

(1991. 9. 25)

MAP トランスポート層 プロトコルの実装
An implementation of MAP Transport protocol瀬戸 康一郎
Koichiro SETO鈴木 靖雄
Yasuo SUZUKI浅野 光春
Mitsuharu ASANO日立電線(株) オプトロシステム研究所
Optoelectronic System Laboratory, Hitachi Cable, Ltd.

要旨

MAP(Manufacturing Automation Protocol)は、米General Motors社によって提唱されている工場内ネットワーク用の通信プロトコルである。MAPはOSI7階層プロトコルの機能プロファイルであり、OSI(Open Systems Interconnection)によるマルチベンダ接続の実証例として注目を浴びている。

筆者等は、このフルMAP ver3.0 に準拠した通信制御ボードを開発した。本ボードは、OSI 第5層までのプロトコルをボード上において処理するインテリジェントタイプである。本稿においては、このボードに実装したISO-TP4(Transport Protocol class4)の処理方式、及び性能測定の結果について報告する。

Abstract:

MAP(Manufacturing Automation Protocol) is the computer communication protocol defined for the multi-vendor interoperability at manufacturing sites. MAP is an OSI functional profile and is one of the earliest experiments of the interconnection by OSI protocol.

We have developed an intelligent network adapter for MAP, on which we implemented the protocol processing software upto Session Layer. This paper focuses on the implementation of Transport Protocol processing software on this adapter.

1.はじめに

MAP(Manufacturing Automation Protocol)は、米General Motors社によって提唱されている工場内ネットワーク用の通信プロトコルである。MAPはISOの規定するOSI(Open Systems Interconnection)7階層プロトコルの機能プロファイルである。バークレー版UNIXの新リリース(BSD4.3-Reno)にOSI下位4層が実装されるなど、OSIによる異機種接続の試みが最近活発化しているが、MAPはマルチベンダ機器を共通プロトコルにより接続するためにOSIプロトコルを使用した最も初期の試みである。

筆者等は、CPUにモトローラ社の68020(20MHz)を用いたインテリジェントタイプのMAP通信制御ボードの試作を行った。インテリジェントタイプの通信制御ボードは、価格が高くなるという欠点がある一方、ホストにかかるプロトコル処理のための負荷を軽減できるため、高速LANに接続されたサーバマシン等への適用に適している。

筆者は、このインテリジェントタイプのMAP通信制御ボードに搭載するトランスポート層プロトコル処理プログラムの開発を行った。OSIモデルの第4層に位置づけるトランスポート層は、エンドシステム間の確実なデータ転送を保証するためのコネクション型プロトコルである。トランスポート層の一般的な役割は、フロー制御やデータ到着の監視、パケット紛失時の再送、送信データの分割/再組立等である。MAPにおいては、トランスポートプロトコルとしてISO8073規定のTP4/CLNS(Transport Protocol class4 over Connectionless Network Service)が採用されている。TP4/CLNSは、前述したバークレー版UNIXにも採用されており、OSIプロトコルの選択の中では、比較的LAN環境に適したプロトコルである。

開発及び移植の容易さを考慮して、処理プログラムは全てC言語により開発した。また、スケジューリングやメモリ管理は、独自に開発したリアルタイムOS(Operating System)にて行う。

処理方式の決定においては、トランスポート層処理の高速化を主眼に検討を行った。トランスポート層の動作には、コネクションの設定/解放、転送データの送受信、再送、再順序づけなど、様々な動作が存在するが、これらの動作の内、トランスポート層処理を高速化するために最も重要なのは、コネクション設定後の転送データ送受信処理の最適化である[1]。そこで、トランスポート層における通常の転送データの送受信処理の高速化を中心に検討した。

高速化のための検討においては、机上の検討と併せて、簡単なプロトタイプ・プログラムによる比較を行い、より高速な処理方式を決定するようにした。又、このような事前の検討と併せて、エミュレータによる実機テストの段階で関数レベルの処理時間の測定を行い、ボトルネックとなる部分の処理アルゴリズムの変更を行った。

又、このようなプロトコル処理高速化の為の工夫に加えて、選択再送や、輻輳制御手順の実装を行っている。本稿においては、高速化のための検討やスループット等の測定結果に加え、これらの実装手順の内容について報告する。

