2016.12.20

情報ネットワーク

Ibaraki Univ. Dept of Electrical & Electronic Eng.

Keiichi MIYAJIMA

TCPとUDP
2

TCPの役割

TCPは複雑

TCPの役割を一言で言うと、
「アプリケーションプログラムを作る人が楽ができる環境を提供する」

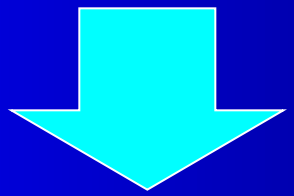
なぜか？

IPには様々な制約があった

- パケットの喪失
- 順番が入れ替わる
- データの破壊

他にも・・・

- ふくそう（輻輳）
- 回線速度が一定でない



これらのことを考えて、アプリケーションを作るのは大変

UDPではこれらのことを全て考えてプログラムを作る必要がある

TCPの役割

TCPは複雑

そこで・・・

アプリケーションを作る人が、ネットワークの細かい挙動について意識しなくても、高性能な通信ができるプログラムをたやすく作ることができるようにしたい

- 一定で無い回線速度でも自動で制御
- パケットの喪失したら再送して復元
- ふくそう（輻輳）が発生したら送信量を自動で減らす

このような機能をもったものがTCP

セグメント

TCPのパケットサイズ：セグメント

TCPが区切るメッセージの最大オクテット長：
最大セグメント長（MSS：Maximum Segment Size）

TCPでは、最大セグメント長を決定してから通信を開始する

最大セグメント長は、第5章（11月29日の講義）でおこなった、IP分割処理（IPフラグメント）が起きない最大の大きさに設定される。

極力、通信の無駄が発生しないように設定される

TCPの再送制御や、ふくそう（輻輳）制御では、最大セグメント長を考慮して処理が行われる

再送制御

IPでは、データが目的のコンピュータまで届いたかどうかを保証しない

届いたことを保証するのがTCPの最も重要な役割

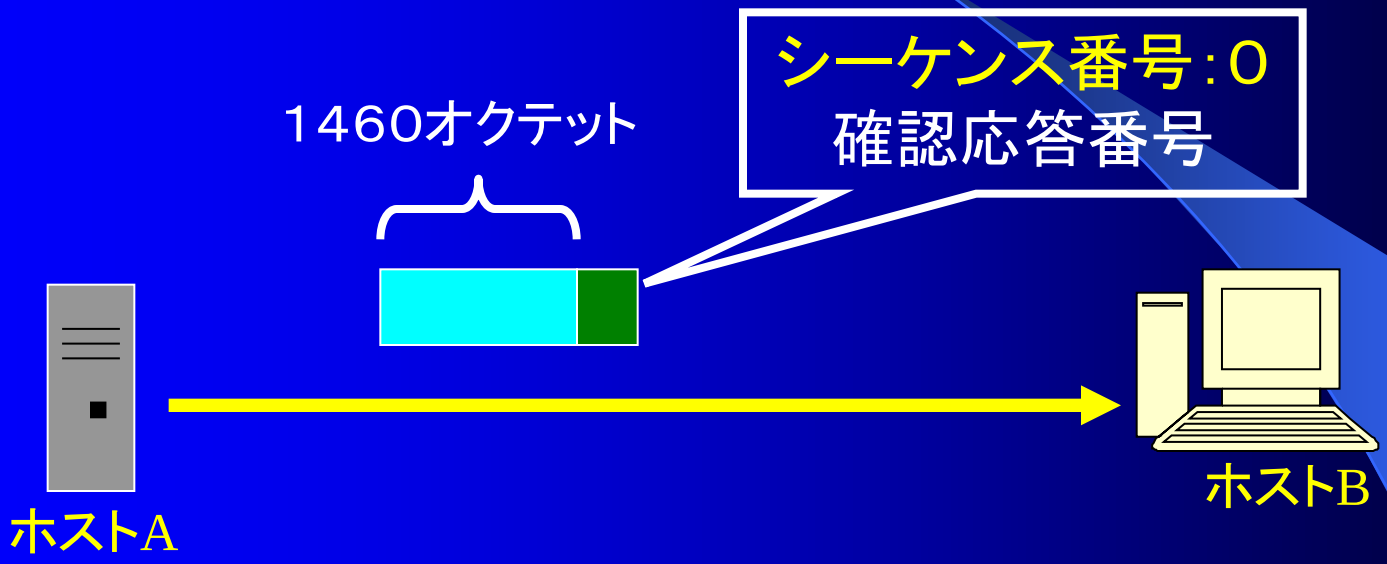
ではどうするか？

端末の両端で確認し合いながら通信すればよい

- シーケンス番号
- 確認応答（ACK）

再送制御

- シーケンス番号
- 確認応答（ACK）

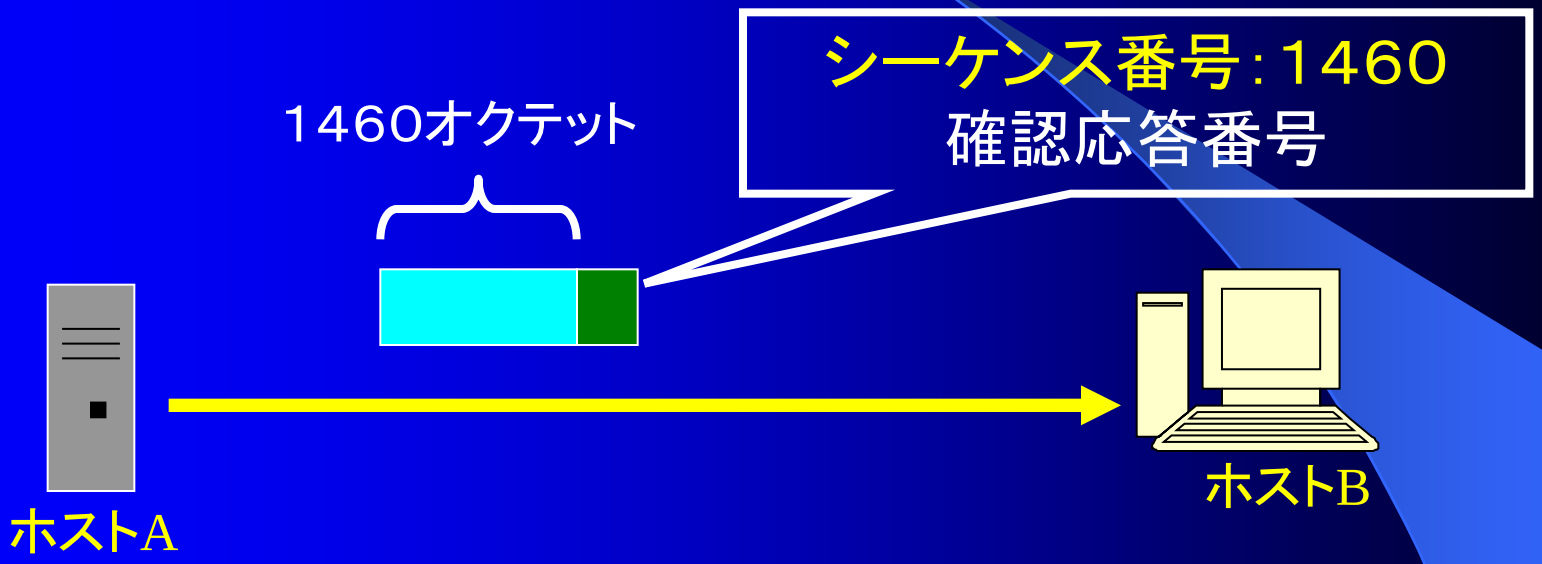


再送制御

- シーケンス番号
- 確認応答（ACK）

再送制御

- シーケンス番号
