

HNAにおける分散処理計算機向け通信管理

プログラムの処理方式

古屋正和^{*}, 柳生和男^{*}, 池場悟郎^{**}, 櫻尾次郎^{*}

^{*}(株)日立製作所システム開発研究所, ^{**}同ソフトウェア工場

1. はじめに

オンライン適用分野の多様化, インテリジェント端末, 分散処理用小形計算機の普及といった最近の動向は, 回線や端末を複数のオンライン業務, T&S, 複数のホスト・コンピュータで共有したいなどのニーズを強めた。このようなニーズに対処するため, HDLC(ハイレベル・データ・リンク制御), パケット交換など最新の通信技術を取り込み, 日立ネットワーク・アーキテクチャ" HNA"(Hitachi Network Architecture)としてオンライン・ネットワークの体系化, 標準化を先に行なった。

"HNA"の利点は, ネットワーク資源(回線, 端末)の有効利用, ネットワーク構成からの独立性, オンライン業務の追加, 変更の容易性などである。

本論文は, 特に最近の分散処理に対するニーズに対処するために, HNAに基づいて開発した分散処理計算機(サブホスト)用通信管理プログラム"HI COP"(Hitachi Network Architecture Subhost Communication Program)について, その設計思想とインプリメント上重要であるソフトウェア方式, フロー制御とバッファ・マネジメント方式及びインプリメントしたプロトコルの検証方式について報告する。

2. HNAプロトコルのサポート

2.1 サブホストに要求されるネットワーク機能

まず分散処理計算機を用いたネットワーク・アプリケーションについて概観する。図1に主なアプリケーションのデータの流を示したが以下この図に従って説明する。

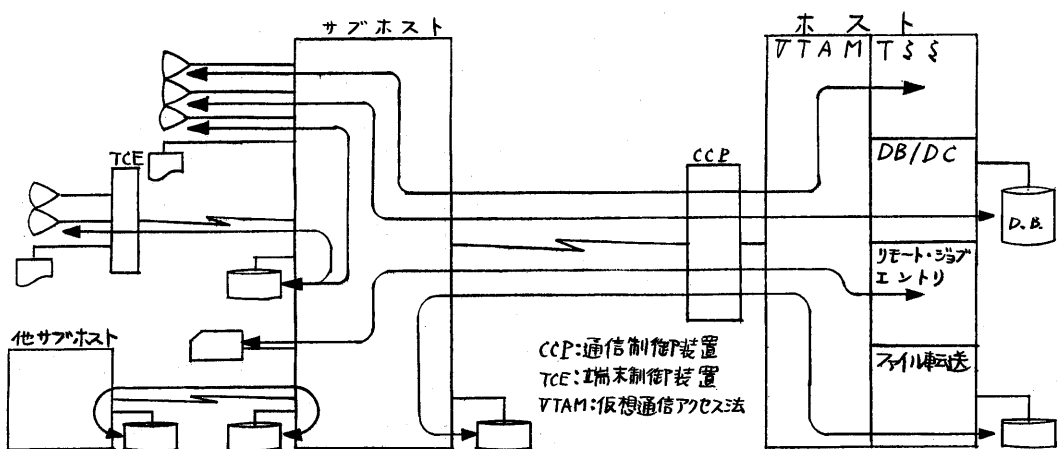


図1 サブホストにおける通信形態

(1) ホストとの通信

(a) リモート・ジョブ・エントリ

大規模な計算処理等をホスト計算機で行なえるよう、サブホストから直接ホストへジョブを転送する。この機能によりホストの処理能力をサブホストから活用できる。

(b) ファイル転送

サブホストでローカル処理したデータ等を一括してホストに送信したり、ホストで処理した結果等のファイルをサブホストに転送する。本機能によりサブホストとホストとの間の処理の分割が容易になる。

(c) アプリケーション間通信

支店、部門固有のファイルやデータ・ベースの間合せはサブホスト内で行ないながら、時にはホストにある全社的なデータ・ベースの間合せをしたときなど会話モードの通信を行う。本機能により処理及びファイルやデータ・ベースの分散が可能となる。

