

プロトコル変換プログラムの変換処理 共通化法の検討

北井 敦 大原 康博 河岡 司

NTT 電気通信研究所

異なるプロトコルに基づく通信システム間を相互接続する有効な一方法として、プロトコル変換技術がある。従来のプロトコル変換の検討は単一レイヤ、低位レイヤの変換が主で、階層化プロトコルの複数レイヤを対象にした変換、複数の通信プロファイルが許されている場合の多種プロファイル間の変換に関する検討はほとんど行われていない。

本稿では以下について考察している：

- ①階層化プロトコル間のプロトコル変換方式及びそれを実現するプロトコル変換システムの構成法、
- ②多種プロファイル間の変換システムをコンパクトに作成するアルゴリズム。

上記の検討を具体的にDCNA-SNAプロトコル変換システムに適用し、その結果コンパクト化のアルゴリズムにより約60%のモジュールが共通化可能であることを示している。

A study of the method making compact a multiple layered
protocols conversion system

Atsushi KITAI, Yasuhiro OHARA and Tsukasa KAWAOKA

NTT Electrical Communications Laboratories, 1-2356,

Take, Yokosuka-shi, Kanagawa, 238-03, Japan

The protocol conversion is a technique which enables an interconnection between heterogeneous computer communication systems. However, the reserach has not been made for the construction of multiple layered protocols conversion systems, especially for the conversion of protocols that contain many communication profiles.

This paper describes the constructing method of a multiple layered protocols conversion system, the minimization algorithm for the conversion of multiple layered protocols that contain many communication profiles, and the DCNA-SNA protocol conversion system implementation. The result shows that about 60 % of the modules are used in common.

1. まえがき

異なるプロトコルに基づく通信システム間を相互接続する有効な一つの方法として、プロトコル変換技術があり、いくつかの検討がされている。従来のプロトコル変換の研究の特徴は、伝送制御手順の変換を中心とした、単一レイヤ、低位レイヤの変換が主であった[1]。階層化プロトコルの変換については、概念レベルの考察は示されているが[2, 3]、具体的なプロトコル変換システム(方式)の構造についての検討はされていない。

本稿ではまず、2つの階層化プロトコル同志の変換を高位レイヤまで行うときの、プロトコル変換方式および、プロトコル変換システムの構成法について検討する。

次に実用的な階層化プロトコルを考えた場合、そのプロトコルに複数の通信プロファイルが許されている場合が多く、従って実用的なプロトコル変換システムではある特定のプロファイル間の変換を考えただけでは不十分であって、できるだけ多くのプロファイルの組み合わせで変換を行わなければならない。反面、複数プロファイル間の変換を実現すると、一般に変換プログラムの規模が増大する[5]。本稿では変換対象プロトコルが多種の通信プロファイルを許すとき、多種プロファイル間の通信を可能とするプロトコル変換システムをコンパクトに作成する一方法を提案し、具体的に、DCNA-SNAプロトコル変換システムに適用した例を示す。

2. 検討対象とするプロトコル変換方式とモデル化

2.1 プロトコル変換方式

(1) 直接変換方式と間接変換方式

一般に異なるプロトコル(XNA, YNA)間のプロトコル変換を行う場合、XNAとYNAを直接変換する方式と、中間プロトコルを介して間接的に変換する方法の二通りが知られていて、それぞれ評価されている[4]。本稿では、変換対象は限定されるが実用性を重視して性能面ですぐれている直接変換方式を前提に考える(図1)。

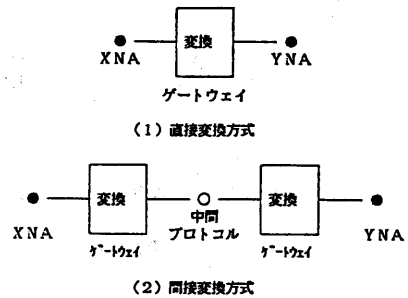


図1 プロトコル変換方式

(2) メッセージ中継方式とレイヤバイレイヤ変換方式

本稿では、OSIプロトコルと同様の階層化されたプロトコル(マルチレイヤプロトコルと呼ぶ)を対象とする。この場合、変換される機能(範囲)の観点から、プロトコル変換の基本方式として次の二方式が知られている[3](図2)。

