

通信網ワイドの交換サービスを実現するためのプログラム制御方式

久保田 稔 丸山勝己

NTT交換システム研究所

オブジェクト指向の考え方に基づく交換処理モデルを示し、本モデルに基づく交換プログラムの実行制御に対する要求条件を示す。交換システムが高い信頼性を要求される超多重実時間システムであることから、一般情報処理システムのOSと比較して要求条件の考察を行う。また複数の交換ノードが関与する通信網ワイドのサービスの最近の普及から、分散処理への適用についても考慮する。次にこれらの要求条件を満たすことにより、通信網ワイドの交換サービスを効率良く実現するために必要なOSの機能について述べる。具体的にはオーバーヘッドを極力押えたプロセス実行制御、ノード間メッセージ転送方式などである。

Execution Control Program for a Switching System to Support Network-Wide Switching Services

Minoru Kubota Katsumi Maruyama

NTT Communication Switching Laboratories

3-9-11, Midori-cho, Musashino-shi, Tokyo 180, Japan

This paper shows requirements for execution control of switching programs, which are implemented based on a object-oriented switching processing model. From the viewpoint that a switching system is a highly reliable, multiple and real-time system, the requirements are described by comparing the execution control program with OS for a ordinary information processing system. The execution control in distributed processing environment is also described considering that network-wide switching services realized by multiple switching nodes are becoming popular these days. Then we show the features of OS to implement these services efficiently, such as process management, inter-node message communication with less overhead.

1. はじめに

情報通信社会の進展に伴い、電話サービスへの要求はますます多様化している。このためインテリジェントネットワーク^{〔1〕}の検討が進められている。インテリジェントネットワークでは、従来の交換機が有していた機能を、データベースを利用する^{〔2〕}こと等により高度のサービス処理を制御する高機能レイヤと、基本的な接続処理を行う伝達レイヤに階層化する(図1)。階層化は機能の複雑さ、実時間性への要求等の観点から行われ、高機能レイヤから伝達レイヤを制御することにより、サービスの追加・変更の柔軟性を向上させることを狙いとする(図2)。フリーダイヤル(図3)やパーソナル移動通信はインテリジェントネットワークの一つのアプリケーションとみなせる。

高機能レイヤには、実時間要求の厳しいサービス処理を行うノード(NSP: Network Service Control PointあるいはSCP: Service Control Point)と、それほど厳しくないサービス管理機能を有するノード(NSSP: Network Service Support PointあるいはSMS: Service Management System)がある。伝達レイヤには高機能レイヤの指示に下がつて交換スイッチ等のハードウェアを駆動するノード(SAP: Service Action PointあるいはSSP: Service Switching Point)がある。

