論 文

EPN을 이용한 통신프로토콜 변환방식 설계

正會員 李 相 鎬*

Design of Commnication Protocol Conversion using EPN

Sang Ho Lee* *Regular Member*

要 約

서로 다른 프로토콜로서 운영되는 네트워크 간의 통신 문제를 해결하는 방법으로서 프로토콜 변환이라는 방법이 사용된다. 본 논문에서는 모형화 능력의 확장 및 변환 프로토콜의 성능 분석이 가능한 모형화 도구 EPN을 제안하고 이를 이용한 프로토콜 변환 방식을 설계한다. 제안한 변환 방식은 프로토콜 변환기의 동기화를 보장하고 오류복구 기능을 자동으로 추가시킨다. 따라서 이 변환 방식은 프로토콜 변환기 설계자의 설계 부담을 경감시키고 프로토콜 변환기 설계의 자동화를 가능하게 하며, 게이트 웨이 설계가 용이해진다.

ABSTRACT

Protocol conversion is to resolve the incompatibility between networks so that users on different protocols can communicate with each other. This paper designs a formal model, EPN, to increase its modeling power and analyze conversion protocol and proposes a conversion algorithm using it. The conversion algorithm has two capabilities : synchronization and automatic error recovery function. Therefore burden of protocol converter design will be reduced, automatic protocol converter design can be possible and implementation for half-gateway will be easy.

I. 서 론

오늘날 컴퓨터 네트워크 환경을 통한 각종 정보처리를 행하는 사용자들의 증가와 이에 대한 기술 발전으로 다양한 종류의 네트워크 구조와 프로토콜을 지닌 많은 네트워크 시스템들이 연구 개발되어지고 있다. 이와같은 네트워크의 구조와 통신 프로토콜의 확산으로 서로 다른 구조의 네트워크에 연결된 사용자들간의 통신을 보장하는 문제가 등장하게 되어[1,2] 이 문제를 해결하여 네트워크 유연성 증가를 통한 네트워크 이용의 극대화를 이룩하는 과제가 등장하였다. 서로 다른 네트워크를 통한 통신을 보장하기 위한 방법으로는 브리지(Bridge), 게이트웨이(Gateway), Intenetwork Protocol 등이 있으며, 이중 프로토콜 변환방식은 서로 다른 환경에서 통신하는 개체들, 예를들어 LAN-PSDN-LAN과 같이 논리적 연결성[3]을 갖지 못하는 개체들 간의 통신을 위한 게

*忠北大學校 컴퓨터科學科
論文番號 : 93-52

이트웨이 설계의 핵심 기술이다.

프로토콜 변환이란 서로 다른 문법에 의해 동작하는 통신 프로토콜 상호간에 전송되는 정보를 수신측에서 이해할 수 있는 정보로 변환시켜주는 과정을 의미하며 이를 수행하는 개체를 변환기(protocol conveter)라 한다. Green[3]은 IBM SNA 네트워크의 20,000 여 가입자, DECnet의 2,000 여 가입자등 현재 운용중인 컴퓨터 네트워크 자원의 방대성 때문에 전세계 공통인 통신 프로토콜의 구현은 사실상 불가능한 실정이므로, 서로 다른 프로토콜로서 운영되는 네트워크간의 통신 문제를 해결하는 방법으로서 프로토콜 변환 이라는 개념을 제안하였다. 그는 프로토콜 변환을 partial conversion, substitution과 complete conversion의 방법으로 구현할 수 있음을 보였다. Groenbak[1]에서는 미국방성의 표준프로토콜인 TCP와 NATO의 표준 프로토콜인 OSI 프로토콜 사이의 상호운영성을 제공하기 위하여 공통적 서비스 기능을 추출하여 변환 기법을 설계하였다. Auberbach[4]은 소프트웨어 라이브러리를 사용하여 변환을 수행하는 일종의 툴키트(toolkit)를 제안하였다. 이 방법은 트랜스포트 추상화 형(transport abstract type)이라는 개념을 이용하여 변환을 수행하나, 많은 프로토콜들에 대한 인터페이스 정보를 표현하는데 있어 한계성을 가진다. Lam[2]은 프로토콜 변환기를 이용하여 두 프로세스간의 상호작동성(interoperability)을 얻기 위하여 프로토콜 투사(projection)이론을 이용한 형식 모형을 제시하였다. Liu[5,6,7]는 두 프로토콜 사이의 의미적으로 동일한 메세지들에 대하여 동기화의 개념을 도입하여 변환 프로토콜을 생성하는 방법을 제안하였다. 이방법은 생성된 프로토콜에 대하여 성능분석등을 위한 모의실험이 어렵고, 오류복구기능을 설계자가 정확하게 기술하여야 한다는 단점이 있다. Okumura[8]는 두 프로토콜 사이에 전송되는 메세지의 의미들을 명시한 변환핵(conversion seed)을 기반으로 하여 변환을 수행하였다. 이 방법은 변환핵의 정의가 설계자의 관점에 의존한다는 점으로 변환기의 완전성을 검증하기가 곤란한 점이 있다. Kristol[9]은 서비스 수준에서의 프로토콜 변환을 위하여 프로토콜의 형식적 명세로부터 프로토콜 합성 기법을 적용하여 프로토콜 변환기를 추출할 수 있는 절차적 방법을 제안하였다. 이와 같은 프로토콜변환에 관한 문제는 1980년 중반에 대두된 것으로서 몇가지 특정 프로토콜 간의 변환기가 구현되었으나 이론적 체계가 미흡한 실정이다 [10,11].

