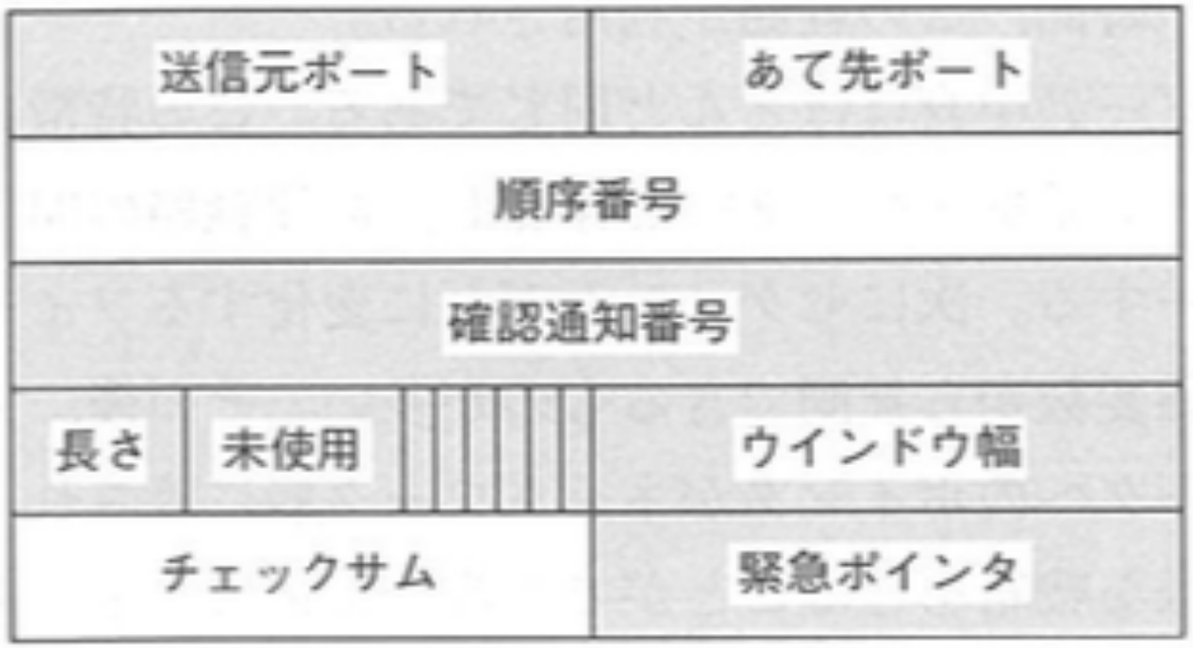
セグメント処理のオーバヘッド

1. セグメントあたりのオーバーヘッド
2. バイトあたりのオーバーヘッド

鍵となる考え方は、通常のうまく行っている場合を特別に扱い高速化すること



高速セグメント処理



TCP

IP

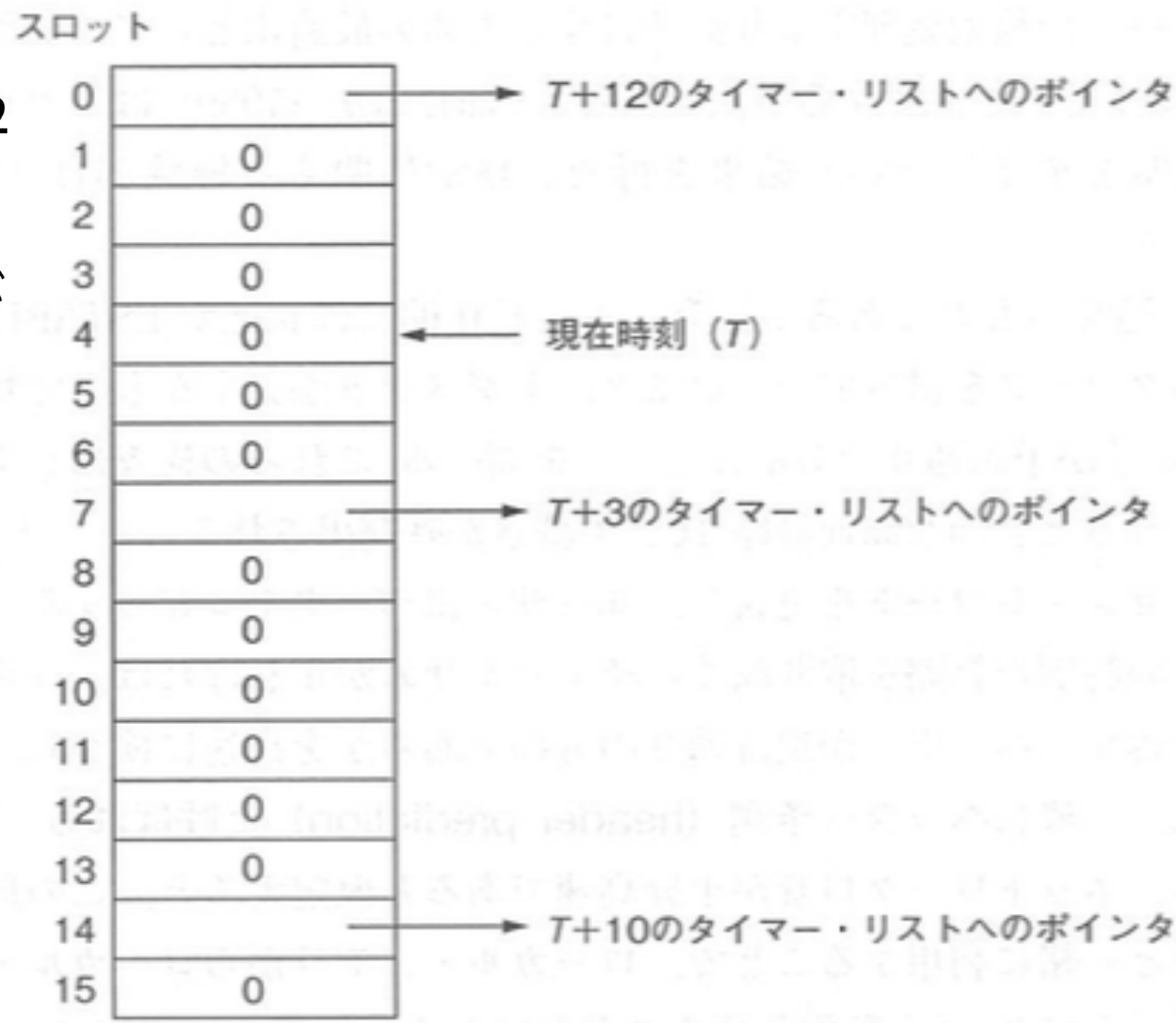
灰色の部分は連続するセグメント間で共通

1. コネクション・レコードはハッシュ・テーブルに保存
2. 送信・受信のIPアドレスとポート番号がハッシュ・キー
3. アドレスとポートを見て正しいレコードか確認
4. 最後に利用したレコードのポインタを保持する

Timing Wheel

タイマーはそれぞれ3, 10, 12単位時間後に切れる

例えば、T+10のタイマーが切れる場合、スロット14のリストが探索され、削除される



ヘッダ圧縮

プロトコルのフォーマットを 詳細に分析して行われる

<u>Van Jacobson (1990)</u>
典型的な 40バイトのTCP/IP ヘッダを3バイトに圧縮

<u>ROHC (RObust Header Compression)</u>
40バイトのIP/UDP/RTPヘッダを1バイトないし3バイトに圧縮
無線リンクで生じるロスにも耐えられる設計

