第 10 回の課題について

（1）通信プロトコルに関する記述のうち、適切なものはどれか

ア　アナログ通信で用いられる通信プロトコルはない
イ　国際機関が制定したものだけであり、メーカが独自に定めたものは通信プロトコルとは呼ばない
ウ　通信プロトコルは正常時の動作手順だけが定義されている
エ　メーカや OS が異なる機器同士でも、同じ通信プロトコルを使えば互いに通信することができる

通信プロトコルは、 物理的なものから、メッセージの内容、通信手順、誤りが生じたときの制御方法などの規定を階層で定めているものですので、これに従って処理することでメーカが違った製品であっても、また OS が違っていても通信が行えます。したがって「エ」が適切であると考えられます。

「ア」については、通信という相手がある処理を考えると、デジタルであってもアナログであっても双方で取り決めが必要であり、プロトコルが必要になります。
なお「イ」については、メーカが独自に定めたプロトコルもあり、それが国際規格になることもあります。（有線LAN（イーサネット）では、Xerox・Intel・DEC が開発した仕様から国際規格 IEEE 802.3 ができています）
「ウ」については、たとえばトランスポート層で誤り制御（正常でない場合の処理）が行われ、その手順が定義されています。

『エ』のように記号だけを記載した解答もありましたが、適切でない選択肢に対して、その理由をつけていただくとよいのではと思います。

（2）ネットワークでメッセージをパケットに分割して送ることの必要性やメリットを述べよ

ネットワーク回線や交換器などの資源を占有することなく、共用し多重化することで利用効率を高めることができます。

データが大きいと伝送路（ネットワーク回線）がそのデータで多く使われてしまい他のデータが送れなくなりますが、それを回避するため大きなデータを流さないよう制限すると利便性がなくなります。

そこで、データを分割し、伝送路を占有せずに送受信を行える仕組みにしたものがパケット交換です。パケットは、ルータが利用できる伝送路を選ぶので、混み合ったルートを避けるなど状況に応じたルート選択が可能で、回線の使用効率や信頼性、性能を高めることが可能になります。また、通信速度の異なるネットワーク間であってもパケットの送受信が可能となります。

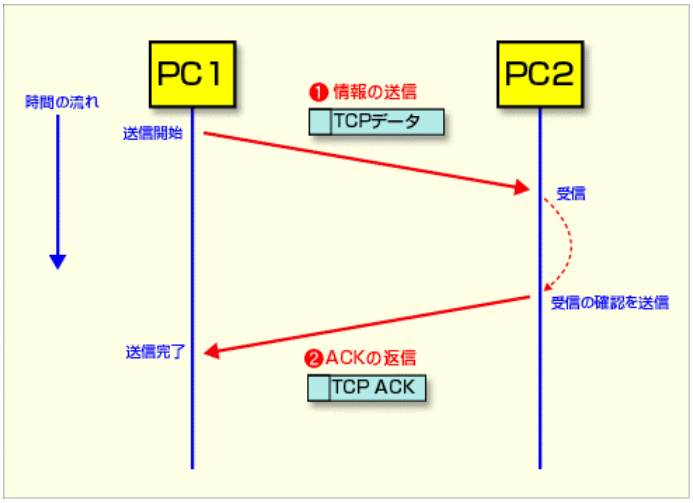
優先順位をつけて処理できることもあります。 たとえば、 IP 電話のパケットの優先度を上げて他のパケットより先に処理することで、 品質を安定できるといったメリットがあります。

なお、パケット消失の可能性がある・パケットの到着タイミングの均一性は保証されない・パケットの到着順序が送出順序と同じであることも一般には保証されない、(すなわち回線交換と比較して QoS の維持が難しい) ということがありますが、逆に通信中にエラーが起きることを前提に、分割されたデータが送信途中で消失しても、全体を再度送り直す必要はなく、そのパケットだけを送り直すだけで済むことなどから、エラーに効率よく対処できます。

以下、これに関連して参考までに、TCP での通信の手順を紹介しておきます。(『@IT 基礎から学ぶWindowsネットワーク 信頼性のある通信を実現するTCPプロトコル』からの抜粋です)

- **TCP における基本的な通信形態**

  TCP では、送信したパケットに対して、必ず確認のための応答を送信することによって信頼性のある通信を実現している。



❶送信するデータを TCP パケットに載せ (送信したデータに TCP ヘッダを付加する)、それを IP パケットに載せて PC2 あてに送信する。

❷TCP パケットを受け取ると、データを受け取ったことを示すために ACK パケット (正確には、ACK フラグがオンになった TCP パケット) を送信する。ACK を受信して始めて送信が正常に終了したことになる。

