

仮想化技法に基づく 公衆パケット網への端末収容について

鈴木健一・浦野義頼・小野欽司
(国際電信電話株式会社・研究所)

渡辺ス雄
(日本通信協力株式会社)

1. まえがき

現在、公衆データ網における端末収容に関しては、各国で仮想端末にもとづく検討がなされている。また、ISD、IFIP等においても、各網におけるインプリメントの進展にあわせて、標準化作業が進展してきている。

仮想端末は、当初、ホスト計算機においてアプリケーションプログラムが実端末の属性に制約されることなく作成できるように、端末ハンドラに標準的な論理端末を想定したものである。その後、ネットワーク技術と分散処理技術の進展に伴い、この端末ハンドラがホスト計算機から独立し、ネットワーク環境下での仮想端末に変貌してきた。

また、ユーザの立場からは、網を介してホスト計算機にアクセスするのみではなく、既存の各種端末とも通信したいというニーズが強く、端末/ホスト間、端末相互間通信いずれをも包含できる仮想端末仕様、プロトコルの設定が望まれている。

筆者等は、上記背景をふまえて、公衆パケット網における端末収容について仮想端末の立場から検討をすすめることにし、IFIPモデル¹⁾をベースとしてその対象を端末相互間通信に拡張することとした。端末相互間通信においては、プロトコルが対称的であることにより従来のホスト主導型のプロトコル(非対称)とは差が生ずると考えられる。また、公衆パケット網における仮想端末を考える際には、それを実現する通信処理装置が集線装置の機能をもつことも重要であると考えられる。

本稿では、仮想化技法にもとづく公衆パケット網を介した端末間通信を実現するにあたり、既存の各種端末を

scroll, paged等の端末として仮想化した際の実端末・仮想端末間のマッピング処理、仮想端末プロトコルおよびX.25パケット網との対応について報告する。

2. 仮想端末モデル

2.1 基本的な仮想

仮想端末は、ホスト計算機内で行なわれていた端末ハンドリングを計算機システムから切り離して、ユーザが選定できる標準的な特性をもたせ、ホスト計算機に対しては標準的なインタフェースを提供する。従って、図1のように、仮想端末は実端末と計算機との間に2つのインタフェースをもつ。この発想によれば、遠隔の計算機は仮想端末のみをハンドリングすればよく、アプリケーションプログラムは特定実端末を意識することなく作成できる。一方すべての実端末は、その特性を仮想端末にマッピングすることになる。

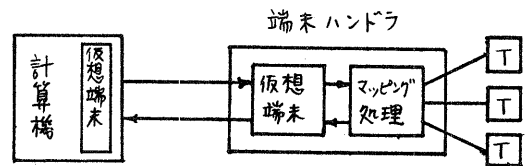


図1. 仮想端末の考え方

2.2 仮想端末における通信形態

仮想端末を想定した端末・ホスト間通信の形態は、図2に示すような3種類が原理的に可能である。

(1). 端末・ホスト間通信

(2). 端末・端末間通信²⁾

(3). ホスト・ホスト間通信

これらの通信形態のプロトコル上の差異は、(1)においては、VT間のネゴシエーションがホスト主導型で行なわれる

ように、プロトコルが非対称型であるのに対し、(2)、(3)では対称型のプロトコルが採用されることになる。

従来検討されている仮想端末プロトコルは主として(1)の範囲であるが、本稿では(2)を中心として検討した。

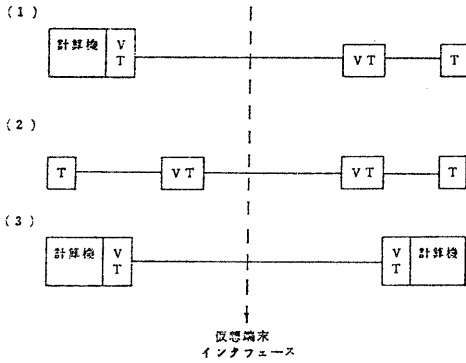


図2 仮想端末における通信形態

2.3 仮想化処理と仮想端末

仮想化処理と仮想端末の位置づけを図3に示す。

(1) Adaptation Unit (AD)