- 確認応答（ACK）

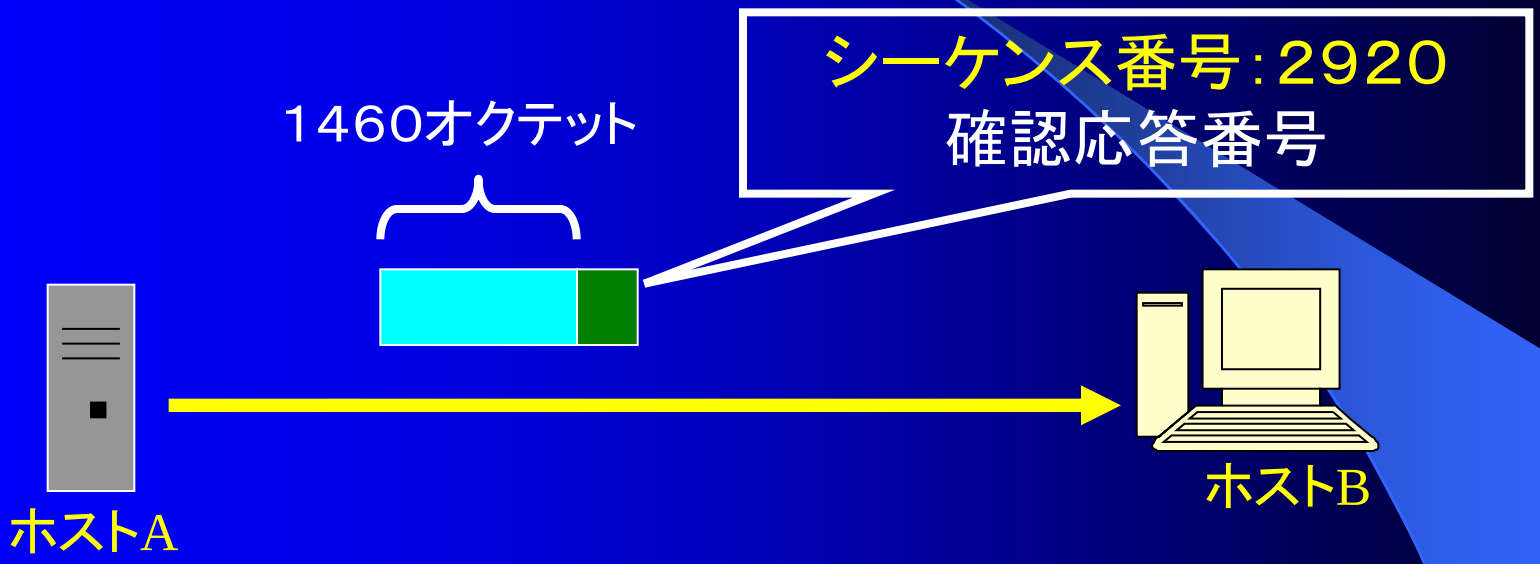


再送制御

- シーケンス番号
- 確認応答（ACK）

再送制御

- シーケンス番号
- 確認応答（ACK）

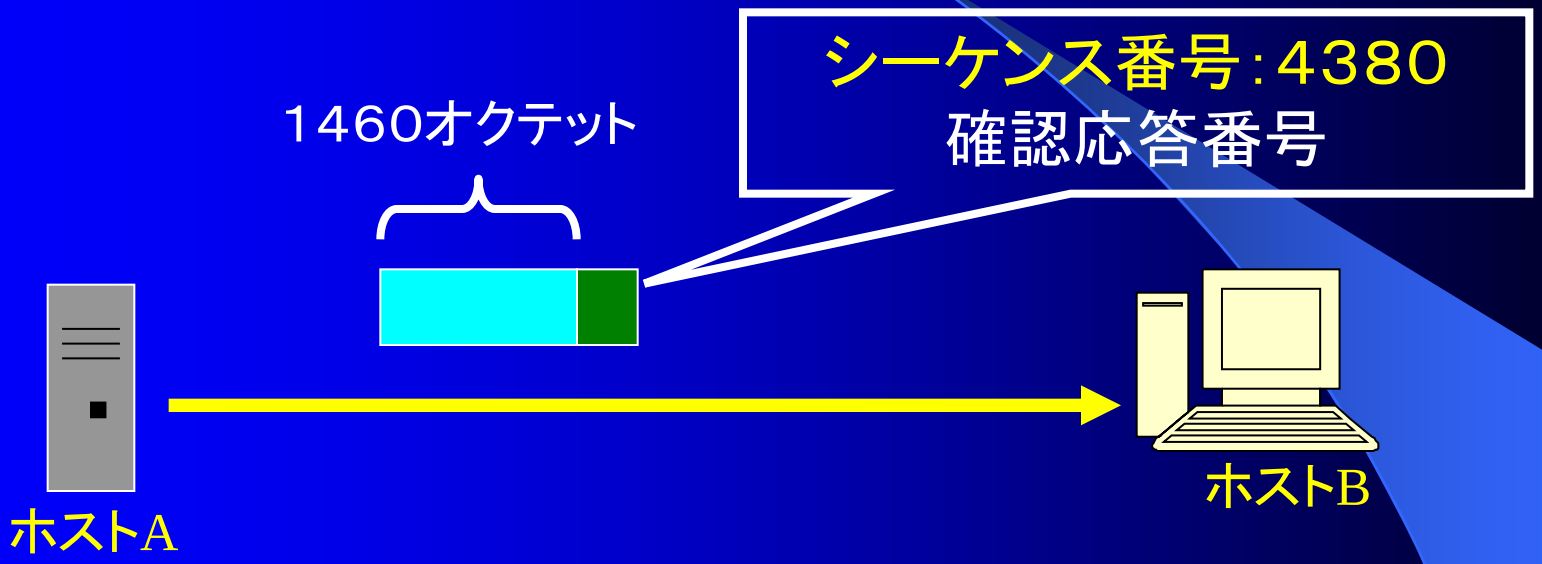


再送制御

- シーケンス番号
- 確認応答（ACK）

再送制御

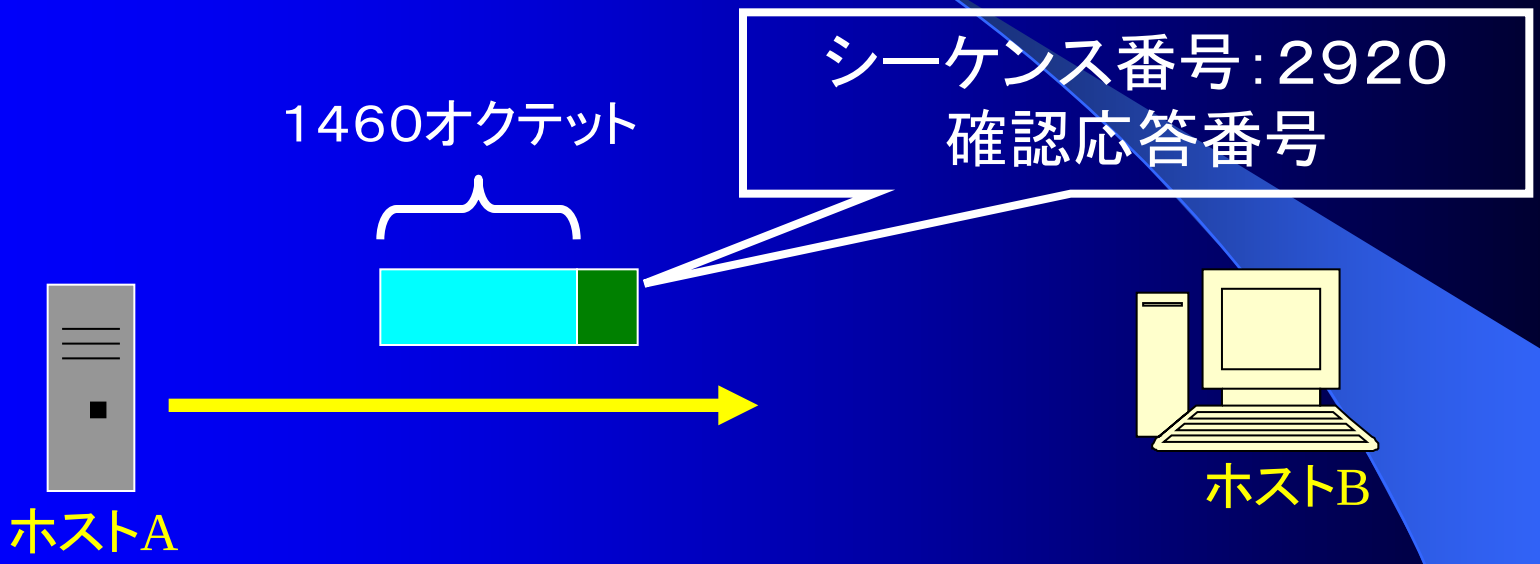
- シーケンス番号
- 確認応答（ACK）



この繰り返し

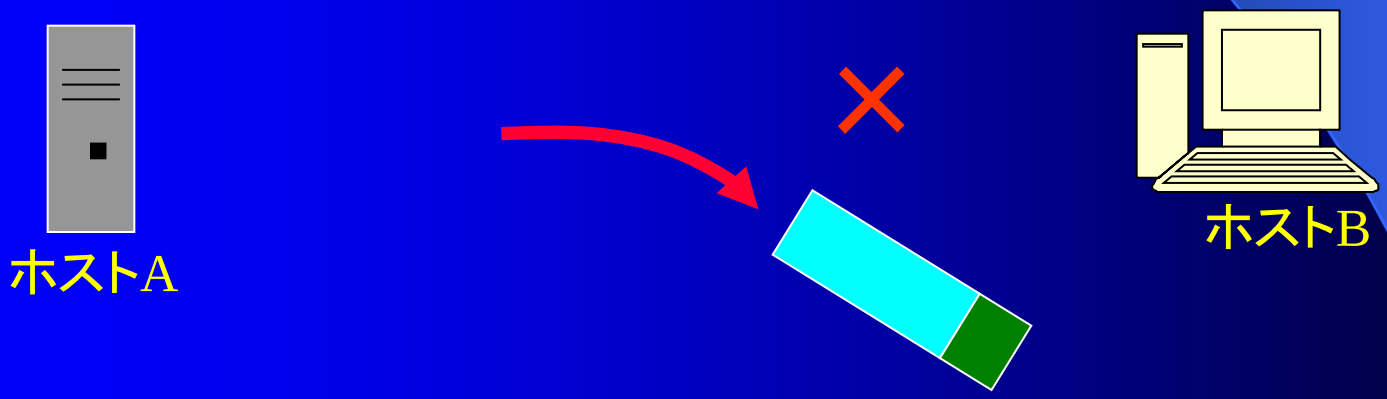
再送制御（パケット喪失時）

- シーケンス番号
- 確認応答（ACK）



再送制御（パケット喪失時）

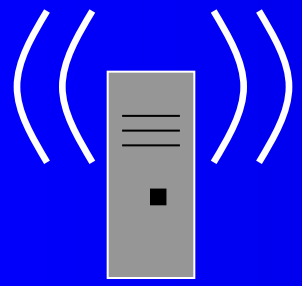
- シーケンス番号
- 確認応答（ACK）



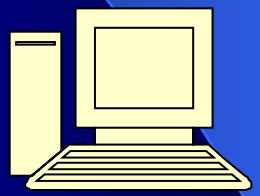
パケットが喪失し、ホストBに届かない

再送制御（パケット喪失時）

- シーケンス番号
- 確認応答（ACK）



ホストA