프로토콜 변환의 기법에는 크게 서비스 수준에서의 변환[4]과 프로토콜 데이타 단위 수준에서의 변환[7]으로 나누어 진다. 서비스 수준에서의 변환 기법은 주어진 2개 이상의 상이한 프로토콜로부터 그들의 상위 계층에서 공통적으로 제공하는 서비스들을 서로 연결시키는 방법으로 이 방법은 장점은 주어진 프로토콜들의 하위 계층에서의 세밀한 PDU의 순서 또는 메세지들의 상호 교환과 연관된 제어를 프로토콜 설계자가 직접 고려하지 않아도 된다는 점이다. 프로토콜 데이타 단위 수준에서의 변환 기법은 주어진 2개의 상이한 프로토콜들 사이의 하위 계층에서 전송될 메세지 또는 데이타 단위 수준에서 나타나는 공통적인 가능성을 찾아서 변환을 수행하게 되며, 이 방법은 부분적인 경우이긴 하지만 생성된 변환기가 완전한 투명성(trasnsparency)을 지닐 수 있다는 점에서 가장 많이 접근을 하고 있는 기법이다. 이 기법에 기반을 두는 방법들은 주로 완전성을 지닌 메세지들의 논리적 표현을 위한 형식모델(formal model)[2]과 네트워크의 구조적 확장성과 유연성[2,10]을 증가 시켜주는 방법을 사용하고 있다.

프로토콜의 기능을 표현하기 위한 그래픽 모형화 도구로는 FSM(Finite State Machine), Petri Net [12], 언어 모형으로는 ESTELLE, LOTOS, SDL 등이 사용되고 있다. FSM은 사용의 편리성은 우수하나 실제 프로토콜의 경우 상태수가 너무 많아지고 채널의 명시적 표현이 어려운 단점이 있고, Petri net는 비동기적이고 비결정적인 특성을 갖는 시스템의 표현 및 분석에 적합하고 채널의 명시적 표현이 용이하나 상태수 문제는 FSM과 유사하다. 언어 모형의 일반적 특성은 표준 언어 그 자체에서는 현재까지는 정당성을 보장하는 구체적 방법이 없는 실정으로 표준 언어 그 자체에서는 표준 언어의 부분 집합을 정의하여 정당성을 검증하는 방법에 대한 연구가 세계적으로 진행중에 있는 실정이다. 본 연구에서는 모형화 능력의 확장성, 모형 검증의 용이성, 프로토콜 성능분석의 가능성 및 통신 채널의 명시적 표현이 가능한 측면에서 Petri Net를 확장하여 이에 통신 프로토콜의 변환에 있어 핵심인 변환기의 동기성을 표현할 수 있는 능력을 가지도록 EPN(Extended Petri Net)으로 정의하고, 이를 이용하여 프로토콜간의 변환 알고리즘을 설계한다. 제안하는 변환 알고리즘은 변환 프로토콜 설계자의 설계 부담을 덜어주기 위하여 변환 프로토콜에 오류복구 기능을 자동 삽입시키는 기능

을 갖는다. EPN을 이용함으로써 프로토콜 변환시 핵심이되는 메세지 변환에서의 동기화가 보장되고 통신 채널의 명시적 표현이 가능하여 기존 연구에서 취급하지 못하였던 프로토콜 변환기의 성능 분석 가능성을 가지며, 설계된 변환 프로토콜에 대한 논리적 정당성을 입증하기 위한 모의 실험 또한 가능하다.

Ⅱ. 프로토콜 변환

2.1 통신 프로토콜의 변환

일반적으로 통신 프로토콜들은 구문, 의미 및 시간의 중요 세가지 요소를 고려하여 표현되어야 한다[3]. 여기서 구문이란 데이타 양식, 코딩 방식과 신호 레벨 등을 기술하는 것이며, 의미란 통신 개체간의 조화있는 동작과 오류 제어 등을 위한 정보를 기술하는 것이다. 또 시간이란 각 통신 개체간의 속도조절, 순서화 및 흐름 제어 등에 관한 사항을 의미한다. 이와같은 특성을 가지는 통신 프로토콜을 표현하기 위한 도구에는 FSM, Petri Net, SDL, LOTOS, ESTELLE 및 프로그래밍 언어 등이 있다. Petri Net는 1962년 독일의 C. A. Petri에 의하여 고안된 모형화 도구로서, 비동기적이고 병렬성을 지니는 비결정적 속성을 갖는 시스템 특히 통신 프로토콜의 모형화 도구로서 많이 이용되고 있다. 본 연구에서는 표현 구조가 시각적으로 이해가 쉬우며 통신 채널의 명시적 표현이 용이한 장점을 가지며, 프로토콜 합성에 적합한 Petri Net를 프로토콜의 변환에서 핵심이 되는 변환기의 동기화 및 드 프로토콜간의 서로 다른 메세지에 대한 사상 기능을 수행하는 변환 함수 기능을 가지는 EPN을 정의한다. 이 EPN은 향후 각 트랜시션의 레이블에 확률 등의 개념을 삽입함으로써 프로토콜의 성능 분석이 가능해진다.

<정의 1 : Petri Net> PN=(P,T,A,μ)는 4-튜플로서 각 속성은 다음의 의미를 가진다.

P 는 플래이스(place)의 집합,

T 는 트랜지션(transition)의 집합,

A 는 입출력 함수로서 아크를 의미하며,

$A \subseteq I \cup O$, 단 $I=\{P \times T\}$, $O=\{P \times T\}$,

μ 는 마킹(marking)으로서 $\mu : P \rightarrow N$ (여기서 N은 음이 아닌 정수)□

PN은 유향 그래프 표현 방법에 근거하여 플래이스와 트랜지션과 아크들로 구성되며 플래이스는 원(circle)으로 트랜지션은 바(bar)로 표현된다. 트랜지션의 기능은 사건의 발생조건이 충족되면 입력 플래이스의 토큰(token)을 소비하여 출력 플래이스와 트랜지션을 연결시킴으로서 특정 시스템의 성질을 표현하게 한다.

