

プロトコル変換の理論

奥村薫

日本アイ・ビー・エム株式会社
東京基礎研究所

異なるプロトコルに基づくネットワーク間を相互接続する為に、プロトコル変換の必要性が認識されてつつある。形式主義の立場から見ると、与えられたプロトコルに対して変換機と成るようなプロトコルを構築することとして問題を捕える事ができる。本稿ではプロトコル変換の形式的定義を与え、それに基づいて幾つかの定理を証明した。特にある条件下では、プロトコル変換機が存在判定ができて変換機を構成する手法があることを示した。

Protocol Conversion Theory

Kaoru Okumura

Tokyo Research Laboratory
IBM Japan, Ltd.
5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan

With the proliferation of heterogeneous networks, the need for protocol conversion has been recognized. From a formal view point, we regard that problem as demanding a protocol which satisfies the properties of the conversion. This paper proposes a definition of conversion and proves some theorems, especially the existence and construction method of converter.

0. 序

近年、情報通信の発達に伴って各種のネットワークが急速に普及しつつあり、そこでは異なったアーキテクチャによるネットワーク同志の遭遇に対応することが求められるようになってきた。この問題に対するひとつの解決法はプロトコルの標準化であり、全てのネットワークが標準プロトコルに従うことであろう。ISOによるOSIを中心とし、各標準化機関等で精力的な活動が行なわれており、かなりの層では標準が提案されつつある。しかしOSIに準拠したものでもプロフィール(機能集合)にはかなり選択の幅があり得るし、一方既に稼働中のネットワークも無視し得ない。標準化のみでは全ての問題が解決されないならば、変換という異なる通信システムをつなぐ手法を併用することになる。このように、複数のプロトコルが存在し今後も存在し続けるだろう状況で、プロトコル変換は重要な役割を担うことになると言えよう。

1. 研究の経緯

プロトコル変換は歴史の浅い分野であるが、最近関心が高まりつつあるよう思われる。個別のプロトコルを対象とした変換方式の提示や考察はかなり以前から行なわれていたが、変換一般に対する考察や理論化は最新の話題の一つである。[Green86]はこの動きにひとつのきっかけを与えた論文である。変換を形式的に記述する試みがTCPとOSI TP4 (level 4 class 4)等について行なわれている[Groenbeak 86]。わが国でもプロトコル変換に関する問題の分類、位置付けや、変換の実現方法の種類が詳しく検討されている[野口 86]、[北井86]。プロトコル変換への理論的側面からのアプローチも始められている。[Lam86]、[Okumura86]

2. プロトコルの表現とその意味

通常プロトコルは種々の仕様記述言語、オートマトン又は自然言語で表現されている。これらの表現は、あいまい性を含んでいる場合もあり得るが、今回プロトコル変換をどれほど形式的に行えるかを論じたいので、各々のプロトコルは形式的な記述法で厳密に表現されたところから出発する。ここでは表記の道具として最も一般的なもののひとつ cfsm¹を採用する。

¹ Communicating Finite State Machine. 有限オートマトンの一種。本稿では cfsm と略記する。

プロトコル変換の理論

もし各々のプロトコルが純粹に形式的に表記してあれば、その状態やメッセージに意味を読むことは不可能である。また一方の或るメッセージが他方のどのメッセージに対応するべきかなども原理的には分からない。そこで何らかのヒューリスティック・アプローチをとって、対話的に解を形造るか、或る形で意味を与えてやることになる。どのような方式で意味を与えるかは議論の有るところである。様相論理の式で、望ましい性質を定義するものもある。S. Lam は彼の projectin 及び image protocol の手法をもってプロトコルに意味を与えた [Lam84]、[Lam 86]。本稿第3節ではプロトコル表記と同じ cfsm でそれぞれのプロトコルの対応への要請を表現すことにする。

3. プロトコル変換の理論

3.1. 準備

定義1. プロトコル $A = [A_0, \dots, A_n]$ は n 個の cfsm の組である。プロトコル A 内の cfsm A_i とは次のような体系である:

cfsm A_i は 4 項組

$$A_i = (\sigma_i, M_i, \delta_i, q_i) \quad (i = 0, \dots, n)$$

であり、

- σ_i は状態 (states) の空でない有限集合。
- M_i は A_i が、やりとりするメッセージの有限集合で、他の cfsm A_j との入力 I_{ij} 、出力 O_{ij} の和である。即ち、

$$M_i = \bigcup_{j=0}^n I_{ij} \cup \bigcup_{j=0}^n O_{ij}$$

但し $I_{ii} = J_{ii} = \emptyset$ 。

- δ_i は $\sigma_i \times M_i$ から σ_i への部分関数であり、状態遷移関数 (transition function) とよばれる。
- q_i は σ_i の元で初期状態を示す。