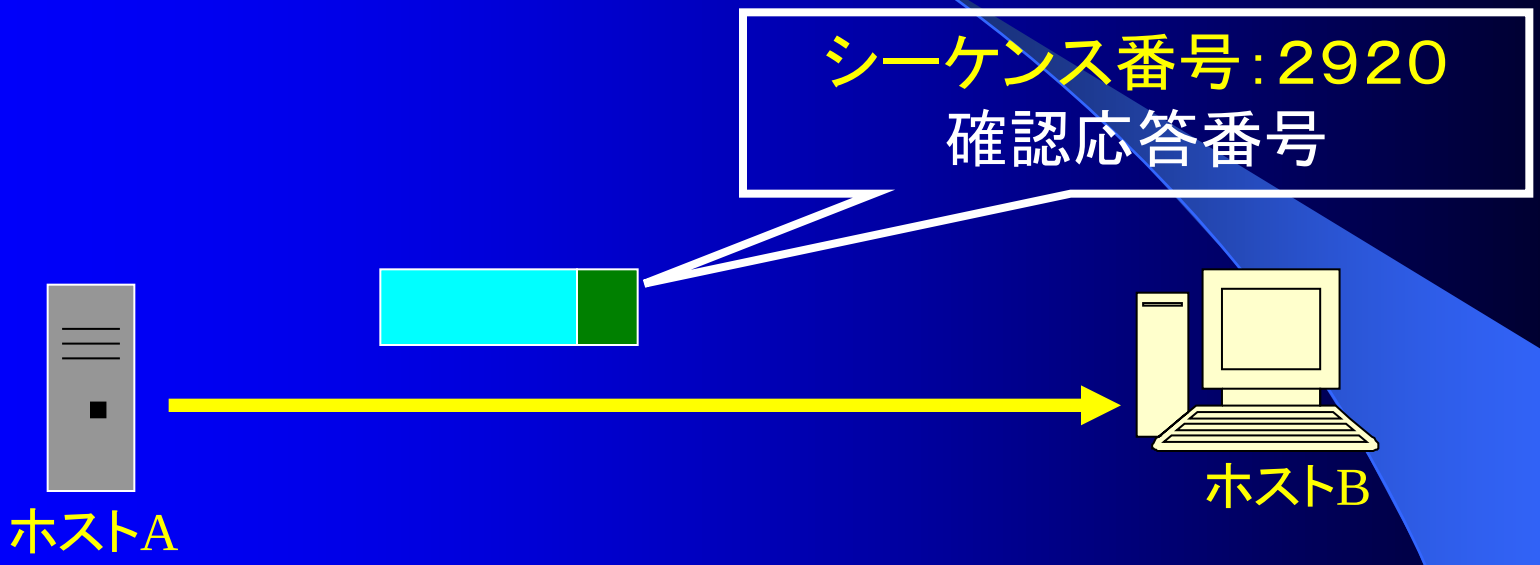


ホストB

一定時間待つ

再送制御（パケット喪失時）

- シーケンス番号
- 確認応答（ACK）



一定時間待っても、確認応答がこないため、再送を行う

再送制御（パケット喪失時）

- シーケンス番号
- 確認応答（ACK）

確認応答が返ってきたら、次を送信

TCPの内部変数（TCB）

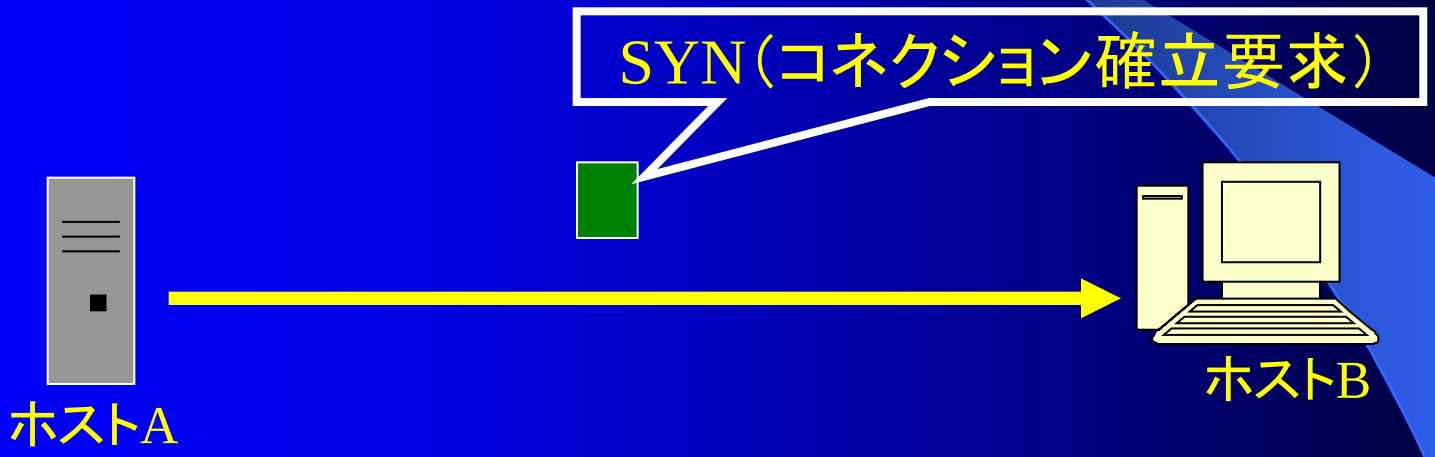
データの到達性を保証するために、TCPモジュール間でシーケンス番号と、確認応答番号の値を記憶しておく必要がある。

TCB（Transmission Control Block）:

snd.una	まだ確認応答されていないシーケンス番号
snd.nxt	次に送るシーケンス番号
snd.wnd	送信ウィンドウ
snd.cwnd	ふくそうウィンドウ
iss	シーケンス番号の初期値
mss	送信セグメントの最大長
rtt	セグメントを送ってから確認応答を受信するまでにかかった時間
rcv.nxt	次に受信するデータセグメントのシーケンス番号
rcv.wnd	受信ウィンドウ
irs	確認応答番号の初期値