一方より効率的かつ高度にネットワークの保守を行うため、作業の保守センタへの集約が進められて

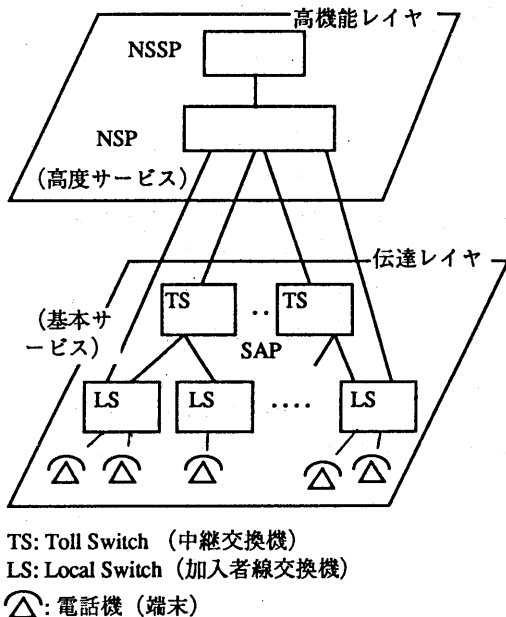


図1. インテリジェントネットワークのアーキテクチャ

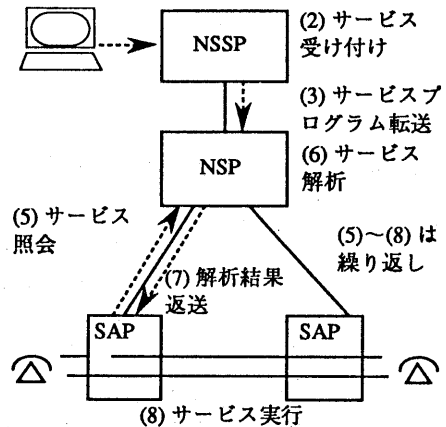


図2. インテリジェントネットワークにおけるサービス制御

いる。すなわち、これまで個々の局で行われていたサービス処理や保守作業が、ネットワークを介した分散処理環境で行われるようになってきている^{〔3〕}。

従来からも、異なる電話局に属している電話機(端末)を接続するため、各電話局間で信号をやりとりしながら処理を進めていた。1つの呼に複数の電話局(交換ノード)が関与するという観点から、電話網は世界的規模の超広域分散処理と言える。しかし送受される信号や情報は、基本的な接続処理を行うことが主目的で、伝送速度も十分でなく、遅延時間も大きいため、前に述べたような分散環境でのサービス、保守を行うためには十分でない。

光ファイバやATM技術により伝送速度の向上は期待できるが、プロトコル処理に関連する遅延時間はさらに少なくする必要があると考える。また局間の通信処理は、物理的な相手位置や通信プロトコルを意識して記述されることが多く、プログラムの流用性を損ねている。このためこれらを隠蔽し、より論理性的の高いプログラムの記述を可能にする必要がある。

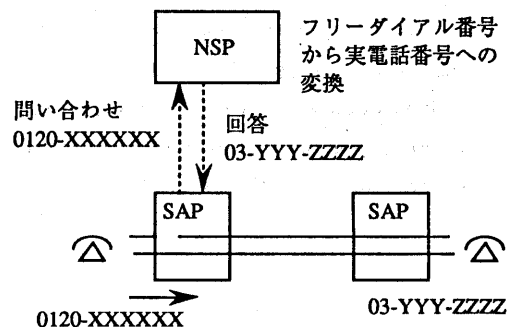


図3. フリーダイヤルサービス

将来の通信サービスでは、通信網ワイドに分散配置された交換ノード、データベース、サービス制御ノード、管理運用ノードなどが、協調してサービスを行う必要があり、高度の分散処理技術が必要である。また交換処理の特徴である超多重処理、実時間効率、高信頼性と安定性、常時成長性等をサポートする機能もあわせもつ必要がある。本稿では、これらの技術をサポートし、ネットワークワイドのサービスの実現を容易にするプログラム制御方式について考察する。

2. オブジェクト指向交換処理モデル

本稿で対象とする交換プログラムは、オブジェクト指向モデルに基づいたものである[4]、[5]。

2.1 交換システム要素とオブジェクト

オブジェクト指向交換プログラムでは、加入者線装置、交換スイッチ等の共通リソース、サービス分析データ、サービス実行制御といった、交換機の論理的要素をオブジェクトとする。オブジェクトはさらに自律的に並列に動作する並列オブジェクトと、並列オブジェクトに従属して動作する直列オブジェクトに分類する。すべての要素を並列オブジェクトとすることがモデルを簡単にすることから望ましい

