

システム高速再開における端末無中断方式†

川原 洋人** 柴 垣 齊**
仲 谷 元** 大 石 和 寛**

本論文では、汎用大型計算機であるホストと前置通信制御処理装置 (FEP) から構成されるオンラインシステムの、高信頼化のための手法である端末無中断方式について述べる。端末無中断方式は、ホスト障害発生時の予備系への高速切替をベースとし、以下の処理を行うものである。(1)予備系ホストへの切替中も FEP で端末とシステム間のコネクションを保持し、FEP までの電文受信とその電文の保留を可能とする。(2)予備系ホストでサービスを再開時に FEP で保持した端末とのコネクションをホストへ再接続し、保留した電文の処理を開始させる。本稿ではこれらを実現するための、ホスト-FEP 機能分担のあり方、予備系ホストでのコネクションの引き継ぎ法について論ずる。また、典型的なバンキングシステムを例にとり、端末無中断方式の効果の評価法を与える。また、大規模オンラインシステムモデルの数値計算例により、大部分の端末については、ホスト障害時にもそれを意識させないことが可能であることを示す。

1. はじめに

オンラインシステムの大規模化に伴い、そのセンタの障害は社会的な大きなインパクトを与える。このためシステム高信頼化技術が種々検討され、実用のシステムに利用されつつある¹⁾。なかでもミニコンをベースとする専用メカニズムを備えたフォールトトレラントコンピュータ (FTC) の商用化が進みオンラインシステムを中心に利用が拡大している^{2), 3)}。一方、銀行のオンラインシステムのように大規模なファイル入出力を伴うシステム分野は、まだ大型の汎用機の守備範囲であると考えられている³⁾が、前述の FTC のメカニズム⁴⁾をそのまま適用するには無理があり、大型システムに適した高信頼化対策が必要である。当研究所では高性能・高信頼を目的とし、汎用大型コンピュータ、および前置通信制御処理装置 (FEP: Front End Processor) を組み合わせた大規模な複合構成システムを実現した⁵⁾。本システムにおいては

- (1) ホスト障害発生時に予備系ホストへ高速に切り替える技術 (システム高速再開)
- (2) ホスト障害時に FEP で端末とのコネクションを保持し、端末から見てホスト障害を見せない技術 (端末無中断方式)

を実現している。端末無中断方式は、利用者から見たオンラインシステムの信頼性をいっそう向上させるた

めに有効な技術である。

本稿では端末無中断方式の構成技術、および実システムに適用する場合の評価法を述べ、大規模オンラインシステムをモデルとし、評価例を示す。

なお、本方式では端末側の特別な機能、端末-センタ間のプロトコルについては何も仮定しない。例えば、利用者にホスト障害を意識させない方式として端末を高機能化する方法もあるが、本方式では端末側にそのような機能を必要としていない。

2. 高速再開方式の概要

2.1 複合構成システムの概要

本方式のベースとなる複合構成システムの概要を図 1 に示す⁵⁾。高速光ループ (100 Mbps) を媒介にシステム全体の運転を司るシステム制御プロセッサ (SCP: System Control Processor), 汎用計算機からなるホスト, 通信制御処理装置からなる FEP, およびそれらの予備が接続可能で、それらすべて間で相互通信を可能としている。システム内ではホット予備方式が採用され、障害時には SCP が提供する自動運転機能により予備への切換えが行われる⁶⁾。予備ホストは現用ホスト n 台 ($n \leq 3$) につき 1 台置くことができ、現用ホスト障害時に共用ファイル経由で、現用ホストの処理内容を引き継ぐことができる。

2.2 ホスト障害時のリカバリと高信頼化施策

障害の発生したホストから予備ホストへの処理の切換えは、図 2 に示す四つのフェーズからなる。

複合構成システムではシステムの信頼性を高めるために、

† A Method to Keep Terminals in Service during System Reconfiguration by HIROTO KAWAHARA, HITOSHI SHIBAGAKI, HAJIME NAKATANI and KAZUHIRO OISHI (NTT Communications and Information Processing Laboratories, Nippon Telegraph and Telephone Corporation).

** 日本電信電話(株)NTT 情報通信処理研究所

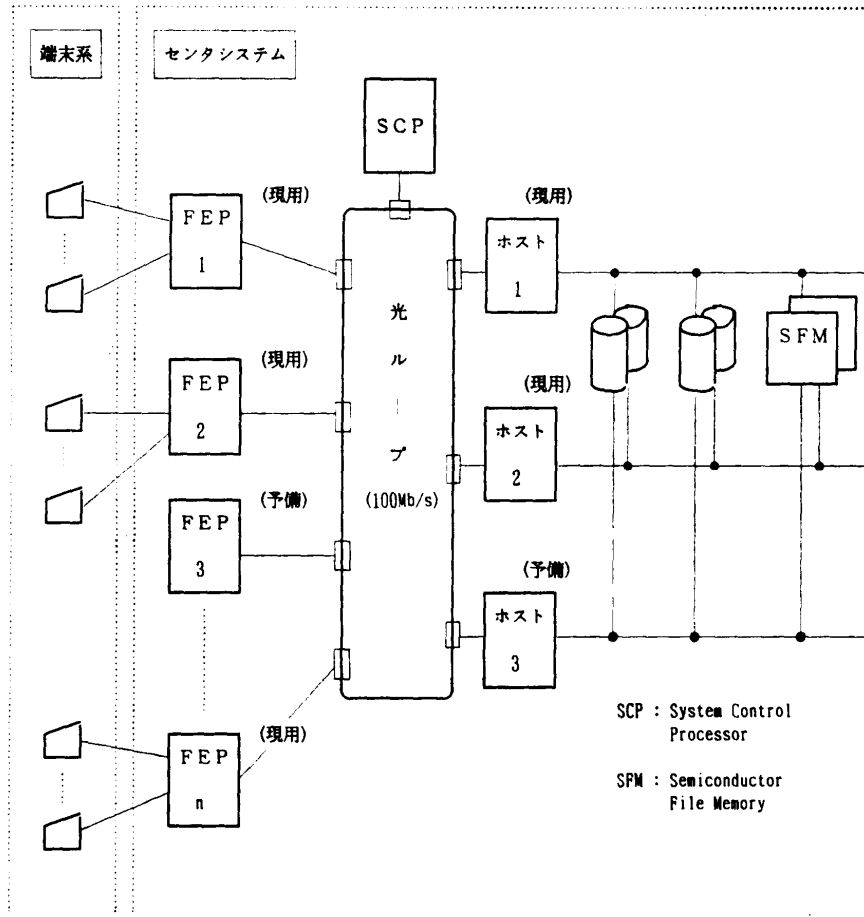


図 1 複合構成システムの概要

Fig. 1 Outline of a computer complex system.

