

ALSを適用したOSI-TPシステムの 2Q-1 実現方式に関する一考察

岩倉 伸行 玉置 政一 松田 栄之

NTTデータ通信株式会社

1. はじめに

OSI (開放型システム間相互接続) の分散トランザクション処理プロトコル (OSI-TP) ⁽¹⁾ の標準化が進められており、旅券発行業務などの分散処理システムに適用できるものとして注目されている。筆者らはさきに報告したように、応用層構造 (ALS) ⁽²⁾ に準拠してOSI-TPの実装を進めてきた⁽³⁾。本稿では、試作システムにおけるALS準拠のプロトコル実装方法についての検討結果とシステムの性能への影響について報告する。

2. プロトコル概要

OSI-TPは、ダイアログやトランザクションブランチなどの論理的な通信路を利用して複数のシステム間に必要なデータの転送や実行されている処理の一貫性を保証するためのプロトコルである。またトランザクション処理では通信効率を高めるため、複数のPDU (プロトコルデータ単位) を一つのPDUとして転送する連結機能や、他のプロトコルで作成されたPDUのユーザデータとしてプロトコルデータを転送する埋め込み機能が規定されている。

OSI-TPプロトコルの記述では、応用層のプロトコル処理モデルを体系化したALSによるモデルを採用している。このモデルでは一本のアソシエーションに関連する機能と状態を制御する単一アソシエーションオブジェクト (SAO) と、複数のアソシエーションに関連した機能を制御する複数アソシエーション制御機能 (MACF) に分けている。さらにSAOは、複数の応用サービス要素 (ASE) と、これらASE間およびプレゼンテーションサービスの利用を調整する単一アソシエーション制御機能 (SACF) により構成される。

3. プロトコル実装方法の検討

次の基本方針にもとづき、ALSに準拠してプロトコル実装を行った。

- ①プロトコル処理モジュールの部品化
- ②OSI-TPやACSEなどのプロトコル実装で必要となる機能の共有化
- ③多様化する応用層プロトコルへの対応

ALSに準拠したモジュール構成を図1 (A) に示す。OSI-TPプロトコルの実装においては、次の機能の実装方法は実装者に任されているため、その実現方法を明確にする必要がある。

- ①ASEの外部インタフェース
- ②プリミティブの連結と埋め込みの実現方式
- ③プリミティブの処理順序制御方式

次にこれらの機能の検討内容と実現方法を示す。また、表1にALSに準拠した各モジュールに実装した機能をまとめる。

3.1 ASEの外部インタフェース

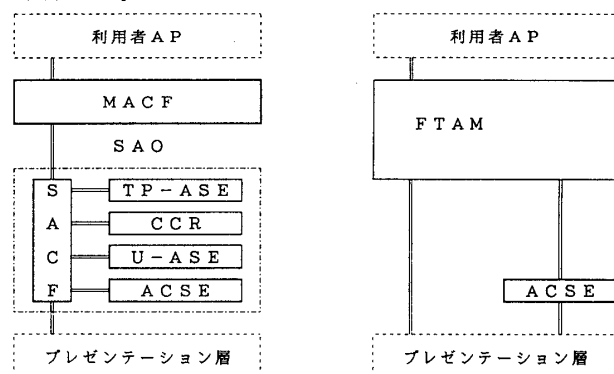
従来、図1 (B) に示すようにALSに準拠せずプロトコル毎に処理モジュールを単独に作成していたため、モジュール間のつながりが強く、上位プロトコルが異なると、そのプロトコルに対応して改造が必要となっていた。これに対しALSに準拠することでASEモジュールは、SACFとのみインタフェースを持つように統一することによって、ASEモジュールに特別な改造を加えずに種々のプロトコル構成に柔軟に対応できるようにした。

ASEの外部インタフェースは従来のモジュール構成で利用していたものを拡張して規定した。これはPDU/プリミティブごとに、①通信路を識別するためのプレゼンテーション・サービスアクセスポイント (PSAP) とPSAP番号、②SACFで転送先を決定するのに必要な情報である発行元/発行先のモジュール識別子、③データの内容を識別するためのPDU/プリミティブの種類、などからなる。これによりSACFでは、PDU/プリミティブが次にプロトコル処理を必要とするモジュールを決定し転送するようにした。

3.2 連結と埋め込みの実現方式

連結や埋め込みの機能を実現するには、複数のASEによるプロトコル処理が必要である。したがって、連結/埋め込みの機能を実現するモジュールには、①各ASE間で認識して処理を行う、②SACFで処理を行う、ことが考えられる。今回の実装では前出の基本方針に従って、ASEモジュールの部品化をはかり、各ASEごとに持つ機能の共通化をはかるために、SACFで実現する (②を採用) こととした。

連結機能の実現方法は、ASEで作成されたPDUをSACFで蓄積し、次にASEから連結されるべきPDUを受信したときに連結操作を行うことにより実現した。また埋め込み機能の実現方法は、SACFで埋め込みの必要なPDUをその種類と発行元モジュール識別子からSACFで判断し、埋め込みを実行するASEに転送することにより実現した。



(A) ALSに準拠した構成 (B)従来の構成の例

図1 モジュール構成

A Study on the Implementation of Application Layer Structure to OSI-TP System

Nobuyuki IWAKURA, Masakazu TAMAKI, Sigeyuki MATSUDA

NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

表1 モジュール実装機能

モジュール名	機能の割当
MACF モジュール	①論理的通信路(アソシエーション、ダイヤログ、トランザクション、ラッチ)の管理 ②SAOの管理 ③TPサービスのプリミティブを各ASEのサービスに対応づけ ④異常なイベント受信時の処理
SACF モジュール	①PDU/プリミティブのモジュール間振り分け機能 ②PDUの連結機能 ③PLモジュールへの情報付加機能(PDV-LIST) ④PDU/プリミティブの処理順序制御機能
各ASE モジュール	①PDU/プリミティブの生成/解釈機能 ②PDUのパラメータチェック

