

## データ交換処理における呼接続方式の検討

有満秀浩 藤田克孝 濱田晃

NTT 交換システム研究所

近年、遠隔地とのオンラインデータ処理等の普及に伴ってデータ交換処理に対する需要が増大している。特に1本の回線上にレイヤ2/レイヤ3の論理リンク/コネクションを多重するプロトコルの場合、データを集中管理するホストでの多重度は非常に大きくなるため、これらが一斉に接続要求を行なうと、交換機には非常に大きな負荷がかかり、接続遅延時間が増大することが大きな問題になっている。本稿では呼接続処理負荷を平滑化し、交換機負荷の軽減と接続遅延時間等のユーザに対するサービス品質を向上するための提案を行なう。

## A call-setup method in data switching systems

Hidehiro ARIMITSU Katsuyuki FUJITA Akira HAMADA

NTT Communication Switching Laboratories

As the demand for data switching services has been increasing in recent years, data switching systems are often congested and call-setup delay rapidly increases when large hosts which are multiplexing many virtual connections on a physical access channel request to establish many calls at a time. In this paper, we describe the load smoothing method in a switching system to decrease call-setup delay and to improve service qualities.

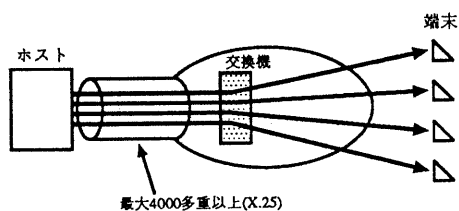


図 1: 接続形態

## 1 はじめに

近年、遠隔地とのオンラインデータ処理等の普及に伴ってデータ交換処理に対する需要が増大している。特に大規模なホストコンピュータでデータを集中管理し、各地の端末が適宜オンラインでデータを参照するような使用形態では概して回線使用効率が低いため、一般にホスト側では一本の回線上に複数のレイヤ2、レイヤ3の論理リンク / コネクションを多重して回線使用効率を上げている。このため、ホスト側では一本の回線上の多重度は非常に大きくなり、例えば X.25 ではプロトコル上のレイヤ3 多重度は最大 4000 以上にも及ぶ。(図1)

端末の高機能化、高処理能力化に伴って分散処理等が今後進行すると思われる。この際にもフレームリレー等の多重化を前提とするプロトコルにより、ホストばかりではなく多くの端末で構成される LAN 間を接続する形態が想定され、やはり多数の相手先との接続が必要になると思われる。このため、今後はますます多くの高多重度端末 / ホストがネットワークに接続される形態が増えると予想される。

このような多重度の高いホストが、制御しているリンクを一斉に接続要求した場合、これを処理する交換機では一時的に呼接続処理比率が非常に高くなり、特に CPU 処理負荷が急激に増大する。処理能力を越える数の接続要求があった場合には接続遅延時間の増大さらには交換機の輻輳が起こるが、この場合の遅延時間の増大、輻輳による影響は当該ホストのみではなく、同一交換機に収容されている他の端末に対しても及ぶという意味で問題が大きい。特にプライベート網などの比較的小規模なネットワークでは交換機処理能力も小さい場合が多く、大群化効果が現れにくいいため、このような呼が突出し、問題は深刻である。

これまでこの問題はネットワーク側では対処できず、ホスト側で接続要求数を一定の値以下に抑える方法しかなかった。しかし逆にホスト側から見れば本来一斉に接続要求可能であるはずの呼をこの制限のために抑えなくてはならず、全ての相手先に接続が完了するまでに時間がかかる、リンクの状態管理が不必要に複雑になるなどの弊害があった。

本稿では、このように1本の回線上で多くのレイヤ2 / レイヤ3リンクを多重して制御している大規模なホストが一斉に接続要求を行なう場合、交換機の呼接続処理負荷を平滑化することで接続遅延時間を短縮し、ホスト側の不要な状態管理を削減する等のサービス品質を向上するための提案を行なう。