(d) 端末データのパススルー

サブホストに接続するCRT、プリンタがあたかも直接ホスト計算機に接続されているように見せかけてホストと通信を行なう。この機能によりホスト計算機のDB/DCパッケージ・プログラムやTSSと直接交信できる。

(2) 配下の端末との通信

会話式データ・エントリやローカルファイル、データ・ベースのアクセス及びジョブ起動等のコマンド投入などの交信を行なう。

(3) 他サブホストとの通信

他の支店や部門とファイルなどの交換を行なう。

2.2 HNAによるサブホスト通信機能の実現

(1) HNA階層構造の概要

図2にHNAの階層構造を示す。

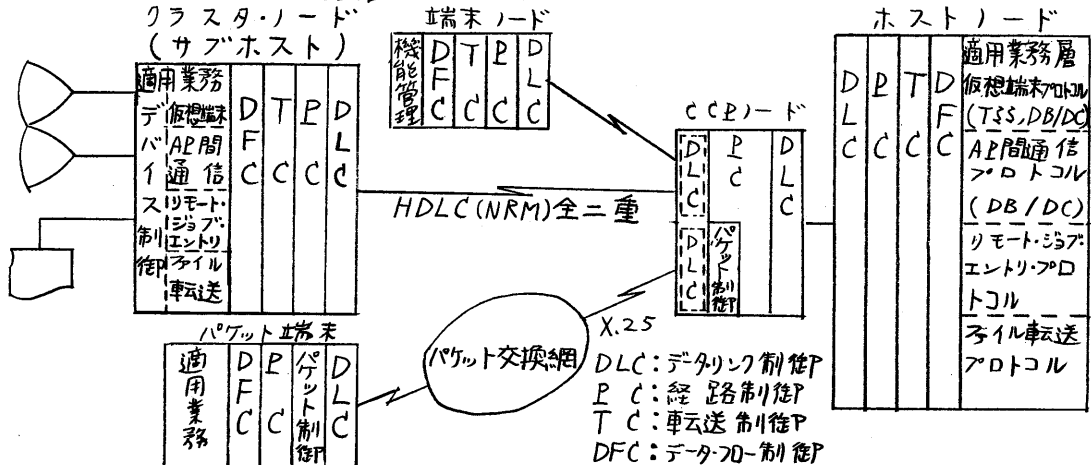


図2 HNAの階層構造

以下各階層の概略を説明する。

(a) DLC (データ・リンク制御)

ISO (国際標準化機構) 及 JIS (日本工業規格) で標準化が進められている HDLC (ハイレベル・データ・リンク制御) に準拠した HDLC を採用している。

HDLC は回線の有効利用を図るため両方向同時通信, 情報の連続転送を可能にし, すべての伝送データに対してチェック情報を付加して, 回線上のエラー検出の機能を強化している。

(b) PC (経路制御)

送信データを回線に出力する単位の長さ(セグメント)に分割したり, ブロッキングを行なう。

またデータ単位に付加されているネットワーク・アドレスを見て振り分け処理を行なう。

(c) TC (転送制御)

送信データのシーケンス番号をチェックしてデータの抜けを監視したり, エンド・ユーザのデータ処理速度と同期をとってデータを送るよう, データの転送を制御する等の制御を行なう。

(d) DFC (データ・フロー制御)

同時に双方のエンド・ユーザがデータを送られた場合 (コンテンツ) の解決を行ったり, 互いにデータの送信権を交換し合うルールを規定する等エンド・ユーザ間のデータ・フローの制御を行なう。

(e) 適用業務層

本階層には適用業務毎に各々プロトコルが定められている。例えば HNA 内の標準端末プロトコル (VTP; 仮称), ファイル転送プロトコル, RJE プロトコル, アプリケーション間通信プロトコル等が規定される。

(2) サブホスト (分散処理計算機) における HNA サポート

図 3 にサブホストの通信機能をサポートするソフトウェア構成を示す。

(a) HICOP (Hitachi Network Architecture Subhost Communication Program)

本プログラムは図 2 に示したクスタ・ノードの DLC, PC, TC, DFC (一部) をサポートする。DFC 層の全規定項目をインプリメントしない理由は, 適用業務によって DFC プロトコルのサブセットの切り出し方が異なり, フルセットをサポートすることが困難なことによる。