ヘッダ圧縮は帯域だけでなく遅延を削減するのにも役立つ

ロング・ファット・ネットワーク

32ビットの順序番号が周回するのにかかる時間

56-kbps: 1 週間
10-Mbps: 57 分
1-Gbps: 34 秒 (< インターネット上でのパケットの生存時間 120 秒)

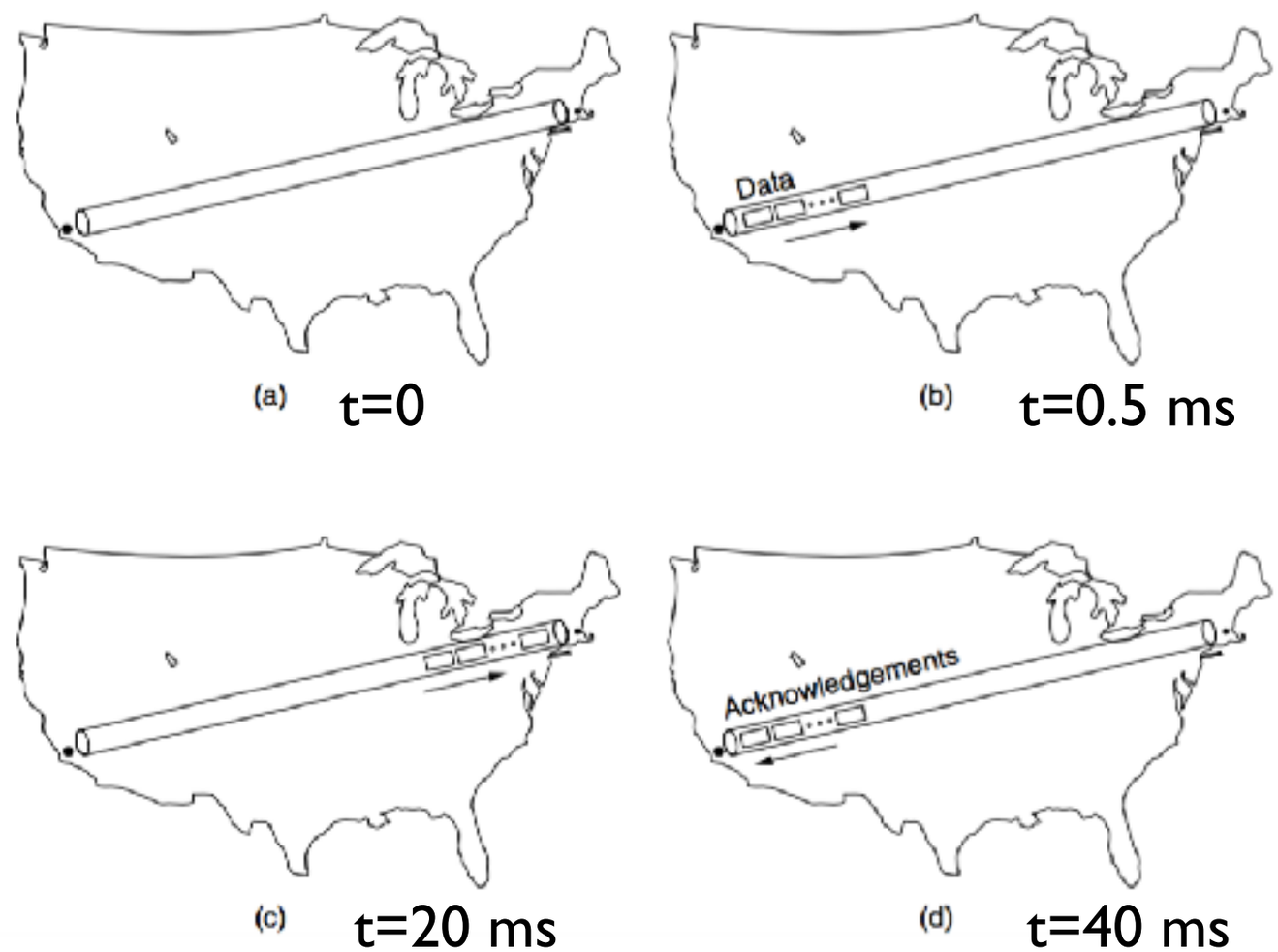
PAWS (Protection Against Wrapped Sequence numbers)