なお、本稿で述べる方式はデータ交換一般に適用できるものと考えており、特定のプロトコルに依存するものではないが、本稿ではデータ交換の中でも現在普及率が高いと思われる X.25 と、今後の発展が予想される Q.933 プロトコルによるフレームリレーを例として方式提案を行なう。

## 2 一斉発呼の特徴

本稿では1台の端末装置が短い時間の間に非常に多くの呼接続要求を行なうことを一斉発呼と定義する。一斉発呼の特徴として考えられるものは以下の通りである。

- 接続相手先が多数になるほどユーザが構築しているシステムは業務上の重要性が増し、またオペレータ等の省力化のために呼接続処理は自動化される。このため、複数の相手先への呼接続要求順序、呼が接続 / 切断される時刻は毎回ほとんど同一であると思われる。
- 一般にプライベート網もしくは公衆網上で企業専用網等を構築する場合には、接続先、使用するアプリケーション、通信形態等は固定化されており、呼接続ごとにファシリティ等の内容が異なるような場合は非常に少ないと思われる。

これらの特徴により、ユーザ側のシステムが稼働を開始し、ある程度安定した時点では一斉発呼の時刻、接続要求内容などを定型化できることが多いと予想できる。また、ユーザのシステム規模が大きく、ホスト

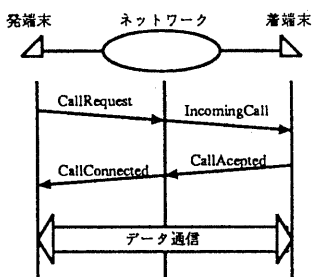


図 2: 呼接続モデル

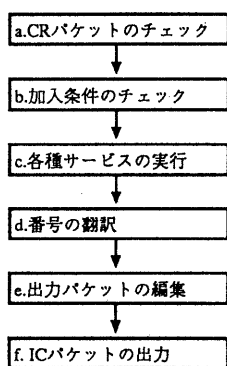


図 3: 一般の交換処理概要

側の多重度が大きいほど、すなわち一斉発呼による同時接続要求数が多いほどこの傾向が強いと思われる。

### 3 呼接続処理平滑化の考え方

本稿では呼接続シーケンスを図 2 のように仮定する。これは信号名など X.25 のシーケンスを例にしたものであるが、他のプロトコルであっても基本的には同様である。

このうち、交換機処理負荷が最も大きいのは Call-Request(CR:発呼要求)パケットを受けとって IncomingCall(IC:着呼)パケットを送出する処理である。このため、一斉発呼による処理負荷を平滑化するために本稿ではこの部分の処理に着目した。

データ交換処理に限らず、この部分の交換機での処理はおおむね図 3 のような手順で行なわれる。

アナログ回線交換での電話、FAX 等に代表されるデータ交換以外のペアラサーサービスでは不特定多数のユーザ間通信が多くを占め、また回線上で多数の呼を多重することができないため、1本の回線で様々な相手先と接続することが多い。そこで接続要求の時刻、接続相手先などの接続要求内容等が毎回大きく異なることが一般的である。従って端末からの接続要求が到着するまでは交換機では何も行なうことができず、接続要求が到着してはじめて処理を開始することが一般的であった。

ただし、アナログ回線交換では多重を行なっていないため、本稿で述べているような多重化したプロトコルと比較すれば同時接続要求数のピーク比率は小さく、多重化を前提としたデータ交換における一斉発呼のような問題は起こりにくい。

一方、データ交換で多重度が高い端末では一斉発呼の問題があるが、2項の一斉発呼の特徴で述べたように接続要求の時刻及び接続相手先等の内容にはほとんど変更を加える必要がないという前提のもとでは、以下のようなサービスが可能であると思われる。

端末装置をネットワークに接続する際に、一斉発呼を行なうと思われる装置の場合は、一斉発呼時刻、接続要求内容等をあらかじめ登録しておくことにより、交換機での一斉発呼時のピーク負荷を事前準備、事後処理等に分散することができる。