①メッセージ中継方式

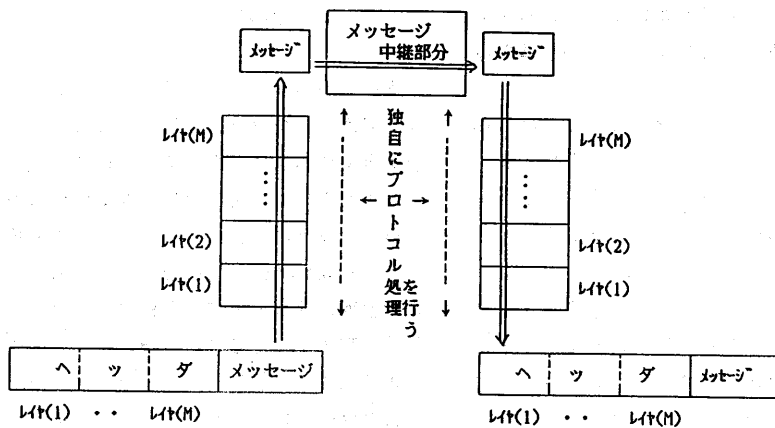
変換システムは最上位レイヤまでプロトコル処理を行いメッセージ単位で中継する。即ち、本方式では各NAのユーザは相手からのメッセージのみを送受信する事を要求する。従って、プロトコル情報(例えば、通信処理の間に発生したエラーは相手側には伝わらない。

②レイヤバイレイヤ変換方式

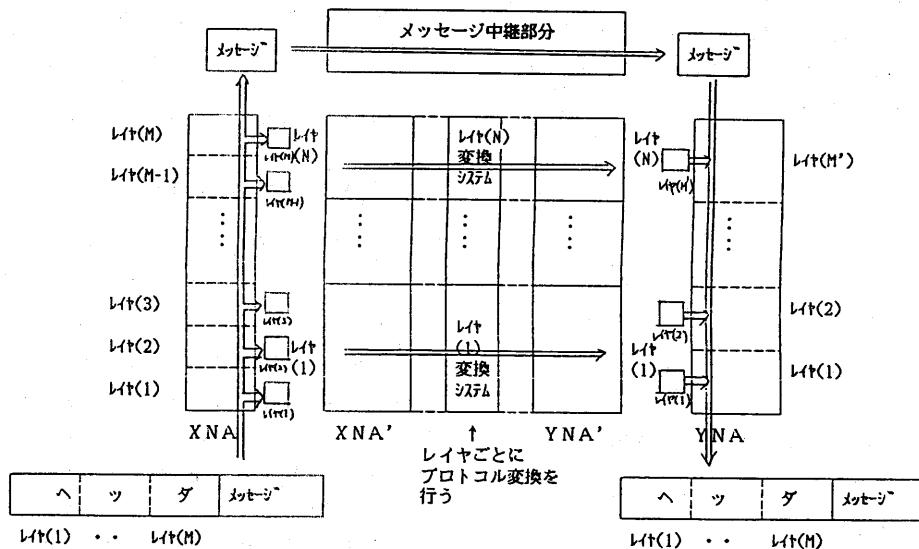
変換システムはレイヤごとにその機能を変換する。即ち、本方式ではプロトコル情報も相手側に伝わる。

マルチレイヤプロトコルではレイヤリングに機能的なずれが生じている可能性があるため、本方式を適用できるようにするため、レイヤの対応が1対1になるようにレイヤの再定義を行う(例えば対応するインタフェースではさまれるように、レイヤをグループ化する)。XNA, YNAからレイヤの再定義を行った変換用プロトコルをXNA'とYNA'とすれば、XNA'とYNA'の変換システムはレイヤバイレイヤ変換が適用できる。なお、本方式においてレイヤ(N)に関する変換の部分をレイヤ(N)変換システムと呼ぶ。

本稿では、②のレイヤバイレイヤ変換方式のプロトコル変換を対象とする。



(1) メッセージ中継方式



(2) レイヤバイレイヤ変換方式

図2 プロトコル変換方式

2.2 プロトコル変換システムのモデル化

一般にレイヤ (N) 変換システムはプロトコル処理部 ((N) PPE_x, (N) PPE_y: protocol preprocessing element of XNA, YNA) とプロトコル変換部 ((N) PCE: protocol conversion element) から構成される。ここでPPEとPCEの相互関係により次の二変換方式が考えられる(図3)。

