

# X.25 パケットレベルプロトコルの 2U-7 Ada<sup>\*</sup> による記述

中山 仁, 荒木 啓二郎, 牛島 和夫  
(九州大学 工学部 情報工学科)

## 1. はじめに

プログラミング言語Ada<sup>5)</sup>は、情報隠蔽、データ抽象化を実現するためのパッケージ、並列動作を陽に記述するタスクなど、豊富な機能を備えた汎用言語である。我々は、Adaを用いて通信システムの開発を行うことにより、こうしたAdaの諸機能が階層的な通信システムの開発に有用であることを確認することができた。<sup>1)</sup>

以上の経験を踏まえて、今回新たな事例として、X.25パケットレベルプロトコルを適用したシステムの開発を試み、まず、Adaによりプロトコルを記述し、単一計算機上でシミュレーションを行った。本稿では、Adaによる通信モデルの記述について、主に動的な多重コネクション(論理的通信路)の取り扱いに注目して考察する。

## 2. X.25の概要

CCITT勧告X.25<sup>4)</sup>はパケットモード端末(DTE: Data Terminal Equipment)と公衆データ通信網(DCE: Data Circuit terminating Equipment)との間のインターフェースを規定する。インターフェースは物理・リンク・パケットの3層(レベル)にわかれており、パケット層は最上位にあたる。パケット層プロトコルはDTE-DCE間において、制御情報とユーザデータとを含んだパケットを交換するためのパケットフォーマットおよび制御手順を規定しており、その提供する機能には次のようなものがある。

- コネクションの管理(設定/解放)
- 障害の検出・復旧
- データパケット転送時の順序制御・誤り制御・フロー制御

パケット層プロトコルを適用した通信システムの特徴のおもなものとしては、1つの端末が同時に複数のコネクションを取り扱え(多重コネクション)、またコネクションは端末間で固定的に存在するのではなく、必要に応じて動的に設定、解放される(動的コネクション)、の2点がある。

## 3. Adaによる記述

X.25パケット層プロトコルをAdaで記述するにあたって、以下のような方針を立てた。

第一には、X.25の階層構造を反映した設計を行うことである。そのため、パケット層の機能はAdaのパッケージとして実現した。この場合、ISOのOSIモデル<sup>3)</sup>におけるサービスアクセス点はパッケージ仕様部で宣言した手続きに、エンティティはパッケージ本体に、それぞれ対応づけることができる。なお、下位の2つの層(物理・リンク)に相当する部分は、複数のパケット層パッケージ間で透過的なパケット転送を行う単一のパッケージとした。

第二に、パケット層では同時に複数のコネクションを取り扱うが、これらのコネクションを一括して管理する(図1(a))のではなく、それぞれのコネクションに対してそれを管理するAdaタスクを1つつ割り当てる方法(図1(b))をとることである。本稿ではこの方法をタスク/コネクションと呼ぶ。コネクションは動的に設定、解放されるので、タスクもそれに対応して動的に生成、消滅するものとした。

また今回は上述のように通信コネクションの取り扱いに主眼を置いたため、実現の対象としたのはX.25のコネクション管理とデータ転送に関する基本的な機能のみであり、インターフェースのリスタートやオプションユーザファシリティ等のサービスは記述しなかった。

なお、本システムの開発は日本データゼネラル社のECLIPSE MV/10000上で稼働するAda処理系を用いて行った。

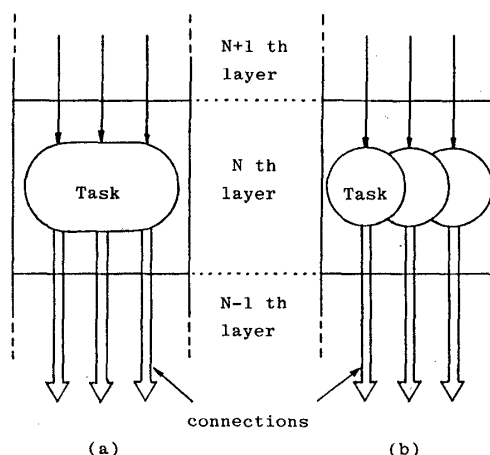


図1. 複数コネクションの管理方式

\*Adaは米国政府(Ada Joint Program Office)の登録商標である。

#### 4. バケット層パッケージの概要

図2にバケット層パッケージの構成を示す。この図の記法はBuhr<sup>2)</sup>のグラフ記法に準ずる。なお図中の矢印のついた丸はユーザデータの流れを表し、白丸が受信データ、黒丸が送信データに相当する。