2 高速化のための検討

2.1 コネクションへの関連付け

(1)検討

コネクションへの関連付けとは、受信TPDU(Transport Protocol Data Unit)に含まれるレファレンスと呼ばれる16bitの識別情報を元に、対応するコネクションの管理テーブル(TCCB;Transport Connection Control Block)を検索する処理である。

レファレンスは、トランスポート層が管理し、コネクションの設定時に、他のコネクションにおいて使用中でない値を割り当てる。

コネクション管理に使用中のTCCBを順に検索して、レファレンスの一致するTCCBを探し出すような関連付け方式の場合、コネクション数が増加するにつれ、検索するTCCBの数も増え、コネクションへの関連付けにかかる時間が増加するといった問題がある。そこで、次に述べるように管理するコネクションの数に関わらず、高速な関連づけができる方式を検討した。

(2)サイクリックレファレンス割当方式[2]

今回開発したトランスポート層には、サイクリックレファレンス割当方式と呼ぶレファレンス割当方式を実装した。すなわち、コネクション管理テーブル(TCCB)を予め固定数だけメモリ領域に確保しておき、各TCCBが使用するレファレンスの下位数ビットを、TCCB毎に固定とした。

このような割当方式を使用することにより、レファレンスの下位の固定ビットから対応するTCCBアドレスを直ちに得ることができるようになり、高速な関連付けを行えるようにした。具体的には、レファレンスの固定ビットの値にTCCBのサイズを掛け、TCCBエリアのベースアドレスに足すことにより、該当のTCCBアドレスを得ることができる(図2-1)。

図2-2に、簡単なプロトタイプ・プログラムによる2方式の処理速度の比較を示す。単純な検索方式によるコネクション関連付けの場合、コネクションの数が増えるにつれ関連付けのための処理時間が増加しているが、本方式による関連付けは、一定時間で関連付けが終了している。ボード上でエミュレータを用いて本方式による関連付け処理時間を測定したところ、約9 μ secという良好な結果が得られた。

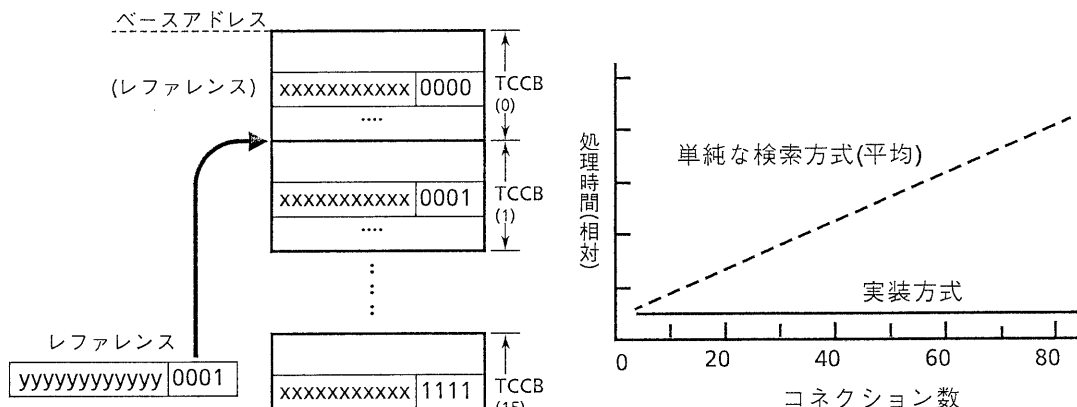


図2-1 サイクリックレファレンス方式による
コネクション関連付け

2.2 タイマの操作

(1)検討

OSI-TP4においては、4種類のタイマを規定しており、実装によっては、更に数種類のタイマを使用する。トランスポート層におけるタイマは、主に再送監視や、無活動監視のためのタイマであり、TPDUの送受信の度にこれらのタイマのスタート/ストップを繰り返す必要がある。例えばDT TPDU受信時には、無活動監視タイマ(I-timer)のリスタート操作を行い、受信したDT TPDUの応答としてAK TPDUを送信する場合には、ウィンドウタイマ(W-timer)のリスタート操作、AK TPDUを直ちに送信せずに、遅延送信を行う場合にはAKホールドバックタイマ(AKHB-timer)のスタート操作を行う必要がある。

このようなTPDU受信毎のタイマ操作を、全てOS(Operating System)の提供するタイマ機構を利用して行った場合、システムコールによるオーバーヘッドが大きくなるといった問題がある。そこで、トランスポート層の内部でタイマ管理を行う方式を検討した。