3.3 プリミティブの処理順序制御方式

MACFがSACFへプリミティブを送信する順序は、SACFからPLへ送信される順序として同一である必要がある。たとえば埋め込み機能が必要なプリミティブが連結を指示されてSACFへ入力されると、埋め込み機能に必要な複数のASEでプロトコル処理が実行される。この間にSACFに入力された他のプリミティブのプロトコル処理が完了して先に連結処理され、送信PDUのシーケンスが変わる可能性がある。このようにPDUの送信順序がSAOで入れ替わる場合を避けるため、MACFからのプリミティブの処理順序についての制御機能を、各ASEの処理結果を受け取るSACFに持たせることとした。

SACFではPDUの処理順序を記録するキューを持ち、MACFあるいはプレゼンテーションモジュール(PL)からPDU/プリミティブの受信順序をキューに登録する。そしてSACFではキューの順序に従って埋め込みや連結を実行し、またMACFやPLへの出力順序を入力順序と同じになるように制御することとした。

4. プロトコル処理時間の考察

4.1 前提条件

ALSの準拠によるプロトコル処理効率への影響を調べることを目的に、単一ノード間を対象を絞って本システムのモジュール別PDU/プリミティブ処理時間を測定した。OSI-TTPプログラムはワークステーション上で実行した。OSI-TTPの機能単位はカーネル、ハンドシェイク、全2重機能単位を選択した。アプリケーショントランザクションを選択した場合の測定結果例を表2、表3に示す。測定値はOSがモジュール間の通信に要する時間を基準値として正規化した値を示す。

表2 プロトコル処理時間例(PDU発行側)

モジュール名 プリミティブ名	MACF	SACF	ASE	SACF
TP-BEGIN-DIALOGUE-req	28.7	2.8	6.0	21.2
TP-DATA-req	21.1	2.1	2.9	20.9
TP-END-DIALOGUE-req	28.3	2.1	2.0	27.9
TP-U-ERROR-req	19.0	2.1	3.0	26.3
TP-U-ABORT-req	32.1	2.1	4.1	28.9
TP-HANDSHAKE-req	18.9	2.2	4.0	27.3
TP-HANDSHAKE-cnf	19.0	2.2	3.1	21.4

注 プリミティブの処理順序はMACFからSACFへの方向

4.2 考察

表2、表3の測定結果から、次のことがわかる。

①PDUの発行側と受信側での処理時間はほぼ等しく、MACF、SACFでの処理時間に比較して、ASEでの処理時間が短くなっている。これはASEで共通に利用される機能をSACFに実装し共有化したため、具体的にはPLとの間でPDUを送受信する時に必要なPDV-LISTの作成、解釈の機能などがある。これらの機能はASEに共通に必要なためSACFに実装した。しかしASE単独の機能を高めるためには、それぞれのASEに共通機能を実装することも考える必要がある。

②MACFでの処理時間がASEの処理時間に比較して大きい。これはダイアログなどの論理的通信路の確立、解放や複数の通信相手との調整に必要な処理、およびOSI-TTPプロトコル利用者からのプリミティブ作成あるいは解釈に伴う処理のためである。複数のノードからダイアログの確立、解放の処理が頻繁に行われるシステムでは、MACFでの処理がネックとなる可能性がある。このためMACFの機能の一部を他のモジュールに移すことなどが考えられるが、通信ノード数が増加した時のシステム性能や対策については今後の検討課題である。

③MACFとASE間のSACFの処理は、一つのPDU/プリミティブのプロトコル処理全体に占める処理時間のうち5パーセント程度である。このことからPDU/プリミティブの転送先モジュールの決定に要する1回の処理時間は、性能に及ぼす影響は少ない。しかし埋め込み機能や連結機能を利用する場合には、異なるASEでのプロトコル処理を実行することにSACFを通過する必要があるため、埋め込み実行時にはSACFによる影響が大きくなる。したがって埋め込み処理の頻繁なシステムでは、ALSに準拠しないローカルなインタフェースを検討するなどにより性能改善をはかる必要がある。

5. あとがき

本稿では、ALSに準拠したプロトコル実装方法についての検討を行い、システムの性能に対する影響を考察した。

本稿の結果からALSに準拠したプロトコル実装を行った場合、SACFの機能が重要であるとともに、ASEとのインタフェースを効率よく実現する必要があることがわかった。今後は本検討の結果をもとに、複数ノード間での通信効率に関する詳細な検討を進める予定である。

[参考文献]

- [1] ISO/IEC 10026(2nd DP), 1988 Dec.
- [2] ISO/IEC 9545, 1989 Dec.
- [3] 岩倉、澤、玉置、"OSIトランザクション処理プロトコルの実現方式に関する一考察"、情処全大第39回。

表3 プロトコル処理時間例(PDU受信側)

モジュール名 プリミティブ名	MACF	SACF	ASE	SACF
TP-BEGIN-DIALOGUE-ind	19.1	2.5	6.0	28.7
TP-DATA-ind	7.8	2.3	2.6	22.9
TP-END-DIALOGUE-ind	30.9	2.1	8.0	20.8
TP-U-ERROR-ind	21.1	2.9	2.2	22.8
TP-U-ABORT-ind	21.5	2.9	2.6	16.5
TP-HANDSHAKE-ind	27.3	2.3	4.8	18.9
TP-HANDSHAKE-rsp	19.1	2.1	3.0	28.4

注 プリミティブの処理順序はSACFからMACFへの方向