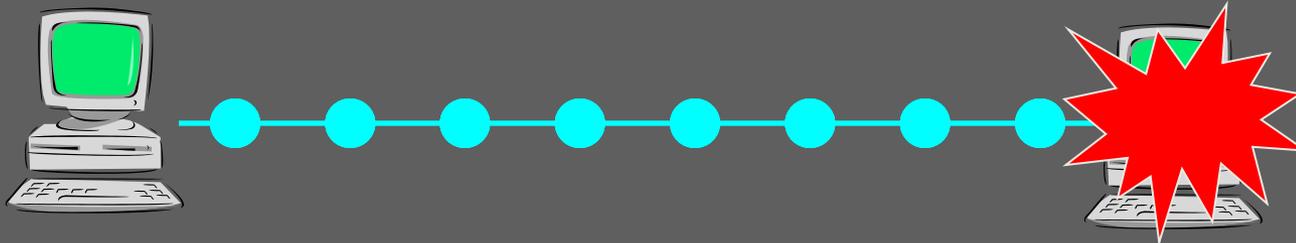


TCPのフロー制御

龍 浩志

TCPでのデータの送受信

- ⇒ 一度に受信できるデータの量は限られている
- ⇒ 受信したデータはすぐに削除できるわけではない
 - データは受信バッファに蓄積されていく



一方がデータを送り続けると..

フロー制御

- ⇒ 受け取り可能なデータ量を相手に伝える
 - 受信したデータの応答で、受け入れ準備ができていないオクテット数を示す
- ⇒ TCPヘッダのフォーマット



- シーケンス番号: 送信データの位置情報
- 応答確認番号(Ack): 次に受信したいデータの位置
- ウィンドウ: 受信可能なデータのサイズ

ホストA→Bへのデータ送信(1)

ホストA,Bの初期受信バッファのサイズを400とする

シーケンス番号 101
応答確認番号 なし
データ 100
ウィンドウ400

バッファ

100
200
300
400



ホスト A

バッファ

100
200
300
400



ホスト B



ホストA→Bへのデータ送信(2)

シーケンス番号 なし
応答確認番号 201
データ 0
ウィンドウ300



ホストA→Bへのデータ送信(3)

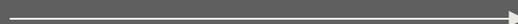
シーケンス番号 201
応答確認番号 なし
データ100
ウィンドウ400

バッファ

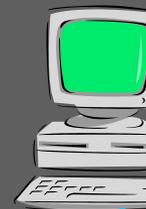
100
200
300
400



ホスト A



ホスト B



バッファ

100
200
300
400

ホストA→Bへのデータ送信(4)

シーケンス番号 301
応答確認番号 なし
データ100
ウィンドウ400

バッファ

100
200
300
400



ホスト A



ホスト B

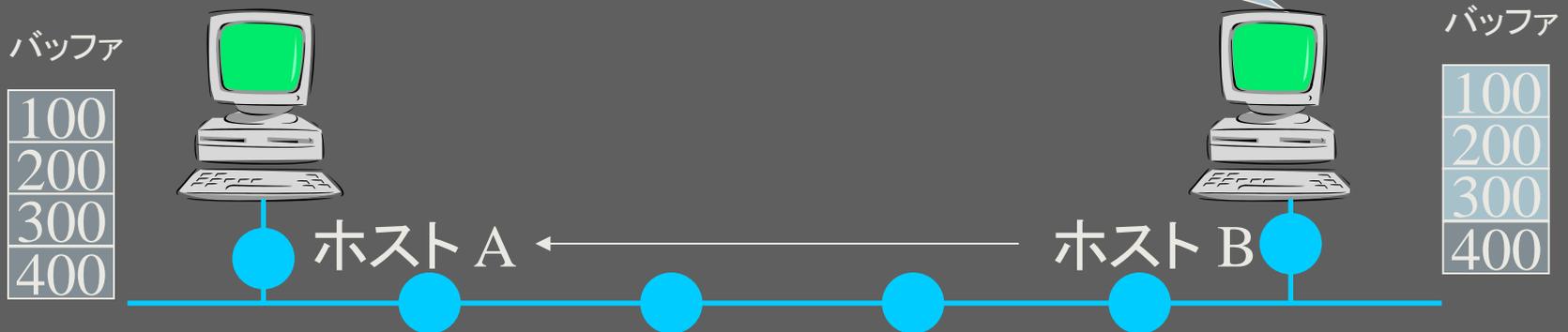


バッファ

100
200
300
400

ホストA→Bへのデータ送信(5)

シーケンス番号 なし
応答確認番号 401
データ 0
ウィンドウ100



受け取りバッファが0になると

- ⇒ 送信側はパケットの送信を停止
- ⇒ 受信側はパケットの処理が進み、バッファに空きができたならウィンドウを通知
- ⇒ 小さいデータを繰り返し送ったのでは効率が悪い
 - 「無能ウィンドウ症候群」

- ⇒ 受け取りバッファが適切なサイズになるまで受信可能であることを知らせない