トランスポート層におけるタイマ管理方式として、タイマ管理ブロックをタイムアウトの近い順にキューイ

ングしておき、最も近いタイムアウトに対してのみ、タイムアウト割り込みを指定しておくタイマキュー方式及びその改良方式が報告されている[3]。しかしながら、タイマのスタート/ストップ操作を行う度にタイマ管理ブロックをエンキュー/デキューする必要のあるタイマキュー方式は、タイマ操作を頻繁に行うトランスポート層においては、エンキュー/デキュー処理によるオーバーヘッドが大きく不向きである。そこで以下に述べるようなカウンタ検索方式の使用を検討した。

(2)カウンタ検索方式

コネクション管理テーブル(TCCB)に各タイマに対応したカウンタを設ける。タイマスタート操作はカウンタに0以上の値を設定することにより行う。タイムアウト検出のため、タイマ周期の何分の一の期間毎にタイマ管理タスクを起動して、全ての使用中のTCCBのカウンタの減算を行う。減算の結果カウンタが0となった場合、タイムアウトが発生したものとしてタイムアウト処理を行う。

本方式によるタイムアウト検出処理は、タイマキュー方式によるタイムアウト時の処理(タイマキューの先頭からタイマ管理ブロックを取り外して、再びキューの末尾にエンキューする処理)の1/10以下のオーバーヘッドで済むことがプロトタイプ・プログラムによる測定の結果判った。即ち、タイマ管理タスクの起床周期を各タイマのタイマ周期の1/10以上に取れば、カウンタ検索方式によるタイムアウト処理とほぼ同様又はより短い処理時間でタイムアウト操作を行うことができる。更に、本方式によるタイマのリスタート操作はカウンタの設定だけなので、タイマキュー方式によるリスタート操作(エンキュー/デキュー操作)の1/10以下のオーバーヘッドで済む。このため、タイマのスタート/リスタートがより頻繁な程、本方式が有利となる。

2.2.3 トランスポートヘッダの解釈/生成

MAPにおいて採用しているIEEE802.4/Token Bus方式のLANにおいては、最大8KByteのパケット長が許されている。8KByteもの送受信バッファを使用した場合、メモリの使用効率が極端に低下することが考えられる。そこで、開発した通信制御ボードにおいては、1024Byteの固定長送受信バッファを使用し、1024Byte以上の送信データを扱う場合には、2つ以上のバッファを、図2-3に示すようにチェーンポイントによって連結して使用している[4]。

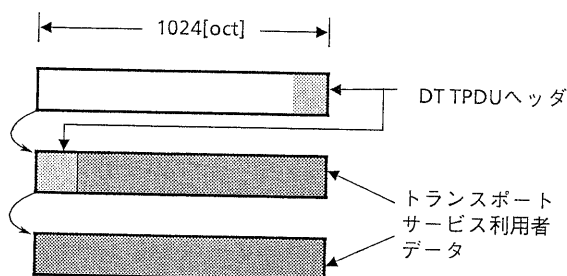


図2-3 送受信バッファの使用法

表2-1 パラメタリード処理時間(改良前)

処理内容	測定結果
1Byte パラメタリード	9.3[micro sec]
2Byte パラメタリード	15.1[micro sec]
4Byte パラメタリード	13.0[micro sec]

表2-2 DT TPDU(512Byte)の受信処理時間

内容	測定結果
改良前	532.0[micro sec]
改良後	404.6[micro sec]

このようなデータ構造の送受信バッファを使用した場合、受信TPDUのトランスポートヘッダが2つのバッファにまたがって設定されている場合が有り得る。トランスポートDT TPDUやAK TPDUのトランスポートヘッダは、4Byte～14Byteと小さいため、ヘッダの解釈/設定そのものにかかるオーバーヘッドは、さほど大きくないと考えられる。しかしながら、各パラメタを読み取り/設定する度に、パラメタが2つのバッファを跨っていないか等の判定を行う必要があり、このためのオーバーヘッドが、ヘッダの解釈/設定の処理において大きな割合を占めていることがわかった(表2-1)。

そこで、この判定付きのパラメタリードを無くすため、TPDUのヘッダ部分を、固定領域にコピーしてから、ヘッダの解釈を行うように変更を行った。この変更により、DT TPDUの受信処理で約24%の高速化を行うことができた(表2-2)。