(1) プリミティブインタフェース方式

PPEとPCEは疎結合であり、PPE_x, PPE_y, PCEは独立した有限状態機械 (FSM) となる。具体的には以下の性質を持つ。

- ・ (N) PPEは (N) SP (サービスプリミティブ) と (N) PDU (即ち (N) PCI と (N) SDU) を受信する。
- ・ (N) PCEは (N) PPE_x が出力する (N) SP_x を入力して (N) SP_y を出力する。逆方向も同様。

(2) PCIインタフェース方式

PCEとPPE_x, PPE_yは密結合で、全体で一つのFSMとなる。具体的には以下の性質を持つ。

- ・ (N) PPEは(N) SPと(N) PDU (即ち(N) PCIと(N) SDU)を受信する、
- ・ (N) PCEは(N) PCI_xを入力して(N) PCI_yに変換し出力する。逆方向も同様。

プリミティブインタフェース方式とPCIインタフェース方式は、それぞれ適応領域が有るが、本稿では、両方式の比較は議論せずそれらの変換FSMの構造について検討する。即ち、プリミティブインタフェース方式のPCE, PCIインタフェース方式のPCE, PPE_x, PPE_yを結合したFSMを変換FSM(図3の斜線部分)と定義し、この変換FSMのコンパクト化を検討する。後述の共通化についての議論はどちらの方式でも同じ議論が行える。

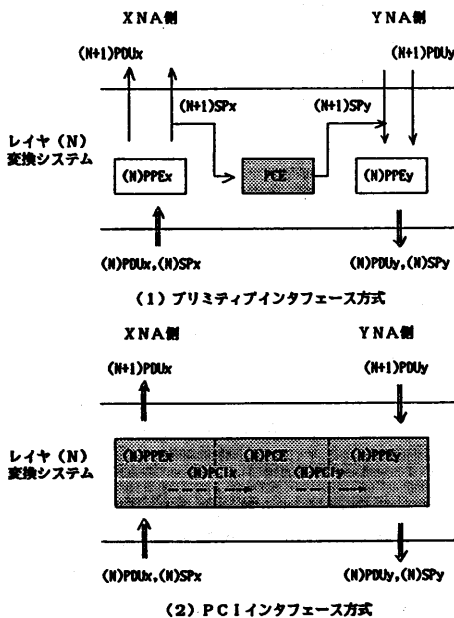
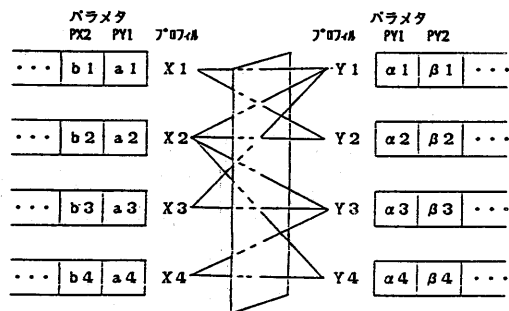


図3 プロトコル変換システムの内部構造

3. インタフェースの共通化の検討

3.1 共通化の考え方

実際の通信システムでは、さまざまな通信形態によって通信プロフィール(通信パラメタの実現値の組)を選択できるようになっているので、実用的なプロトコル変換では多種プロフィール間の変換を実現する必要がある(図4)。しかし、一方の変換対象プロフィールが変われば、その変換システム間では一般に変換論理は異なるので、多種プロフィール間の変換を同時に実現するときは、変換システムの規模が大変大きくなる。しかし、プロフィールが違っても変換論理は部分的に異なるだけで、共通化できる部分も存在すると考えられる。



[注] ai, bi, ai, βiはそれぞれPx1, Px2, Py1, Py2の実現値

図4 多種プロフィール間の変換

本稿では、変換システムの論理的動作を変換FSMとしてとらえ、その変換FSMの構成について考えている。変換FSMはコマンド/ヘッダ列を入力して、対応するコマンド/ヘッダ列を出力する有限状態機械と考えられる。即ち、次の定義・仮定をおく。

[定義] 対応するコマンド/ヘッダ対を変換単位と定義する

[仮定] 変換FSM全体の構成は、変換単位ごとにモジュール化されている

本稿では、多種プロフィール間の変換FSMを効率的に作成する方法として、あるプロフィール間の変換でそのなかのあるモジュールが他のプロフィール間の同一変換単位のモジュールと同じ変換論理のときに、そのモジュールを共有することにより全体の規模をコンパクト化する方法を提案する(図5)。