接続要求内容を交換機で予測できるとすれば、図 3 の処理手順中では、前回の CR パケットを参照することによって 'a' ~ 'e' までの処理を端末からの接続要求前に行なうことができる。'f' の IC パケット出力については実際の接続要求がないと行なうことはできないため、それまで待つ必要がある。

このように、あらかじめユーザからの呼接続要求に対する準備を交換機で行なうことによって一斉発呼が到着する前に処理時間の大きいほとんどの処理を事前に行なっておくことができる。実際に一斉発呼が到着した時にはそれまでに準備しておいた処理内容が正しいものであるかどうかをネットワークで確認し、IC パケットを出力するのみでよいいため、一斉発呼時の交換機ピーク負荷を従来よりも大きく削減できると考えられる。

本稿ではこのような考え方による処理方式を ACC (active call control)、従来の方式を PCC (passive call control) と呼ぶ。

本方式はサービスとしては、X.25、フレームリレーでのPVCサービスとの類似点が大きいが、以下の点で異なっており、PVCとは別のサービスとして位置付けられる。

- PVCは接続先が固定であり、端末側の一時的な接続先変更は困難である。
- ACCでは代表選択、着信転送サービス等が使用でき、PVCよりも融通性が高い
- CRパケット等で発呼時にユーザデータを送信できる

## 4 ACC手順

### 4.1 ACCの概要

前述の図2のようなモデルについて考えた場合、ACCの手順は以下ようになる。

1. 交換機は登録内容に従って一斉発呼を行なう装置の呼接続要求時刻、要求内容を内部に記憶する。
2. 1で記憶しておいた一斉発呼時刻の前に、交換機は記憶しておいた接続要求内容に従って内部の接続準備を行なう。ただし、このときユーザの端末装置とのインタフェースがPCCの場合と異なるようなことがあってはならないため、端末もしくは他ネットワーク等へのイベントは送信しない。
3. 端末からの実際の接続要求を待ち、要求内容と登録されていた接続要求内容を比較し、同じならば2で用意しておいた接続処理を継続し、外部にイベントを送出する。
4. 接続要求内容が登録しておいた内容と異なる場合は前項で準備しておいた処理結果は無視し、通常の呼接続処理を行なう
5. 登録されていた時刻をある程度過ぎても端末からの接続要求が来ない場合は、前記の処理結果は廃棄し、当該リンクを空き状態とし、補足していた各種のリソースを解放する。

### 4.2 ACCの詳細手順

手順の概略を図4に示す。

図中の番号部分についての処理詳細を以下に記す。

#### 4.2.1 ネットワークへの登録内容

一斉発呼に対する準備を行なうためには、PCCの呼接続処理で必要になる全ての情報、すなわち接続相手先番号、X.25の場合には様々なファシリティパラメータ、Q.931/Q.933では様々な情報要素まで含めた呼接続要求内容が必要である。通常、多重を前提にしたプロトコルの場合には図4のCRパケットのように接続要求内容は発呼時の最初の1パケットに集約されており、これに呼接続に必要な全てのデータが含まれることが多く、X.25の場合にはCRパケット、Q.931/Q.933の場合にはSETUP信号がこれにあたる。

ACCの呼接続準備の際にできる限りPCCの呼接続との処理の共通化を図る観点から、このCRパケットをそのまま登録することを考えた。ただし、ユーザデータを含むことが可能なプロトコルの場合には、ユーザデータが毎回呼接続ごとに異なる可能性が高いため、これを除いた呼接続に必要な情報に限定する必要がある。

また、一斉発呼時刻についてはユーザ側の使用状況に応じて、時間、日、週、月、年等の様々な周期で一斉発呼を行なうと考えられるため、これらに対応できるような登録方法である必要がある。例えば多くのユーザは平日には接続を行なうが、週末、休日には接続を行なわない。また月曜に発呼を行ない、金曜の夜に切断を行なうようなユーザがいればACCは月曜のみ起動すればよい。