<정의 2 : 점화 가능> 트랜지션 t가 마킹 μ에서 점화가능(enable)이란 트랜지션 t의 모든 입력 플래이스 p_i에 대하여 $\mu(p_i) \geq \#(p_i, I(t))$일 때이다. 여기서 $\#(p_i, I(t))$는 트랜지션 t의 플래이스 p_i에 대한 입력 아크의 수이다.

<정의 3 : 점화 규칙> 임의의 트랜지션 t에 대한 점화규칙(firing rule)은 다음과 같다.

1) 트랜시션 t는 점화가능인 경우에만 점화된다.

2) 점화 가능 트랜지션 t가 점화되면 t의 입력 플래이스로부터 입력 아크 갯수만큼의 토큰이 제거되고 t의 출력 플래이스에 출력 아크 갯수만큼의 토큰이 추가된다.□

따라서 트랜시션 t의 점화후의 마킹 μ'은 다음과 같이 된다.

$$\mu'(p_i) = \mu(p_i) - \#(p_i, I(t)) + \#(p_i, O(t))$$

<정의 4 : EPN> EPN=(P,T,A,μ,F,L)는 6-튜플로서 각 속성은 다음의 의미를 가진다.

P 는 플래이스의 집합,

T 는 트랜지션(transition)의 집합으로 메세지 송신(−) 또는 수신(+)을 나타내는 레이블을 가짐,

A 는 입출력 함수로서 아크를 의미하며

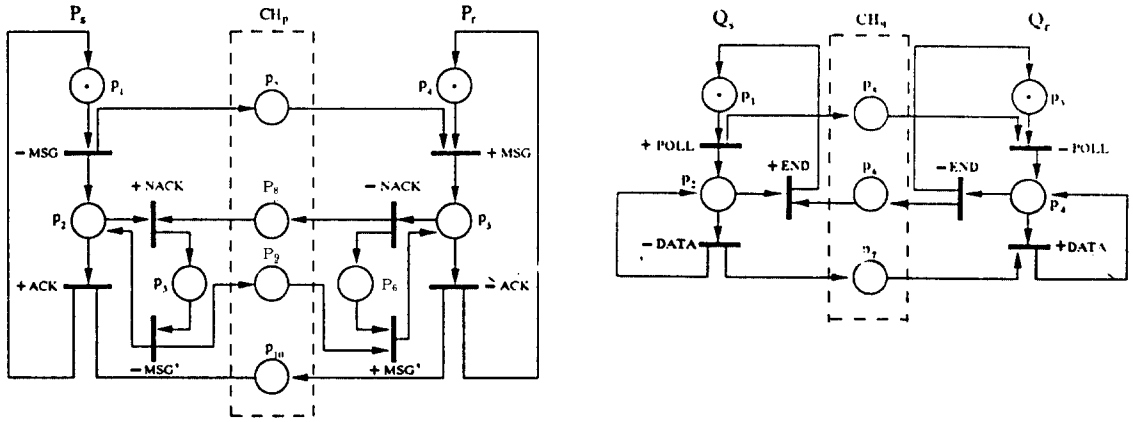
$A \subseteq I \cup O$, 단 $I=\{P \times T\}$, $O=\{P \times T\}$,

μ 는 마킹(marking)으로서 $\mu : P \rightarrow N$(여기서 N은 음이 아닌 정수),

F 는 메세지 변환 함수 PUT(m), GET(m/c)로서 트랜지션의 레이블. (여기서 m과 c는 메세지)

L 은 레이블 함수로서 L : 레이블 → T. □

그림 1은 통신 프로토콜의 EPN 표현이다[6]. 그림 1에서 두 프로토콜 $P=[P_s, CH_p, P_r]$과 $Q=[Q_s, CH_q, Q_r]$는 각각 단순 Stop-and-Wait와 Poll-End 프로토콜이다. 여기서 CH_p와 CH_q는 각각 프로토콜 P와 Q상의 통신 채널을 의미하고 P_s는 프로토콜 P의 송신부분을, P_r은 수신 부분을 나타낸다. 또한 Q_s는 포로토콜 Q의 송신부분을 Q_r은 수신부분을 나타낸다.



(a) P=[Ps, CHp, Pr]

(b) Q=[Qs, CHq, Qr]

그림 1. 두 프로토콜 P와 Q
Figure 1. Two protocol P and Q

2.2 프로토콜 변환의 정의

계층적 통신구조에서 사용자 프로세스가 논리적 연결성(logical connectivity)을 가지지 못하는 경우, 서로 다른 네트워크를 사용하는 이용자의 통신을 보장하기 위하여 프로토콜 변환은 필수적 과제이다. 프로세스 P_1과 P_2가 논리적으로 연결되어 있다는 것은

1) $P_1 = P_2$ 이거나 (각 프로세스는 자기 자신에게도 논리적으로 연결되어 있음),

2) P_1 과 P_2가 물리적 채널에 직접 연결되어 상호작용하거나

3) P_1 과 P_2가 상호 작용하여 P_1 과 P_2를 논리적으로 연결시키는 중간 프로세스 P_i ($P_1 \leftrightarrow P_i$, $P_i \leftrightarrow P_2$)가 존재하는 경우이다.

논리적 연결성이란 물리적 연결보다도 더 강한 조건으로서 공통 프로토콜이 아니며 서로작용하여야 하는 P_1 과 P_2 사이에 물리적 경로를 따라 중간 프로세스들이 존재하는지의 여부를 조사하는데 유용한 개념으로 쓰인다. 따라서 두 프로세스 P_1 과 P_2가 논리적으로 연결되지 않은 경우 중간 프로세스 P_i 를 통하여 P_1 과 P_2 가 통할 수 있도록 하는 일이 필요하며 이러한 일을 프로토콜 변환이라 한다[3].

2.3 프로토콜 변환 이론

프로토콜 변환 방식을 설계하기 위하여 메세지 사상 집합, 변환기에 대한 정의를 다음과 같이 한다.