- (1) 上記リカバリ処理をできるだけ高速に行うこと
- (2) リカバリ処理中に端末にホスト障害を意識させないこと

を目標としている。フェーズ1～3の時間短縮については、文献 5)～7) に述べられている。本稿ではフェーズ4のみ対象とする。

2.3 端末無中断方式の概要

従来の方式では、ホスト障害時にはそのホストに収容されている端末とのコネクションは切断され、予備ホストでの処理引き継ぎ後、端末とのコネクションが新たに再接続されるのが一般的である。端末とのコネクションが切断された時は、端末ではアラームランプ点灯等により、利用者にシステムの障害が見える。バンキングシステムに例を取れば、店舗担当者がお客様への対応を迫られる、またセンタオペレータが問合せへの対応に迫られるなどの実務上の混乱を招く。

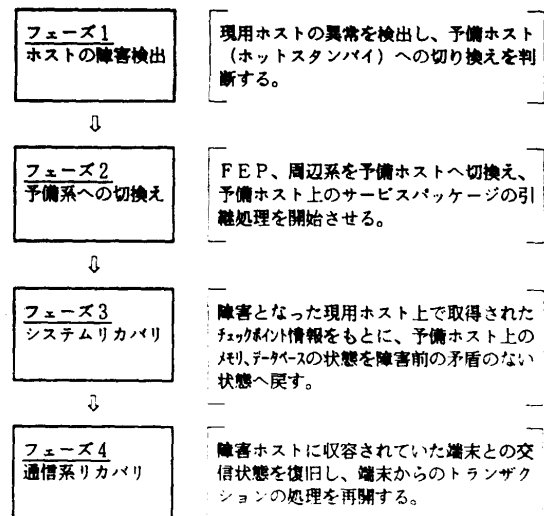


図 2 障害時のシステム復旧処理の概要

Fig. 2 Outline of the system recovery process.

端末無中断方式の概要を図3に示す。本方式では、ホスト障害中も端末からFEPまでのコネクションを保持、FEPまでの電文受信を可とすることにより、以下の効果を狙う。

(1) 端末から見て、ホスト障害中もサービスが継続しているかのように見せ、端末側の混乱を最小限に抑える。(多少応答時間は遅延する。)

(2) 端末との回線の再オープンおよびコネクション再設定のための処理を省略することにより、フェーズ4の通信系リカバリ時間を短縮する。

ホスト障害中、FEPで保留された電文についてはホスト切換えを迅速に行い、端末から見た応答時間を許容範囲内におさめることを目標とする。(端末側では応答時間監視を実施し、一定時間(厳しい場合で60

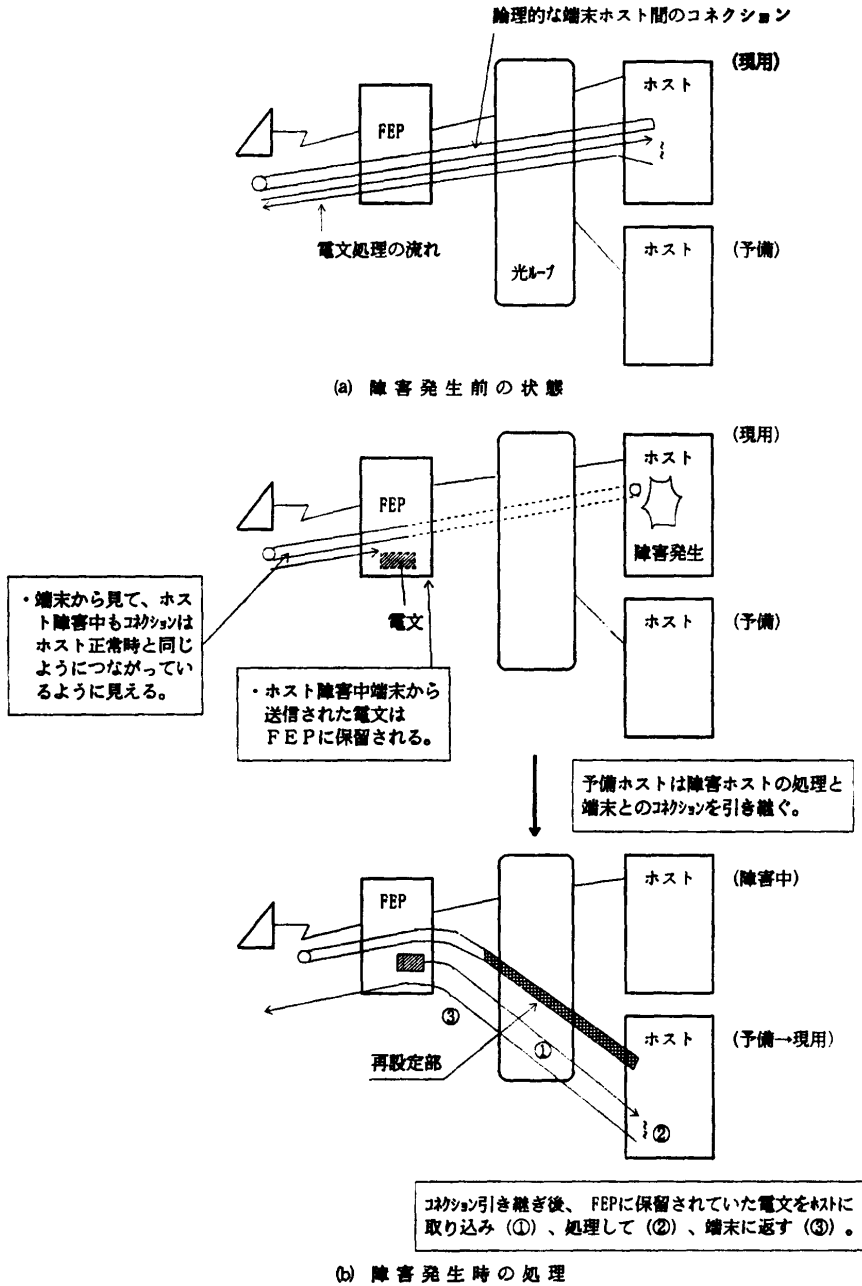


図3 端末無中断方式の概要

Fig. 3 Outline of the method to keep terminals in service during system reconfiguration.