実端末を仮想端末にマッピングする。具体的には Presentation Unit (スクリーン・プリンタ), Input Unit (keyboard), ミグナリング機能 (alarm, interrupt) と VT 間のインタフェースをとる。

(2) Data Structure (DS)

仮想端末を規定する各種パラメータ (表2参照)・エリアヒテキストデータを

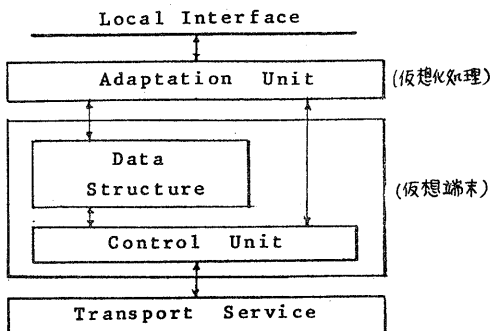


図3 仮想化処理と仮想端末

蓄積する文字 array からなる。Data Structure には Presentation/Input Unit と補助デバイスに対応して Primary および Secondary Data Structure がある。

(3). Control Unit

Transport Service とのインタフェースをとり、仮想端末プロトコルを操作する。

2.4 仮想端末の Class

仮想端末の Class は通常 Scroll, Paged Data entry, Non standard の4種類に分類されるが、本稿では Scroll, Paged を対象にしている。これらの Class は Data Structure の構造と 1:1 に対応する。

(1) Scroll Class

ハードコピー、ディスプレイ装置等で Scroll 型の端末を対象にしており Data Structure は line を意識する。

(2) Paged Class

Data Structure は有限の二次元 array であり、Page を意識する。Page 内の位置が指定できる。

3. 仮想端末プロトコル

仮想端末では、Data Structure (DS) での実端末の特性が仮想化されており、仮

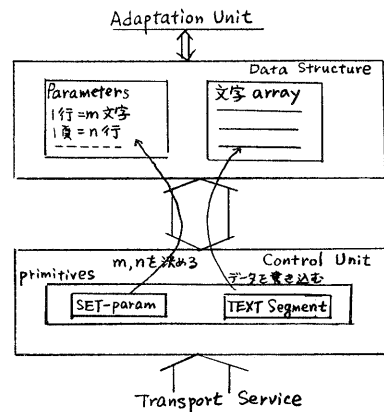


図4 下位 layer からの Data Structure へのアクセス

表1 プリミティブ一覧

Primitive name	type		block header (hex)	item code (hex)	length field (octet)	data field
	Q	mode				
text-segment	M	text	0X	80	n	アリ
new-line (NL)	M	text	0X	81	0	/
start-of-line (CR)	M	text	0X	82	0	/
delete-all	M	text	0X	83	0	/
position	M	text	0X	84	2	アリ
call	M	control	11	C7	16	アリ
accept	M	control	11	C8	0	/
clear	M	control	11	CF	0	/
synchronized attention mark (SYN-M)	M	control	11	C0	0	/
purge mark (PU-M)	M	control	11	C1	0	/
pause mark (PA-M)	M	control	11	C2	0	/
request parameter range (R-PARA)	M	negotiation	21	E0	2P	アリ
indication parameter range (I-PARA)	M	negotiation	21	E1	2P	アリ
set parameter value (S-PARA)	M	negotiation	21	E2	2P+1	アリ
turn	M	negotiation	21	E3	0	/
agree	M	negotiation	21	E7	0	/
disagree	M	negotiation	21	E8	0	/
echo	M	diagnostic	31	F0	n	アリ
asynchronous attention (ASY-ATT)	I	ASYN	4X	/	/	/
please	I	ASYN	50	/	/	/
resume	I	ASYN	51	/	/	/
synchronized attention (SYN-ATT)	I	SYNC	6X	/	/	/
purge	I	SYNC	70	/	/	/
pause	I	SYNC	71	/	/	/

(注) length fieldのPは pairを意味する

表2 パラメータ一覧

Parameter name	Default value
Terminal Class	scroll
Auxiliary Data Structure	main
Dialogue Mode	free running
Erase/Overprint	undefined
Line Length	undefined
Page Size	255