3. その他の実装機能に関する検討

3.1 選択再送/輻輳制御

ルータ、ブリッジ等の中継装置におけるオーバフローやビット誤りによるフレーム紛失の回復手順として、紛失フレーム以降の全てのフレームを再送する「全再送手順」と、紛失フレームのみを再送する「選択再送手順」がOSI-TP4のオプションとして許されている。

選択再送手順は、処理が複雑となる。しかしながら、中継装置におけるオーバフローによるフレーム紛失の

場合、紛失フレーム以降の全てのフレームを再送する全再送手順では、中継装置の輻輳を助長する可能性がある。そこで、今回開発したトランスポート層プログラムにおいては、選択再送手順を実装した。

選択再送手順は、DT TPDU受信処理モジュールにおいて、以下の処理を追加することにより実現した。

- (1)正しいシーケンス番号と、受信クレジット1を設定したAK TPDUを返送する。同時に正しいシーケンス番号より大きなシーケンス番号を持ったDT TPDUを受信した場合、これを保持しておく。同時に正しいシーケンス番号と、受信クレジット1を設定したAK TPDUを返送する。
- (2)正しいシーケンスのDT TPDUが到着したら、この正しいシーケンスの受信DT TPDUの受信処理を行い、次シーケンスのDT TPDUが保持されていないか確認する。もしも、次シーケンスのDT TPDUが保持されている場合は、このDT TPDUの受信処理を行う。

尚、(1)において1に減少させた受信クレジットは、誤り回復後、受信クレジットの値に比例した数だけの正しいシーケンスのDT TPDUを相手局から受信する度に、(制限値まで)1つつ増加させることにより回復を行う。このようなクレジットの回復方法は、中継装置等における輻輳制御のために有効であることがR.Jain[5]により報告されている。簡単なシミュレーションを行ったところ、同様の結果(図3-1、図3-2)が得られたため、本方式による再送手順を実装した。

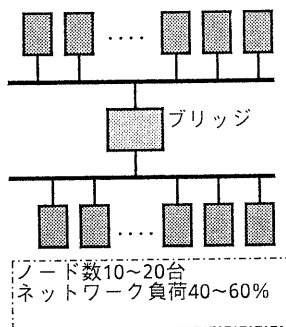


図3-1 シミュレーションモデル

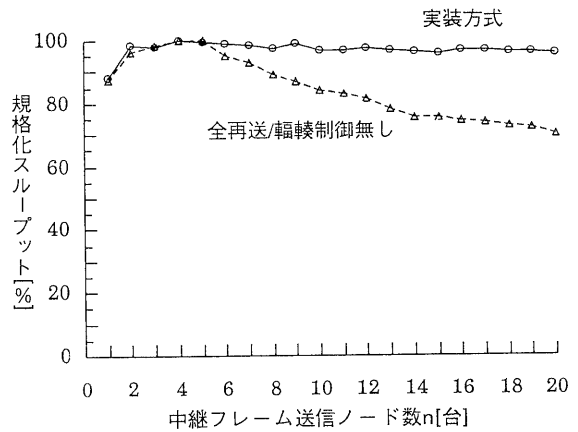


図3-2 シミュレーションの結果例

3.2 AK TPDUの連結送信

TCP等においては、受信データの確認情報を、送信するデータのヘッダに含ませPiggyback-Ack(肩寄せ確認)として返送できる。このような機能により、確認のためのフレームを単独で送信する場合に比べて小さな処理負荷により受信データの確認動作を行うことができる。OSI-TP4においては、DT TPDUのヘッダにこのようなPiggyback-Ackのためのフィールドが無い。そこで、AK TPDUを連結して送信することによりこのPiggyback-Ackの機能を実現した[6]。

AK TPDUの連結送信は、DT TPDUの送信モジュールに以下の処理を追加することにより実現した。

- (1)送信DT TPDUのヘッダ設定処理終了後に、未確認の受信DT TPDUがコネクションに存在しないか確認する。
- (2)未確認の受信DT TPDUが存在する場合、DT TPDUのヘッダを設定した送信バッファの先頭にAK TPDUを設定する。

トランスポート層において連結AK TPDUの送信処理にかかる時間は、AK TPDUを単独で送信する場合の約15[%]程度で済むことが評価の結果得られた(表3-1)。下位層の送信処理も考慮すると、大幅な効率改善となり得る。