秒)を越えた場合、アラームとなる。)

3. 高速再開の処理方式

3.1 フェーズ1~3の時間短縮

フェーズ1~3については以下に述べる技術により、「ホスト障害発生後25秒以内」をほぼ達成している⁷⁾。

- (1) システム制御プロセッサ (SCP) による障害発見、切換え操作の自動化技術
- (2) 予備ホストでのサービスプログラムの事前初期化 (共用ファイルの事前オープンも含む)
- (3) 予備系での障害復旧処理に必要なチェックポイント情報取得媒体として半導体ファイルを利用し、系間共用化することによるリカバリ処理の高速化

3.2 端末無中断方式の実現技術

3.2.1 ホスト-FEPの機能分担

スタンドアロン形態の通信系およびその利用者のソフトウェア構成 (図4) をベースに、端末無中断方式を実現するためのホスト-FEP間の機能分担のあり方を述べる^{8),9)}。

- (1) コネクションの復旧の観点から見た機能分担

ホスト-FEP形態のシステムではホスト側の通信管理プログラムで、OSIのセッションレイヤ相当以上の機能を分担するのが一般的である。この場合、予備ホスト上でコネクションの状態を、端末、FEP上での状態と矛盾なく復旧するために、ホスト上ではプロトコル状態遷移ごとにそれをファイルへ退避する必要がある。そのためのオーバーヘッドは禁止的であり、また障害のタイミングによってはファイルに取得できない。

したがって通信管理プログラム部分は、FEP側に配置することとした (機能分担線は図4のA以上)。

- (2) ソフト共通化の観点から見た機能分担

通信管理プログラムの上位の利用者は複数存在する (例: オンライン処理向けサービスパッケージ、TSS処理向けサービスパッケージ)。通信管理プログラム

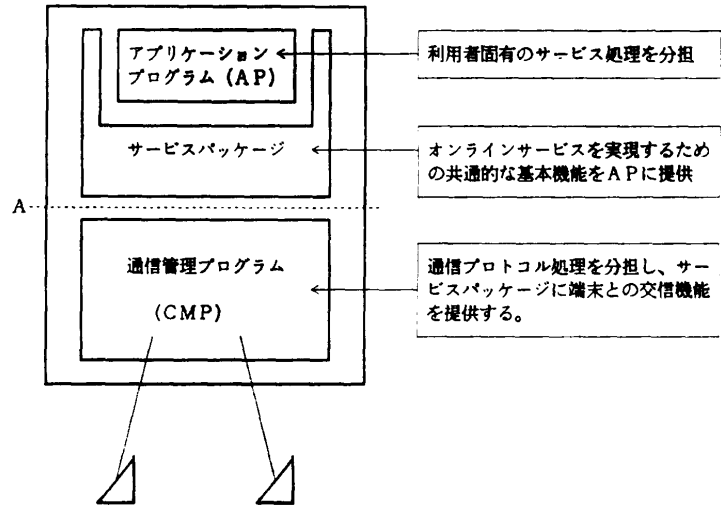


図4 スタンドアロン形態でのソフトウェア構成
Fig. 4 Software structure in a stand-alone system.

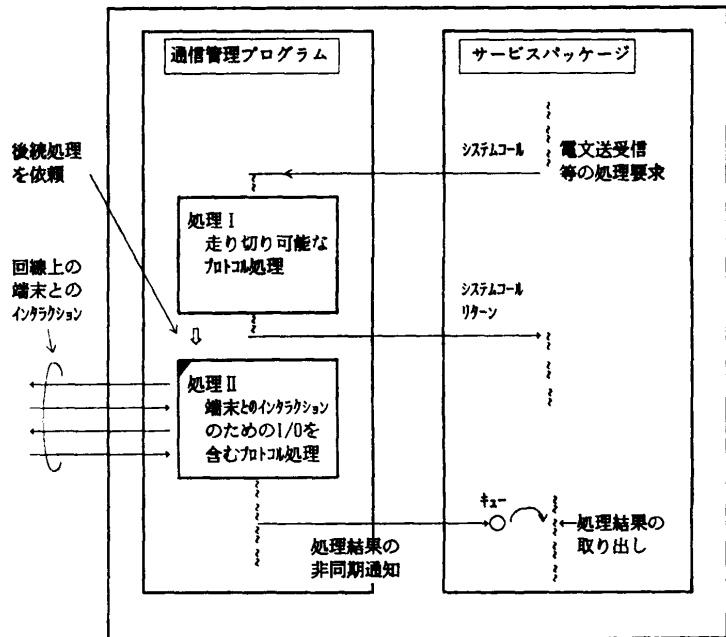


図5 スタンドアロン形態のサービスパッケージと通信管理プログラムのインタフェース

Fig. 5 Communication management program interfaces in a stand-alone system.

より上位でホスト-FEPの機能分担を切るならば、それら個々にホスト-FEPインタフェースメカニズムが必要となる。これはソフトウェア生産性の観点から好ましくない。ゆえに機能分担線は図4のA以下に設定すべきである。

以上(1)、(2)に述べたことにより機能分担は通信

管理機能は FEP に、サービスパッケージはホストに置き、ホスト-FEP 間のインタフェースを司るパッケージとして FIP (FEP Interface Package), HIP (Host Interface Package) を、ホスト上、FEP 上にそれぞれ置くこととした。

具体的なインタフェースを決めるにあたって、利用者(サービスパッケージ)から見たインタフェースがスタンドアロン形態とホスト-FEP 形態とで同一になるように考慮した。

スタンドアロン形態での通信管理プログラムに対する、その利用者の処理要求の手順は一般的には図 5 のように非同期型である。これは電文送受信処理等の回線 I/O を伴う処理が多く、同期型であれば処理がブロックされ、多数のコネクションを並行処理できないためである。ホスト-FEP 形態でもこの非同期型インタフェースを実現する必要があるが、その方法を図 6 に示す。要点は以下のとおりである。

① FIP は、サービスパッケージに対し、通信管理プログラムのインタフェースを疑似する。HIP は通信管理プログラムに対しホスト上のサービスパッケージの処理要求を代行する。FIP, HIP はシステムコール

のパラメータ、処理結果、および電文をフレームの形に編集してやり取りする。端末とのコネクション上のプロトコル処理は、FEP 上の通信管理プログラムに任される。

② HIP と FIP 間の交信プロトコルは、レイヤ 2 相当までとし、オーバーヘッドを極力減らす。(HIP と FIP 間のリンクは、単にフレームの運搬路であり、端末とのコネクション上のプロトコルとは独立である。)

上記により、FIP は利用者からみてスタンドアロン形態と同一の通信管理プログラムインタフェースを提供できることを示した。また、FEP 上の通信管理プログラムはスタンドアロン形態の通信管理プログラムと同一にできる。なお、HIP, FIP は次節に述べる障害時のコネクションの引き継ぎ処理のための処理を行う(ただしコネクションごとのオプション)。

3.2.2 障害時の端末とのコネクションの引き継ぎ処理

概要を以下に述べる。(なお、以降、システムコールによる通信管理プログラムへの処理要求をコマンドと称する。)