想端末プロトコルは、このDSへのアクセスという形で定義する。

3.1 Data Structureへのアクセス

DSは各種パラメータエリアとテキストデータと蓄積する文字arrayからなる。各種パラメータは文字arrayの属性(ライン長、ページサイズ等)を規定し、プリミティブ*は文字arrayおよびパラメータエリアへのアクセスを規定する。図4は下位layerからのDSへのアクセス例を示している。同時に Adaptation Unitからのアクセスも認められている。(Write/Read) (*プリミティブの中にはVTのフェーズ切替、制御用もある)

3.2 プリミティブとパラメータ

VT相互間の通信はプリミティブによって行われる。プリミティブには通常のデータを転送するためのMessageと、割り込み情報を転送するInterruptの2つのtypeに分類される。Message typeのプリミティブはその特性により4 mode (text, control, negotiation, diagnostics)に分類されている。Interrupt typeのプリミティブは asynchronous と synchronous の mode に分類される。これらプリミティブの一覧を表1に示す。

表2に Data Structure の属性を表わすパラメータと Negotiation 不成立時の Default valueを示す。

3.3 フォーマット

VT間で転送されるプリミティブは block を単位としている。

(1) Block 構成

block は Octet を基本単位とし、block header 部と item 部からなる。(四) item 部を伴う場合は VT 間のプリミティブを message type といい、伴わない場合を interrupt type という。両者の識別は block header 部の Block Qualifier による。

図6にユーザが通信するメッセージ(ユーザメッセージ)とBlock, Itemの関係を示す

(2) Block header部の構成

図7にBlock header部の構成を示す。

[Format Identifier] (bit 7)

将来のBlock header部の拡張用

0: block header長が1 Octet.

1: 拡張用

[Block Qualifier] (bit 6)

該当blockがmessage typeかinterrupt typeのprimitiveかを示す。

0: message type, 1: interrupt type

[Primitive mode] (bit 5,4)

blockに含まれているprimitiveの特性を示す。

Block Qualifierの値により意味は異なる

Block Qualifier = 0 (message type primitive)

00: text mode primitive

01: control mode primitive

10: negotiation mode primitive

11: diagnostic mode primitive

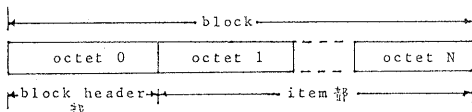
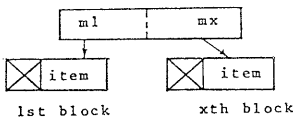


図5 Block 構成

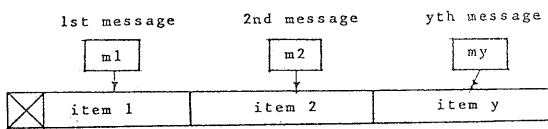
(1) 1 block = 1 message



(2) x blocks = 1 message



(3) 1 block = y messages



m, m1, m2, mx, my -- message

☒ -- block header

図6 Message, Block, Item の関係

Block Qualifier = 1 (interrupt type primitive)

00: asynchronous 拡張用

01: asynchronous mode primitive

10: synchronized 拡張用

11: synchronized mode primitive

[Turn indication] (bit 3)

end-to-endの送信権を示す

0: my turn (送信権の維持)

1: your turn (送信権の放棄)

[EOM] (bit 2~0)

該当Blockがユーザメッセージの最終blockであるか否かを示す。

000: continue

001 } : ユーザメッセージ終了条件用パラメータ

111 }

またBlock Qualifier = 1のときbit 3-0はinterrupt data fieldと見なされる。

(3) Item部の構成

item部は1つ以上のitemから構成される。各item部は転送されるユーザメッセージの各要素と対応し、Code, length dataの3つのfieldによって表わす(図8) item部で表わされるmessage要素は、全て同一のprimitive modeに属しており、異なるmodeのmessage要素がitem部に含まれることはない。