図の最も上側にある長方形の枠がパッケージ仕様部の手続き群を表す。ユーザはこれらの手続きを呼び出すことによってこのパッケージのサービスを利用することができる。平行四辺形はタスクを表しており、その辺に接して描かれた枠はエントリを示す。VIRTUAL\_CIRCUIT\_MANAGER タスク（以下VCMと略す）はバケット層プロトコルが規定する通信動作のほとんどを実行するタスクであり、アクティブなコネクションごとに1つつ存在する。VCMはコネクション設定の要求が発生すると、ディスパッチャ（DISPATCHER）タスクが動的に生成する。ディスパッチャタスクはコネクションの識別子である論理チャネル番号（LCN：Logical Channel Number）とそのコネクションを管理するVCMを指示するアクセス変数（ポインタ）との対応表を管理する。ROUTER\_IN、ROUTER\_OUT、BUFFERの各タスクはVCMと下位層パッケージとの間でバケットの受渡しを行うタスクである。これらのタスクを介在させることにより、VCMタスクと下位層のタスクとが直接同期を取る必要がなくなり、両者の

並列性を高めることができる。

次に、タスク/コネクション方式の構成を実現するための方法について述べる。まず、VCMタスクは、動的に生成するために、アクセス変数によって指示されるタスク型の算体とした。新たなコネクションの接続要求が発生すると、ディスパッチャは新しいVCMタスクを生成し、そのアクセス変数を、LCNとともに対応表に追加する。通信手順が終了しコネクションを解放する場合には、VCMはそのことをディスパッチャに通知して終了し、通知を受けたディスパッチャは対応表からそのコネクションとVCMとの組を取り除く。また、コネクションにアクセスする場合には、ユーザはディスパッチャにそのコネクションのLCNを送り、対応するVCMへのアクセス変数を受け取ってそのVCMに対してサービス要求を行う。

ところで、VCMタスクの終了とディスパッチャにおける対応表の更新とが完全に同時には行われなため、VCMを呼び出す側が、終了したVCMタスクを呼び出す恐れがある。そこで、VCM呼び出しを行う部分には局所的な例外ハンドラを置いて、障害が発生した場合にこれを解決するようにしている。

#### 5. まとめ

今回の試みによってX.25システムのような多重・動的コネクション構成の通信システムを開発する際にもAdaの諸機能が有用であることが確認できた。とくにタスク/コネクションを自然な形で実現できたことは、Adaのタスク処理の強力さに負うところが大きい。また、タスク/コネクションの構成は、1つのタスクが1つのコネクションにだけ注目していればよいので、タスクの構造が簡単になり、記述が容易であった。またプロトコルとの対応も素直である。現在のところタスク/コネクション方式の効率についての評価は行っていないので、実用的なシステムに適用できるかどうかの判断はできないものの、プロトタイプングのアプローチとしては優れているといえる。

今後は、今回実現しなかった機能の拡充、および実際のシステムへの実装を行って行く予定である。

#### 参考文献

1. 荒木, 牛島: 通信システムのAdaによる統合的開発事例, 情報処理, Vol.27, No.3, pp.244-253, 1986.
2. Buhr, R. J. A.: System Design with Ada, Prentice-Hall, 1984.
3. 勅使河原: 開放型システム間相互接続 (OSI) の参照モデル, 情報処理, Vol.26, No.4, pp.299-309, 1985.
4. CCITT: Recommendation X.25, 1984.
5. United States Department of Defence: Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A, 1983.

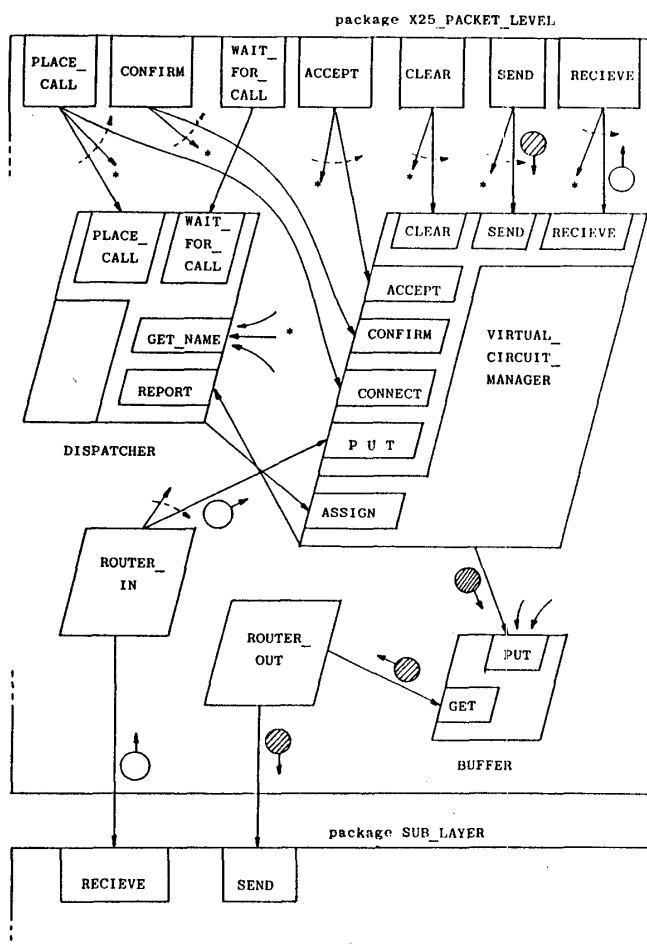


図2. バケット層パッケージの構成