#### 4.2.2 一斉発呼に対する準備

呼接続準備のための負荷が通常の呼接続処理を圧迫するようなことがあってはならない。このため、例えば処理に優先度を設け、ACCによる呼接続準備の優先度を落とす等の方法を用いて準備のための負荷が通常呼接続処理を妨げないようにする必要がある。

このとき、多重化されたデータ交換と回線交換の相違として、以下の点を考慮する必要がある。回線交換では一般に呼接続処理時にトランクなどの物理的な装置を割り当てる。これらは高価なハードウェアリソース

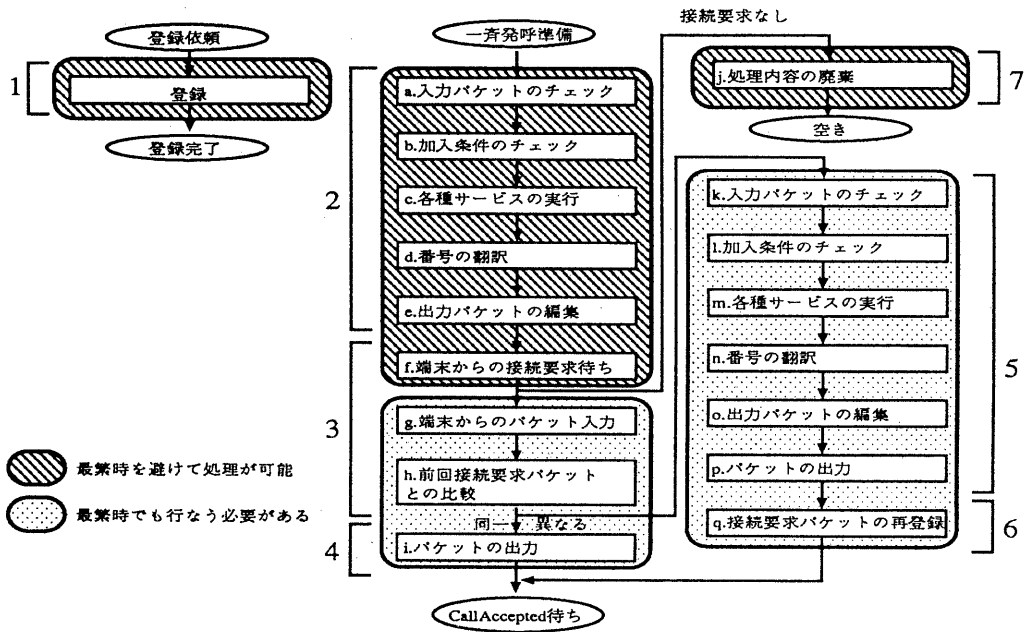


図 4: ACC手順の概略

スであり、コストを抑えるためには無効保留時間では  
 できる限り小さくする必要がある。これに対し、中継網  
 の多重化を前提にしたデータ交換では論理的には呼接  
 続時に何らかのリソースを割り当てる必要があるが、  
 多重化されているために回線交換ほどには無効保留  
 時間は大きな問題にならない。ここでいう多重化され  
 たプロトコルの論理的なリソースとは主にメモリであり、  
 今後さらなるコスト低下が予想される。そこで、  
 特にデータ交換処理においては呼接続準備の開始時刻  
 等を決定するためのスケジューリングとして多少の無  
 効保留時間等の無駄があっても確実に登録された時刻  
 までに準備が終了するような方法を選択することが望  
 ましい。これには例えば一日のうちでもほとんど負荷  
 のない夜中に行なう等の方式が考えられるが、より効  
 率の良い方法について今後考察する必要がある。

交換機であらかじめ呼接続準備を行なうことによ  
 ってユーザに対するインタフェースが変更されることは  
 許されない。呼接続準備の段階ではユーザからの発呼  
 があるかどうか不確定であるため、実際の呼接続要求  
 が来るまではネットワークの外部にイベントを送るよ  
 うな処理を行なうことはできない。