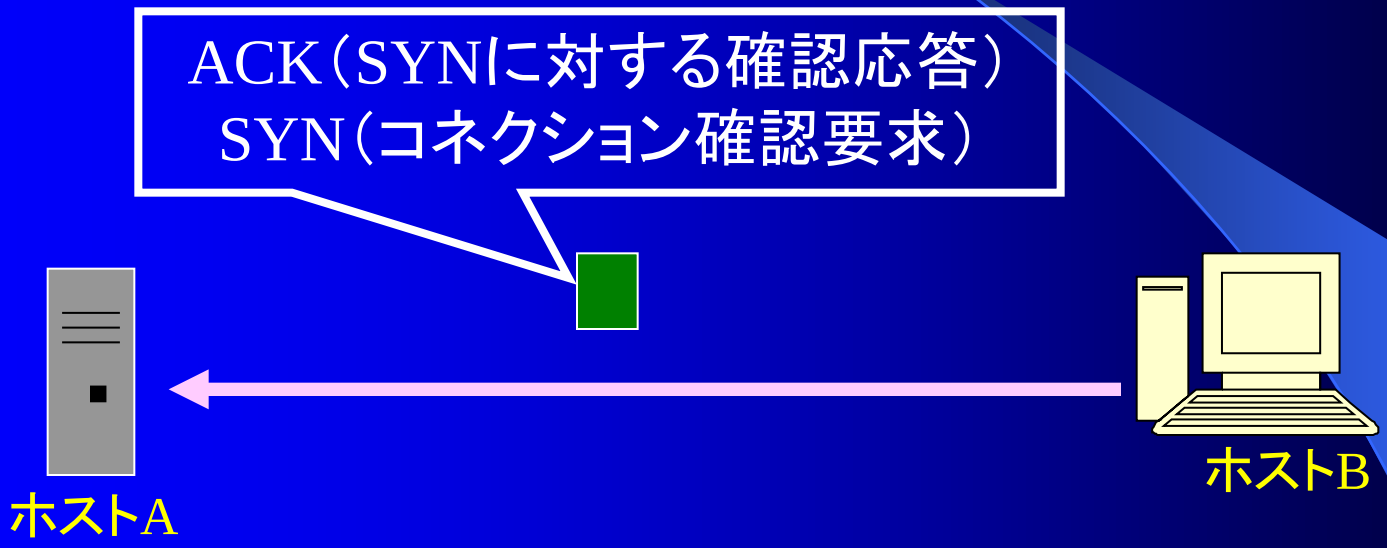
コネクションの管理

TCPでは、データセグメントの送信を開始する前に、通信相手との間にコネクションを確立する



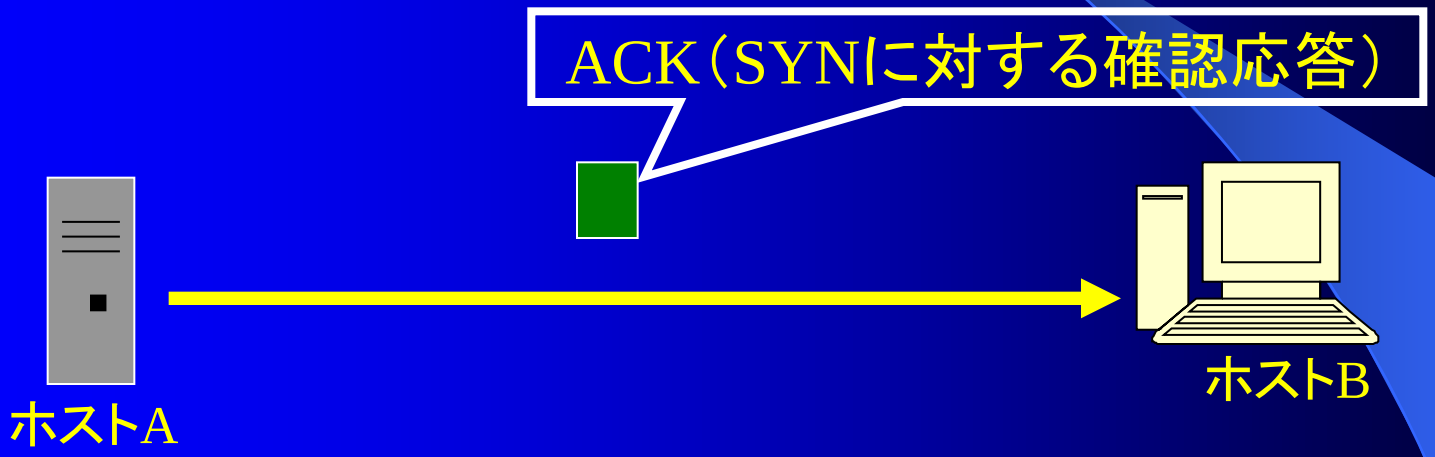
コネクションの管理

コネクションの確立



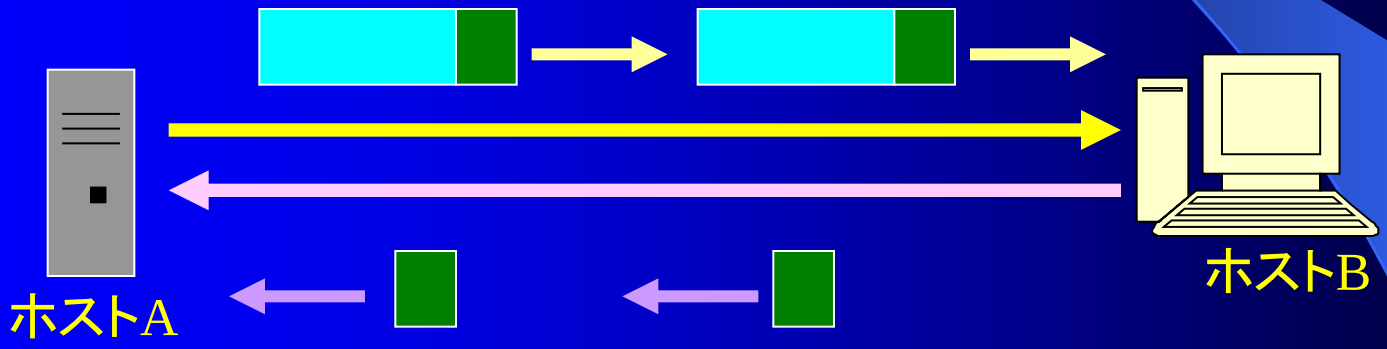
コネクションの管理

コネクションの確立



コネクションの管理

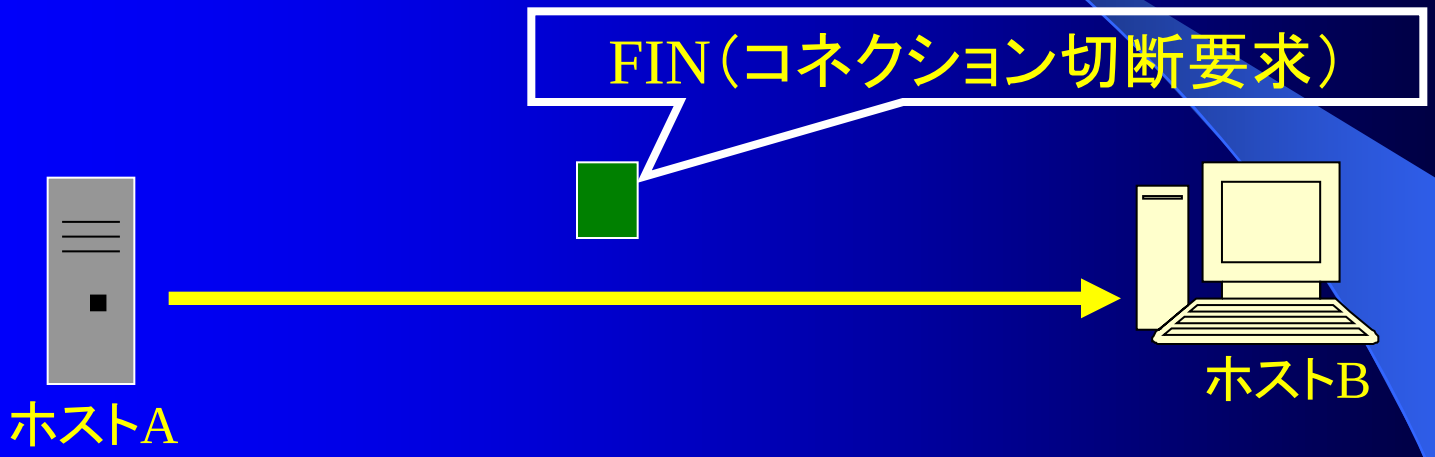
コネクションの確立