<정의 5 : 메세지 사상 집합>두 통신 프로토콜 P와 Q의 메세지 집합을 각각 M_P와 M_Q라 하면, 메세지 사상 집합 T는 $\{<m_p, m_q> | m_p \in M_p, m_q \in M_q\}$로 정의되고, m_p, m_q는 각각 P와 Q의 중요 메세지 즉 변환이 필요한 메세지이다. T의 각 $<m_p, m_q>$는 정확하게 하나의 수신 메세지와 하나의 송신 메세지의 쌍으로 구성되며 서로 상대의 이메지(image)가 된다. 또한 m_p, m_q는 서로 의미적으로 일치하여야 한다. □

정의 5의 메세지 $<m_p, m_q>$는 EPN에서 트랜지션의 레이블 중 두 프로토콜에서 메시지 변환이 필요한 것들의 쌍들이다.

<정의 6 : 변환기> 변환기 C는 두 통신 프로토콜 P와 Q의 사이에 위치하며 EPN으로 표현되어 동기화를 보장하기 위한 GET, PUT함수를 가지며, 메세지 사상 집합 T에 대한 사상 기능을 수행한다. 즉 변환기와 두 통신 프로토콜의 관계는 $[P_s, P_r, C_f, Q_s, Q_r]$ 혹은 $[P_r, P_s, C_b, Q_r, Q_s]$로 정의될 수 있다. 여기서 C_f는 P에서 Q방향으로의 변환을 수행하는 변환기로서 $[P_r', Q_s']$이다. PUT(m) 함수는 P_r' 과 Q_s'의 동기화를 보장한다. □

프로토콜 변환 방식을 설계하기 위하여 서로 다른 문법을 지닌 프로토콜 $P = (P_s, CH_p, P_r)$와 $Q =$

(Q_s, CH_q, Q_r)에 의해 작동되는 2개의 상이한 네트워크 환경을 가정하여 프로토콜 변환의 개념을 제시하면 그림 2와 같다.

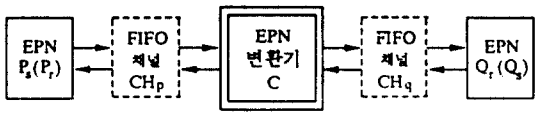


그림 2. 프로토콜 변환기의 개념
Fig 2. Concept of a Protocol Converter

Ⅲ. 변환기의 설계

변환기의 설계는 정의 5로부터 통신하고자하는 서로 다른 두 프로토콜 P와 Q를 이용하여 프로토콜 P와 Q간의 의미있는 메세지들의 모임인 메세지 사상 집합 T를 정의하여 이들로부터 동기화가 보장되고 오류회복기능을 갖는 변환기 C를 생성하는 일이다. 따라서 프로토콜 변환기의 설계 문제는 P_s로부터 P_r'을 또 Q_r로부터 Q_s' 및 P_r로부터 P_s'을 또 Q_s로부터 Q_r'을 올바르게 생성하는 일로 정의할 수 있다.

3.1 변환기의 속성

두 프로토콜 P와 Q에 대한 변환기 C는 두개의 프로토콜 C_f와 C_b로 구성된다. 즉 $C=[C_f, C_b]$ 이다. 여기서 프로토콜 C_f는 프로토콜 P의 메세지를 프로토콜 Q에서 이해할 수 있는 형태로 변환시키는 기능을 수행하며, 프로토콜 C_b는 프로토콜 Q의 메세지를 프로토콜 P에서 이해할 수 있는 형태로 변환시키는 기능을 기본으로 한다. 또한 변환기 C는 정상적으로 동작하여야 하며, 통신에 있어서의 오류 발생에 대한 회복 기능을 가져야 한다. 따라서 프로토콜 변환기 C가 가져야할 속성을 정리하면 다음과 같다.

1)C는 프로토콜 P로 부터 임의의 메세지를 송신(혹은 수신)할 수 있어야 하며, 또 프로토콜 Q로 부터 임의의 메세지를 송신(혹은 수신)할 수 있어야 한다.

2)C는 EPN의 메세지 변환 함수 PUT(m)과 GET(m/c)를 이용하여 어떤 변형을 가하여 이를 Q에 전달하고 또한 이것이 Q에 의하여 의미적으로 이해될 수 있도록 하여야 한다.

3)무정의 수신(unspecified reception) 오류가 없어야 한다.

4)무교착(deadlock free)이어야 한다.

<정의 7 : 전역 마킹>임의의 프로토콜 $P=[P_1, P_2, \ldots, P_n]$의 전역 마킹(marking) G는 ($[s_1, s_2, \ldots, s_n]$, $[q_1, q_2, \ldots, q_n]$)로 표현하고, s_i는 P_i의 현재 마킹을 q_i는 수신 P_i의 수신 채널의 메세지 내용을 뜻한다. 초기 전역 마킹 G_0는 모든 P_i의 초기 마킹들의 전역 마킹이며, G'은 G의 다음 마킹(next marking)으로 G>G'이 되기위하여는 각 P_i들이 점화가능하고 점화되어야 한다. □

<정의 8 : 다음상태 함수> 마킹이 μ이고 트랜지션 t에 대하여 다음상태 함수(next-sate function) δ: $N^n \times T \rightarrow N^n$는 모든 플래이스 p_i가 $\mu(p_i) \geq \#(p_i, I(t))$일 때 정의되고 $\delta(\mu,t)$가 정의되면 $\delta(\mu,t)=\mu'$이다. 단, $\mu'(p_i)=\mu(p_i)-\#(p_i, I(t))+\#(p_i, O(t))$이다. □

<정의 9 : 도달 가능> 전역 마킹 G_k는 초기 전역 마킹 G_0로 부터 $G_0>G_1>G_2>\ldots>G_k$이어야 도달 가능(reachable)이며, $G_0>^* G_k$ 로 표현한다. 또 순서(sequence)$<G_0, G_1, G_2, \ldots, G_k>$는 $G_0>G_1>G_2>,\ldots,>G_k$ 이고 $G_0=G_k$인 경우 P의 실행(execution)이다. □

<정의 10 : 무교착> 전역 마킹 $G=([s_1, s_2, \ldots, s_n])$, $[q_1, q_2, \ldots, q_n])$가 프로토콜 P의 도달가능 마킹인 경우, 모든 G가 점화 가능이면 무교착이다. □

3.2 변환 알고리즘

두 통신 프로토콜 P와 Q를 입력받아 프로토콜 변환기 C를 생성하는 변환 알고리즘의 개요는 다음과 같다.