#### 4.2.3 準備内容と実際の発呼要求の比較

準備しておいた処理内容が正しいものであるかど  
 うかを判定する方法としては、呼接続内容の登録時に  
 何を登録するかによっていくつかの方式案が考えられ  
 る。上で述べたようにCRバケットをそのまま記憶し  
 ておくような方式ではこれを実際のCRバケットと単  
 純に比較することで対応できる。この方法では、接続  
 要求内容が全く同一であっても複数のオプションパラ  
 メータがあった場合にそれらの指定順序が入れ替わる  
 だけで別内容の要求であると思われるが、2項  
 の一斉発呼の特徴で述べたようにCRバケットの編集  
 を自動化しているような場合は順序の逆転が起こる確  
 率は非常に小さいと考えられる。一方、各パラメータ  
 を分析し、パラメータ毎に比較を行なうような方式も  
 考えられるが、パラメータの指定順序によらないかわ  
 りに処理時間が大きくなるため、望ましくない。

正常性確認は通常の呼接続動作中にはなく、また一  
 斉発呼時には各CRバケットごとに必要な処理であ  
 るため、処理時間はできる限り小さくしなければならない。  
 今後、呼接続時には様々なサービスが提供され、

また Q.933 によるフレームリレー VC サービスなどのさらに複雑なプロトコルがサポートされると、それに伴って呼接続時の処理時間増加が予想されるため、パケット比較に必要な処理時間の割合はさらに小さくなると思われる。

#### 4.2.4 準備内容と合致した場合の呼接続

受信した呼接続要求と登録されていた要求内容を比較した結果、同じものであれば準備しておいた内容と合致したと判断できる。この場合には、準備段階で作成しておいた IC パケットを実際に出力するのみである。(本稿では本ケースを"ACCのヒット"と呼ぶ)

ただし、例えばユーザデータを含む呼接続要求の場合には、受信した CR パケットからユーザデータ部をコピーする必要がある等、どこまでを準備段階で編集できるかはプロトコルに依存するため、インプリメントする時には注意が必要である。

#### 4.2.5 準備内容と異なる場合の処理

CR パケットを比較した結果、同じものでなければユーザは登録しておいた内容とは異なる接続を望んでいると思われる。従って、それまでに準備した内容は廃棄し、新たに通常の呼接続処理を行なう必要がある。(本稿では本ケースを"ACCのミスヒット"と呼ぶ)

#### 4.2.6 CR パケットの再登録

ACC がヒットしなかった場合には、次回以降に同じ内容で ACC 準備を行なうかどうかの問題がある。つまり、ミスヒットがたまたま 1 回だけのものであるのか、次回以降は連続して新しい CR パケットが来るのか交換機では予測できない。

複数の CR パケットを保存しておけばより正確な判断ができると思われるが、本稿では簡単のため、接続要求が自動化されている場合、次回以降は新規の接続要求内容になるものとして、新規に CR パケットを再登録し、次回はこの新規接続要求を使用して ACC を行なうこととした。

#### 4.2.7 呼接続要求がなかった時

登録された時刻をある程度過ぎても呼接続要求がなかった場合、準備しておいた内容は無駄になる可能性

が高い。そこで、できる限りすみやかに一斉発呼への準備のために保持していた様々なリソースを解放する必要がある。しかし、ここでも物理的なリソースを保持する必要がないというパケット交換の特徴から、解放処理にはある程度余裕を持つことができる。

従って、リソースの解放を行なう際には各リソースの解放処理が他の呼接続処理を妨げないために、優先度を落とし、ピーク時間帯からずらすようにして処理を行なうことが可能である。

## 5 考察

### 5.1 ACC ヒット率

本方式では、ユーザからの呼接続要求が前回と異なるか、もしくは呼接続要求がなかった場合にはそれまでに準備しておいた処理は全く無駄になるため、一斉発呼への準備が実際の呼接続要求と合致する確率を ACC ヒット率として定義し、この確率をできる限り高くする必要がある。