論理的なコネクションが確立されたら、データが転送される

コネクションの管理

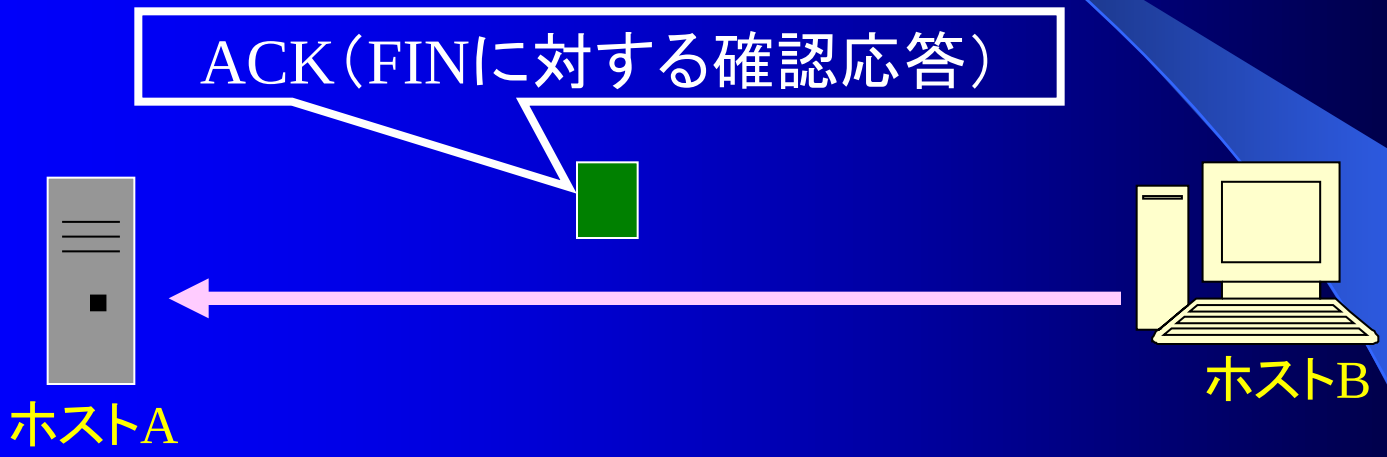
コネクションの切断



コネクションを切断するときは、これ以降送るデータがないことを意思表示し、切断する

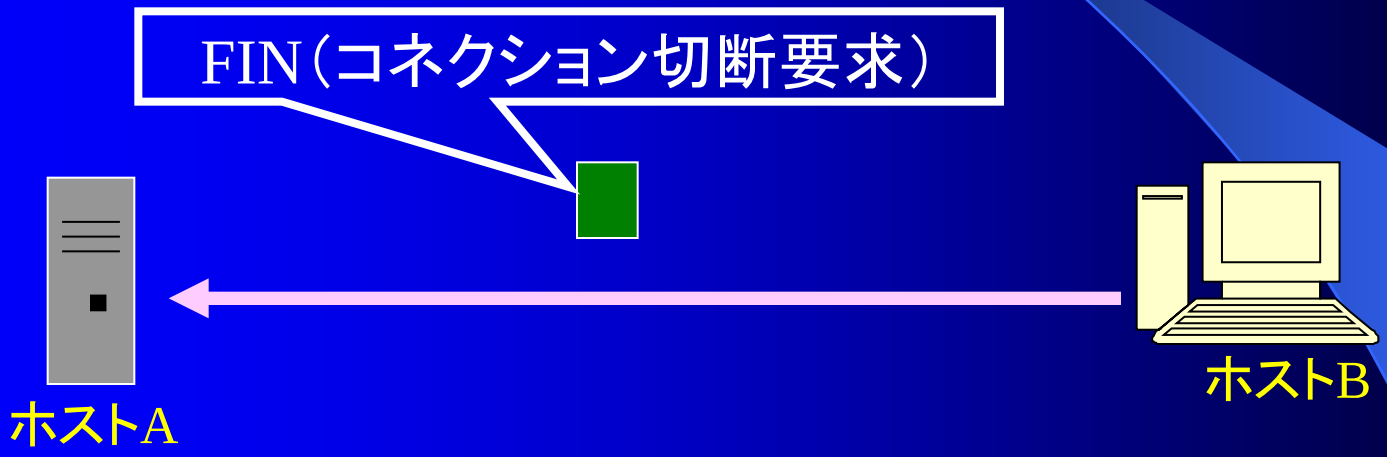
コネクションの管理

コネクションの切断



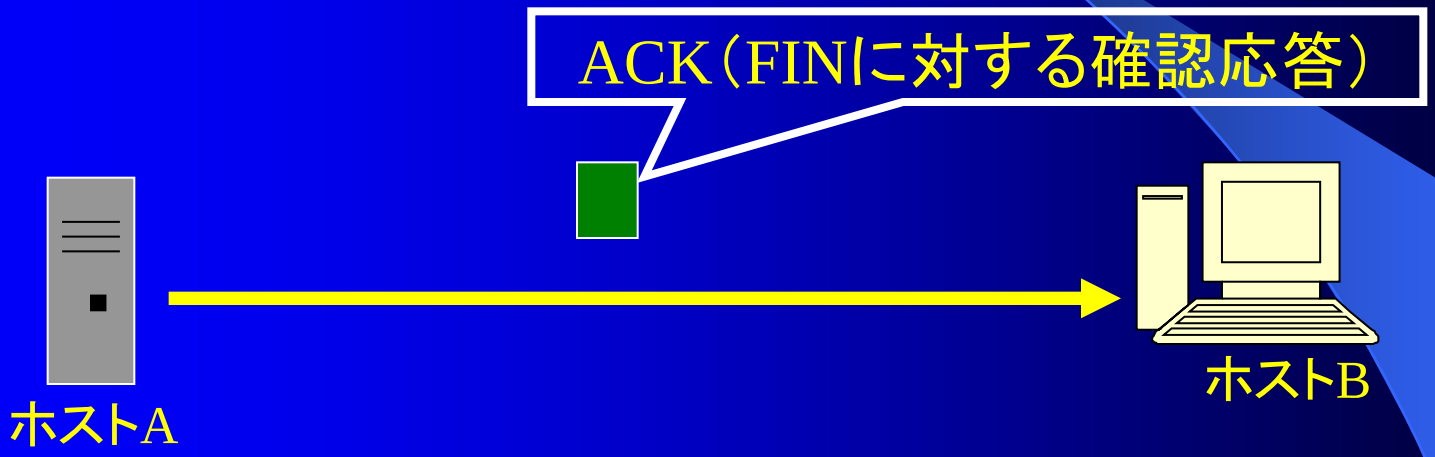
コネクションの管理

コネクションの切断



コネクションの管理

コネクションの切断

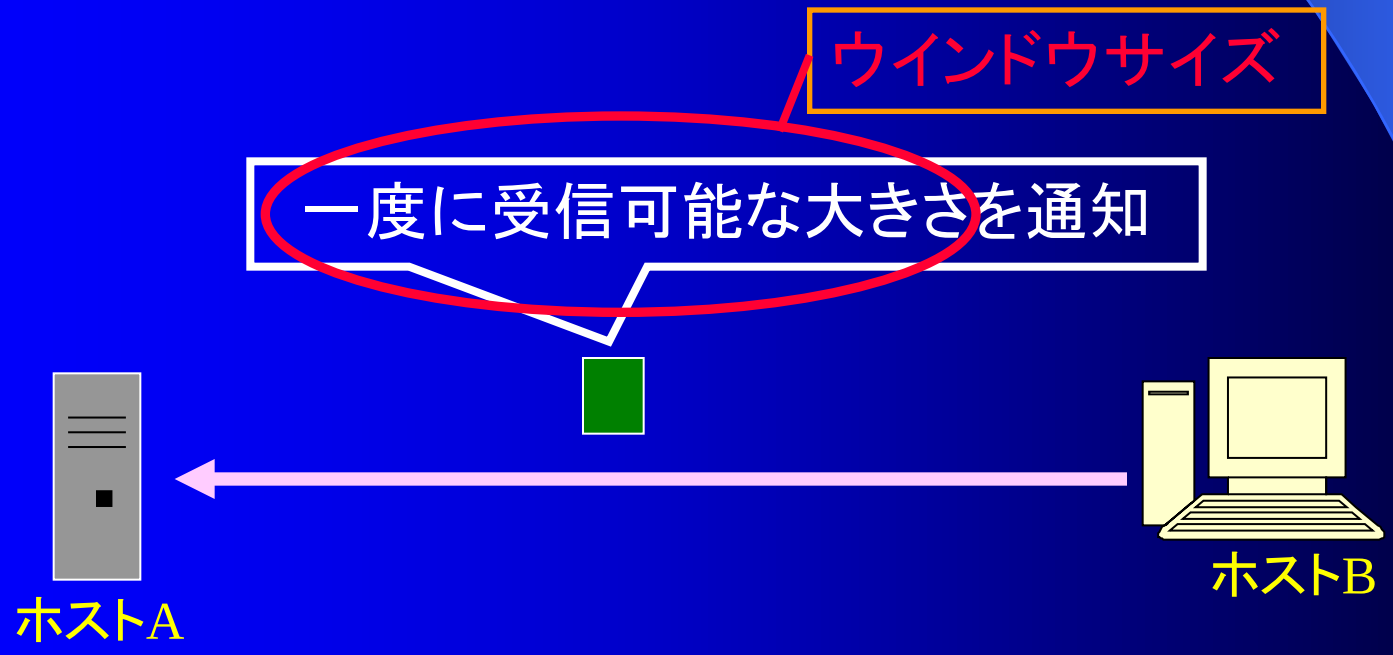