順序番号周回回避

帯域幅遅延積

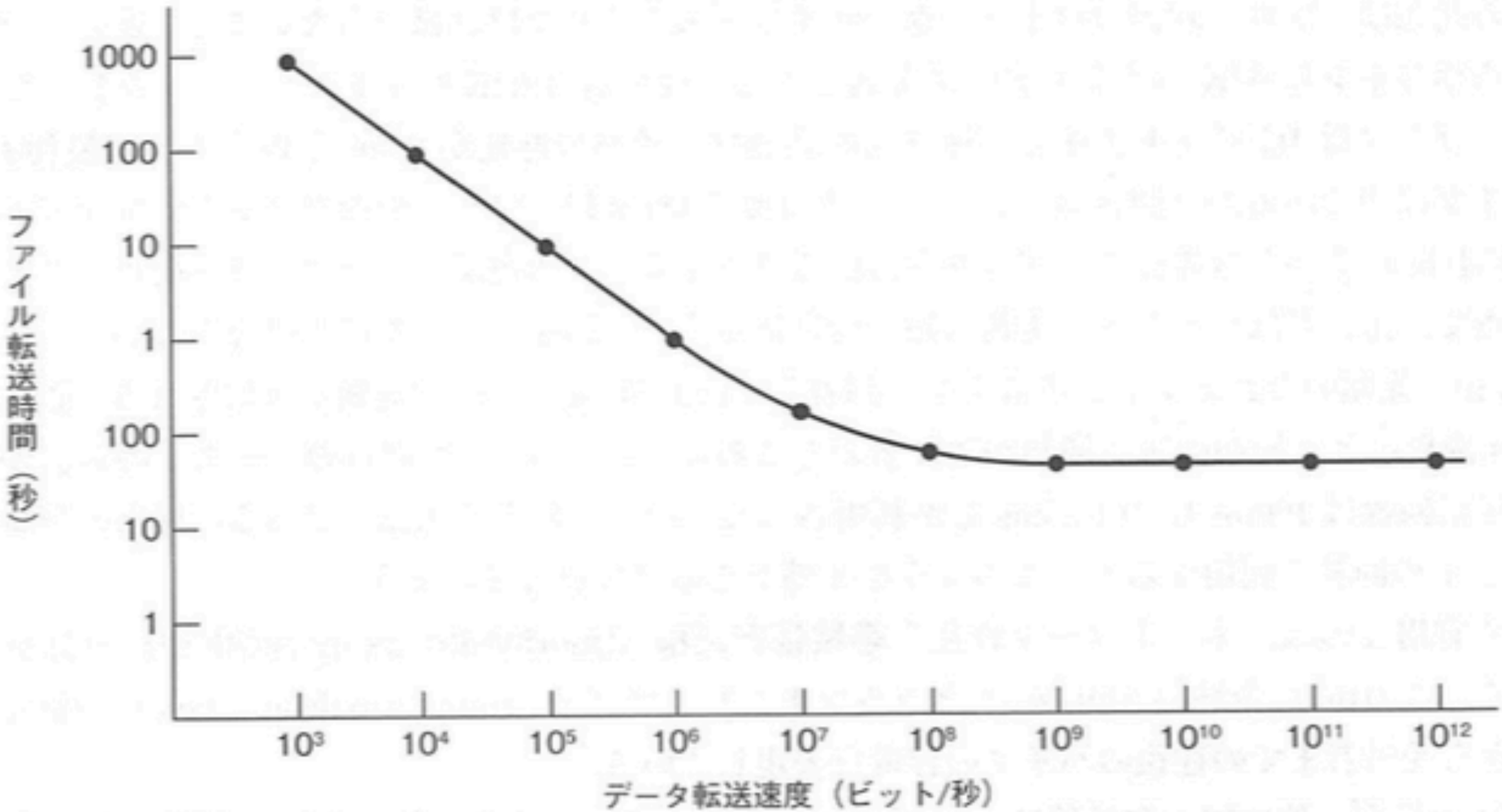
(帯域幅) x (往復遅延時間)
=40 Mbits

受信ウィンドウは少なくとも帯域幅遅延積よりも大きくなくてはならない



ロング・ファット・ネットワーク

ギガビットの線は帯域幅よりも遅延によって制限されている



遅延耐性ネットワーク (DTN: Delay-Tolerant Networking)