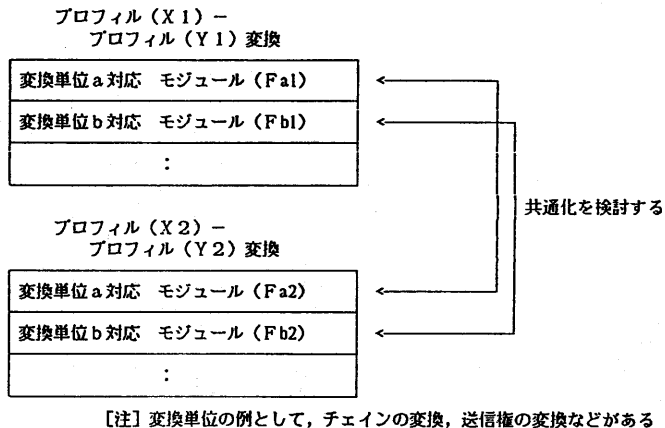


図5 変換プログラムのモジュール単位の共通化

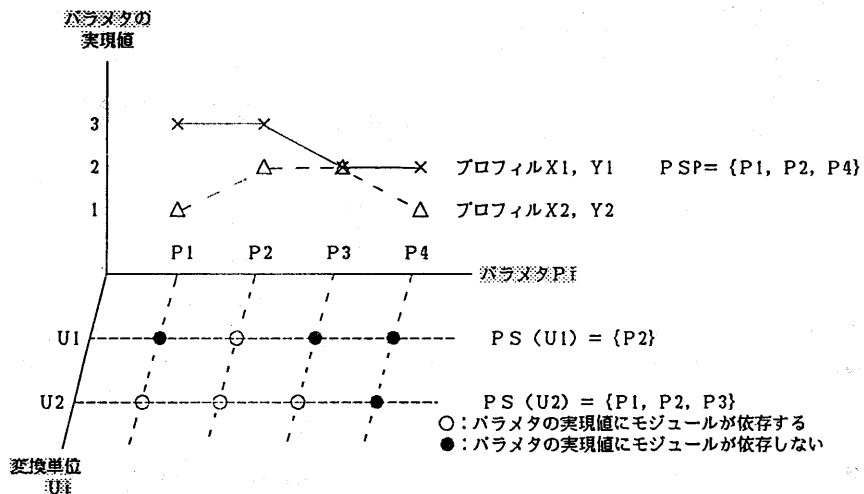


図6 モジュールとパラメタの実現値の関係

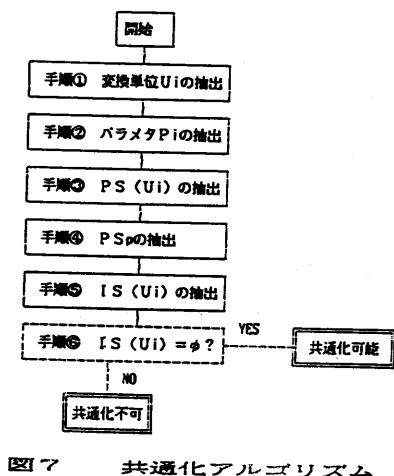
3.2 共通化アルゴリズム

変換単位に対応するモジュールは、プロフィールを構成するパラメタを引数とする。これらのパラメタは、実現値がモジュールに影響を与えるパラメタと与えないパラメタに分けられる。このうち、影響を与えるパラメタの実現値が異なるプロフィールを対象とする変換の間ではモジュールの共通化はできない(図6)。

2つのプロトコル(X; Y)の多種プロフィール(X1, X2等; Y1, Y2等)が与えられたとき、変換システム1(X1-Y1変換)と変換システム2(X2-Y2変換)で共通化可能な変換単位を全て求めるアルゴリズムは以下のようになる[5]。

【アルゴリズム】(図6, 7)

- 手順①ヘッダ情報/コマンド種別から変換単位(Ui)を抽出する。
- 手順②各プロフィールで選択可能なパラメタ(Pi)を抽出する。
- 手順③各変換単位: Uiごとに実現値の影響を受けるPiの集合(PS(Ui))を抽出する。
- 手順④プロフィールX1, X2間, Y1, Y2間で異なるPiの集合(PSp)を抽出する。そしてPS(Ui)とPSpの共通集合IS(Ui)を求める。
- 手順⑤IS(Ui)が要素を持たなければ、変換単位(Ui)に対応するモジュールは共通化できるIS(Ui)が一つでも要素を持てば、モジュ