フローの制御（ウィンドウフロー制御）

1パケット（セグメント）毎に確認応答セグメントをやりとりしていたのでは、効率が悪い（大きなスループットを得られない）。

そこで、・・・

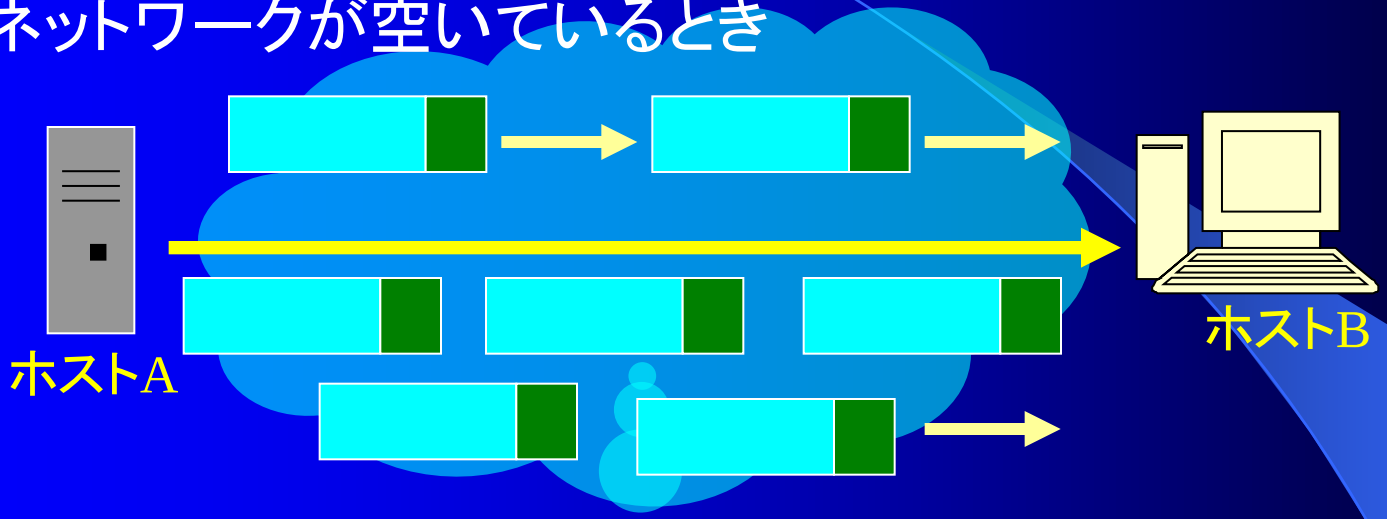
一度に複数のデータセグメント（パケット）を送ってしまえばよい。



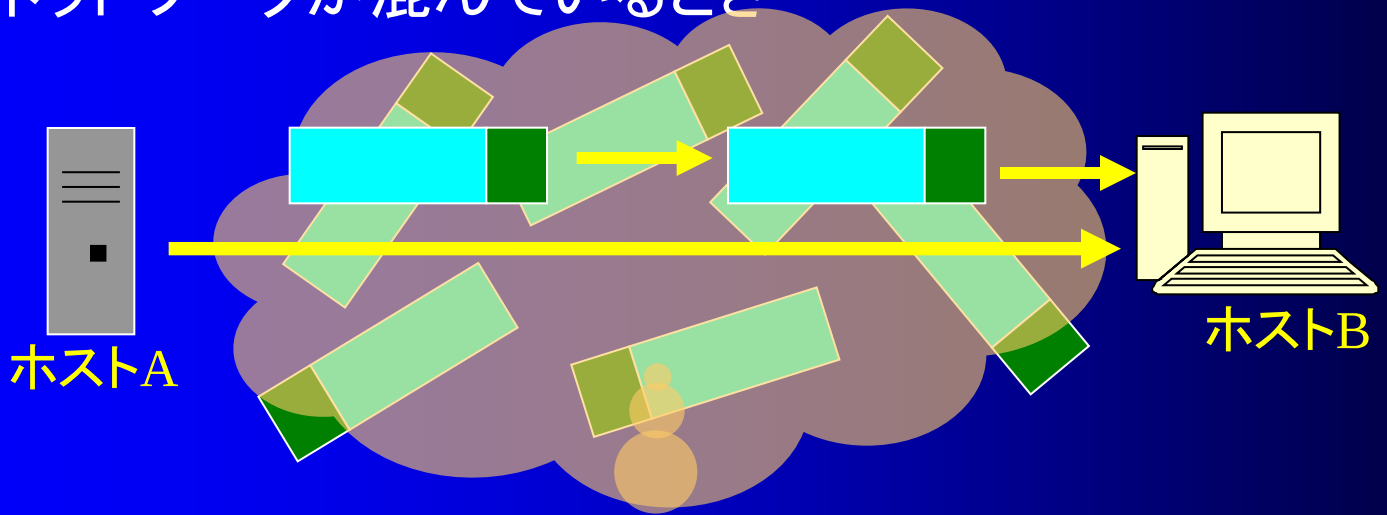
ふくそう（輻輳）制御

ネットワークの混雑度を判断して、送信データ量を調整する

- ネットワークが空いているとき

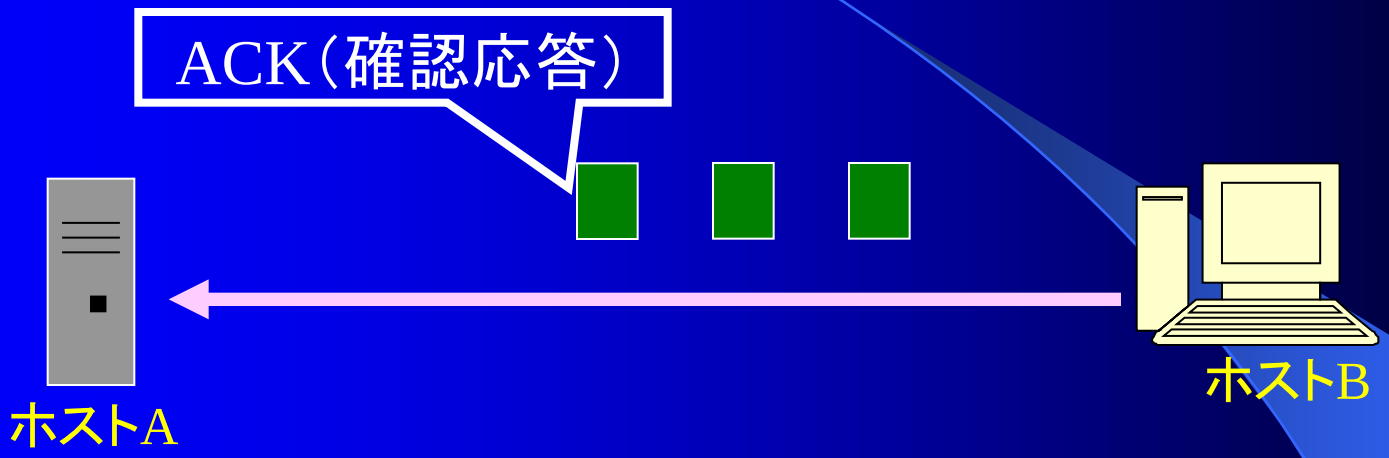


- ネットワークが混んでいるとき



ふくそう（輻輳）制御

混雑度をどうやって判断するか？



ACK（確認応答）がどのくらいきちんと返ってくるかで、
混雑度を判定

データの信頼性

ヘッダ中のチェックサムで信頼性を保つ

本日のまとめ

TCPとUDP 2

- TCP

  TCPの役割、セグメント、再送制御、
  TCPの内部変数（TCB）、
  コネクションの管理、フロー制御、
  ふくそう（輻輳）制御、信頼性

本日の課題

1．TCPの主な役割について、説明しなさい。（基本　改）

2．TCPのヘッダ部分（疑似ヘッダではない）に含まれている情報を全て記せ（基本 類）