本プログラムは上位に位置するプログラムに対しセッション (論理リンク) の開始/終了, データの送受信等の要求マクロを提供する。本プログラムによってホスト及び他のサブホストとの通信が可能となる。

データの流の観点からは, 本プログラムがデータの蓄積伝送 (Store and

forward) 機能を持っている。すなわちHICOPの上位プログラムからの送信データは一担バッファに蓄積され、送信可能状態になったとき回線に出力される。よってランダムに発生する会話型トランザクションを平滑化して回線に出力する等の機能があり、高価な回線リソースを有効に活用する。

(b) HICOP ユーザ・プログラム

HICOPを用いてホストとの通信を行なう、HNA適用業務レベルのプログラムには以下のものがある。

○ 9415DSC (Data Stream Compatibility)

本プログラムによってサブホストに接続する端末が、直接ホストに接続するようホストのVTAMユーザ・プログラムに見せかける。本プログラムはサブホストに接続する非HNA端末を、プロトコル変換することによってHNA標準端末に見せかけている。

○ HIFIT (High level File Transmission Program)

ファイル転送を行なうプログラムである。サブホスト、ホスト相互からファイルの送信及び受信要求を発行できる。またデータの圧縮や、ファイルを送信中障害が発生した場合、続きから転送を再開できるリスタート機能などを持っている。

○ RESP (Remote Batch Station Program)

リモート・ジョブ・エントリを実現する。

○ CUTE (Customer's Terminal Equipment Support)

データ・エントリ、問合せ応答等が簡易な言語でプログラムできるオンライン・パッケージであり、アプリケーション間通信プロトコルをサポートしホストDB/DCパッケージプログラムと交信する。

本プログラムを用いてホスト計算機のデータ・ベースを検索したり、ホストからサブホストのデータ・ベースにアクセスすることができる。

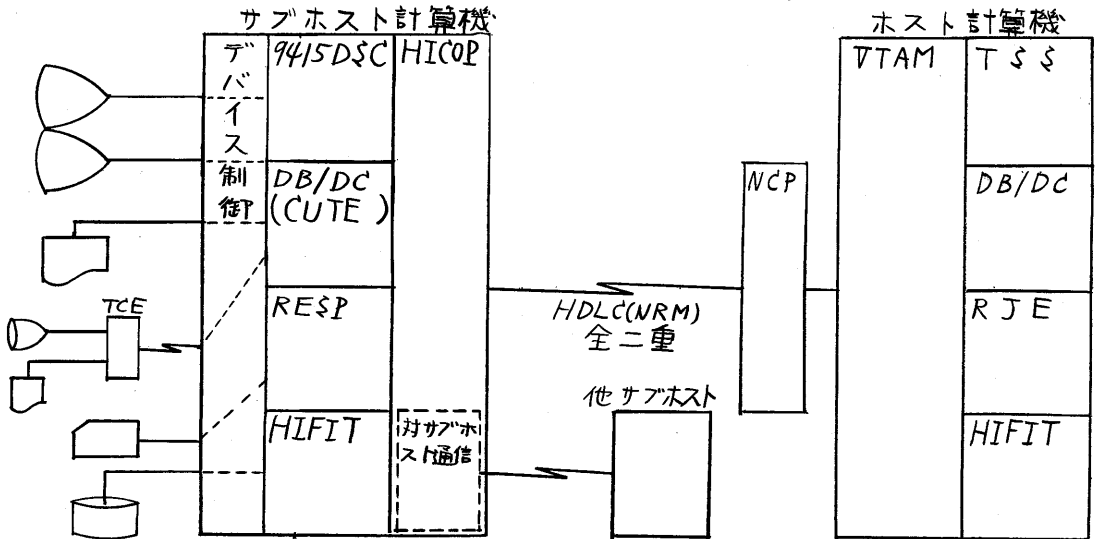


図3 サブホストにおけるHNAサポートソフトウェア

3. サブホスト通信管理プログラム(HICOP)のインプリメント

3.1 ソフトウェア方式

(1) 設計方針

以下に示す設計方針をした。

(a) OSの仮想化

HICOPの各プロトコル処理モジュールが直接OSを意識しなくても良いようにする必要がある。これはサブホストに位置する計算機のOSを複数設定しており、一部のモジュールの修正によってこれらのOSに適用できるようにするためである。