表3-1 トランスポート層処理時間測定結果

	AK TPDU送信	AK連結処理
平均	551.0	83.9

単位[μsec]

4. 性能評価

4.1 評価方法

図4-1に、今回用いたソフトウェアの性能評価系を示す。10Mbpsの伝送路によって接続した2局のボード間の送受信において性能評価を行った。また、一方方向遅延の測定等のため、伝送路の一端にLANアナライザを接続し、2局間においてやり取りされるフレームをトレースした。

性能評価のためのテストプログラムは、ボード上に搭載した。すなわち、ボードの上位システムバスを介し

たボードの上位システムとのやり取りは、評価の対象外となっている。

4.2 測定結果及び考察

(1) 一方向遅延

一方向遅延の測定結果を図4-2に示す。

一方向遅延の測定結果は送信データ長1[Byte]で最短、7.0[msec]であった。

一方向遅延の増加は、送信データ長1[Byte]~32[KByte]にわたってほぼ直線的である。また、送信データ長の約8[KByte]毎に、一方向遅延が一度に3~4[msec]増加している。これは、送信データの増加8064[Byte]毎に、トランスポート層における送信データの分割が発生するためである。

送信データの分割が発生した際の一方遅延の増加には、トランスポート層における送受信処理時間の増加(0.9~1.4[msec])及び、ネットワーク層以下の送受信処理回数が増えることによる送受信処理時間の増加が含まれる。