① まず、ホストは端末とのコネクションをオープン

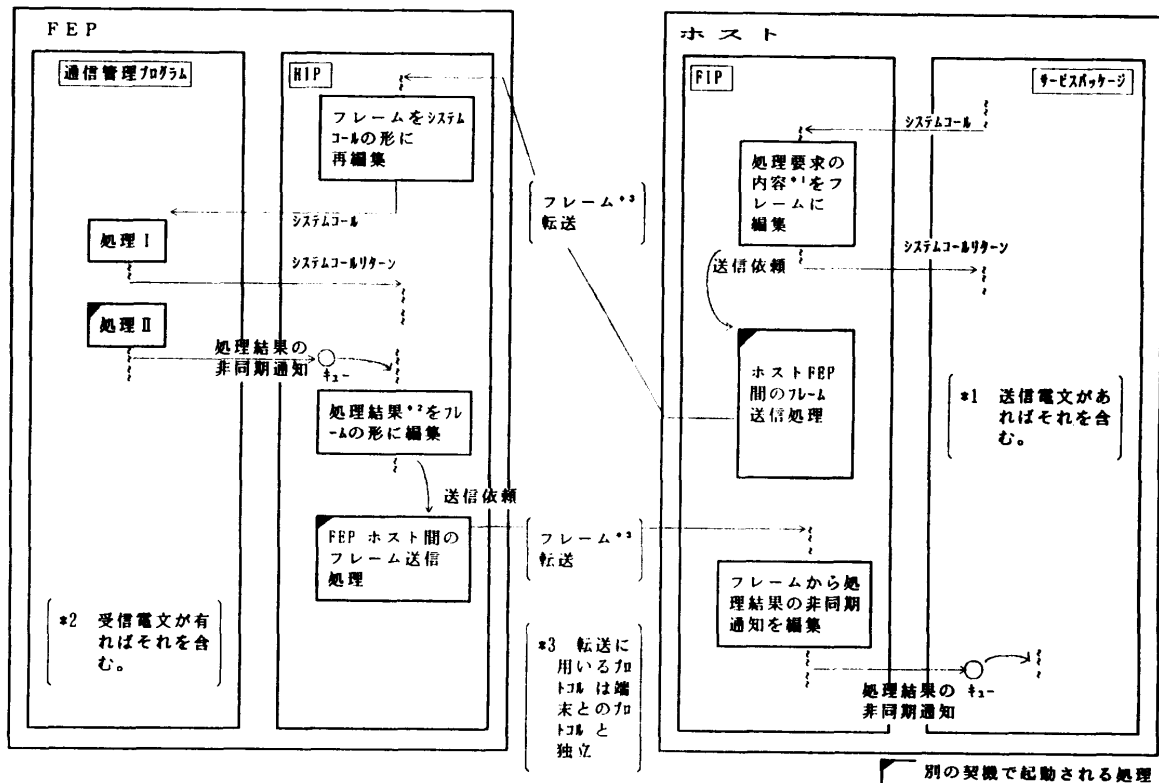


図 6 新しい機能分担に基づくホスト-FEP インタフェースの概要

Fig. 6 Communication interfaces between a host and a FEP in the computer complex system.

したとき、その FEP-ホスト間でユニークに決められるコネクション番号を予備系ホストと共有するファイルへ退避する。

② 交信中にホストがダウンした時、FEP はダウン直前のホスト-FEP 間の交信状態（ホストからの処理要求）を予備ホストへ引き継ぐため、それをメモリ上に保持する。（保持する交信状態を表1の「FEP からの状態通知」に示す。）ホストダウン中も端末との交信は続行され、FEP で受信した電文はそこで保留される。

③ 予備ホストはシステムリカバリが終了した時点で FEP へサービス開始を通知する。これを契機に FEP は保留していたホスト-FEP 間の交信状態を全コネクション分一括してホストへ通知する。

④ ホスト上ではあらかじめ退避してあるコネクション番号により、ホスト上の処理と端末とを結び付け、端末との交信状態の復旧処理を行う。

上記④の復旧処理について図7に示すような照会型電文処理を例に取り、具体的に説明する。

照会型電文処理のホスト上での処理¹⁰⁾を図8に、FEP からの状態通知によって障害前のホストの処理を継続する方法を表1に示す。表1を一般化すれば、1コネクション上の処理を再開できるのは、FEP からの交信状態通知により、チェックポイント取得後からホスト障害発生までの処理内容が判定できる場合である。照会型電文の場合は、図8に示した時間経過の中で、処理内容の判定ができる区間B（すなわち入力電文待ち）が大部分を占めるため、ほとんどの場合、コネクションの引き継ぎ処理は成功することになる。

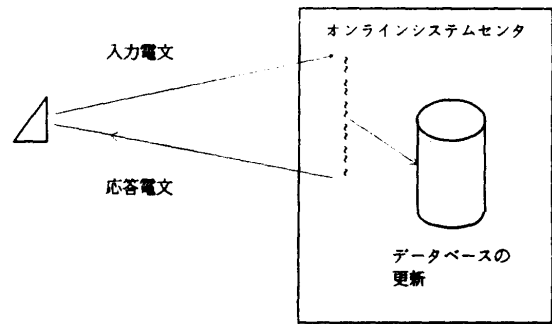


図7 オンラインシステムにおける照会型の電文処理
Fig. 7 Inquiry data processing in an on-line system.

4. 評価

ここでは、定常交信状態の時にホストの一つ（ホスト1とする。）が障害となり、予備ホストへ切り換える場合について種々の評価を加えることとする。（なお、以降、誤解のおそれのない場合はホスト1および予備ホストを単にホストと称する。）

4.1 評価の前提条件

評価対象となるシステムモデルの諸元および評価式に用いる記号を表2に示す。また、評価を行う上での前提条件を以下に述べる。

- (1) センタシステムへの電文の到着はポアソン過程とし、その到着率はホスト正常/障害中を問わず、一定とする。（端末数が十分大きければこの仮定を置いて差しつかえない。）
- (2) 1ホストに収容されるコネクション数、トラフィックは現用 FEP に均等に分散されているも

表1 FEP からのコネクション状態通知とホスト上での復旧処理

Table 1 Status notification from the FEP and corresponding connection recovery process on the host.

FEP からの状態通知	障害状態の解釈	ホストでの当該コネクションについての復旧処理
ホストからの SEND コマンド実行中 (SEND: 電文送信要求)	<ul style="list-style-type: none"> ●ホストが図8のAの区間で障害となったことを示す。 ●当該 SEND コマンドが正常に終了すれば引続き次電文処理の継続が可能である。 	<ul style="list-style-type: none"> ●SEND 完了通知を受け取るため SEND コマンドを再発行する。 ●SEND完了通知が正常終了以外を返せば当該端末とのコネクションをリセットする。
ホストからの RECEIVE コマンド実行中 (RECEIVE: 電文受信要求)	<ul style="list-style-type: none"> ●ホストが図8のBの区間で障害となったことを示す。 Bの時点は前電文処理が正常に終了し、新たな次電文の処理にはいったことを示す。したがって単に電文読み込みにより継続が可能である。 	<ul style="list-style-type: none"> ●RECEIVE コマンドを再発行し電文読み込み結果を FEP から受け取る。 (FEP に受信電文が保留されていれば、コマンド発行後直ちに電文を受け取れる。
上記以外 (割込み処理中異常処理を含む)	<ul style="list-style-type: none"> ●ホストはどの時点で障害となったか判断できない。 	<ul style="list-style-type: none"> ●端末とのコネクションをリセットする。

