

TCP의 계층적 페트리 넷 모형

임재걸*, 김환근*

*동국대학교 컴퓨터멀티미디어학과
e-mail:{yim, ducice}@dongguk.ac.kr

A Hierarchical Petri Net Model of TCP

Jaegel Yim*, Hwanken Kim*

*Dept of Computer and Multimedia, Dongguk University

요 약

무선통신 환경에서 TCP의 효율성을 개선하기 위한 ECN 방법을 하나의 페트리 넷 모형으로 표현한 사례가 있다. 이 경우에는 모형이 너무 복잡하여 ECN 방법의 구조를 한눈으로 파악하기 어렵다는 단점이 있다. 본, 논문은 기존의 모형을 추상화하여 계층적으로 표현한 결과를 보인다. 계층적 표현은 시스템 전체 구조를 이해하기가 용이하다는 장점이 있다..

1. 서론

트랜스미션 콘트롤 프로토콜 (TCP:Transmission Control Protocol)은 프로세스간의 신뢰성 높은 연결을 구축해 주는 통신 규약으로, 현재 인터넷 통신량의 대부분이 TCP정보이다. 무선 인터넷, 무선 LAN, 무선통신이 보편화 되어감에 따라 무선통신 환경에서 TCP의 효율성을 개선하기 위한 연구가 활발히 진행되고 있다. 이러한 연구는 TCP의 효율성을 개선하는 방안을 제안하고, 제안한 방안이 정말로 효율성을 개선한다는 것을 증명하는 방식으로 진행된다. 효율성 개선을 증명하는 방법으로 수식을 이용하는 방법, 제안하는 방안을 프로그램으로 구현하는 방법, 시뮬레이션 방법 등이 주로 사용된다.

시뮬레이션 방법에서 주로 사용되는 도구에는 ns-2[1]가 있으나, 일반적인 모델링 및 시뮬레이션 도구인 페트리 넷 [2]이 쓰이는 경우도 적지 않다. 최근에 Design/CPN [3]이라는 페트리 넷 모델 구축 및 시뮬레이션 소프트웨어를 이용한 TCP의 ECN(ECN: Explicit Congestion Notification) [4-6] 방식의 모형 구축 및 시뮬레이션 분석 결과가 [7]에 소개되었다.

[7]에 소개된 페트리 넷 모형은 무선통신 환경에서 TCP의 효율성을 개선하기 위한 ECN 방법을 하나의 페이지에 표현하여, ECN 방법의 구조를 한눈으로 파악하기 어렵다는 단점이 있다. 본, 논문은 [7]에 소개된 모형을 추상화하여 계층적으로 표현한다. 계층적 표현은 시스템 전체 구조를 이해하기가 용이하다는 장점이 있다.

지면의 제한으로 페트리 넷 관련 기본 용어와 개념

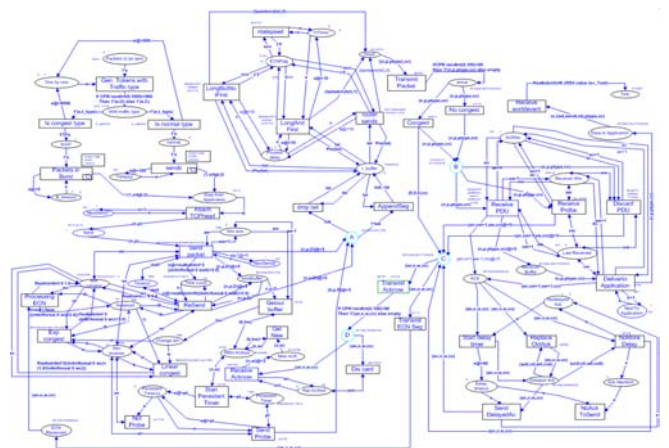
에 대한 소개는 생략한다. 관심 있는 독자는 참고 문헌 [2]와 [3]을 참조하기 바란다.

2. 기존의 연구

본 절에서는 기존의 연구로 [7]에 소개된 모형과 페트리 넷 계층화 방법에 대하여 소개한다.

2.1 기존의TCP모형

기존의 TCP 모형은 [그림 1]과 같이, 세그먼트 생성부, 송신 TCP, 네트워크부, 수신 TCP로 크게 구분될 수 있으나, 하나의 페이지에 복잡하게 그려져 있어 이해하기가 어렵다. 본 논문에서는 이 모형을 보다 간결하고 단순하게 표현할 수 있도록 계층 구조로 변환한다.



[그림 1] TCP의 페트리 넷 모형

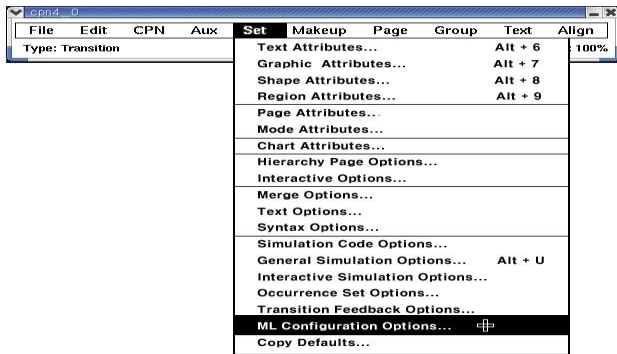
2.2 페트리넷 계층화 방법

Design/CPN을 설치하기 위한 운영체제로는 리눅스와 솔라리스가 있다. 본 연구에서는 레드햇 리

눅스 9.0을 채택한다. 레드햇 리눅스는 이미 배포 판이 웹상에 많이 있으므로 다운로드 경로[8]와 설치에 대한 정보는 생략한다. Design/CPN[3] 설치는 [3]을 방문하여, 'DesignCPN-4.0.5-1.i386.rpm'을 다운로드하여, 더블 클릭하면 된다.