ここで、 A_i の A_j への出力 O_{ij} と A_j の A_i からの入力 I_{ji} は等しく、その他の場合は O_{ij} 、 I_{ij} 、 σ_i は共通の元を持っていないものとする。

定義2. cfsm $X = [\sigma, M, \delta, q]$ において、メッセージの列 $\alpha = m_1 \dots m_k \in M^*$ が²

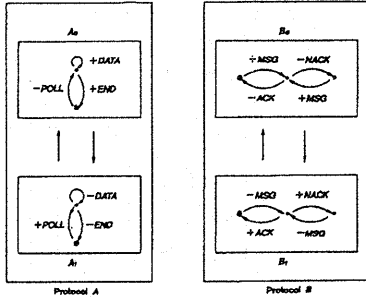


図 1: プロトコルの例 $A = [A_0, A_1]$, $B = [B_0, B_1]$.

$$\begin{aligned} \delta(q, m_1) &= s_1, \\ \delta(s_1, m_2) &= s_2, \\ &\vdots \\ \delta(s_{k-1}, m_k) &= s_k. \end{aligned}$$

を満たすとき、 α を cfsm X のパスと言ひ、 $\delta^*(q, \alpha) = s_k$ 又は単純に $\alpha \in \delta^*$ と書く。

定義 3. プロトコル $A = [A_0, \dots, A_n]$ の状態とは cfsm A_i の状態の組 $S = (s_1, \dots, s_n)$ ($s_i \in \sigma_i$, $i = 0, \dots, n$) である。プロトコルの状態遷移を次のように定める。プロトコルの状態 $S' = (s'_0, \dots, s'_n)$ が、ある i, j について、

$$\begin{aligned} s'_i &= \delta_i(s_i, -m), \\ s'_j &= \delta_j(s_j, +m), \\ s'_k &= s_k \quad (k \neq i, k \neq j). \end{aligned}$$

となるときに³, s' は s の次の状態であると言つて、 $\Delta(S, m) = S'$ と記す。これは cfsm A_i から cfsm A_j へメッセージ m が送られて、状態 S から S' に変わったことを表している。プロトコルのメッセージ列 $\alpha \in (\bigcup_i M_i)^*$ について、次のような状態の列 $S_1 \dots S_{k-1}$ が在るときに α は実行可能であるという： $\alpha = m_1 \dots m_k, S_0 = (q_0, \dots, q_n)$ (初期状態の組) とすると、

2 M の元を重複を許して並べた任意の有限列の集合を M^* と記す。

3 メッセージ m について $-m$ は m を送信する事、又 $+m$ は m を受信する事を示す。

プロトコル変換の理論

$$\begin{aligned} \Delta(S_0, m_1) &= S_1, \\ \Delta(S_1, m_2) &= S_2, \\ &\vdots \\ \Delta(S_{k-1}, m_k) &= S_k. \end{aligned}$$

この時特に $\Delta^*(\alpha) = S_k$ 又は $\alpha \in \Delta^*$ と記す。プロトコルの状態 S について、初期状態から S に到るメッセージ列が存在するときに、 S は到達可能な状態 (reachable state) であると言ふ。

定義 4. プロトコルの状態 S が次の状態を持たない時、即ち $\Delta(S, m)$ がどんなメッセージ m についても存在しない時に、 S はデッドロック状態であると言ふ。又どんな到達可能な状態もデッドロック状態でない場合に、そのプロトコルはデッドロックフリーであると言ふ。

定義 5. cfsm の状態 s について、 $\delta(s, +m)$ が定義されている時、状態 s はメッセージ m を受理可能であると言ふ。プロトコルの状態 $S = (s_0, \dots, s_n)$ で、任意の j ($j \neq i$), $m \in O_{ji}$ について $\delta_j(s_j, -m)$ が存在すれば $\delta_i(s_i, +m)$ も対応して存在する時に、cfsm A_i は状態 S で受理可能性を満たしている (unexpected input free) と言ふ。さらに、プロトコル A の任意の到達可能な状態で各 cfsm が受理可能性を満たしている時に、プロトコル A は受理可能性を満たすと言ふ。