のとする。

(3) 電文は照会型 (図7) のみとし、一電文当りのホスト CPU 処理量は一定であるとする。また、ホスト上の処理が完了し、FEPへ応答電文が出力される順は、入力電文のホストへの到着順にほぼ等しいものとする。

(4) ホスト上に多くの電文が滞留する時、CPU はアイドルになることはないし、かつ I/O ボトルネックが生じることもないものとする。これは複数の電文を平行処理するために、同一 AP 処理を実行するタスクを複数起動すること、およびデータを複数 I/O デバイスに分散配置することにより実現できる^{10),11)}。

(5) ホスト再開処理において FEP および、ホスト-FEP 間の転送路にボトルネックは生じない。

なお FEP 障害*については、本方式の対象外であり、評価の対象としない。

4.2 端末無中断性の評価

予備ホストが再開後、前の処理が引き継げず、リセットせざるを得ない端末とのコネクション数を評価する。表1によれば、

ホスト障害時に、電文処理状態が図8の RECEIVE 完了から SEND コマンドまでの間にあるコネクション、すなわちホスト上で電文処理中のコネクションは処理が引き継げない。ゆえに、障害発生時にホスト上で処理中の電文数を求めればよい。(割込み処理中、

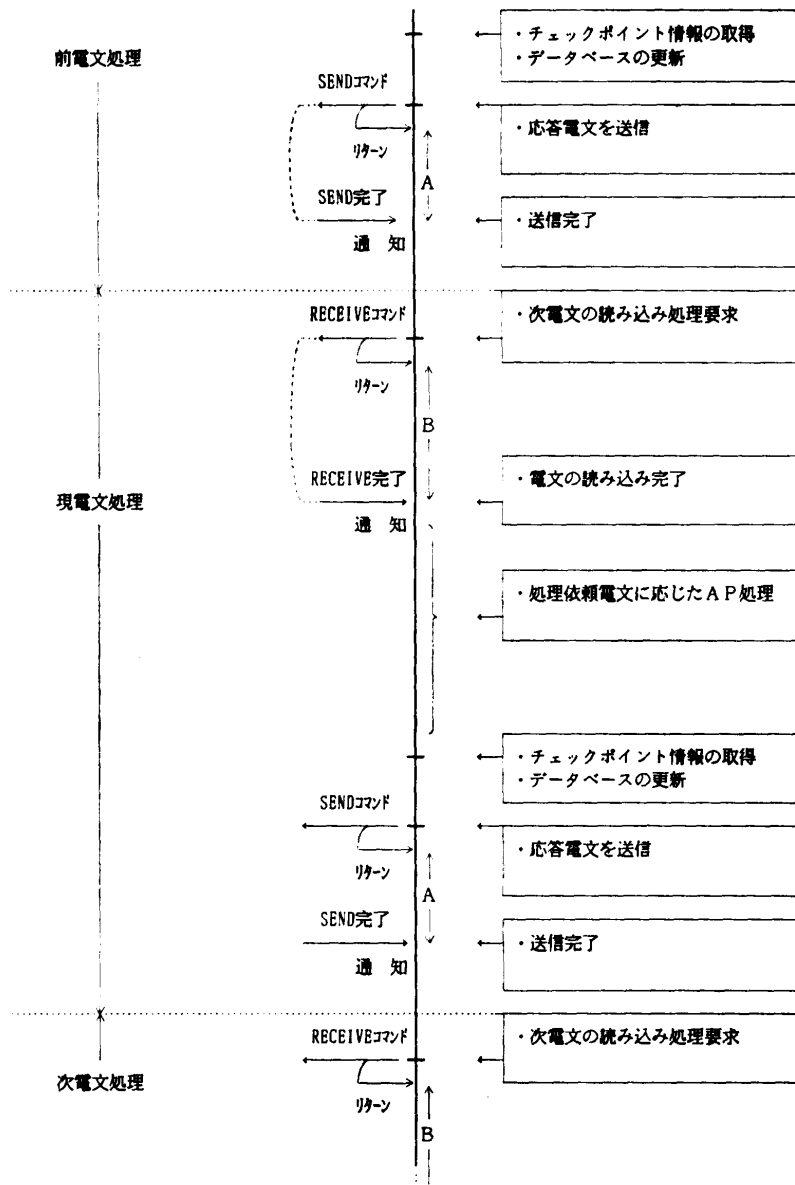


図8 一つのコネクションに着目した電文処理の流れ
Fig. 8 Outline of on-line transaction processing.

異常処理中のコネクションの数は無視できるものとする。) ホスト上での電文の平均滞在時間を T_p とする。 T_p は CPU 時間 (20 msec, 表 2), I/O 時間 (約 60 msec) およびそれらの待ち時間から構成される。 CPU 使用率 80% (繁忙期), I/O の使用率 0.4 以下¹¹⁾で考え、 T_p は 200msec 程度*である。 繁忙期の平均電文到着率 λ は 40 電文/秒 (表 2) であるからホスト上に滞在する平均電文数はリトルの公式¹²⁾より

$$\lambda \cdot T_p \leq 40 \times 0.2 = 8 \quad (1)$$

* 本稿の対象とするシステムでは FEP 障害に対し、予備への高速切替機構を用意しているが、切替えの際、端末とのコネクションは切断・再設定される。

FEP の収容端末数は一般にホストに比べ少ないこと、および、ハード、ソフトがシンプルであり、障害率がホストより低いことから、これ以上の高信頼化対策の必要性は比較的薄い。

* $20 \cdot (1 - 0.8)^{-1} + 60 \cdot (1 - 0.4)^{-1} = 200$ と概算。

表 2 記号およびモデルとしたシステム諸元
Table 2 Notations and their example values in the model system.

項目	記号 (単位)	モデル値	備 考
ホスト数	—	3	現用 2, 予備 1
FEP 数	—	8	現用 6, 予備 2
1 ホスト当りの端末数	N_T	4,500	コネクション数でもある
1 ホストの CPU 能力	M (命令/秒)	$D_s/M=0.02$	ホスト上のみ
一電文処理に要する命令数 (動的ステップ数)	D_s (命令)		
システムリカバリ時間	T_R (秒)	25	フェーズ 1~3 に要する時間
交信状態一括読込処理時間	T_A (秒)	1	
端末とのコネクション再開処理に要する命令数 (1 コネクション当り)	D_A (命令)	$D_A/M=0.0023$	端末とのコネクションをホスト上で再設定し, SEND/RECEIVE コマンド再発行まで
FEP および回線上の電文送信または受信時間 (往復)	T_B (秒)	4	主として回線上的値
1 ホストでの定常状態のトラフィック	N (電文/秒)	30~40	30: 通常トラフィック 40: 繁忙期トラフィック
電文受信後, AP 処理待ちキューへ接続するまでの処理に要する命令数 (動的ステップ数)	D_B	$D_B/M=0.0012$	ホスト上 D_s のうちの一部