そのために登録段階で様々なユーザの発呼形態に対応した登録方法を考慮することが必要であるが、その他にもヒット率の低い端末に対してはサービスの提供を行わない等の考慮が必要となる。

### 5.2 ヒット率と ACC の効果予測

本方式を適用することによって、一斉発呼による呼処理バースト性がどの程度平滑化できるかを示すために、図 5 に一回の一斉発呼時の処理負荷の概念図を示す。

ACC, PCC での処理ステップ数の比較は、呼接続プログラムのインプリメントによって値が大きく異なる。a を IC パケット出力を除いた PCC 処理時間、

表 1: ACC、PCC の処理時間比較

	全体	一斉発呼到着時
PCC	$a + b$	$a + b$
ACC(ヒット)	$a + b + c$	$b + c$
ACC(ミスヒット)	$2a + b + c + d$	$a + b + c + d$

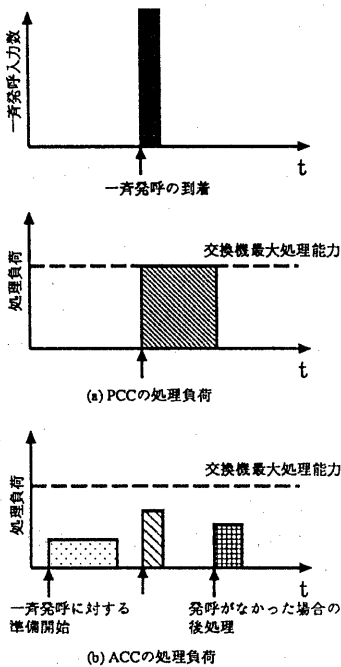


図 5: 一斉発呼時の処理負荷

$b$  を IC パケット出力の処理時間、 $c$  を CR パケットの比較に要する処理時間、 $d$  を CR パケットの再登録に要する処理時間とすると、各処理時間は表 1 のようになる。ここで、全体処理時間とは一斉発呼到着時の処理時間と ACC 準備処理時間の合計である。

3 項で述べたようにこれらの処理時間の間には一般的に  $a \gg b$ 、 $a \gg c$  の関係があること及び端末からの一斉発呼に対して直接問題となる接続遅延時間が表 1 の一斉発呼到着時処理時間に従うことから、ACC ヒット時は PCC よりも大きく減少すると思われる。X.25 パケット交換機の例では、ACC ヒット時のと PCC 時の処理時間比である  $(b+c)/(a+b)$  は 1/30 程度である。

ここで、ACC ヒット率を  $\alpha$  とすれば、一斉発呼時に、PCC 処理時間よりも ACC 処理時間平均値が小さくなるための  $\alpha$  の条件は

$$\alpha(b+c) + (1-\alpha)(a+b+c+d) < a+b$$

より、

$$\alpha > (c+d)/(a+d)$$

となる。

現在の X.25 パケット交換機処理から予測すると、 $(c+d)/(a+d)$  は約 1/50 程度にすることが可能であり、一斉発呼到着時の  $\alpha$  が本稿の前提のように大きければ十分適用可能である。また今後、フレームリレーなどのさらに複雑なプロトコルの普及にともなって  $a$  の値が大きく増加するため、この値はさらに小さくなると思われる。

今後は、ACC 準備を行なうための処理負荷と、接続要求がなかった場合の後処理負荷を考慮した考察を行なう予定である。

## 6 ACC 準備処理の削減

これまで述べた方式では一斉発呼が到着した時の最繁忙時負荷は平滑化することができる。しかし、表 1 で示すように準備段階まで含めた全体のステップ数が PCC よりも増加することは避けられない。そこで、ACC 準備の処理時間を短縮するために前回の ACC 準備結果を保存しておく方式が考えられる。準備結果とは、呼状態等を管理する呼制御メモリと呼接続処理の結果得られた IC パケットである。