(b) 拡張性の考慮

新データ網への加入等によりパケット制御層の追加, DLCの変更等の機能追加, 修正に対処する必要がある。

(c) ソフトウェアの高信頼化

非同期に発生するトランザクションを扱うためタイミング障害が発生しやすい。また複数のユーザ・プログラムが同時にHICOPに対して送受信要求を発行するが1つのユーザ・プログラムが異常終了しても他に影響を与えない方式が要求される。

(2) 処理方式

上記の設計方針を実現するためHICOPではPSS (Process Schedule Service) 方式を採用した。PSS方式とは図4に示すように、汎用OSの下にトランザクション処理向けの一種のOS (PSS) を設定し、その下でプロトコル処理を行なうプロセス (一種の疑似タスク) を稼働させる方式である。

プロセスは図5に示す状態遷移図に従ってPSSに制御される。例えばユーザ及び回線からトランザクション処理要求が発行されるとマクロ受付ルーチン及び割込分析ルーチンから該当するプロセスに対しHISCHEDマクロを発行する。

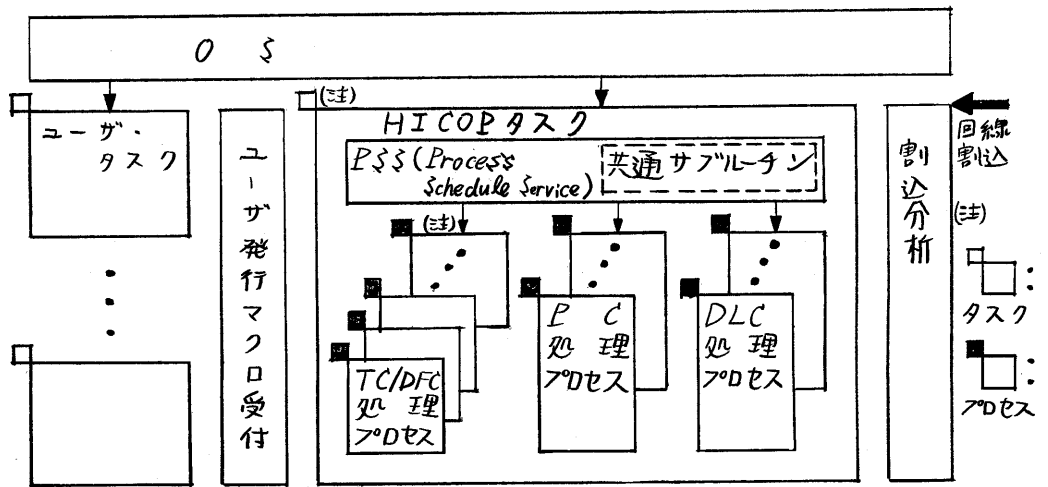


図4 HICOPのモジュール構造

このマクロはプロセスの状態管理テーブルを処理要求キューにチェーンする(スケジュール状態)。HI COP タスクが POST されると P S S はスケジュール状態のプロセスをキューからとり出し、順番に制御を移す(ディスパッチ状態)。ディスパッチされたプロセスはトランザクション処理を開始し、終了すると HI EXIT マクロを発行して P S S に終了したことを知らせる(アイドル状態)。P S S は次のディスパッチ状態のプロセスをとり出しに行く。もし無ければ WAIT する。プロセスも、もしバッファが取れない等の理由で処理を続行できなくなったと HI WAIT マクロを発行して WAIT できる(ウェイト状態)。

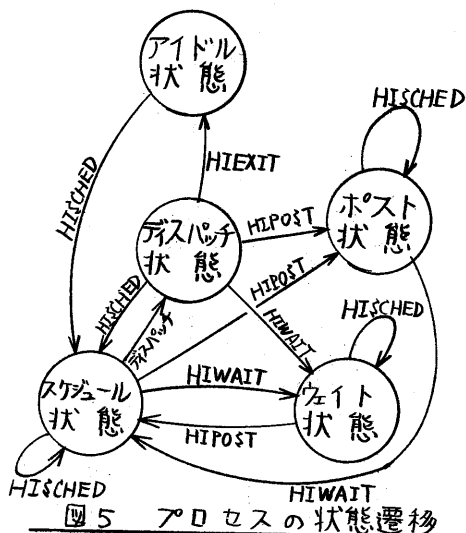


図5 プロセスの状態遷移

以上の P S S マクロは割込出力ルーチン等からも発行されるから処理の追起し等に十分注意を払う必要がある。HI COP ではこのような処理の競合を P S S マクロレベルで解決しているので、プロセスのコーディングは処理の競合を意識しなくて良い。