1)오류 복구 기능이 명시되지 않은 두 통신 프로토콜 P_s(혹은 P_r)와 Q_r(혹은 Q_s)을 읽는다.

2)메세지 사상 집합 T를 기술한다.

3)메세지 사상 집합에 대한 조사를 수행한다.
메세지 사상 집합 T의 각 원소 $<m_p, m_q>$에 대하여 메세지 m_p의 다음 트랜지션들과 메세지 m_q의 다음 트랜지션의 집합이 서로 의미적으로 일치하지 않으면 메세지 사상 집합은 올바로 정의된 것이 아니므로 오류 메세지를 출력하고 끝낸다.

4)P_s(혹은 P_r)와 Q_r(혹은 Q_s)에 대한 프로토콜 P_r' (혹은 P_s')과 Q_s' (혹은 Q_r')을 생성한다. 이때 동기화가 보장되도록 메세지 사상 집합 T의 각 원소 $<m_p, m_q>$에 대하여 변환 함수 GET(m/c) 또는 PUT(m) 함수를 삽입한다.

5)오류복구 기능을 삽입한다.

6)생성된 P에서 Q(또는 Q에서 P) 방향으로의 변환기 C를 출력한다.

3.3 변환 프로토콜의 생성

입력된 두 프로토콜 P_s와 Q_r을 이용하여 변환 프로토콜 P_r' 과 Q_s'을 생성하는 방법은 다음과 같으며, 이 방법을 P_r 과 Q_s를 이용하여 변환 프로토콜 P_s' 과 Q_r'을 생성하면 변환기 C를 얻게 된다.

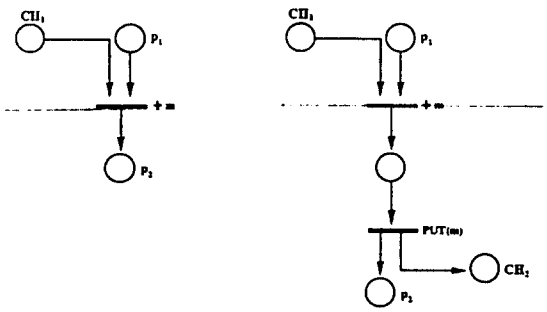
1)기본 프로토콜 P_s와 Q_r에 대한 각 트랜지션에 대하여 송신은 수신으로 수신은 송신으로 트랜지션의 레이블을 변경하여 1 단계 P_r' 과 Q_s'을 얻는다.

2)변환 함수 GET(m/c)와 PUT(m)은 내부 변환 프로토콜 P_r' 과 Q_s' 사이의 의미있는 메세지 변환을 의하여 변환기 상에 삽입하게 되며, 이들 함수를 삽입함으로써 두 변환 프로토콜 사이에서 동기화를 보장 받게 된다[5]. 변환 함수의 삽입 방법은 다음과 같다.

i)PUT(m) 함수는 의미있는 메세지(m)들이 주어진 변환 프로토콜 P_r' 과 Q_s' 사이에서 의미적 동치성의 보장을 위하여 추가된다. 즉, 변환 프로토콜 중 프로토콜 P_r'이 메세지 사상 집합 T에 명시된 의미있는 메세지(m)들을 수신하여 이를 상대 변환 프로토콜 Q_s'에 올바로 송신하도록 추가된다. 따라서 P_r' 의 의미있는 메세지의 수신 즉 +m 레이블을 가진 트랜지션의 바로 뒤에 PUT(m) 트랜지션을 추가 한다. 그림 3에서 플래이스명 CH_1은 기본 프로토콜 P_s와 변환 프로토콜 P_r'사이의 통신 채널을 의미하며, 플래이스명 CH_2는 변환 프로토콜 Q_s'과 기본 프로토콜 Q_r사이의 통신 채널을 의미한다.

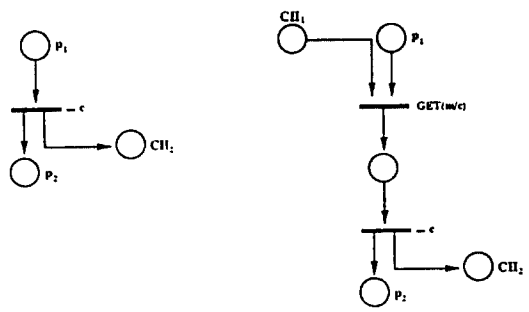
ii)GET(m/c) 함수는 의미있는 수신 메세지에 대하여, 변환 프로토콜 P_r에서 PUT(m)함수를 사용하여 채널을 통해 전송된 메세지를 수신측 변환 프로토콜 Q_s이 수신하여 기본 프로토콜의 수신측(Q_r)에서 사용할 수 있는 메세지로의 문법적 변환(syntatic transformation)을 수행하기 위하여 삽입한다. 따라서 Q_s'의 의미있는 메세지 수신 즉, -c레이블을 가진 트랜지션의 바로 앞에 GET(m/c) 트랜지션을 추가 한다. 그림 4의 (a)에서 플래이스명 CH_2는 변환 프로토콜 Q_s'과 기본 프로토콜 Q_r사이의 통신 채널을 의미하며, (b)에서 플래이스명 CH_1은 변환 프로토콜 P_r' 과 Q_s' 사이의 통신 채널을 의미하며, 플래이스명 CH_2는 변환 프로토콜 Q_s과 기본 프로토콜 Q_r사이의 통신 채널을 의미한다.