3.2 プロトコル変換の定義

さてここでプロトコル変換のモデルを与えよう。始めに異なる2つのプロトコル $A = [A_0, A_1]$ と $B = [B_0, B_1]$ があるとす。 (図 2 参照)。プロトコルの変換とは、異なるアーキテクチャに従っている通信主体が (この図では A_0 と B_1 とする) 互いにコミュニケーションするのを可能にすることと考えて、プロトコル変換器 C を置こう。 C によって、 A_0 があたかも A_1 と通信している様に ($C + B_1$) と通信し、 B_1 にとっては ($C + A_0$) が B_0 の様に見えて、かつ互いに有意義な情報交換をすれば C は有効なプロトコル変換機であると言えよう。

これらを形式的に議論する為にプロトコル $P = [A_0, C, B_1]$ の性質をいくつか定める。

定義 6. 今プロトコル $P = [A_0, C, B_1]$ のなかで、cfsm A_0 (B_1) がプロトコル A (B) のなかにあるのと同様に動くことを要請したい。 P の任意の実行可

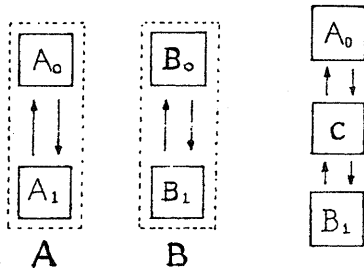


図 2: プロトコル変換のモデル

能なメッセージ列 α について、 A_0 に係わる部分 $\alpha|_{A_0}$ が 4 プロトコル A でも実行可能なメッセージ列であれば、(即ち $\alpha \in \Delta_P \rightarrow \alpha|_{A_0} \in \Delta_A$) C は外部的等価性を満たしているとする。 5

プロトコル A と B の意味的対応を次の様にして cfsm で表現する。まず、A, B のメッセージの中から意味的に重要であると思われるものを M_K と置いて ($M_K \subseteq M_i \cup M_j$) M_K の望ましい現れ方を cfsm K で定める。この K を核 (conversion seed) と呼ぶことにする。 6

定義 7. プロトコル A, B と核 K が与えられた時、プロトコル $P = [A_0, C, B_1]$ の任意の実行可能なメッセージ列 α について $\alpha|_K$ が cfsm K に受け入れられるとき、即ち $\alpha \in \Delta_P$ ならば $\alpha|_K \in \delta_K$ であるときに、C は意味的等価性を満たしているとする。

これら 2 つの等価性と、プロトコルが支障なく稼働するための要件、ここではデッドロックフリーと受理可能性をもってプロトコル変換機への要請としよう。

定義 8. 与えられたプロトコル A, B と核 K に対して、以下の 4 つの条件を満たすときに cfsm C がそれらのプロトコル変換機であると定める。

1. cfsm C は外部的等価性を満たす。
2. cfsm C は核 K について意味的等価性を満たす。

4 メッセージ列 α のうちプロトコル A のメッセージ M_A の元のみを取り出して並べた列を $\alpha|_A$ と書く。

5 ここではプロトコル A で実行可能なメッセージ列の全てがプロトコル P でも実行可能な事までは主張していない。

6 プロトコル A, B の意味を cfsm K で与えるのは、 M_K をアルファベットとする言語が正規言語程度の複雑さで実際の要求を表現できるだろうと考えるからである。

プロトコル変換の理論

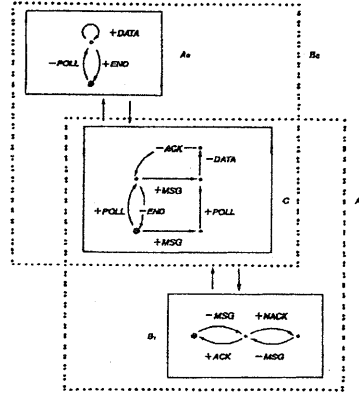


図 3: プロトコル A, B に対する変換機 C の例。

3. プロトコル $P = [A_0, C, B_1]$ はデッドロックフリーである。
4. プロトコル $P = [A_0, C, B_1]$ は受理可能性を満たす。

