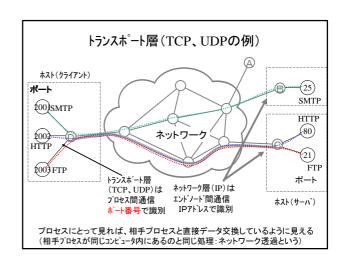
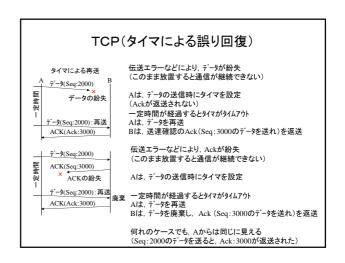
第10回 トランスポート層

エンドプロセス間の通信 (プロセスの識別,多重化,TCPとUDP) 授業で使用した教科書の図表 表4.3,表13.1、図13.1~13.8



TCP(送達確認) 以下、データ長は全て1000オクテットとして説明 送信データ $\begin{array}{ccc} & & & & & & & \\ \hline & & & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & &$ A シーケンス番号 B 送信シーケンス番号 送信データの先頭オクテット番号 1000オクテット → データ(Seq:2000) ± −7(Seq:2000) ACK(Ack:3000) 受信データの最終オクテット番号+1 (どこまで受信できたかを示す) -7 −**9**(Seq:3000) -ACK (Ack:3000) SYNおよびFINは 1オクテットのデータとして扱う 7 −9(Seq:11000,Ack:4000) ピキーハック 1000オクテット ACK番号 次は3000オクテット目から送れ → ¬¬*-¬¬(Seq:3000) (2999オクテット目まで届いた) 3998 -ACK (Ack:4000) 考え方はHDLCと同じ TCP:シーケンス番号はオクテット番号を表す HDLC:シーケンス番号はフレーム番号を表す 4001 4002



TCP(重複Ackによる誤り回復) 重複ACKによる再送 Aは、Ack:3000を受信後、4個のデータを連続して送信 (Seq:3000、Seq:4000、Seq:5000、Seq:6000の4個) そのうち、Seq:3000のデータが伝送誤りによって、廃棄 Bは、Seq:3000を受信していないのに、 Seq:4000、5000、6000を受信したので、それぞれに対し τ̄*-\$(Seq:2000) ACK(Ack:3000) 7°-9(Seq:3000) 7-9(Seq:500) 7-9(Seq:500) 7-2(Seq:660) Ack:3000 (Seq:3000を送れ)を返送。 同じ番号のAckを連続して受信することを<mark>重複Ackと呼ぶ</mark>. 重複Ackを連続して3回受信すると、そのデータを再送する。 このように、大量のデータを送信する場合は、タイマを使う ACK (Ack: 3000) よりも短時間で誤り回復ができる Bは、再送されたSeq:3000のデータを受信した後、Ack:7000を返送(Seq:6999までのデータが累積して届いた). データ(3000):再送 このようなAckの返送方法を累積Ackと呼ぶ ACK(7000) 累積ACK

プロトコル	TCP	UDP
通信方式	コネクション型 (通信開始時にコネクション を設定,終了時に解放)	コネクションレス型 (コネクションの設定・解放 をせず, データ送信のみを 行う)
ヘッダ長	20オクテット	8オクテット
誤り回復	再送による回復	なし
フロー制御	ウィンドウ制御	なし
用途	品質保証が必要な通信で 使用	効率を重視した通信で使用
アプリケーションプロトコ ルの例	HTTP, SMTP, POPなど	DNS, RTP, TFTP

TCPとUDPの比較

重要

p.115