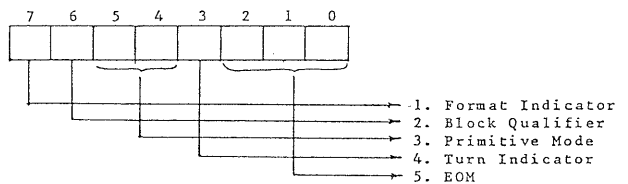


図7 Block header部の構成

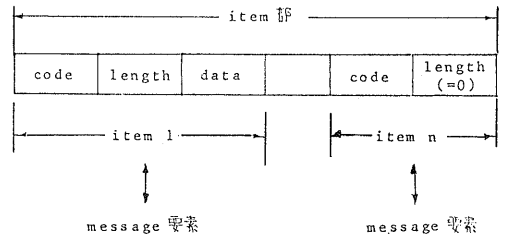


図8 Item部の構成

3.4 Negotiation機構

Negotiationは、End-to-Endの通信を保証するため、VT間でDSの属性を統一させるもので、initial negotiationとrenegotiationに分けられる。initial negotiationはprimary端末どうしで行なわれ、起呼側からset parameter value primitiveを送出しagreeを期待する。disagreeの場合にはdefault値で通信が可能となる。

renegotiationは、補助デバイスの指定解除およびパラメータの部分的な修正等を契機として行なわれ、signalling phaseより移行する。

Negotiation時におけるコンテンションを解決するため、S-PARA primitiveには10 Octet分ランダム変数のfieldがあり、コンテンション発生時には、その値を比較する。

3.5 Signalling機構

端末からの割込みにより、データ転送を中断して、renegotiationを行わせるために両VTを同期させる機構である。通常 interrupt primitiveにより同期開始の指示が行なわれ、Mark primitiveを交換することにより同期させる。Synchronized signallingの場合のフェーズ移行を図9に、伝送制御シーケンス例を図10に示す。

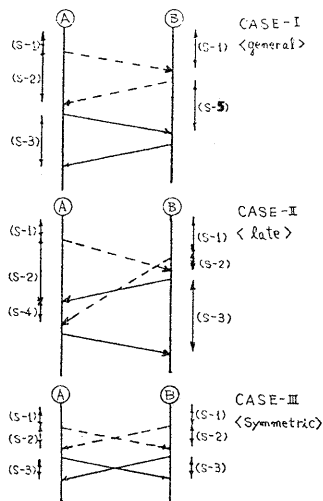


図10 Interrupt/Markの伝送制御シーケンス例

点線: interrupt S-1, S-2, S-3: 図9参照
実線: Mark

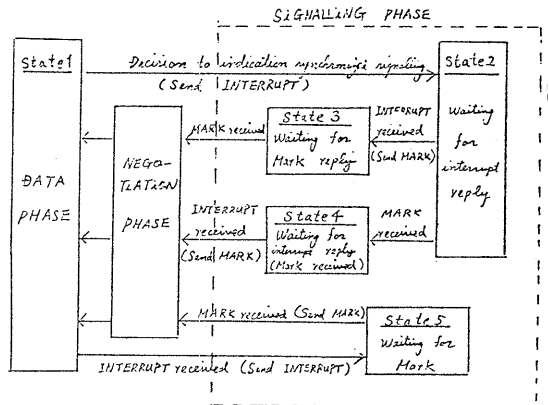


図9 Synchronized signallingにおけるフェーズ移行

3.6 VTのフェーズ管理

VTはコネクション、ネゴシエーション、シグナリングおよびデータの4フェーズで管理されている。

4. 仮想化技法にもとづく公衆パケット網を介した端末間通信

上記の仮想端末プロトコルをもとに公衆パケット網を介した端末間通信を実現した。

4.1 システム構成

実験システムの構成を図11に、収容した実端末を表3に示す。

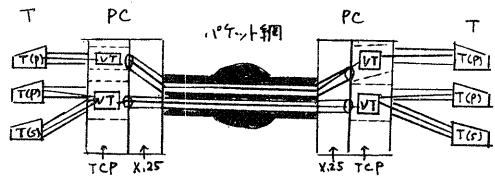


図11 システム構成 PC: プロトコル変換装置 TCP: Terminal Control Program

表3 収容端末一覧 (*補助デバイスも使用)