3.3. プロトコル変換機の実在について

この節では、プロトコル変換機の実在判定や構築の為に有用な定理を導く。

始めに cfsm 上の関数 ' \bullet ' 及び ' $*$ ' を定義する。

定義 9. 今 cfsm X 及び Y を

$$X = (\sigma_X, M_X, \delta_X, q_X)$$

$$Y = (\sigma_Y, M_Y, \delta_Y, q_Y)$$

とする。 $M_X \cap M_Y = \emptyset$ の時に $X \bullet Y$ が定義できて

$$X \bullet Y = (\sigma_X \times \sigma_Y, M_X \cup M_Y, \delta_{X \bullet Y}, (q_X, q_Y))$$

但し、 $\delta_{X \bullet Y}((s_X, s_Y), m)$ は X のメッセージ $m \in M_X$

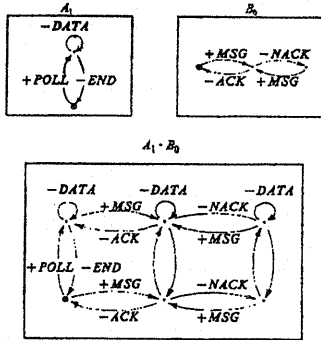


図 4: $A_1 \bullet B_0$

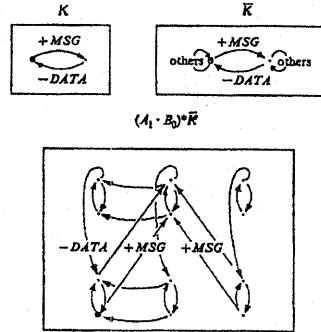


図 5: $\text{cfsm } (A_1 \bullet B_0) * \bar{K}$.

については $(\delta_X(s_X, m), s_Y)$, Y のメッセージ $m \in M_Y$ については $(s_X, \delta_Y(s_Y, m))$ を与えとする。
 また、 $M_X = M_Y = M$ の時に $X * Y$ が定義でき、

$$X * Y = (\sigma_X \times \sigma_Y, M, \delta_{X*Y}, (q_X, q_Y))$$

但し、 $\delta_{X*Y}((s_X, s_Y), m)$ は $\delta_X(s_X, m)$ と $\delta_Y(s_Y, m)$ がともに存在するときに限って定義され、

$$\delta_{X*Y}((s_X, s_Y), m) = (\delta_X(s_X, m), \delta_Y(s_Y, m)).$$

である。

関数 '•' の例として $A_1 \bullet B_0$ を図4に掲げる。

今、核 K に対してその入出力メッセージ M_K を $M_A \cup M_B$ まで自然に拡張して \bar{K} としよう。つまり、

$$\delta_{\bar{K}}(s, m) = \begin{cases} \delta_K(s, m) & \text{if } m \in M_K \\ s & \text{else} \end{cases}$$

とする。

7 cfsm Y が cfsm X の sub-cfsm であるとは、 Y の状態集合、メッセージ集合、状態遷移関数が、それぞれ X の状態集合、メッセージ集合、状態遷移関数の部分集合で、初期状態が等しい事を意味する。

$K = ((s, s'), (MSG, DATA, NACK),$
 $(\delta(s, MSG) = s', \delta(s', DATA) = s), s)$

プロトコル変換機が存在を調べる上で重要な性質を定理 1 で証明する。これは変換機を探すときの或る種の上限 $((A_1 \bullet B_0) * \bar{K})$ を与えるものである。(図 5 参照)

定理 1.

プロトコル A, B 及び核 K が与えられた時に、もしプロトコル変換機が存在するならば、 $(A_1 \bullet B_0) * \bar{K}$ の sub-cfsm⁷ でプロトコル変換機である cfsm D が存在する。

証明. cfsm C の全てのパスの集合を λ とし、 λ から $(\sigma_{A_1} \times \sigma_{B_0} \times \sigma_{\bar{K}})$ 上への写像 Γ を考える。

$$\lambda = \{\alpha \mid \alpha \in M^*, \alpha \in \Delta_C^*\}$$

$$\Gamma(\alpha) = (\delta_{A_1}^*(\alpha|_{A_1}), \delta_{B_0}^*(\alpha|_{B_0}), \delta_{\bar{K}}^*(\alpha))$$

cfsm の任意のパス α について $\Delta(\alpha)$ を定義可能である。何故なら、外部的等価性によって $\alpha|_A$ は A_1 ($\alpha|_B$ は B_0) に受入れられ、意味的等価性によって $\alpha|_K$ は K に受入れられるからである。 $(A_1 \bullet B_0) * \bar{K}$ の sub-cfsm D を