(2) 送信スループット

送信スループットの測定結果を図4-3に示す。

送信データ長8[KByte]において4.7[Mbps]、送信データ長32[KByte]において6.5[Mbps]の送信スループットが得られた。

図より、一方向遅延の測定結果と同様、トランスポート層において送信データの分割が発生するたびに送信スループットが落ち込んでいることがわかる。

これらの落ち込みは、トランスポート層における送受信処理時間の増加及び、ネットワーク層以下の送受信処理回数が増えることによる送受信処理時間の増加によるものである。

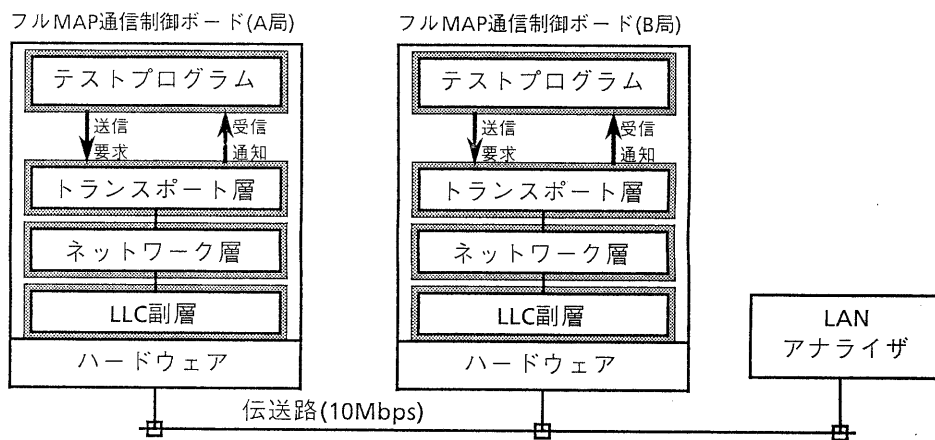


図4-1 フルMAP用通信制御ボード性能評価系

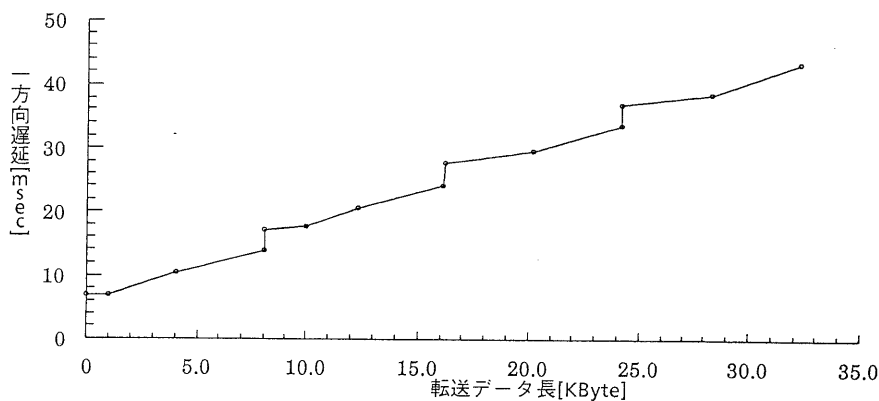


図4-2 転送データ長 - 一方向遅延

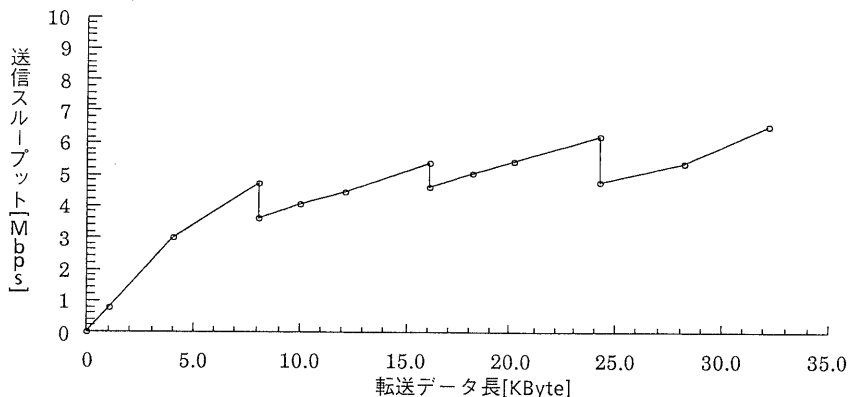


図4-3 転送データ長-送信スループット

5. 結果と考察

今回のトランスポート層プロトコル処理プログラムの実装においては、トランスポート層処理の高速化を中心に検討を行った。これらの検討に基づき、以下のような実装を行った。

- (1)コネクションへの関連づけ方式として、レファレンスの下位ビットからコネクション管理テーブルのアドレスを得ることのできる「サイクリックレファレンス管理方式」を実装した。
- (2)タイマ管理方式として、タイマ管理タスクが、一定周期毎にコネクション管理テーブルを検索し、タイマカウンタの減算を行う「カウンタ検索方式」を実装した。
- (3)TPDU受信の際には、受信バッファからトランスポート層のヘッダを一旦固定領域にコピーしてから解釈するようにした。
- (4)再送方式として、選択再送を実装した。再送発生後の受信クレジットの回復は、受信クレジットの値に比例した数だけの正しいシーケンスのDT TPDUを相手局から受信する度に、1つつ増加させることにより行う。
- (5)受信データの確認フレーム(AK TPDU)を送信するデータフレーム(DT TPDU)の先頭に連結して送信するPiggyback-Ack機能を実装した。

以上のような実装の結果、一方遅延遅延最小7.0[msec]、スループット最大6.5[Mbps]の性能を得た。

6. おわりに

本稿では、インテリジェントタイプのMAP通信制御ボードに実装したトランスポート層処理プログラムの実装方式を報告した。

今回試作したインテリジェントタイプのMAP通信制御ボードは、その後第5層(セッション層)までのソフトウェアの実装を完了した。しかしながら、製品化のためには、上位層の実装やホスト側ドライバ、層管理機能の整備等の課題を残している。又、近年のホスト側の環境の変化に伴い、インタフェースバス(Multibus-I)やCPU等のハードウェア仕様の再検討も必要になって来ている。今後これらの事項の見直しを行い、本試作ボードの製品化を進めたい。

参考文献

- [1]David D. Clark:An Analysis of TCP Processing Overhead, IEEE Communications Magazine, p.p.23-29, June 1989
- [2]瀬戸 他:トランスポート層クラス4プロトコルにおけるレファレンス管理方式の提案, 信学会秋季全国大会(1990年)B-492, 1990
- [3]木村 他:通信処理に適したタイマ管理方式の提案, 情報処理学会第39回全国大会7T-3, 1989
- [4]鈴木 他: Full-MAPプロトコルにおける層間インタフェース方式の一検討, 信学会春期全国大会(1990年)B-616, 1990
- [5]Raj Jain:A Timeout-Based Congestion Control Scheme for Window Flow Controlled Networks, IEEE Journal on Selected Areas in Communications, vol.SAC-4, No.7, p.p.1162-1167, Oct. 1986
- [6]瀬戸 他: 連結AKによるOSIトランスポート層処理の効率化, 情報処理学会春期全国大会(1991年)7T-6, 1991