以上 P S S 方式の特徴をまとめると次のようになる。

- (a) プロセスは直接のインタフェースを意識しなくて良い。
- (b) プロセスの追加、変更は他のプロセスに影響しないため拡張性に優れる。
- (c) 非同期に発生するトランザクション処理要求をシリアライズし、タイミングバグの発生を防止している。このためプロセスは処理中の割込み等非同期処理を意識しなくてよい。
- (d) ユーザ・タスクの延長で走る部分が少ないので、1つのタスク異常が他へ影響することを防止しやすい。
- (e) 1つのタスクでトランザクションの並列処理ができる。

3.2 フロー制御とバッファマネージメント

(1) 論理リンク(セッション)の優先制御

前節で示したように HI COP はリモート・ジョブ・エントリ、ファイル転送のように大量のデータを送受信するタイプのセッションと T S S、問合せ処理のような会話形トランザクションが流れるセッションを同時に扱い、これを1本の物理回線に乗せる。このためバッチタイプのフローが問合せタイプのフローのレスポンスタイムを低下させないようなフロー制御が必要となる。

HI COP では HNA の全階層(転送制御、経路制御、データ・リンク制御の各層)を通じ、問合せタイプのトランザクションがバッチトランザクションを追い越すよう、セッションを問合せ型とバッチ型に分け、フローの優先制御を行なった。

図6に優先制御方式の概要を示す。まずセッションを HI COP のゼネレーション時に問合せタイプとバッチタイプに分類しておく。ユーザからの送信要求があると HI COP では転送制御層のプロトコル処理を行なった後経路制御

層においてセッションタイプをチェックし、データ・リンク制御層に優先度パラメタ付き送信要求を発行する。データ・リンク制御層では送信データを優先度別にキューしておき、送信可状態になったら優先度の高い問合せデータから回線に出力する。

本方式による送信遅延時間（ユーザから送信要求が発行されてから回線への出力が完了するまでの時間）短縮の効果を図7に示す（ただし問合せトラフィック量が少ない場合）。

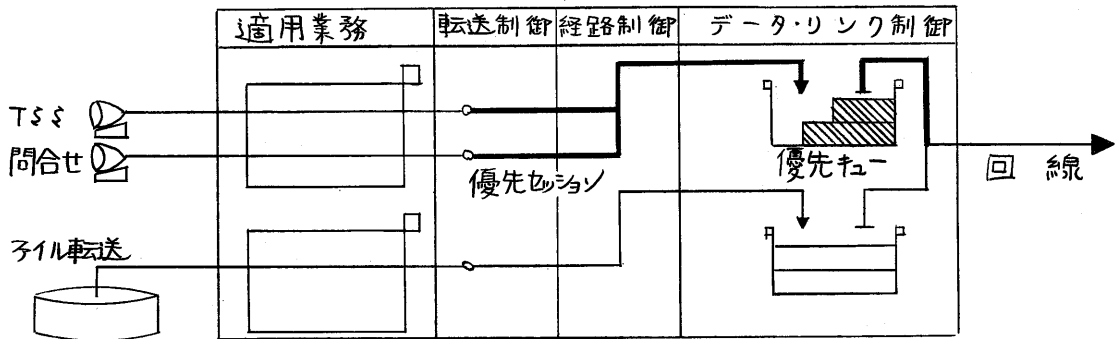


図6 優先制御方式の概要

(2) バッファ・マネージメント

HICOPでは信頼性向上及び処理の単純化のため送信データは一担システムの用意するバッファに移してから回線に出力する。

また受信データもユーザからの受信要求と実際のデータのとりこみが非同期であることから一担システム内のバッファにとりこむ。このときバッファの使用効率を向上させるため送受信とも同レベルとした。