$$D = (\Gamma(\lambda), M, \delta_D, \Gamma(\phi))$$

とする。 $M = M_A \cup M_B$ 。状態遷移 δ_D は D の状

態 $\Gamma(\alpha)$ について、 $\alpha, m \in \delta_C^*$ の時に $\delta_D(\Gamma(\alpha), m) = \delta_C^*(\alpha, m)$ とする。プロトコル $[A_0, D, B_1]$ で実行可能なメッセージ列はプロトコル $[A_0, C, B_1]$ におけるものと同一なので、 D は外部的等価性、意味的等価性を満たす。又、 $[A_0, D, B_1]$ がデッドロックフリー及び受理可能性を満たす事も $[A_0, C, B_1]$ がそれらの性質を満たしている事から証明される。

よって $(a_1 \bullet b_0) * \bar{K}$ の各 cfsm がプロトコル変換機の条件 (1)-(4) を満たすか決定できれば、プロトコル変換機が存在するかどうか判定でき、又同時に変換機を与える事ができる。まず、受理可能性と関連した 2 つの性質を定義する。

定義 10. cfsm X がプロトコル Z 内で X の全てのパスが実行可能である、即ち、任意の $\alpha \in \delta_X^*$ に Z のパス $\beta \in \Delta_Z^*$ が存在して $\beta|_X = \alpha$ となる時、 X は Z においてエフェクティブ (effective) であると言う。さらに、プロトコル Z のすべての cfsm がエフェクティブである時に、プロトコル Z はエフェクティブであると言う。

定義 11. 2 つの cfsm X, Y があって、条件:

1. 全てのパス α について $\alpha \in \delta_Y^*$ ならば、 $\alpha \in \delta_X^*$ 。
2. 全てのパス α 、メッセージ m について $\alpha \in \delta_Y^*$ 、 $\alpha \in \delta_X^*$ ならば、 $(\alpha + m) \in \delta_Y^* \iff (\alpha + m) \in \delta_X^*$ 。

が成り立っている時、 Y を X の還元的 cfsm と呼ぶ。

上記の 2 つの性質と受理可能性について次の事が成り立つ。

定理 2.

- (1) cfsm X がプロトコル Z で受理可能性を満たしていると仮定する。この時 cfsm Y が X の還元的 cfsm であれば、 Y はプロトコル $Z[X/Y]$ ⁸ において受理可能性を満たす。
- (2) cfsm X がプロトコル Z でエフェクティブであると仮定する。この時 X の sub-cfsm Y がプロトコル $Z[X/Y]$ において受理可能性を満たしていれば、 Y は X の還元的 cfsm である。

この証明は各性質の定義を忠実に追うことによって得られる。次の定理 3 は定理 2 から直ちに得られる。

定理 3.

cfsm X がプロトコル Z 内で受理可能性を満たしエフェクティブであると仮定する。このとき X の sub-

cfsm Y がプロトコル $Z[X/Y]$ で受理可能性を満たす必要十分条件は Y が X の還元的 cfsm であること。

次にデッドロックフリーの判定に関する定理を述べる。

定理 4.

プロトコル A, B がエフェクティブで受理可能性を満たしており、cfsm C が $(A_1 \bullet B_0)$ の還元的 cfsm であるとする。プロトコル $[A_0, C, B_1]$ の到達可能な状態 (s_A, s_C, s_B) がデッドロック状態である必要十分条件は、 s_C が次の状態を持たない事である。

定理 5.

プロトコル A, B がエフェクティブかつ受理可能性を満たしている時、 A, B と与えられた核 K についてプロトコル変換機が存在するか決定できる。

証明. プロトコル変換機を求めるには定理 1 により、 $(A_1 \bullet B_0) * \bar{K}$ の sub-cfsm のみを調べれば十分である。 $(A_1 \bullet B_0) * \bar{K}$ の sub-cfsm は既にプロトコル変換機の条件 (1) 外部的等価性と (2) 意味的等価性を満たしている。 A_1, B_0 が A, B でエフェクティブで受理可能性を満たすので、 $(A_1 \bullet B_0)$ がプロトコル $[A_0, A_1 \bullet B_0, B_1]$ でエフェクティブかつ受理可能性を満たす事が判る。定理 3 によって C が受理可能性を満たすかどうかは、 C が $A_1 \bullet B_0$ の還元的 cfsm であるかによって定まり、又定理 4 からプロトコルがデッドロックフリーかは C に次の状態を持たない状態があるかに帰着される。これ等の条件は皆計算可能である。 $(A_1 \bullet B_0) * \bar{K}$ の sub-cfsm は有限なので、その中にプロトコル変換機が無ければ、与えられた A, B, K について変換機が存在しないと結論する事が出来る。