TCP パケットを受信した側では、そのパケットからデータを取り出して上位アプリケーションに届けるだけでなく、送信側に対して自動的に確認応答を返信する。確認応答の返信は、上位アプリケーションの指示とは関係なく、TCP のプロトコルスタック内で自動的に行われる。このような仕組みになっているため、上位アプリケーションは、TCP プロトコルにおけるパケットのやりとりをまったく意識することなく、信頼性のあるストリーム通信機能を利用することができる。

確認応答のことを TCP では「Acknowledge」と呼ぶ (以下では ACK と略記)。TCP のパケットを受信した側では、データを受け取ったことを表すため、ACK 応答を返送する (図中の ❷)。

送信側では、この ACK を受け取って初めて送信が完了したとみなし、次の動作に移る。

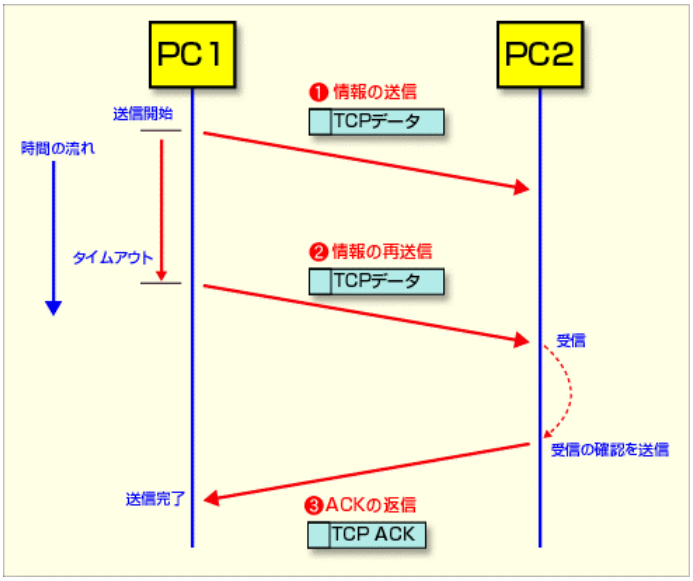
しばらく待っても ACK が受信できなければ送信が失敗したものとみなし、次のように

再送動作を行う。

- **TCP パケットの再送信**

再送信する TCP パケットの内容は、(通常は) 最初に送信した TCP パケットの内容とまったく同じである (再送であることを示す特別な情報は含まれていない)。ACK を受信するまで何度か TCP パケットを再送するが、このような再送動作に対しても上位アプリケーションは何ら指示・関与する必要はない。 すべて TCP/IP のプロトコルスタック内で自動的に処理される。

もし、あらかじめ規定された回数 (もしくは時間) が経っても ACK を受信できない場合は、TCP コネクションが切れてしまったと判断し (通信相手がダウンしてしまったような場合も含む)、上位アプリケーションに対してエラーなどを通知することになる。



❶ 最初の TCP パケットの送信。

❷ 一定時間待っても ACK パケットが戻ってこない場合、先ほどと同じ TCP パケットを再送信する。

❸ ACK パケットを受け取れば送信完了。これが実際に ❶ のパケットに対する応答なのか、それとも ❷ のパケットに対する応答なのかは関係ない (どちらのパケットも同じなので、どちらに応答してもよい)。

そこで、OS の中では受け渡しを行う最大のサイズでバッファを確保しておく必要があります。以下は、「オペレーティングシステム入門」(並木 美太郎 著) 第 10 章 プロトコルスタックでの記述です。

AP 層のデータは，送信時に装置の最大パケットサイズに分割されて送信され，受信時のパケットの順序性も IP 層では保証されていない。これらのパケットの分割と再構成もプロトコルスタックの TCP 処理では重要となる．特に，一つの接続に対して，最大ウィンドウサイズ分のバッファメモリを用意しておく必要がある．送信側では，正しくデータ受信されたことを示す ACK が返るまで，送信データをバッファに保持する必要がある．また，バッファで保持しているデータがウィンドウサイズを越える場合は，それ以上送信しないように，一時停止を行う．

受信側では，途中でパケットが到達しなかったときのために，順序の前後のパケットを保持しておき，順序として抜けているデータが到着するまで待つように，バッファにデータを保持しておく必要がある．(図 10.2)