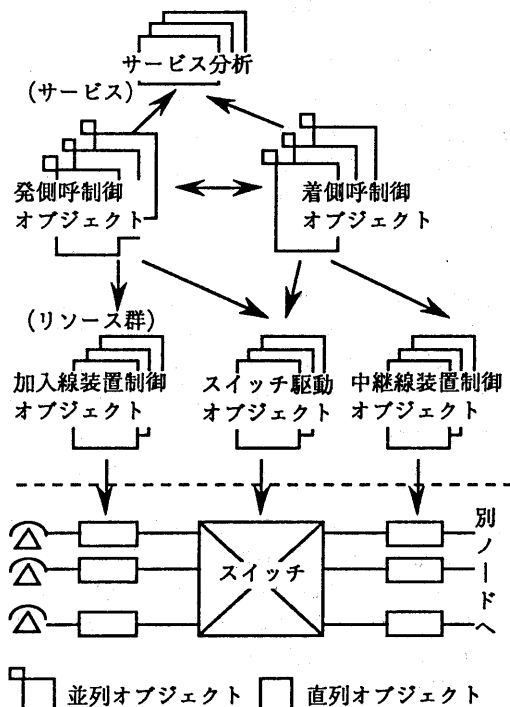


図4. オブジェクト指向交換プログラムモデル

が、オーバーヘッドの原因となる並列処理の単位を少なくする観点から、並列オブジェクトに従属して走行する直列オブジェクトを設けた。後で述べるように交換機も処理能力に対する要求が厳しいためである。また各オブジェクトはネットワーク内で一意に定まる識別子を持つことが可能で、ネットワーク内の他のノードのオブジェクトに直接メッセージを送ることができるものとする[6]。

本交換プログラムでは、図4に示すようなオブジェクトが協調して、電話の接続を行う呼処理を行う。発呼を検出すると、発呼側の端末を制御してサービスを実現する呼制御オブジェクトを割り付ける。これは加入者線装置、交換スイッチ、中継線装置等のリソースオブジェクトを用いて着呼側までのチャネルを確立し、また着呼側の端末を制御するサービス実行制御オブジェクトを生成する。呼のサービスは発呼側オブジェクトと着呼側オブジェクトの間でメッセージをやりとりし、それらがリソースを制御することにより進められる。

従来の交換プログラムでは、実行効率を重視していたためプログラムの構造化が十分でなかったが、オブジェクト指向交換プログラムでは、サービス制御を司るサービス階層、交換機の各種リソースを制御するリソース階層、さらにこれらの階層内のオブジェクトの実行を制御する実行制御階層に区別される（図5）。本稿ではこの実行制御階層に必要な機能について考察する。

2.2 オブジェクト間通信

オブジェクト間の通信について考察する。コンピュータ分野でのネットワークサービスはクライアント・サーバモデルをとり、通信プリミティブとしてリモートプロシージャコール（RPC）をとることが多い。これに対し、通信しあうオブジェクトが対等な関係にあるモデル（本稿ではパートナーモデルと呼ぶ）を考えることができる。以下に両者の両者の特徴を示す（図6）。

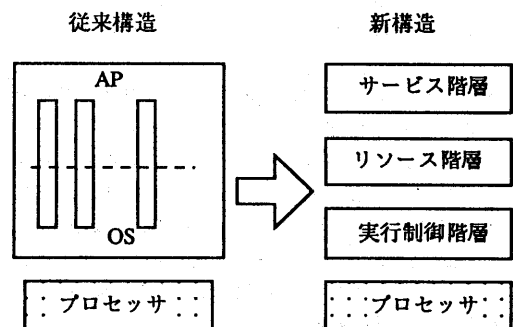
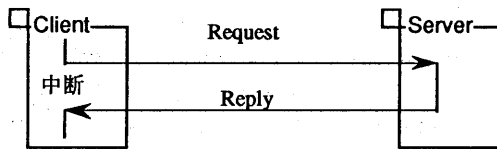
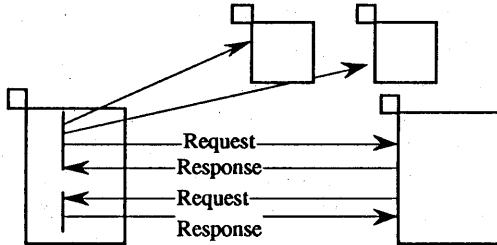


図5. 交換プログラム構造



(1) サーバ・クライアントモデル



(2) パートナモデル

図6. オブジェクト間通信モデル

(1) クライアント・サーバモデル

クライアントがサーバに処理を依頼する形になり、立場が対等でない。RPCが基本機能であり、通常の手続き型言語のセマンティクスを用いて記述できる。この場合、通信は同期的であり、クライアントとサーバの間で並列性が要求されない。

(2) パートナーモデル

通信しあうオブジェクトは対等な関係にあり、サーバ的な役割とクライアントの役割を合わせもつことが可能であり、多様な処理モデルに対応できる。通信を非同期に行えば両者は並列走行可能である。