ールは共通化できない。

手順⑥手順④から手順⑤をすべての交換単位Uiについて行う。

4. 適用例 (DCNA-SNA変換)

現在NTTではDCNA [6]と各種IBMプロトコルの交換システムを開発中である (図8)。このうちDCNA-SNA [7]変換では、多種プロファイル間の交換を対象とする。即ち、

- (1) DCNA: 仮想端末プロトコル (VTP), メッセージ転送プロトコル (MTP) を対象とする。更にその中でも可能なセッションパラメタの組合せを対象とする,
- (2) SNA: FMプロファイル3, 4, TSプロファイル3, 4の組合せを対象とする。

このDCNA-SNAプロトコル交換システムの設計に共通化アルゴリズムを適用した例のうち、DCNAのDUC層とSNAのDFC層変換 (OSIのレイヤ5相当のレイヤ)での共通化を述べる。なおDCNAとSNAの機能の対応を図9に示す。

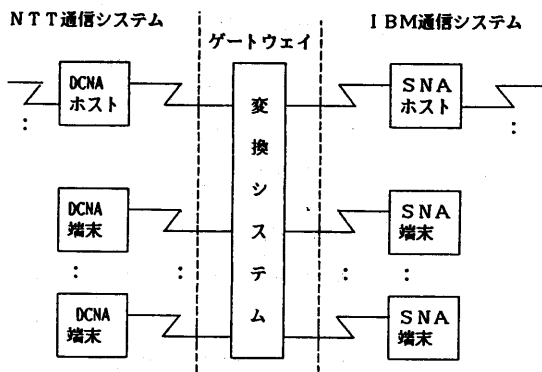


図8 適用モデル

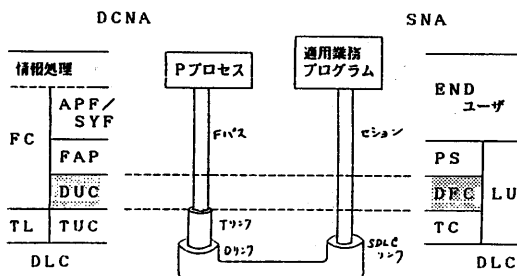


図9 DCNAとSNAの対応

4.1 共通化アルゴリズムの適用

3. 2のアルゴリズムを適用した例を示す。

手順①: 交換単位の抽出

(1) ヘッダ

ヘッダの各フィールド (チェーン, ブラケット, 応答要求) を1つの交換単位Uiとする。

(2) コマンド

変換するコマンド対を交換単位とする。

【例】QECコマンドの変換を図10に示す。図10から、QECの変換は、①VRSTAT, ②プロセス割込み, ③ALTERの3つのシーケンスに分類する事ができる。従って、QEC-VRSTAT, QEC-プロセス割込み, QEC-ALTERのそれぞれを交換単位Uiとする。

本稿の例ではUiの数は約40個であった。

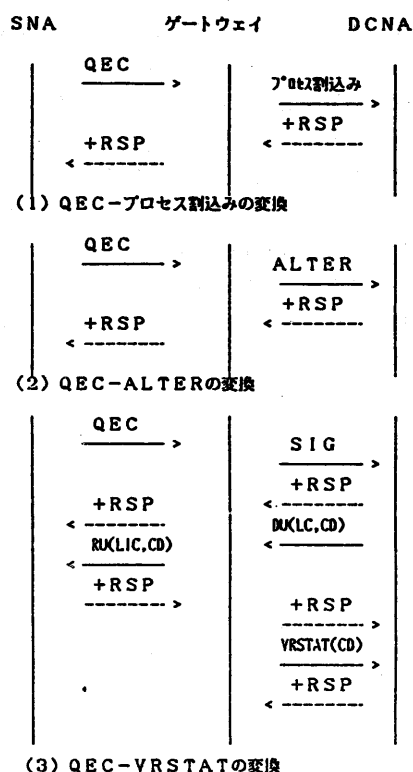


図 10 QECの変換

手順②：パラメタ抽出

(1) SNA側

SNAのセッションパラメタを取る。これは、BIND時に指定されるTS (Transmission Service) プロファイル、FM (Function Management) プロファイル、ブラケット使用法から成る。