3)오류 복구 기능은 변환기 내의 변환 프로토콜에



(a)삽입전 (b)삽입후

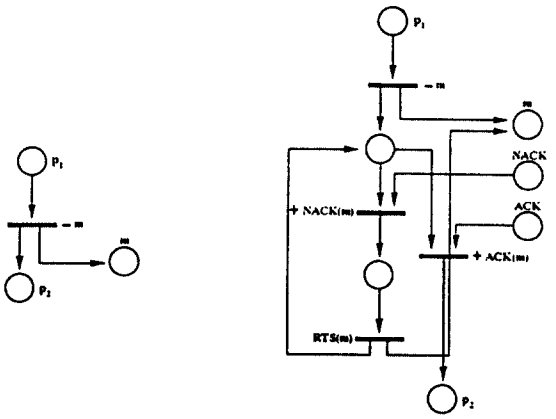
그림 3. PUT 함수의 삽입
Fig 3. Insert a PUT function



(a)삽입전 (b)삽입후

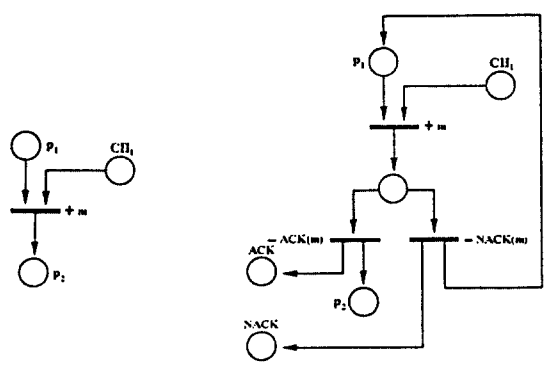
그림 4. GET 함수의 삽입
Fig 4. Insert a GET function

ARQ(Automatic Repeat Request) 기법에 기초하여 자동적으로 생성하여 삽입한다. ARQ 방식에 기초한 오류 복구 기능을 지닌 변환기는 의미있는 메세지의 송신, 수신과 동기화를 위하여 추가된 GET(m/c) 및 PUT(m)에 대하여 오류 복구 함수 ERF ={ACK(m), NACK(m), RTS(m)}를 변환 프로토콜 P_r' 과 Q_s' 상에 그림 6과 7과 같이 삽입한다. 여기서 ACK(m)은 메세지 m에 대한 양의 응답을, NACK(m)은 음의 응답을, RTS(m)은 메세지 m에 대한 재전송 요구를 수행하는 함수이다. 그림 5와 6에서 플래이스명 m, ACK, NACK는 모두 통신 채널로서 각각 메세지, 양의 응답 및 음의 응답을 나타낸다. 또한 그림 6의 플래이스명 CH_1은 변환 프로토콜과 기본 프로토콜 사이의 통신 채널을 의미한다.



(a)삽입전 (b)삽입후

그림 5. 오류복구 기능의 삽입(송신)
Fig 5. Insert an error recovery function(send)

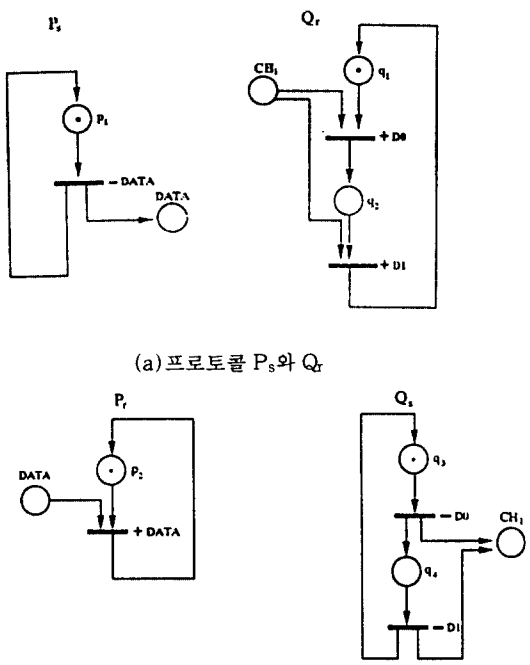


(a)삽입전 (b)삽입후

그림 6. 오류복구 기능의 삽입(수신)
Fig 6. Insert a error recovery function(receive)

Ⅳ. 변환 방식의 적용

Liu[5]의 서로 다른 윈도우 크기를 갖는 두 프로토콜을 이용하여 제안한 변환 알고리즘을 적용하기로 한다. 프로토콜 P에서 Q방향으로의 프로토콜 P_s와 Q_r을 생각하면 그림 7(a)와 같다. 이 그림에서 플래이스 DATA와 CH_1은 각각 프로토콜 P와 Q 내의 통신 채널을 나타낸다. 그림 7에서의 메세지 사상 집합 T는 {<DATA, D_0>, <DATA, D_1>}로 정의되며, 그림 7(a)의 프로토콜 P_s와 Q_r에 대하여 제안 변환 알고리즘을 적용한 결과인 변환 프로토콜은 P_r'과 Q_s'로서 P에서 Q방향으로의 변환기이며 내용은 그림 8과 같다. 완전한 프로토콜 변환기는 P에서 Q방향뿐만 아니라 Q에서 P 방향으로도 작동하여야 하므로 Q_s, P_r에 대하여도 3.3절의 알고리즘을 적용하여 완전한 프로토콜 변환기 C를 얻게 된다. 본 연구에서는 이 두 과정이 적용 방법에 있어 동일한 것이므로 P_s에서 Q_r의 변환기 즉, P_s로 부터 DATA를 수신(P_r')하여 이를 Q 프로토콜이 인식할 수 있도록 의미 변환을 시킨후 D_0 혹은 D_1을 Q_r로 송신 (Q_r')시키는 [P_r', Q_s']만을 생각하였다. 그림 8의 [P_r', Q_s']이 얻어지는 주요 생성 과정을 3.3절의 변환 프로토콜의 생성 알고리즘을 이용하여 기술하면 다음과 같다.