NAME	SPEED	DATA LINK	CODE
TI-700 TI-745 HP-2645A M345FT	300 bps	S/S Free	ASCII even
Intecolor 8001	300 bps	S/S Free	ASCII no
IBM 3767	300 bps	S/S 2741	PTTC/BCD odd
* WANG 20/30	2400 bps	BSC 3780 3780C	EBCDIC
* TI-700 CMT	300 bps	S/S Free	ASCII even

4.2 ソフトウェア・モジュール

(1). モジュール構成

端末間通信プログラム(TCP)は既に開発済のX.25プログラムとともに、プロトコル変換装置上にタスクとして登録した。図12にソフトウェア構成およびISO7層モデルとの対応を示す。各モジュールの機能概要を以下に示す。

①. RDH (Real Device Handler)

実端末と仮想端末間のマッピング⁴⁾処理(仮想化処理)を行う。(例、伝送制御手順の吸収、キャラクタセットの変換)

② LLC (Local Link Controller)

RDHとVT間の通信路、インタフェースの管理。端末ユーザとVT間のLocalなサービス(コマンド/レスポンス、多重アクセス)の管理。

③ VTP (Virtual Terminal Program)

Data Structureの管理。VTのフェーズ⁵⁾管理。VTコマンドの解析とレスポンス

④ VTC (Virtual Terminal Controller)

相手VTとのコネクション管理、複数VTの制御、VT間ポリミティブの送/受信管理、VTP、VTC間のインタフェースおよび通信路の管理

⑤ VCP (Virtual Circuit Control Program)

CCITT勧告 X.25 レベル3 の管理

⑥ LCP (Link Control Program)

CCITT勧告 X.25 レベル2 の管理

OSI N227

Application
Presentation
Session
Transport
Network
Data Link
Physical

TCP

RDH			L
V	T	P	LC
---V T C---			
V	C	P	
L	C	P	

図12 ソフトウェア構成

(2) RDHにおける仮想化処理⁴⁾

RDHにおける仮想化処理は具体的には①コード変換、②伝送制御手順の吸収が主な機能である。①では、EBCDIC、

ASCII, IA5等の相互コード変換を行う。

②では、無争順, IBM2741, バーニック, BSC 3780, 3780C手順等における伝送制御キャラクタの削除, 作成によりメッセージの転送を保証する。

実端末とRDHおよびVT間では図13のマトリックススイッチの方式を用い、フレキシビリティをもたせた。

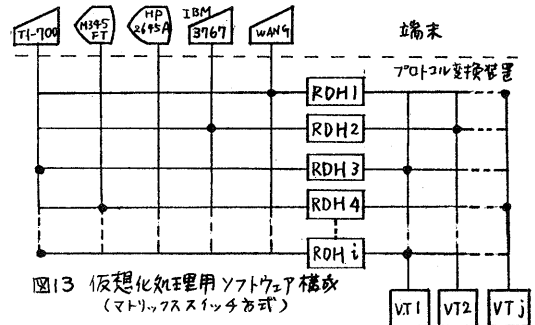


図13 仮想化処理用ソフトウェア構成 (マトリックススイッチ方式)

(3). 複数VTの制御

プロトコル変換装置上で仮想端末を取り扱うにあたって、VTP内で複数の論理VTを実現している。

4.3 VTポリミティブとX.25パケットの対応

message type primitiveのうち、call, Accept, Clearは各々CR, CC, CQパケットと対応させ、コネクション確立、切断時のオーバーヘッドを軽減した。その他のmessage type primitiveはDTパケット、Interrupt type primitiveはITパケットにマッピングした。この際、Mark関係のprimitiveはIFパケットと対応させることも考えたが、伝送路上に残っているパケットをdiscardするため、今回はDTパケットで行なった。

4.4 ユーザ・インタフェース

本プログラムは、実端末とVT間のローカル・コネクションと相手VTとのVT・コネクションに分け各々コマンド/レスポンスによるユーザ・インタフェースをもつ。

(1). ローカル・コネクション

- (a). Log on ローカルコネクションの確立
- (b). CALL アプリケーションプログラムの起動
- (c). OVER ローカルサービスの停止

(d) HELP ローカルサービスのガイダンス

(h) TURN 送信権の強制反転

(2) VT-コネクション

(i) SHOW パラメータ値リストの出力

(a) 割り込み VTにおけるコマンドフェーズへの移行
(break(ロングスペース)の端末固有コード)