交換機の接続処理においては、発信側と着信側の制御オブジェクトが互いに対等にメッセージをやりとりして処理を進める。また交換機では多重度が高いため、あるオブジェクトが複数のオブジェクトに同時に多数のメッセージを送る処理が多い。これを同期的に行うと、コンテキスト切り替えが増大し、オーバーヘッドが大きくなる。たとえば多数のリソースを同時に制御したり、端末からの要求を走査してまとめて呼制御オブジェクトに送るなどの処理がある。これらの点から考えて、交換処理では、オブジェクト間通信としてパートナーモデルが基本と考える。

ネットワークワイドの交換サービスでは、データベースアクセスなど、クライアント・サーバモデルが適切な処理もある。パートナーモデルでこれらの処理を記述するためには、メッセージ通信の多様なタイプ（例えば、ABCL^[7]における、now型、past型、future型等）をサポートする必要があるが、本稿ではパートナーモデルの基本通信形式であるpast型の非同期通信を検討対象とする。

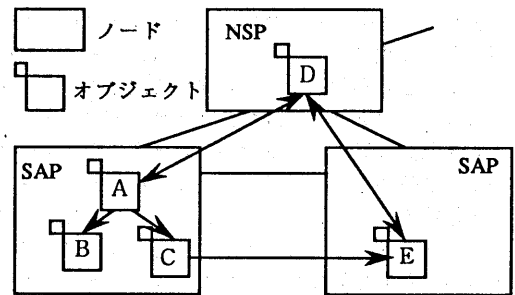


図7. 走行環境

2. 3 走行環境

本モデルによるプログラムの走行環境は、交換機能をもつノードが通信ネットワークで結合されたものである（図7）。ノードとしては加入者交換機他、前に述べたサービス制御ノード等が含まれる。オブジェクトは1つのノード上に存在し、別ノード上にあるオブジェクトに直接メッセージを送ることができる。

3. 交換システムの要求条件

3. 1 交換機単体として

一般の情報処理システムと比較すると、交換システムは次のような特徴がある。

(1) 高い多重性

一般の加入者線交換機においては、通常数万の加入者線及び数千の局間の回線を収容し、同時に数千の呼を処理している。前述したように一つの呼の発呼側と着呼側に並列オブジェクトを割り当てると、数千～数万の並列オブジェクトが同時走行することになる。

(2) 実時間性

数10ミリ秒から数秒以内に完了することを要求される処理が多い。例えば、電話の受話器をあげると500ミリ秒以内に発信音を返さなければならない。それぞれを単独に実現するのは困難ではないが、上記の多重性の条件を同時に満たすには工夫が必要である。

(3) 安定性・高信頼性

設置条件により異なるが、信頼度基準として交換機の不稼働率はおよそ1時間／20年が設計目標になっている。また高機能サービスの導入が進められているが、これらの障害が基本サービス処理部に影響を与えないことが必要である。

(4) 常時成長

サービスに対する要求が多様化しており、これに

対処するためプログラムの追加・更新が頻繁に行われる。サービスの継続性が強く求められており、プログラムの更新時にも、サービスを中断しないか、中断しても中断時間を最低限にとどめることが必要である。

上記の他に次のような事項についても考慮する必要がある。高い信頼性が求められるため交換機の部品として市販品をそのまま使えない場合がある。また交換機自体の数も全国で約8000ユニットで、そのうち更改されるのは多くても年に1000ユニット程度である。ワークステーションは年間数万台が出荷されており^[8]、その他のコンピュータまで含めると交換システムの製造台数は一般情報処理システムと比べて2桁以上少ない。マルチベンダ化によるコストの削減は一般情報処理システムほど大きくない、と考えられる。このため、極めて効率の良いソフトウェアで、コストパフォーマンスをあげる必要がある。保守・運用関係のソフトウェア等で性能条件の厳しくないものは流用できるが、呼処理関係のものは独自に開発しなければならない場合が多い。