を得る。したがって、処理を引き継がないコネクション、すなわち端末の割合は繁忙期においても 8/4500 で 0.2% 以下と評価できる。

4.3 端末応答時間の遅れについての評価

図 9 にホストダウン時からのタイムチャートを示す。以降、記述を簡単化するため $T_R, T_n (n=1, 2)$ 等は時間の長さとともに時間の区間をも表すものとする。また、時間区間 T_n に FEP 上での処理が終了し、ホストへ転送可能となった電文全体を T_n 群電文と称する。

T_R は予備ホスト上で各コネクションの処理状態をチェックポイントまで回復を行う、フェーズ 1~3 のシステムリカバリ時間である。 T_A は通信系再開を FEP に対し宣言し、ホスト-FEP 間の交信状態を FEP からホストへ一括転送するまでの時間である。時間区間 T_c の終了点は以下の二つの処理がホスト上で行わ

れ、FEP 上の保留電文がなくなった時点とする。

① ホスト上でコネクションの引き継ぎ処理を行い、表 1 に従い、SEND/RECEIVE コマンドの発行、またはコネクションのリセットを行う。この処理を全コネクションについて行う。

② SEND/RECEIVE コマンド (再発行) を受け取ると、FEP は障害前に発行された該コマンドの処理を対応させ、すでに終了しているものは完了通知をホストへ転送する。4.2 節で述べたようにこのうちのほとんどは RECEIVE コマンドの完了通知であり、この場合、ホスト上の後続処理としては受信電文を業務対応の応用プログラム (以下 AP と略記) 処理待ちキューへつなぐまでの処理が行われる。

以下では、①、②の処理は AP よりも優先的にスケジュールされるよう設計されていること、および FEP (複数) からホストが保留電文を受け取る間隔が、ホ

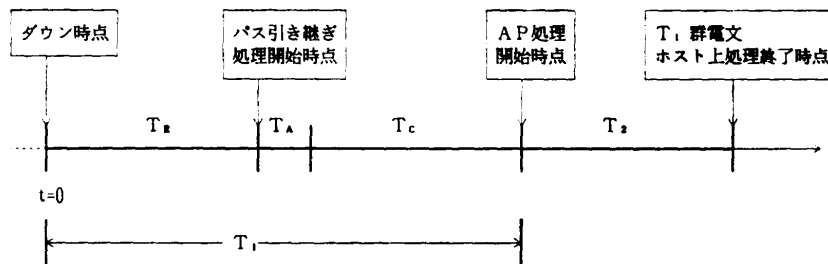


図 9 ホスト障害後の時間経過
Fig. 9 Time chart of the system recovery process.

スト上での②の電文の CPU 処理間隔より短い*ことから、時間 T_c には AP 処理はスケジュールされ得ない*と考慮して解析することとする。また T_c 区間中にも、システムに電文が受信されている。これらを考慮した T_c の計算法を付録 1 に示す。

T_2 は T_1 群の電文のホスト上の処理が終了するまでの時間である。 T_2 には、 T_1 群電文についての AP 処理以降、次電文読み込み要求 (RECEIVE コマンド) 発行までの処理 (図 8 参照) と、 T_2 群電文を FEP から受信し、AP の処理待ちのキューへ接続する処理 (AP 処理より優先度が高い) で構成され、これらの処理で CPU はアイドルになることはない (前提条件 (4))。これらに基づく T_2 の算出法を付録 2 に示した。

次に、 T_1 群電文のうち応答時間 60 秒を越える電文数 (期待値) を N_{out} とおき、その算出法を述べるがその際の留意点は以下のとおり。

- (イ) T_1 群電文は時間区間 T_2 に FEP へ出力されるが、その出力時刻の分布は近似的に一様であると考える。
- (ロ) T_1 群電文の FEP からホストへの転送順、したがって応答電文の出力順は、電文のセンタシステムへの到着順とは独立**と考える。

実際、FEP では RECEIVE コマンドを受け取ると、そのコネクシオンに保留電文があれば、直ちにそれをホストへ転送するので、ホストへの転送順は RECEIVE コマンド受信順に支配される。

算定は以下の場合に分けて行う。

I. $T_1 + T_2 \leq 60 - T_B$ の場合

最初に到着した電文が T_1 群の最後の出力電文となったとしても、その応答時間は $(T_1 + T_2 + T_B) \leq 60$ であるので応答時間が 60 秒を越える端末はない。

II. $T_1 + T_2 > 60 - T_B$ の場合 (ただし $T_1 \leq 60 - T_B$ とする)

時刻 t ($0 \leq t \leq T_1$) にホスト転送待ちとなった電文が制限時間 (60 秒) を越える確率を $P_{out}(t)$ とおくと、留意点 (イ), (ロ), および図 10 より

$$P_{out}(t) = \left. \begin{aligned} & \{ T_1 + T_2 - (t + 60 - T_B) \} / T_2 \\ & (0 \leq t \leq T_1 + T_2 - (60 - T_B)) \end{aligned} \right\} (2)$$

* ホストの処理能力が高くなるとこれは必ずしも成立しない。この場合には、仮に AP 処理が抑止される、すなわち CPU がその分アイドルになるとして評価を行えば安全側の (実際の方が評価より良い) 結果を得る。

** T_c に受信される電文については若干到着順とホストへの転送順に相関があるが、これも独立であると考えることにより安全側の評価を得る。

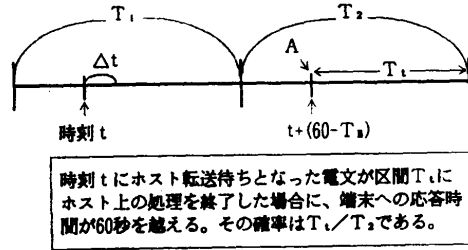


図 10 1 分以内に応答が端末へ返される確率
Fig. 10 Probability that response time is within 1 minute.

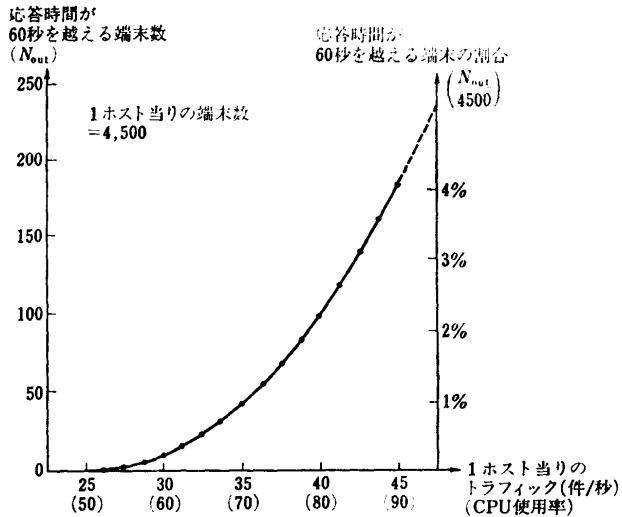


図 11 トラフィックと応答時間が 60 秒を越える端末数との関係
Fig. 11 Number of terminals in which response time is over 60 seconds vs. traffic.