(a)프로토콜 P_s와 Q_r

(b)프로토콜 P_r와 Q_s

그림 7. 대상 프로토콜 P와 Q
Fig 7. Object protocol P and Q

1) 정의된 메세지 사상 집합 {<DATA, D_0>, <DATA, D_1>}에 대하여 DATA의 트랜지션 -DATA와 D_0, D_1의 트랜지션 $+D_0$, $+D_1$이 서로 의미적으로 일치하여 이들이 메세지의 변환 대상이다.

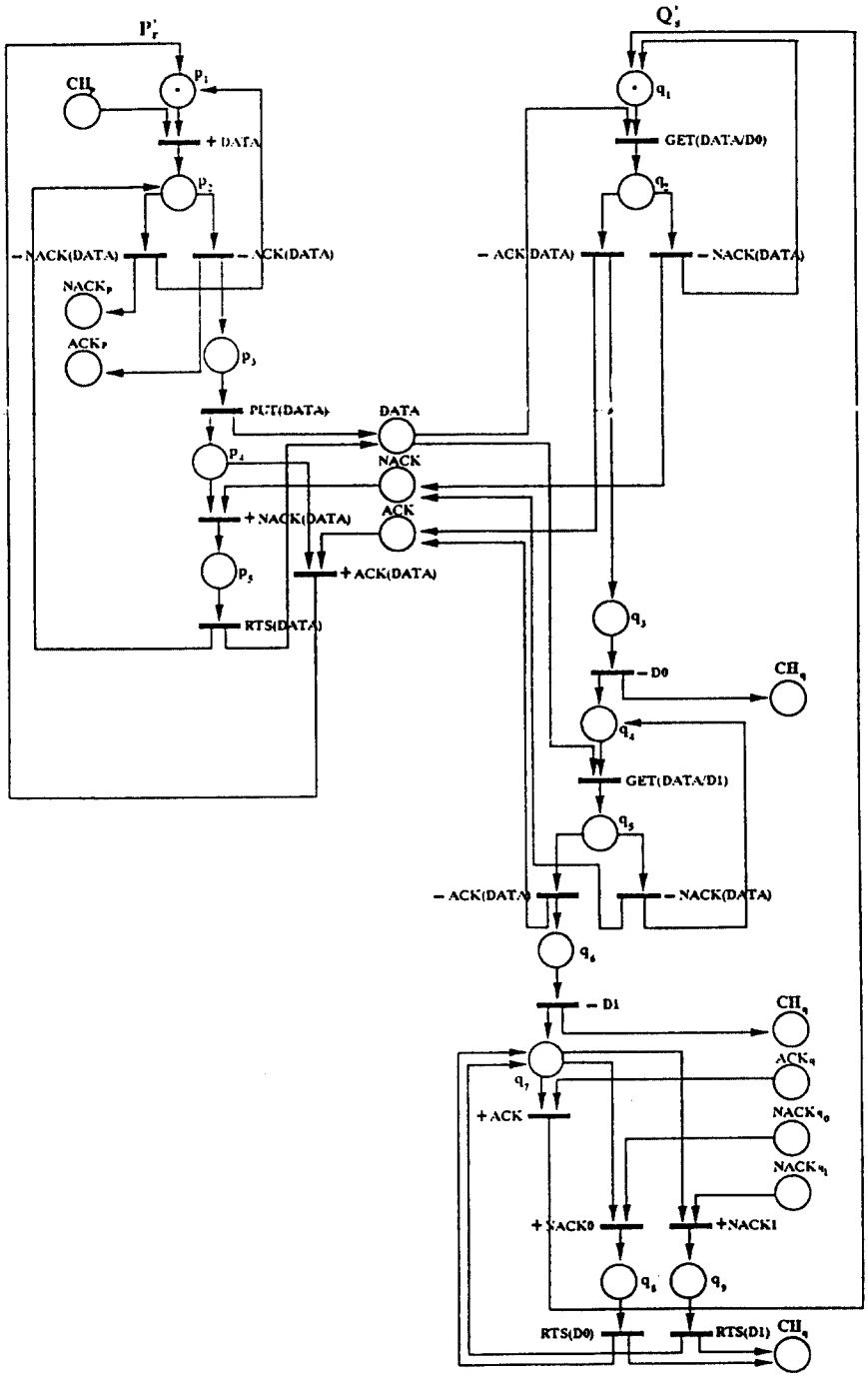


그림 8. 변환 프로토콜 P_r'와 Q_s'

Fig 8. Conversion protocol P_r' and Q_s'

2) 주어진 P_s와 Q_r에 대하여 각각 변환 프로토콜 P_r' 과 Q_s'을 생성하며 P_s와 P_s', Q_r과 Q_s'의 관계는 역함수 관계로 송신은 수신으로, 수신은 송신으로 변환시킴으로써 얻어진다.

3) 단계 2)에서 얻어진 P_r'과 Q_s'에 대해 각각 +DATA 레이블을 가진 트랜지션 바로 뒤에 PUT (DATA) 트랜지션 추가하고 $-D_0$와 $-D_1$ 레이블을 가진 트랜지션 바로 앞에 GET(DATA/D_0)와 GET(DATA/D_1) 트랜지션을 각각 추가 시킨다.

4) ARQ기법에 기초한 오류 복구 기능은 P_r'의 +DATA 트랜지션이나 Q_s'의 GET(DATA/D_0), GET(DATA/D_1) 트랜지션의 바로 뒤에 그림 6과 같이 수신용 오류 복구 기능을, P_r'의 PUT(DATA) 트랜지션이나 Q_s'의 $-D_0$, $-D_1$ 트랜지션의 뒤에 그림 5의 송신용 오류 복구 기능을 추가시킨다.