3.4. プロトコル変換機の構築

定理 5 においてプロトコル変換機が存在判定法を示したが、 $(A_1 \bullet B_1) * \bar{K}$ の sub-cfsm を全てチェックするのは有限とはいえ極めて非能率的であろう。実際に変換機を求めるにはより効率的な手法を取り得る。

(手法 I)

$(A_1 \bullet B_0) * \bar{K}$ から始めて、デッドロック状態や受理不能と成りうる状態を取り除いていく。これを繰返し適応してデッドロックフリーでかつ受理可能性を満たす cfsm が得られれば、それがプロトコル変換機である。この手法は変換機が存在するか否かの判

⁸ プロトコル $Z[X/Y]$ とはプロトコル Z の中の cfsm X を cfsm Y で置き換えたプロトコルを意味する。

定に向いている。何故ならバクトラックを行う必要が無く、変換機が存在しない場合には比較的早く全ての状態が取り除かれて空の cfsm に到るからである。又すぐわかる様に、与えられたプロトコル A, B と核 K に対するプロトコル変換機は一意では無い。できるだけ小さな変換機を求めるには、この手法はあまり適していない。

(手法 II)

意味的等価性を満たす1つのループに始まって、(少なくとも1つは K のメッセージを含んでいるとする) デッドロック状態を回避し、又受信し得るメッセージを受信できる様に状態や枝を付け加えてゆく。この手法は極小の変換機を求めるのに適しているだろう。但し試行錯誤を伴うので、探す順序によって又変換機が存在しなかった場合には多大な計算時間がかかることもあり得る。

4. 考察

以上、図 1, 2 のモデルと定義 8 のプロトコル変換機の定義を仮定して、変換機に関する定理、存在の判定と構築の指針について論じてきた。今後、これらの理論をより実用に近付けるには、プロトコルの意味的対応の表現方法と共に、より効果的なプロトコル変換機の構築方法やその為のツール作成等の開発と協同する必要があるだろう。

通信のモデルには cfsm 間に FIFO 待ち行列⁹を置くものも良く用いられている。これは待ち行列上のメッセージが有限個に限られているならば待ち行列無しのモデルに帰着できる事が知られているが、その際状態数の急激な増加をまねいてしまう。さらにこのモデルではエフェクティブでないプロトコルで表現されている事が多い。その分 cfsm がコンパクトになるからであろう。これ等のモデル上での理論も今後進めて行きたい。

(なお、この原稿の版下作成には、当研究所研究員山内長承氏による JANUS の漢字タグを利用した。)

文献

- [Green 86] P. E. Green, Jr.: Protocol Conversion. IEEE Transactions on Communications, Vol.COM-34, No.3 (March 1986).
- [Groenbeak 86] I. Groenbeak: Conversion Between the TCP and ISO Transport Protocols as a Method of Achieving Interoperability Between Data Communications Systems. IEEE Journal on Selected Areas in Communications, Vol.SAC-4, No.2 (March 1986).
- [北井 86] 北井敦, 大原康博, 河丘司: プロトコル変換プログラムの変換処理共通化の検討. 情報処理学会マルチメディア通信と分散処理研究会 MDP-30 (1986年7月).
- [Lam 84] S. S. Lam and A. U. Shankar: Protocol Verification via Projects. IEEE Transactions on Software Engineering, Vol. SE-10, No.4 (July 1984).
- [Lam 86] S. S. Lam: Protocol Conversion -- Correctness Problems. Proc. of the ACM SIGCOMM '86 (Aug. 1986).
- [野口 86] 野口正一, 白鳥則郎: プロトコル変換. 電気通信学会誌 Vol.69, No.2 pp.145-152 (1986年2月).
- [Okumura 86] K. Okumura: A Formal Protocol Conversion Method. Proc. of the ACM SIGCOMM '86 (Aug. 1986).

⁹ FIFO: First In First Out (先着順サービス)