$$P_{out}(t) = 0 \quad (T_1 + T_2 - (60 - T_B) < t)$$

を得る。上式のうち 2 番目の条件は、図 10 において A 点が区間 T_2 の右へ出る場合であるので確率は 0 となる。区間 $(t, t + \Delta t)$ においてホスト転送待ちとなる電文数の期待値 $(= N \cdot \Delta t)$ に式 (2) の確率を掛けたものの、区間 T_1 についての総和を積分で近似することにより、

$$N_{out} \approx \int_0^{T_1} P_{out}(t) \cdot N dt = (N/2T_2) \cdot (T_1 + T_2 + T_B - 60)^2 \quad (3)$$

を得る。(なお、 $T_1 > 60 - T_B$ の場合の N_{out} の算出については付録 3 に略述した。) なお、 T_2 群以降の電文のホスト上の処理時間は、4.4 節で評価を行うが、高々 T_2 ($< T_1$, 付録の式 (A. 6)) であるので上記条件のもとでは、 N_{out} に寄与しない。 N_{out} を表 2 のモデルで計算しプロットしたものを図 11 に示す。本モデルでは障害ホストに收容されている端末全体から

みると応答監視時間を越える端末の割合は繁忙期でも2%程度である。

4.4 応答時間の遅れの回復についての評価

ホスト処理が再開された直後はホスト上に大量の電文が処理待ち/処理中の状態で滞留するため、しばらくは応答時間が長い。この時の単位時間内にホストから出力される電文数を N_x とおけば、前提条件(4)により付録2の式(A.5)が成立するので、

$$N_x = (M - N \cdot D_B) / (D_s - D_B)$$

である。また、ホストに入力される単位時間当りの電文数は N である(ただし、 $t = T_1$ 以降)から、単位時間当りのホスト上の電文数の減少は

$$(M - N \cdot D_B) / (D_s - D_B) - N$$

となる。時刻 t におけるホスト上の電文数を $N_H(t)$ とおく。時刻 $t = T_1$ におけるホスト上の電文数は $N \cdot T_1$ であるので

$$N_H(t) = \{N - (M - N \cdot D_B) / (D_s - D_B)\} \cdot (t - T_1) + N \cdot T_1 \quad (T_1 \leq t) \quad (4)$$

を得る。また、前提条件(3)により、時刻 t にホストに入力された電文の、ホスト上の処理時間を含む滞在時間 $D(t)$ は、それより以前に到着した電文が全部出力されるまでの時間で近似できるから、 $N_H(t) / N_x$ で近似できる。ゆえに

$$D(t) = N_H(t) / N_x = N_H(t) \cdot (D_s - D_B) / (M - N \cdot D_B) \quad (5)$$

を得る。特に、 T_2 群の最初の電文のホスト上の滞在時間は $D(T_1) = T_2$ を得る。次に応答時間が定常状態に復するまでの時間 T_s を評価する。式(4)、(5)はホスト上の電文数 ($N_H(t)$) が少なくなったときには成立しないが、その時は定常状態に近づいたことを示す。したがって、仮に $N_H(t) = 0$ とおき、 t について解くことにより、 T_s は近似できると考えてよい。すなわち

$$T_s = T_1 + N \cdot T_1 \cdot \{(D_s - D_B) / (M - N \cdot D_B)\} \quad (6)$$

を得る。表2のモデルでの T_s とトラフィックの関係を図12に示す。

4.5 通信系再開時間の短縮効果

通信系再開時間をフェーズ4の通信系再開始時点から、予備系ホストと全端末との接続を再設定するまでの時間とする。端末無中断方式を採用せず、障害ホストに接続されていた接続をいったんリセットする場合には、再開ホストからの接続

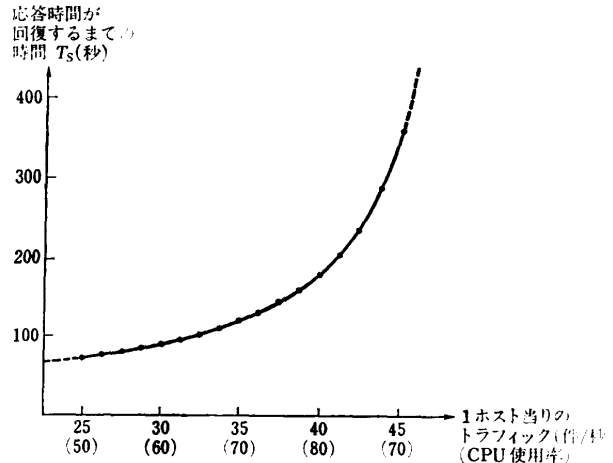


図12 応答時間が回復するまでの時間 (T_s) の推移
Fig. 12 Time necessary for response time recovery vs. traffic.

ン再設定要求に応じて FEP 上で端末とのプロトコル処理を実行する必要がある。表2のモデル(1FEP 当りの再設定接続数 750)では、その処理は FEP 上の CPU 処理時間のみで約2分程度である。

一方端末無中断方式では、通信系再開時間は4.3節で述べた $T_A + T_c$ に含まれる。この値は表2のモデルで14秒(トラフィックが40電文/秒の時、付録式(A.3)参照)である。

端末無中断方式の通信系再開時間の短縮効果は当然ながら顕著である。

5. おわりに

本稿では、端末に対し、ホスト障害発生時にそれを意識させないための端末無中断方式について、それを実現するためのホスト-FEP間の機能分担、および電文処理の再開処理手法について述べた。さらに、ホスト障害時に接続が中断し、障害を意識せざるを得ない端末の割合、応答時間の遅延についての評価法を与え、大規模オンラインシステムの評価例をあげ、方式の有効性を示した。

本稿では、照会型電文処理を中心に論じてあるが、一端から他端末(または計算機)へ処理結果を転送する交換型の電文処理についても議論は同様に展開できる。会話型処理のように一つの処理が複数の電文交換から構成される場合には、チェックポイントをこまめに取ること、コマンドとコマンド完了通知の間の引継情報(これにより、障害発生時に接続の処理状況を、ある程度、予備ホストに引き継げる)を工