一般情報処理システム（特にプログラム開発用）では、不特定多数のユーザがプログラムを開発し、走行させるので、誤ったプログラムがシステムを破壊しないように、システム側のガード機構が必須である。しかしガード機構の強化はオーバーヘッドの増大につながる。交換プログラムは一度作成されると長期にわたって使われるので、運用前の試験・検証を徹底することにより、オーバーヘッドの増大を招く過剰なガード機構より効率的な実行制御機構を優先すべきである、と考える。

3.2 ネットワークワイドサービスをサポートする観点から

(1) 多重性と実時間性の両立

前節で述べた多重性と実時間性に関する条件の一つのノードで実現するのは困難ではないが、複数のノードにまたがって処理を分担する分散環境においては（特にネットワークワイドの）通信遅延を極めて小さくしないと、数10ミリ秒～数秒の遅延時間内での処理の完了を実現することは難しい。たとえばダイヤル番号の解析のためにネットワークワイドのデータベースアクセスなどが伴う場合、ダイヤル終了後数秒後には接続を完了することがサービス基準として要求されるため、メッセージ通信の遅延を極力押える必要がある。

ネットワークワイドのサービスの普及に伴いさらに多数の呼を扱い、それに比例して信号（メッセージ）が送られるために、高いスループットを満たさなければならない。図8に示すような構成でパーソ

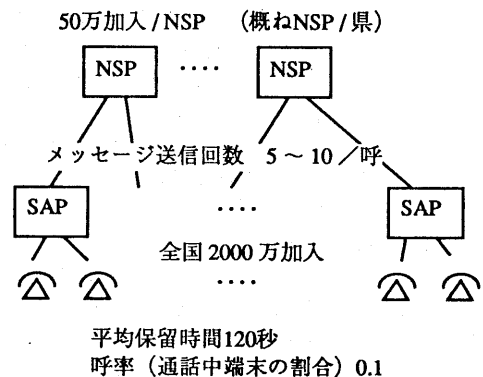


図8. パーソナル移動通信におけるNSP-SAP間メッセージ通信（見積もり）

ナル移動通信を実現する場合を例にとり、どの程度のメッセージ処理のスループットが必要か見積もってみる。図8に示す値は仮想的なものであるが、これを用いて計算すると次のようになる。

呼率0.1から1加入者で1時間あたり平均6分通話中になり、さらにこれを保留時間120秒で割ることにより、1時間に平均3回発呼することが求められる。加入者50万、5～10メッセージ/呼から、NSPが処理しなければならない1時間のメッセージ数は7,500,000～15,000,000となり、これを1秒あたりに直すとおよそ2000～4000となる。

従来と同様の接続遅延条件を満たすためには、NSP-SAP間のメッセージ転送遅延は、数ミリ秒～数十ミリ秒程度でなければならない。これに加えて極めて高い信頼性が要求される。

現在電話網で制御情報を転送するための通信網の物理的な速度は最大48kbpsであり、十分とは言えない。今後ネットワークワイドのサービスに関する呼数が増えるにつれ、より高速の回線、プロトコル処理が望まれる。

分散OSのRPCの性能を表1に示す^[9]、^[10]、^[11]。これらは測定条件が異なるので、単純な比較はできないが、通信速度のオーダが概ねどの程度か

表1. 分散OSにおけるRPC（リモート）の性能

OS	遅延時間 (ms)
Amoeba	1.4
Sprite	2.8
V	2.5
Sun RPC	12.2

ワークステーションSun3/75相当
Ethernet 10Mbit/sec
転送データ0～4バイト

を知ることができる。個々に見ると電話網にも適用可能な数値に見えるが、物理的なネットワークがEthernetであり、これを全国規模のネットワークにはそのまま適用できない。さらにすでに述べたように1秒間に数百という多数のメッセージを扱わなければならない、これに耐えるだけのスループットがあるかどうかは不明である。

(2) 安定性と常時成長

一つのノードの障害が、他のノードに伝播したり擾乱を与えない機構が必要である。インテリジェントネットワークでのサービスは、従来のものと比べて別ノードに依存する程度が大きいため、一つのノードの障害の影響範囲が大きくなると考えられる。別ノードの障害によりノード機能がすべてダウンするのではなく、機能を縮退させてサービスを続行させることが望まれる。