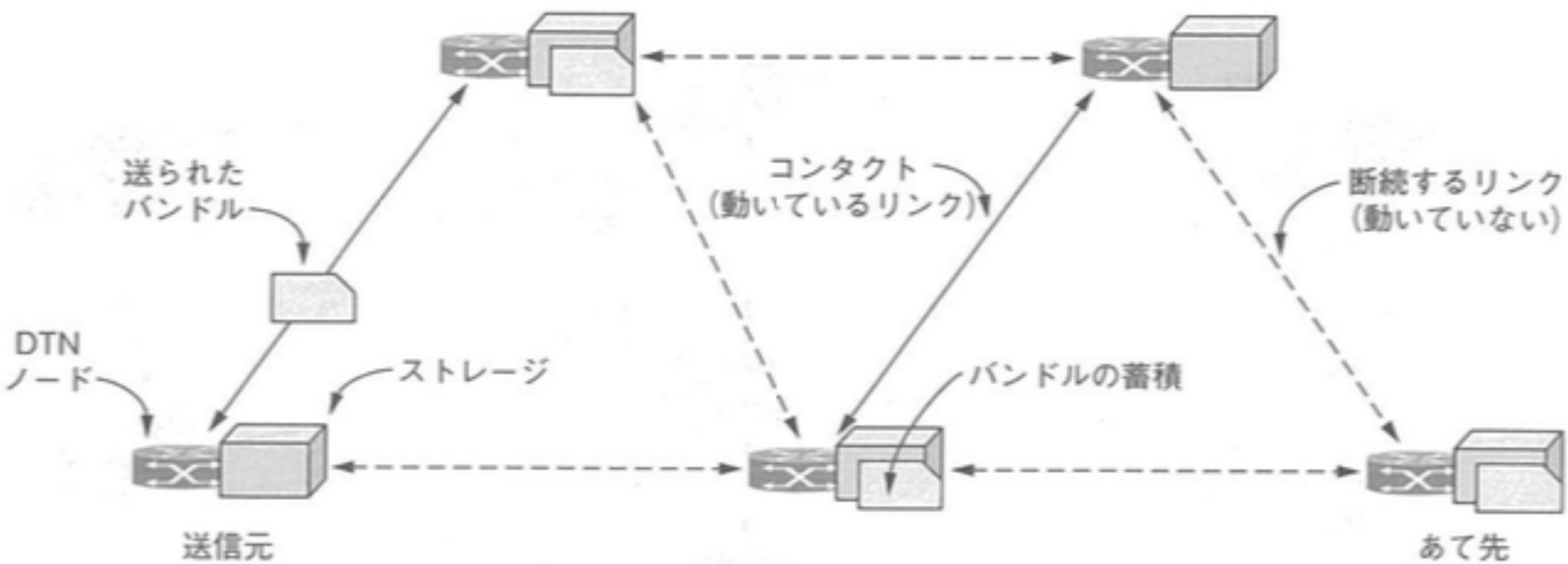
人工衛星、移動端末

メッセージ交換

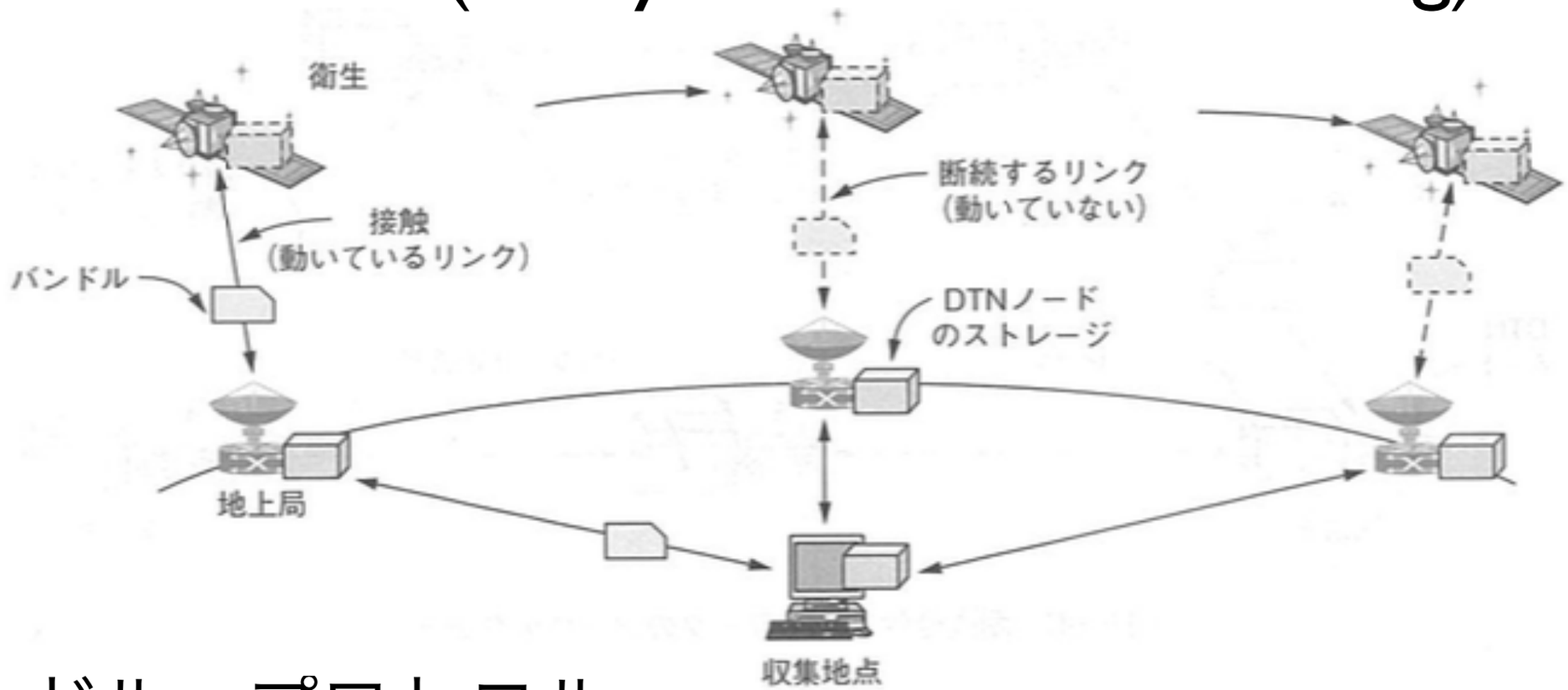
データは一旦ノードに保存され使用可能な経路が現れたら転送される

DTN アーキテクチャ

DTNノードにバンドルは何時間も保存できる
ノードは移動できる

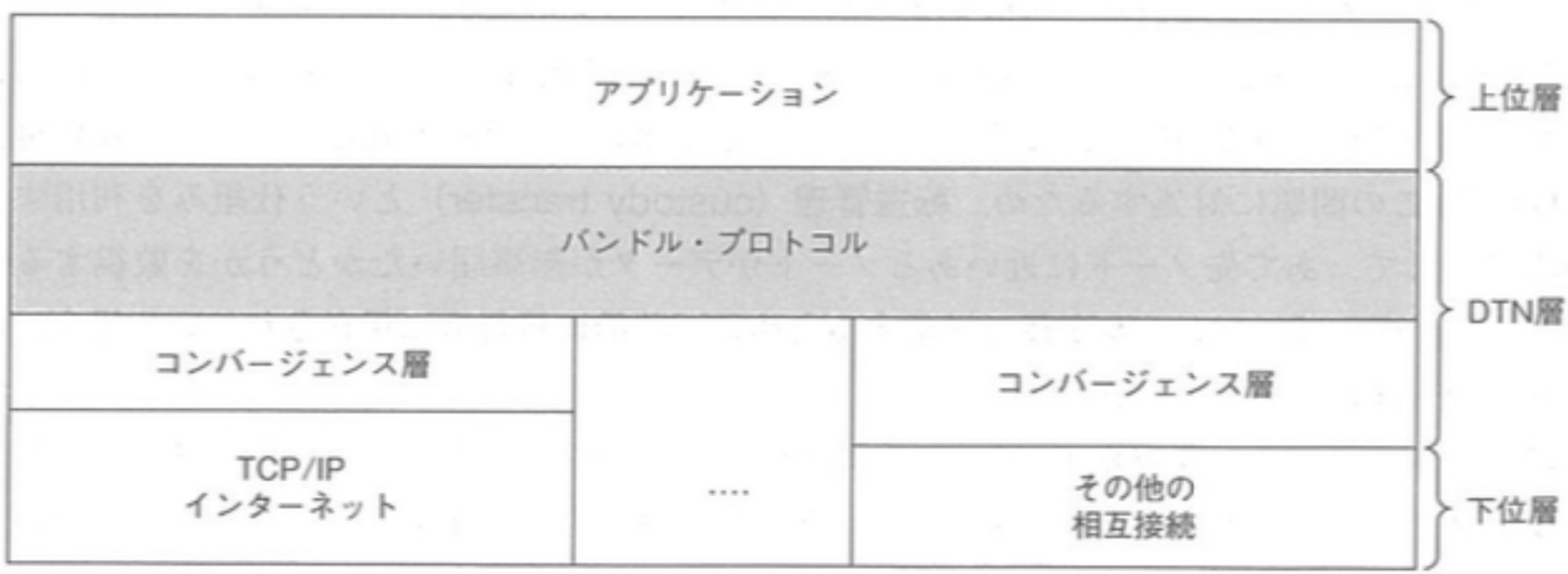


DTN (Delay-Tolerant Networking)



バンドル・プロトコル

TCP/IPの上の階層で動作する



講義日程

		授業計画		課題
05/12	第９回	ネットワーク層１ ルーティング・輻輳制御	５章	ルーティングの種類を理解し 輻輳制御手法を説明できる
05/16	第１０回	ネットワーク層２ インターネットとサービス品質	５章	インターネットの制御プロトコルを理解し ネットワーク間の接続について説明できる
05/19	第１１回	トランスポート層１ トランスポート・プロトコルの要素	６章	誤り制御とフロー制御を理解し 輻輳制御について説明できる
05/23	第１２回	トランスポート層２ UDP と TCP	６章	TCP の信頼性を理解し TCP のコネクション管理を説明できる
05/26	第１３回	アプリケーション層 DNS, 電子メール, www	７章	DNS, 電子メール, www のしくみを理解し ストリーミング，P2P について説明できる
05/30	第１４回	ネットワークセキュリティ１ 対称鍵暗号，公開鍵暗号	８章	暗号アルゴリズムを理解し SHA-1,2 と RSA について説明できる
06/02	第１５回	理解度確認総合演習 (期末試験) 第９回から第１４回までの内容の演習形式による確認		第９回から第１４回までの理解度確認と 到達度自己評価