(2) DCNA側

DCNAのセッションのパラメタを取る。これは、BIND時に指定されるDUC層プロファイル（送受信モード、応答モード等）から成る。

本稿の例ではパラメタ数は約10個であった。

手順③：依存するU_iの抽出

各変換単位毎に、U_iが変換論理に依存するかどうかを調べる。

【例】QECコマンドをVRSTATに変換する変換単位は、送受信モード、応答モード、SIG/VRSTATコマンドサポート/未サポートの各P_iで依存している。従って、PS (QEC-V

／VRSTATコマンドサポート/未サポートの各P_iで依存している。従って、PS (QEC-RSTAT) = {送受信モード、応答モード、SIG/VRSTATコマンドサポート/未サポート}となる。また、PS (QEC-プロセス割込み) = {SIGコマンドサポート/未サポート}

手順④：

プロフィールX1, X2間, Y1, Y2間で異なるP_iの集合 (P_{Sp}) を抽出し、共通集合IS (U_i) を求める。

【例】X1, X2, Y1, Y2が表1のようなプロフィールの場合、P_{Sp}は {送受信モード, TSプロフィール, FMプロフィール}、従って、手順③よりIS (QEC-VRSTAT) = {送受信モード}、IS (QEC-プロセス割り込み) = φとなる。

手順⑤：共通化の判定

IS (U_i) が空であれば、共通化できる。

この例では、X1-Y1変換システムとX2-Y2変換システムでは、QEC-プロセス割込み変換モジュールは共通化でき、QEC-VRSTAT変換モジュールは共通化できないと判定できる。

表1 プロフィールX1, X2, Y1, Y2の差

プロフィール	DCNAの差 (X1,X2の差)	SNAの差 (Y1,Y2の差)	
	送受信モード	TS プロフィール	FM プロフィール
X1, Y1	全二重	3	3
X2, Y2	半二重 アップ・アップ	4	4

4.2 アルゴリズム適用の結果

今回のDCNA-SNAプロトコル変換システムで許されている全てのプロフィール間変換の組合せについて上記の検討を行った結果、全体で約60%のモジュールが共通化可能であることが分かった。

5. おわりに

本稿では、多種のプロファイルを対象にした変換システムのコンパクト化を図る、システムティックな一方法を提案した。この方法は、以下のようないくつかの仮定をおいて議論を進めた。

- ① 個々の変換論理は既に与えられている。変換ロジックを代えれば更に共通化できる可能性がある。
- ② 変換単位に対応するモジュール単位の共通化である。
- ③ 同一プロトコル内の多種プロファイル間の変換システムの共通化を検討した。プロトコルが異なると、2つの変換システムが同一の変換単位を持つという前提を満たさないため。本稿の共通化アルゴリズムは適用できない。
- ④ モジュール単位の共通化の判定を変換単位が依存する項目／依存しない項目に分離して行う。ここで、依存／非依存は既に与えられている。依存／非依存を見つけるアルゴリズムの検討が今後の課題である。

これらの仮定を更に検討して、共通化アルゴリズムを詳細化することが今後必要となる。

また、本稿で示したモジュールの共通化は、階層化プロトコル同志の変換に限定した共通化であった。非階層化プロトコルにも適用できるように共通化法を拡張する検討が今後の課題である。

本稿の方式は、各種既存プロトコルとOSIの変換に適用可能であり、プロトコルのOSI化の推進に役立つものである。

参考文献

- [1] C. A. Sunshine : "Interconnection of computer networks", Computer Networks, 1, pp.175-195(1977).
- [2] 野口, 白鳥 : "プロトコル変換", 信学会誌, vol.69, No.2, pp.145-152(1986).
- [3] 梶原, 大原, 太田, "プロトコル変換方式の一検討", NTT研究実用化報告, 31, No.8, pp.1579~1590(1982).
- [4] 勝俣, 石川, 河岡, "OSIプロトコルを媒介とする異種ネットワーク間接続方式の検討——1. 変換の基本方針——", 第30回情報処理学会全国大会, pp.1045-1046(1985).
- [5] 北井, 大原, "汎用ミニコン上に実現可能なプロトコル変換プログラム構成法の検討", 61年度信学会全国大会, vol.8, p.9(1986).
- [6] NTT, "技術参考資料 DCNAマニユアル 全9冊", 日本データ通信協会, (1983).
- [7] IBM, "System Network Architecture Format and Protocol Reference Manual Architecture Logic"(1978).