夫することなどにより端末無中断を実現できるが、オーバーヘッドと高信頼性のトレードオフの問題を考慮する必要がある。

謝辞 本研究の遂行に当たり、有益な助言をいただいた NTT 情報通信処理研究所石野所長、西川応用システム研究部長、鶴保基本システム研究部長に感謝いたします。

参 考 文 献

- 1) 栗田昭平: 24時間, 365日連続稼働システムをめぐる攻防, *bit*, Vol. 18, No. 11, pp. 4-16 (1986).
- 2) Serlin, O.: Fault-Tolerant Systems in Commercial Applications, *Computer*, August, pp. 19-30 (1984).
- 3) 木崎健太郎: 利用が広がるフォールト・トレラント機, 日経コンピュータ, No. 167, pp. 67-79 (1988).
- 4) 渡辺栄一編訳: ブックフォールト・トレラントシステム, マグロウヒル, 東京 (1986).
- 5) 鶴保証城ほか: DIPS 複合システムのソフトウェア構成法, NTT 電気通信研究所研究実用化報告, Vol. 36, No. 8, pp. 1021-1030 (1987).
- 6) 五十川隆文ほか: DIPS 複合システムにおける運転管理方式, NTT 電気通信研究所研究実用化報告, Vol. 36, No. 8, pp. 1031-1040 (1987).
- 7) 十倉建二ほか: DIPS 複合システムにおけるホット予備方式, NTT 電気通信研究所研究実用化報告, Vol. 36, No. 8, pp. 1041-1048 (1987).
- 8) 川原洋人ほか: 高信頼システムのための FEP・ホスト機能分担手法について, 第 29 回情報処理学会全国大会論文集, pp. 315-316 (1984).
- 9) 大石和寛ほか: 高信頼システムのための FEP 機能, 第 30 回情報処理学会全国大会論文集, pp. 1017-1018 (1985).
- 10) 上田恭雄: オンラインシステム, pp. 93-109, 昭晃堂, 東京 (1987).
- 11) 木ノ内康夫ほか: オンラインファイル記憶系装置のコスト評価に関する一考察, 情報処理学会論文誌, Vol. 27, No. 6, pp. 621-629 (1986).
- 12) Kleinrock, L.: *Queueing Systems Volume I*, pp. 10-19, John Wiley & Sons, New York (1975).

付 録

1. T_1, T_C の算出

時間 T_C 中にホストに転送されるのは T_1 群電文であり, その数を N_1 とおく. 時間 T_C にはホストの CPU は間断なく 4.3 節の①, ②の処理に占有され, それぞれの処理量は表 2 から, $D_A \cdot N_T, D_B \cdot N_1$ である. (②において RECEIVE コマンド完了通知以外の後続処理についてはわずかなため, 無視できる.) ゆ

えに

$$(D_A \cdot N_T + D_B \cdot N_1) / M = T_C \quad (A. 1)$$

を得る. 一方 N_1 は $T_1 (= T_R + T_A + T_C)$ にホスト 1 あてに転送待ちとなる電文数であるから

$$T_1 \cdot N = (T_R + T_A + T_C) \cdot N = N_1 \quad (A. 2)$$

(A. 1), (A. 2) より, T_C, T_1 を得る. すなわち

$$T_C = \{D_A \cdot N_T + D_B \cdot N \cdot (T_R + T_A)\} / (M - D_B \cdot N) \quad (A. 3)$$

$$T_1 = \{M \cdot (T_R + T_A) + D_A \cdot N_T\} / (M - D_B \cdot N) \quad (A. 4)$$

2. T_2 の算出

T_2 においては, ① AP 処理待ちキューに溜った T_1 群電文の処理と, ② T_2 にホストへ転送待ちとなる電文をホストが受信して AP 処理待ちキューへ接続する処理が行われる. 処理②は処理①より優先度が高い. T_2 においては, ホスト上に多くの電文が滞留しているので前提条件(4)を適用し, ホストの CPU はアイドルではない. したがって, 単位時間内にホストが処理を完了する T_1 群の電文の数を N_x とおくと

$$N_x \cdot (D_S - D_B) + N \cdot D_B = M \quad (A. 5)$$

を得る. ここで, 左辺の第 1 項は処理①, 第 2 項は処理②に対応する.

$T_2 \cdot N_x = N_1, N_1 = T_1 \cdot N$ であるから

$$T_2 = \{N \cdot (D_S - D_B) / (M - N \cdot D_B)\} \cdot T_1 \quad (A. 6)$$

を得る. なお, 定常状態での CPU 使用率は 1 より小さくなくてはならないので, $N \cdot D_S < M$ であり, これより容易に $T_2 < T_1$ を得ることに留意されたい.

3. $T_1 > 60 - T_B$ の場合の N_{OUT} 算出

図 10 で行ったと同様の考察を行うことにより以下の結果を得る.

[$T_2 - (60 - T_B) < 0$ の場合]

$$P_{OUT}(t) = 1 \quad (0 \leq t \leq T_1 - (60 - T_B))$$

$$P_{OUT}(t) = \{T_1 + T_2 - (t + 60 - T_B)\} / T_2 \quad (T_1 - (60 - T_B) \leq t \leq T_1 + T_2 - (60 - T_B))$$

$$P_{OUT}(t) = 0 \quad (T_1 + T_2 - (60 - T_B) < t)$$

ゆえに,

$$N_{OUT} \doteq \int_0^{T_1} P_{OUT}(t) N \cdot dt = (N/2) \cdot \{2T_1 + T_2 - 2 \cdot (60 - T_B)\}$$

[$T_2 - (60 - T_B) \geq 0$ の場合]

$$P_{OUT}(t) = 1 \quad (0 \leq t \leq T_1 - (60 - T_B))$$

$$P_{OUT}(t) = \{T_1 + T_2 - (t + 60 - T_B)\} / T_2 \quad (T_1 - (60 - T_B) \leq t \leq T_1)$$

ゆえに,

$$N_{\text{out}} \doteq \int_0^{T_1} P_{\text{out}}(t) N \cdot dt = (N/2T_2) \{2T_1 \cdot T_2 - (60 - T_B)^2\}$$

なお、上記は T_1 群電文のみの N_{out} への寄与分であるが、この場合は、 T_2 群電文にも N_{out} への寄与分があり、それについては式(5)から容易に計算できる。詳細は省略する。

(昭和63年6月22日受付)

(昭和63年11月14日採録)



川原 洋人 (正会員)

昭和24年生。昭和47年東京大学理学部数学科卒業。昭和49年同大学院修士課程修了。同年NTT入社。現在、NTT情報通信処理研究所勤務。応用プログラム、およびDIPS通信管理プログラムの研究・開発に従事。電子情報通信学会会員。



柴垣 斉 (正会員)

昭和31年生。昭和53年東京大学工学部電子工学科卒業。昭和55年同大学院修士課程修了。同年、NTTに入社。現在、情報通信処理研究所に勤務。主に、DIPS通信管理プログラム、DIPS共通OSの研究・開発に従事。電子情報通信学会会員。



仲谷 元 (正会員)

昭和26年生。昭和49年北海道大学工学部電子工学科卒業。昭和51年同大学院修士課程修了。同年、NTTに入社。現在、情報通信処理研究所に勤務。主に、DIPSオペレーティングシステムの研究・開発に従事。電子情報通信学会会員。



大石 和寛 (正会員)

昭和32年生。昭和55年北海道大学工学部電子工学科卒業。昭和57年同大学院修士課程修了。同年、NTTに入社。現在、情報通信処理研究所に勤務。主に、DIPS通信管理プログラムの研究・開発に従事。電子情報通信学会会員。