本方式を実現する上での問題点

として、バッチタイプのセッションがバッファを使い切ってしまうことの防止とデッドロック対策の2つがある。

HICOPでは優先度別バッファ使用制限と送信専用バッファの設定を行なってこれらの問題に対処した。

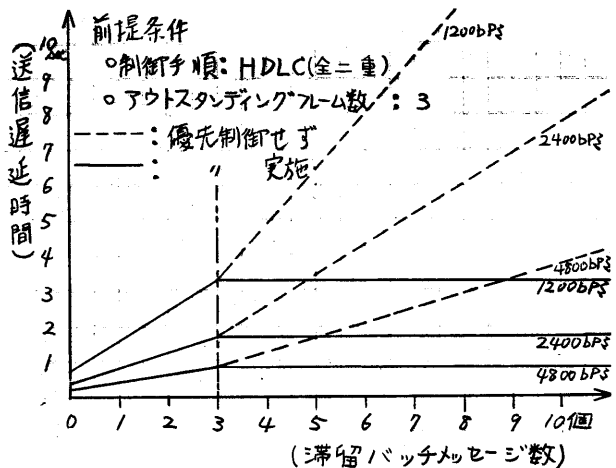


図7 優先制御方式の効果

3.3 プロトコルの検証

できあがったプログラムがプロトコルとして規定した通り動作するかどうか確認することをおプロトコルの検証という。ここではHICOPの開発で行なったプロトコルの検証方式について説明する。

(1) プロトコル検証の基本方針

- (a) 各階層別に独立して検証を進められること。
- (b) 検証対象プログラムの変更部分を極力なくす。
- (c) 全規定項目を順序立ててチェックできること。

(2) プロトコル検証ツール

上記基本方針に従ったプロトコル検証ツールを図8に示す構成で作成した。

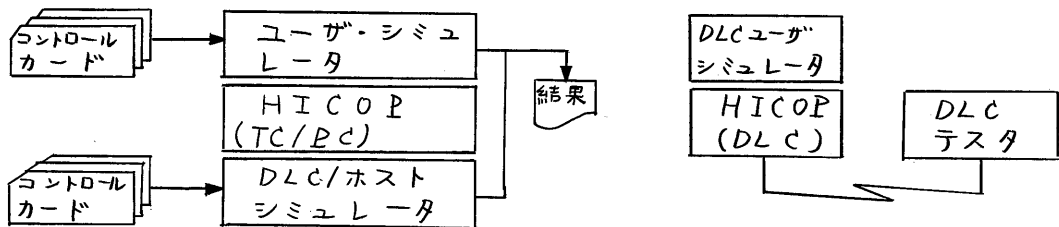


図8 プロトコル検証ツールの構成

この図に示すようにデータ・リンク制御層と経路制御/転送制御層は分けて検証を行なう。この理由はTC/PCとDLCインタフェースが単純であること及びDLCはテストに回線が必要でありテストマシンが限られるため独立して検証ができた方がよい事等である。

以下プロトコル検証ツールの機能を示す。

- (a) 任意のデータフォーマットの作成。
- (b) 動的なシステムの任意の状態のセット、リセット機能。
- (c) 任意の時点でシステムの状態及びデータのリスト処理。
- (d) データの送信/受信 タイミングの制御。

これらの機能により各階層に存在する有限状態マシン(状態遷移図によって表現されるプロトコル処理モジュール)の状態遷移をそれぞれ独立に実現できる。

4. おわりに

HICOPは分散処理計算に要求される通信機能をHNAに基づいて実現したものである。HICOPはHDLC(全二重)を採用し、バッチデータを問合せ型データの間隙を抜って送る(優先制御)等の処理によって回線リソースを効率よく活用できるようにした。またHNAの上位プロトコルをサポートするソフトウェア群によって処理、ファイルの分散が容易に実現でき、ホストの計算能力やデータ・ベースを活用することも可能である。

今後さらに分散処理ニーズに合わせて機能を強化して行く予定である。

<参考文献>

- (1) 櫻尾 他: 日立ネットワーク・アーキテクチャ "HNA" 日立評論1978-12
- (2) 古屋 他: コンピュータ・ネットワークにおけるフロー制御とバッファ管理の手法について, 情報処理学会第20回全国大会