(j) RESUME バッファ解放要求

(b) CALL オペランドに指定された端末への発呼

(k) PLEASE メッセージ送信の催促

(c) CLEAR VTコネクションの切断

(l) ECHO 相手VTとのループ検査

(d) HELP VTコマンドのガイダンス

(m) OVER VTサービスの終了

(e) USE 入出力補助デバイスの指定

4.4 シーケンス・ダイアグラム

(f) DISUSE 指定補助デバイスの解除

図14に端末間通信のシーケンス・ダイアグラム例を示す。

(g) SET パラメータ値の変更指定

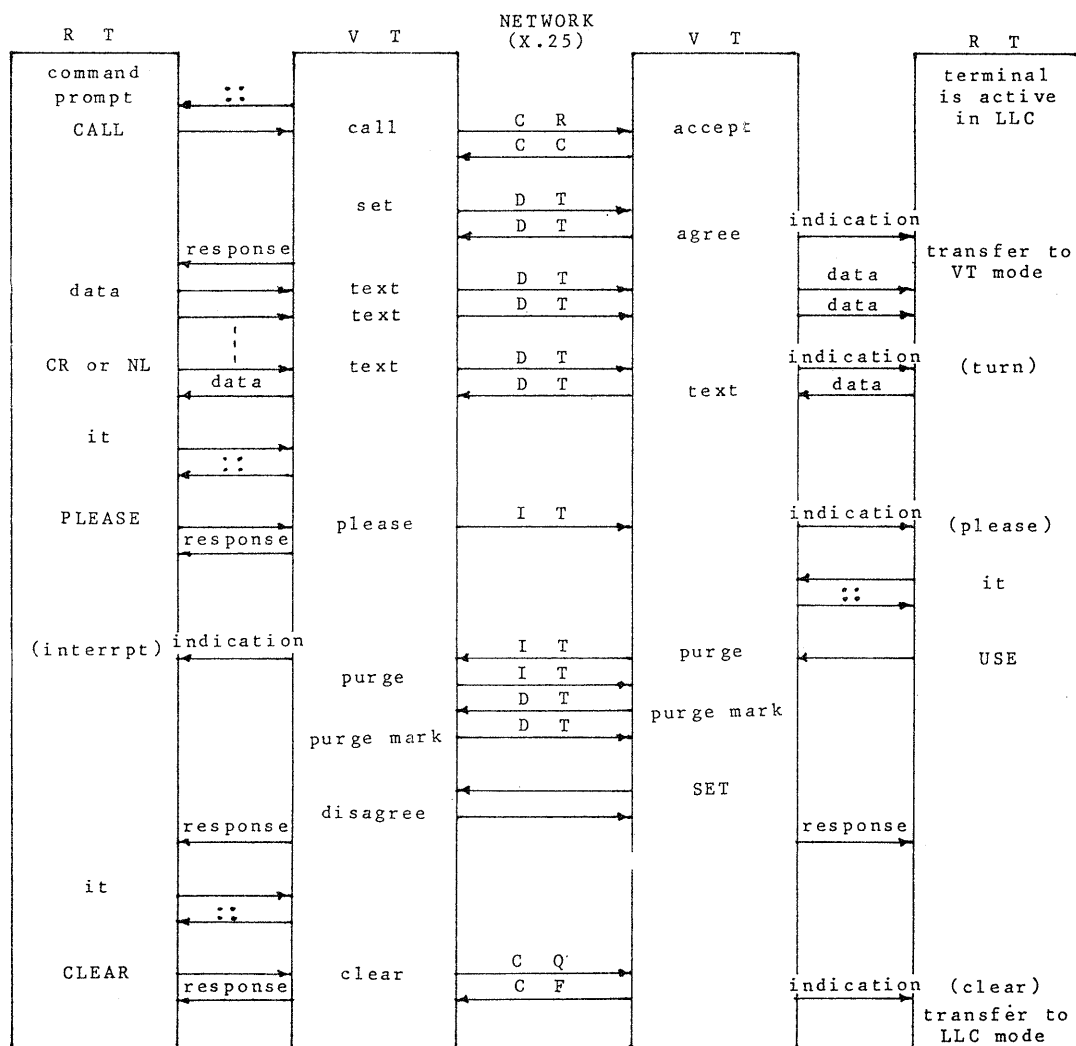


図14. 仮想端末を介した端末間通信シーケンスダイアグラム
(注. 図中 RR/IF パケットは省略した)