그림 8에서 두 플래이스명에 대한 의미는 다음과 같다. CH_p, $NACK_p$, ACK_p는 기본 프로토콜 P_s와 변환 프로토콜 P_r' 사이의 통신 채널을 나타내는 플래이스이고, DATA, NACK, ACK 변환 프로토콜 P_r'와 Q_s' 사이의 채널을 나타내는 플래이스들이다. 또한 플래이스 CH_q, ACK_q, $NACK_{q0}$, $NACK_{q1}$은 변환 프로토콜 Q_s'과 기본 프로토콜 Q_r 사이의 채널을 의미한다. 그림 8의 변환 프로토콜은 Liu의 연구 결과[5]의 EPN 표현 결과와 일치하므로 그 정당성이 입증된다.

V. 결 론

정보 사회의 발달에 컴퓨터 네트워크가 차지하는 비중은 매우크다. 따라서 서로 다른 구조를 가지는 컴퓨터 네트워크 접속된 사용자들 간의 원할한 통신을 보장하는 문제는 필수적으로 해결하여야 할 과제이다. 서로 다른 네트워크를 통한 통신을 보장하기 위한 방법중 게이트웨이가 대표적 방법으로, 프로토콜 변환은 게이트웨이 설계의 핵심 기술이다.

본 논문에서는 통신 프로토콜의 비동기성, 통신 채널의 명시적 표현 능력 및 성능 분석 가능성 등을 고려하여 PN을 모형화 도구로 선택한 후 이를 확장하여 변환 프로토콜을 보다 잘 표현할 수 있는 EPN을 제안하였다. 또한 EPN을 이용하여 주어진 기본 프로토콜의 메세지 전이만을 가지고 자동적으로 오류 복구 기능을 삽입하고, 변환 함수 GET과 PUT을 이용함으로써 동기화를 유지할 수 있는 변환기를 생성해 낼 수 있는 프로토콜 변환 알고리즘을 설계하였다.

이 방법의 주된 장점은 이 기종간의 통신을 위한 게이트웨이상에서의 핵심이 되는 변환기의 설계가 손쉬워 짐에 따라 게이트웨이 개발 비용이 절감될 수 있을 것이며, 프로토콜 변환기의 성능 분석이 가능해진다. 또한 다수의 프로세스간의 통신 환경에 대한 통신 프로토콜의 설계 및 검증에도 활용될 수 있을 것이다. 그러나 이 방법은 실제의 통신 시스템이 복잡하게 될 경우에는 EPN의 속성에 의하여 생성되는 변환 프로토콜의 크기가 방대해질 수 있다는 점이 단점이다. 앞으로 다자간 통신에서의 프로토콜 변환기 설계, 프로토콜 변환기 설계의 자동화 및 성능 분석 등이 연구되어야 할 과제이다.

이 논문은 1991년도 교육부 지원 학술진흥재단의 자유공모 (지방대 육성) 과제 학술 연구 조성비에 의하여 연구되었음

참 고 문 헌

1. I. Groenback, "Conversion between the TCP and Transport Protocols as a Method of Achieving Interoperability between Data Communication Systems," IEEE Joural on Selected Areas in Communications, Vol. SAC-4, No. 2, pp. 288-296, Feb. 1986.
2. S.S. Lam, "Protocol conversion," IEEE Trans on Software Engineering, Vol. 14, pp. 288-353-362, Mar. 1988.
3. P.E. Green, "Protocol Conversion," IEEE Trans on Communication, Vol. 34, No. 3, pp. 288-296, Mar. 1986.
4. J. Auberbach, "A Protocol Conversion Software Toolkit," Proceeding ACM SIGCOM '89, Austin, Texas, pp. 259-270, Sep. 1989.
5. J.C. Shu and M.T. Liu, "A Synchronization Model for Protocol Converison," Proceeding of IEEE INFOCOM '89, Ottwa, pp. 276-284, Apr. 1989.
6. Y. Yao, W.S. Chen and M.T. Liu, "A Parallel Model For Constructing Protocol Converters," Proceeding of IEEE GLOBECOM '90, San Diego, pp.1902-1907, Dec. 1990.
7. Y. Yao and M.T. Liu, "Constructing Protocol

Converters with Guarnteed Service," Proceeding of IEEE INFOCOM '91, Bal Harbour, pp. 960-969, Apr. 1991.
8. K. Okumura, "A Formal Protocol Conversion Method," Proceeding of ACM SIGCOM '86, Stow, pp. 30-37, Aug. 1986.
9. D.M. Kristol, D. Lee, A.N. Netravali and K.M. Sabnani, "Efficient Gateway Synthesis From Specifications," Proceeding of ACM SIGCOMM '91, Switzerland, pp.89-97, Sep. 1991.
10. M. Rajagopal and R.E. Miller, "Sythesizing a Protocol Converter from Executable Protocol Traces," IEEE Trans on Computers, Vol. 40, No. 4, pp. 487-499, Apr. 1991.
11. G. Bochman and P. Mondain-Monval, "Design Principles for Communication Gateways," IEEE Journal on Selected Areas of Communications, Vol. ASC-8, No. 1, pp. 12-21, Jan. 1990.
12. J.L. Peterson, Petri Net Theory and Modeling of System, Prentice-Hall, 1981.

李 相 鎭(Sang Ho Lee) 終身會員
1976年 2月 : 崇實大學校 電子計算學科 卒業(工學士)
1981年 2月 : 崇實大學校 大學院 電子計算學科(工學碩士)
1989年 2月 : 崇實大學校 大學院 電子計算學科(工學博士)
1976年~1979年 4月 : 韓國電力公社 電子計算所 勤務
1981年 3月 以後 現在 : 忠北大學校 컴퓨터科學科 副敎授
1992年 9月 以後 現在 : CANADA UBC POST-DOC.
※關心分野 : 프로토콜 工學, 暗號學 및 시뮬레이션 等