したがって，送受信側それぞれで最大ウィンドウサイズ分のバッファ用のメモリを用意しておく必要がある．

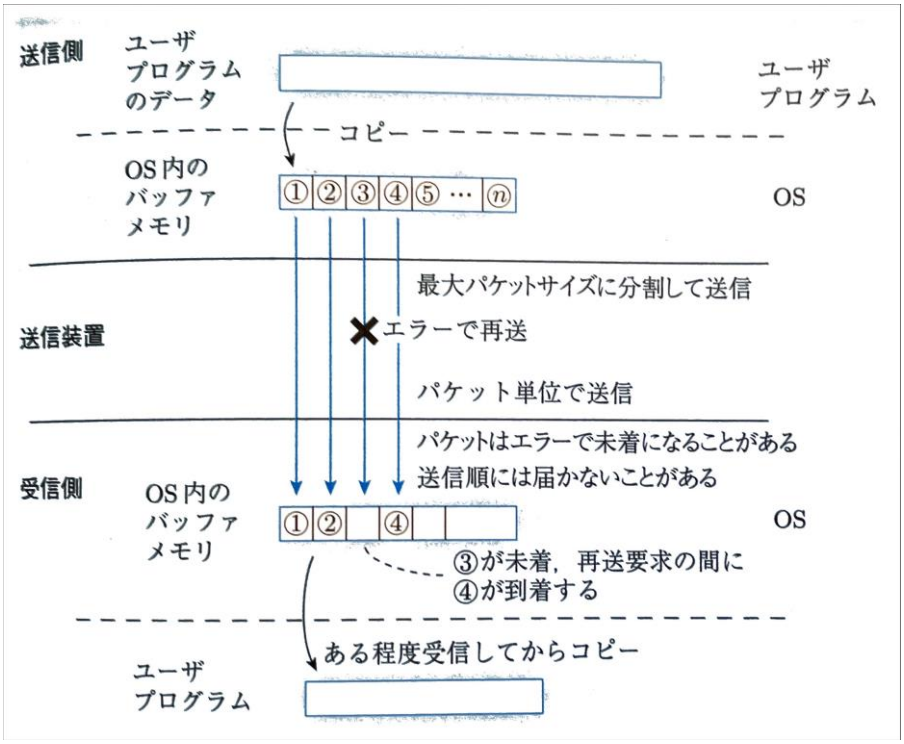
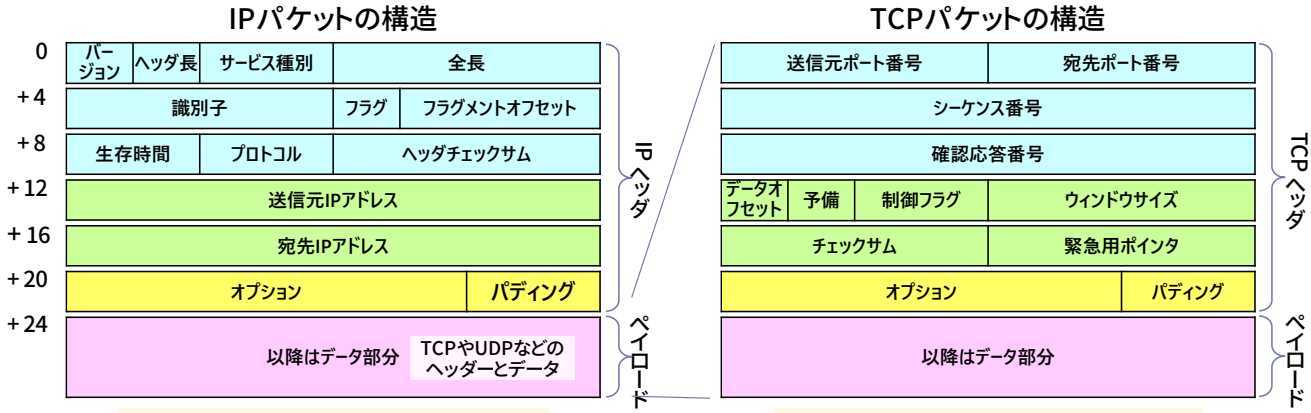


図 10.2　送受信バッファ

なお、前回の講義スライドで紹介したIPパケットやTCPパケットの構造を以下に示します。



IPパケット(IPデータグラム)の構造は上図のようになっており(IPv4の場合)、先頭から24バイトがヘッダで、それ以降がデータ部分(ペイロードと呼ばれます)になります。ここに、先のカプセル化で見たようにTCPなどのデータが、やはりヘッダとペイロードとして入れ子になって配置されて構成されます。

TCPパケット(TCPセグメント)の構造は図のようになっており、先頭から24バイトがヘッダで、それ以降がデータ部分になります。ヘッダに、ポート番号やシーケンス番号が配置されます。