5. 考察

(1). IFIP 仮想端末プロトコルとの比較
本仮想端末プロトコルと、ベースにした IFIP モデルとの相違は、概略以下のとおりである。

①. VT モデル

IFIP モデルでは、Adaptation Unit の位置づけを VT 内部に置いているが、実端末と仮想端末間のマッピング処理は、VT の外部と考え、本モデルの VT は Data structure および Control Unit とした。

②. Block フォーマット

IFIP モデルでは、message primitive と interrupt primitive で Block 構成が異なっているが、統一的な Block 構成として、Block header の Block Qualifier により message type, Interrupt type primitive の識別を可能とした。また、1 block 内に複数のユーザ・メッセージを含むことを認めた。

③. Phase 概念の導入

本モデルでは phase の概念を明確にし、IFIP モデルで議論されていない Connection Phase を採用した。また synchronized signalling の場合、signalling phase から data phase への遷移は negotiation phase を経由することにした。

④. Primitive の拡張

X.25 パケットの CR, CC, CQ の契機となる call, accept, clear primitive 等の Primitive を追加した。

(2). 仮想端末仕様と RDH

仮想端末仕様で scroll, Paged を対象としている範囲では、フラスの設定は限られた範囲のパラメータで決定するため、実端末と仮想端末のマッピング処理が大きな比重を占める。従って RDH のパラエティをどれだけ保持できるかにより通信処理装置の性能が特徴づけられる。今回採用したマトリックス・スイッチ方式を用いれば、RDH をあらかじめ回線数以上に含むことができ、収容端末の変更に対しても柔軟に対応できる。

(3). マルチスフロー・クラス

WANet model 20/30 端末を補助デバイスとして採用した実験例では、3780/3780C 手順の特性として Max. 5/2 キヤラクタがひとまとめに送信される。これらの端末を扱うにあたっては、Data structure として 1 line の意識をかつ scroll と Page の間で中間的にマルチスフロー・クラスを設けることにより scroll up, down で line 前後の内容をチェックできる。

(4). 集線装置としての機能

論理的に複数の VT を同一通信処理装置上に構築し、仮想端末仕様にもとづく集線装置を実現するため、VT primitive の送受信管理はリエントラントタイプの共通モジュールとした。本来この機能は、layer 的には VTP 内で処理されるべきだが、インプリメント上、VTC で実現した。

(5). 対称型プロトコル

端末・ホスト間通信の形態では、コネクション確立、切断が一方向性の非対称型プロトコルであったのに対し、端末間通信では双方向性の対称型プロトコルとなる。この場合、起呼、切断要求等の衝突も考慮する必要がある。従って、端末間通信における仮想端末プロトコルではコネクション・フェーズ管理が重要となる。

6. あとがき

本稿では仮想端末仕様にもとづく端末間通信に焦点をあて、その仮想化技法およびプロトコルについて検討を加えた。今後アーキテクチャ的な検討を深めるとともに、CCITT PAD との融合性についても検討したい。最後に日頃、御指導いただいた KDD 研究所 鍛冶所長、寺村副所長、高橋次長に感謝します。

- [参考文献] (1) IFIP/WG 6.1 "Proposal for standard VTP" IN WG PROTOCOL 91 (FEB. 1978)
(2) 鈴木、浦野、"端末仮想化技法による PAD-PAD 通信の拡張" 昭和三十五年電気通信学会全国大会 1606
(3) ISO TC 97 SC 16 N227
(4) 小野地 "端末仮想化処理におけるプロトコル変換" 昭和三十五年電気通信学会全国大会 1607