新サービスの導入により、新たにノードを加える場合が多くなることが予想される。新しいノードの追加が既存ノードに極力影響を与えない方式が必要である。

4. ネットワークワイドサービスをサポートする交換プログラム用OS

2節で述べたプログラム処理モデルに基づき、3節で述べた要求条件をサポートするために必要なOS機能について考察し、我々が検討を進めている方式について述べる。

4.1 プロセス制御

並列オブジェクトは、プロセスあるいはスレッドによって実現できる。交換プログラムでは多重性が高くかつ一つの並列オブジェクトが連続して走行する時間は短い。概算であるが、基本呼処理として約50Kステップ走行する間、入出力処理やプロセス間通信のため、コンテキスト切り替えが40～50回起こる。

交換プログラムでは高速コンテキスト切り替えの機能が必須である。これは一つのノード内の処理においても必要とされるが、分散環境においては、大量のノード間通信を扱う必要があるため一層重要となる。これを実現するため、プロセス(スレッド)の制御ブロックに格納する情報は極力少なくし、付加的な情報は別に扱う手法をとる。

プリエンブションの回数をなるべく少なくするなどスケジューラの工夫により、コンテキスト切り替えの回数を少なくすることも重要である。現交換プログラムでは、障害時を除いてプリエンブションが周期的にしか起きない。

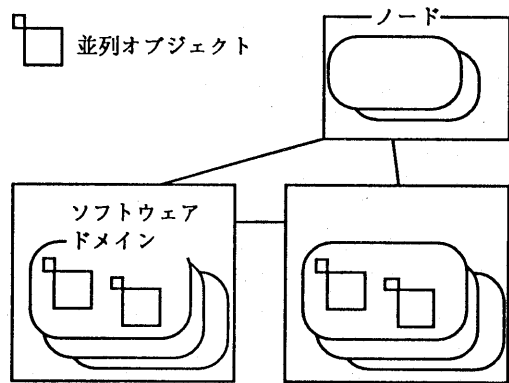


図9. ソフトウェアドメイン

4.2 メモリ管理

仮想記憶を用いて並列オブジェクトに対応するプロセス毎に論理空間を割り付けると、不当なメモリアクセスやプログラムの暴走等に対するプロテクション機構として有効である。しかし処理効率、所用メモリ量増大の観点から次のような問題がある。

呼制御を行う並列オブジェクトに必要なスタックのサイズは数Kバイトであり、1ページ4Kバイトとすれば1ページで十分である。各並列オブジェクトにページ変換テーブルを割り付けるのは無駄が多い。またコンテキスト切り替え毎に論理空間の切り替えが伴い、かつそれが頻繁に生じるため、TLB等のキャッシュの効果が薄れ、性能の低下につながる。またスワップが生じるとさらにオーバヘッドは大きくなる。オブジェクト用のメモリはメモリ常駐とする必要がある。

処理効率重視の観点から、並列オブジェクトの走行環境として実メモリ空間を使うか、多数の並列オブジェクトに一つの論理空間を割り付けるか、のいずれかが望ましいと考える。前者の場合スタックオーバーフローの防止のために、スタックの範囲を示すリミットレジスタのような機構を検討している。後者の場合は、リソース階層、基本サービス階層を論理空間(ソフトウェアドメイン)とし、高機能サービスを別論理空間とする。ソフトウェアドメイン単位に各モジュールに分散配置する(図9)。

4.3 周期処理

多重性が高いことから、交換プログラムではノード間の通信も含めた入出力要求の数は極めて多い。これらを個別に割り込みを用いて処理すると、割り込み処理を扱うためのオーバヘッドが大きく、著しく性能を低下させる。このため、割り込みをマスクしておき、周期的に入出力要求の有無を調べ、あれ