この方式はプロトコル、プログラム構造に依存する部分が多く、CR パケットに加えて呼制御メモリ、IC パケットを記憶する必要があるため、必要メモリ量が非常に大きくなる。しかし ACC がヒットしている限り呼接続処理を行なう必要がないため、ACC 準備に要するステップ数を大幅に削減することが可能である。(図 6)

手順を以下に示す。

1. 1 回目の準備時にはこれまでに述べたように ACC 準備を行なう。ここで、ACC 準備処理を毎回行なう必要をなくすために、呼状態等を管理する呼制御メモリと、ACC 準備の結果得られた IC パケットを交換機内で記憶しておく。
2. 2 回目以降の ACC 準備段階では、本稿で述べたように実際の呼接続処理を行なうのではなく、1 で記憶した呼制御メモリ内容を実際に使用する呼制御メモリにコピーする。
3. ACC がヒットしている間は 1 で記憶した呼制御メモリと IC パケットを使用して ACC 準備を行なう。ミスヒットした時に前述の CR パケット

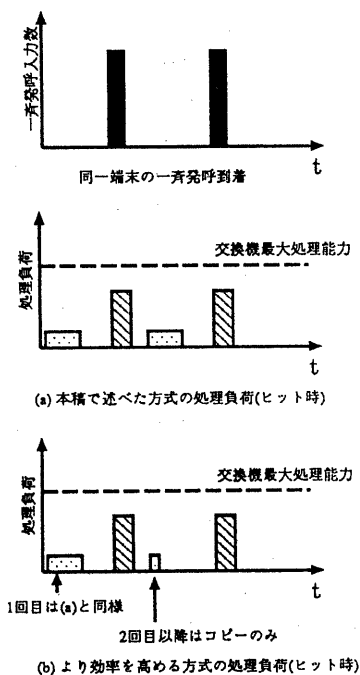


図 6: 処理負荷の比較

再登録を行なうのと同様に新規に呼制御メモリを記憶しなおすか、これまで使用していたものを使用するかは今後の課題である。

その他、本方式の課題として本稿で述べた ACC 方式に加え、以下のものがある。

- PVC サービスに近くなるため、今後普及が予想される IN 等の高度サービスのうち、提供できないものが出る可能性がある
- メモリの増加量

## 7 今後の課題

端末に特別な処理を行なわせることなく交換機の処理負荷を平滑化する方法を提案した。特に企業専用網などの比較的小規模なネットワークではこの方式の有効性が増すと思われる。本稿では実際のトラヒック状況を考慮した評価を行なうことができず、定性的な評価にとどまったが、今後は定量的な評価を含め、以下

の項目を課題として引続きより詳細に検討を行なっていく予定である。

- シミュレーション等によって ACC 準備まで含めた遅延時間等の予測を行なう必要がある。
- 今回はサービスの一部として位置付けたが、プライベート網等の交換処理能力が小さいものほど今後一斉発呼に対する対策が重要になると思われるため、あらかじめ登録することなく、端末の接続状況を監視することによって一斉発呼を検出するような方法についても考察を加える。
- 今回は現在大きな問題になっている一斉発呼の交換処理負荷を平滑化することを考えたが、多重度の低い端末にも本方式を適用することで最繁忙時間帯の処理負荷を全体的に平滑化することも可能であると考えられる。
- プロトコルに依存するシーケンスで本方式の適用可能な例としては、X.25 のリスタートがあげられる。必ずしも時間的な周期性はないが、全ての呼を一度に切断する必要があり、さらにその後で端末が一斉発呼してくる確率が高いことから、非常に交換機での処理負荷が集中する。ここにも ACC を適用することを考える。

## 参考文献

- 1) CCITT, RECOMMENDATIONS X.25
- 2) CCITT, RECOMMENDATIONS Q.931, Q.933
- 3) 日本電信電話株式会社, パケットサービスのインタフェース (PT 編), 日本データ通信協会