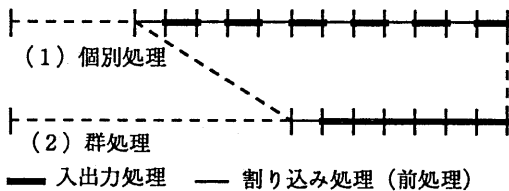


図 10. 個別処理と群処理

ばそれをまとめて処理する方式（群処理と呼ぶ。図 10）が必要となる。交換プログラム用 OS はプログラムを周期的に起動する機構をサポートする。

4. 4 通信制御

ノード間のメッセージ通信のためのプロトコル層として、(1)物理ネットワーク層、(2)ノード間通信層、(3)オブジェクト間通信層を考える。ノード間通信層は概ね OSI モデルのレイヤ 2、3、4 に相当する。前節で述べたように、ノード間のオブジェクトの通信では小さい遅延時間と高いスループットが同時に要求され、信頼性も保証しなければならない。このため次のような対策をとる必要がある。

(1) コネクションレス通信

接続、復旧のための遅延を防止するため、オブジェクト間通信はコネクションレスの通信機構をサポートする。

(2) メッセージサイズによる通信方式の切り替え

個々の呼処理において、ノード間で送られるメッセージの長さは短い（通常数 10 バイト、高々数百バイト）ので、頻繁に送られるメッセージを送るのに十分な長さの固定長メッセージにより、メッセージ処理のオーバーヘッドを防止する。また大量のメッセージに関しては、一度にすべてを転送するのではなく、必要になった時点で必要な部分だけを転送する手法をとる。

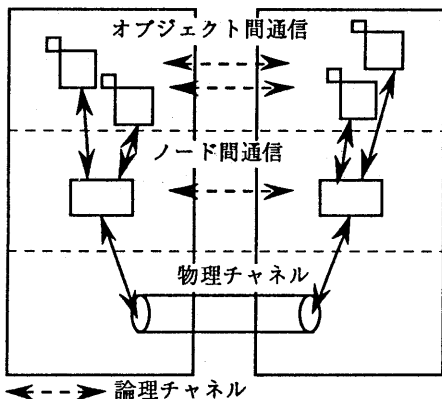
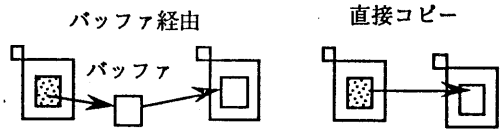


図 11. オブジェクト間通信

ノード内



ノード間（受信側）

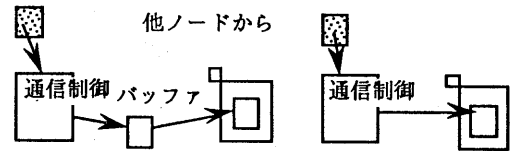


図 12. バッファ管理

(3) データコピーの削減

通信処理の各階層毎にバッファの内容をコピーしていくほうが、各階層の独立性が高まり、安全性が高くなる。しかし階層間でのバッファの内容のコピーはプロトコル処理におけるオーバーヘッドの大きな要因の一つである。交換プログラムにおいては、処理能力を優先して考えると、必要なバッファを最初に割り付けてそれを各階層で共通に使うことによりデータコピーの回数を削減する手法をとる。

またオブジェクト間通信において、通信用のバッファ経由で受信側のオブジェクトのデータ空間にコピーするのではなく、オブジェクト空間に直接データコピーすることによりデータコピーの回数を削減する（図 12）。

(4) 通信保証

通信保証（メッセージの到着確認、再送、順序制御等）をオブジェクト間通信層、ノード間通信層のいずれで行うかという問題がある。下記の理由から、ノード間通信層で行うのが適当と考える。

関連するノード間でのメッセージ更新回数は高くはないので、受信確認メッセージを返送する必要がある。多重度の高さからノード間では、頻繁にメッセージが送られるので、通常のメッセージに受信確認の役割を持たせることができる。このため、メッセージを送るパケット数が減少し、処理効率があがる（図 13）。また大群化効果のため、遅延時間も少なくなる。

この方式では、通信エラーについて、ノード間通信層で完全に隠蔽できない場合が問題となるが、確率が極めて小さいと予想される。より高い信頼性を必要とする通信を行うためには、送信毎に受信確認を行う手段も提供する。

(5) 分散隠蔽

プログラムの維持・管理性の向上のため、オブ

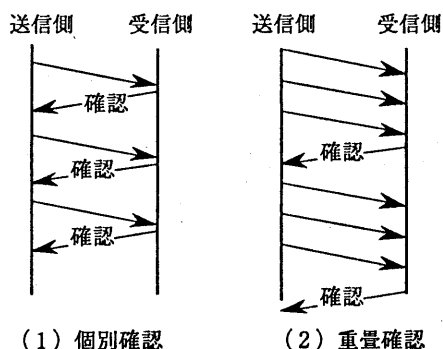


図13. メッセージの通信保証

ジェクト間通信はノード内かノード間にまたがる通信かを意識しないで記述できることが望ましい。このため、オブジェクトを指定する識別子から、オブジェクトが存在するノードを決定できる機構を提供する。

4.5 プログラム更新

サービスを中断することなく、プログラムを更新・追加する機構が必要である。更新・追加する単位としては、先に述べたソフトウェアドメイン、オブジェクトが考えられる。前者はサービスを中断させずに更新・追加するのが難しいが、ソフトウェアドメイン間はメッセージ通信のみによる疎な結合でかつ論理メモリ空間が独立しているため、置き換え・追加の機構は簡単になる。オブジェクト単位の置き換えはきめ細かな制御が可能であるが、置き換えによる影響範囲を局所化することが難しい。いずれにせよ、アプリケーションだけでこれらの処理を行うことは難しく、OSのサポートが必要と考える。今後の検討課題である。

4.6 高信頼性安定性

ネットワークワイドでの耐擾乱性強化機構（障害波及閉じ込めとプロテクション）、あるいは自己治療機構といったものがOSに要求されたと考える。今後の検討課題である。

まとめ

一般情報処理のOSと比較しながら、交換処理における分散処理まで含めた実行制御に対する要求条件を示し、通信網ワイドの交換サービスを効率良く実現するためのOSの機能について述べた。本稿では詳細に述べなかったが、ノード内の処理については上記のOS機能は実現されている。しかし、効率を最優先にしたため、呼処理部分との分離性がよく

なく、プログラムの維持管理性を損ねている。実行効率を低下させることなく、呼処理からのOS機能の独立性を高め、かつ分散処理への対応を図っていくことが重要課題となっている。現在これら機能の実現手法の基本的な検討とともに、これらを実証するためのシミュレーション環境^[12]の構築を進めている。これらの検討結果を元に交換処理における最適な分散制御方式について明らかにする予定である。

謝辞

本検討を進めるにあたって有益な意見、コメントをいただいた当研究所伝達ソフトウェア研究部の富田研究部長、田中、大崎両社員に感謝します。

参考文献

- [1] 青木他：インテリジェントネットワーク、信学会誌、Vol.73、No.7、pp.737-743、1990.7
- [2] 水沢：データベースを用いた新電話サービス、信学会誌、Vol.71、No.3、pp.251-254、1988.3
- [3] 山下他：トータルオペレーションを支えるネットワークアーキテクチャ、NTTR&D、Vol.38、No.12、pp.1499-1508、1989
- [4] 甲斐他：オブジェクトモデルによる交換プログラム構成、信学技報、SE87-147、pp.1-6、1988.1
- [5] 渡部他：実時間オブジェクト指向交換プログラム構成の考察、信学技報、SSE88-83、pp.25-30、1988.7
- [6] S. Tanaka et.al.: Execution Control for Distributed Object-Oriented Switching Programs、信学技報、SE89-107、pp.71-76、1989.10
- [7] 米澤：並列オブジェクト指向言語ABC L/1による並列処理記述とその枠組みの研究、信学会論文誌、Vol.J71-D、No.8、pp.1415-1422、1988.5
- [8] 日本情報処理開発協会：情報化白書、1990
- [9] D.R. Cheriton: The V Distributed System、Comm.ACM、Vol.31、No.3、pp.314-333、1988
- [10] A.S. Tanenbaum et.al.: Amoeba, A Distributed Operating System for the 1990s、IEEE、Computer、May 1990
- [11] J.K.Ousterhaut et.al.: The Sprite Network Operating System、IEEE、Computer、Feb. 1988
- [12] 田中他：分散処理交換システムのシミュレーション環境の実現、第39回後期情処全国大会