Design/CPN을 실행하면 [그림 2]와 같은 주메뉴바가 화면에 나타난다. [그림 2]는 주메뉴 중 Set가 선택되고, Set의 풀다운 메뉴 중 'ML Configuration Options...'이 선택된 상태를 보인다. 모델을 구축한 다음 컴파일을 하려면 'ML Configuration Options...'를 선택하여 다음과 같은 사항들을 지정해 주어야 한다.

- 'Local' or 'Remote' : 인터프리터를 가지고 있는 컴퓨터의 위치를 가리키는 것으로 지역형과 원격형으로 나뉜.
- Hostname : 인터프리터를 가지고 있는 컴퓨터의 주소. (ex: localhost.localdomain)
- Port number : 인터프리터를 실행하기 위한 포트 중의 하나.(ex: 2097)
- Runtime : 시뮬레이션 실행 파일 경로 (ex :/usr/lib/cpn/run.x86-linux)
- ML image: 메타언어의 인터프리터가 있는 위치 (ex: /usr/lib/cpn/cpn.ML)

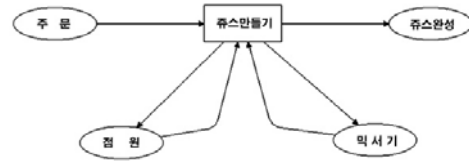


[그림 2] Design/CPN의 주메뉴

계층화한 주어진 페트리 넷의 일부(부페트리 넷)를 하나의 트랜지션으로 대체하는 것이다. 이때, 대체되는 트랜지션을 대체 트랜지션 (substitution transition)이라고 하며, 하나의 변천으로 대체된 부페트리 넷은 하위페이지(subpage)라고 일컫는 다른 페이지에 분리되어 표현된다.

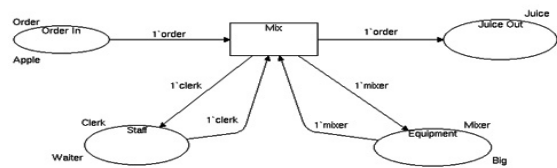
예를 들어, 다음과 같은 사건이 있다고 가정하자: [사과주스 전문점에서 손님의 주문을 받으면, 여러 종업원 중 하나가 믹서를 사용하여 사과주스를 만들어 컵에 포장하여 내준다.]

위와 같은 관계를 페트리 넷으로 도식해보면 [그림 3]과 같다.



[그림 3] 사과주스 전문점의 주문받기 과정

[그림 3]을 Design/CPN을 이용하여 보다 구체적으로 표현하면 [그림 4]와 같다.



[그림 4] Design/CPN을 이용하여 구체화한 모형

손님(Order In)이 사과주스를 주문하면, 종업원 한 사람이 믹서를 차지하여 사과주스를 만들어 한 컵을 생산한 다음, 종업원과 믹서가 다음 일을 기다리며 쉬는 상태로 되돌아가는 것을 나타낸다.

여기서 우리가 집중해야 할 부분은 'Mix'라고 이름 지어진 트랜지션이다. 'Mix'는 종업원이 사과를 가지고 와서, 껍질을 벗기고, 육질부분을 깎뚝썰기하여, 믹서에 넣고 갈고, 거르는 일련의 사과주스 만드는 공정이 함축되어 있다. 이는 이러한 과정이 필요 없다는 뜻이 아니라 과정을 숨긴다는 뜻이다. 바로 이것이 계층화이다.

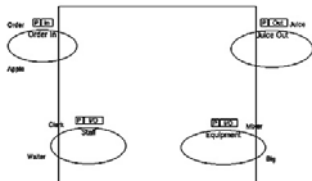
Design/CPN에서 계층화하는 과정은 다음과 같다.

① **Create Decomposition Page:** [그림 4]에서 트랜지션 'Mix'를 클릭하여 선택하고, 선택한 상태에서 주메뉴 'CPN'의 풀다운 메뉴 'Move to Subpage'를 클릭한다. 'Move To Subpage'는 'Decomposition Page'를 생성한다. Decomposition Page란, prime page의 한 트랜지션의 구체적인 구조를 명시하는 하위페이지이다. 즉, 트랜지션 'Mix'에 대한 구체적인 구조를 명시할 수 있는 페이지이다.

② **Set Subpage:** 트랜지션 'Mix'를 더블클릭하여, [그림 5]과 같은 'Subpage'를 열 수 있다.

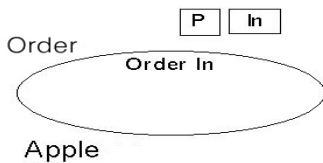
③ **Set Page Attributes:** 트랜지션 'Mix'에 대한 구체적인 구조를 명시하기 전에 주메뉴의 'Set'의 풀다운 메뉴에서 'Page Attributes'를 선택하면,

나타나는 대화상자에서 다음과 같은 페이지의 속성을 지정할 수 있다: Current Page Name, Current Page Number, Palette Page, Change Current Page, Save as Defaults, Border Size, Page Borders



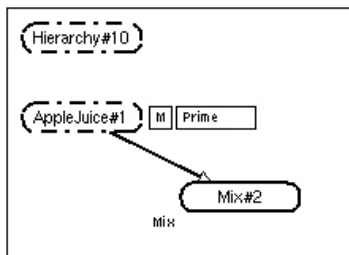
[그림 5] Subpage

④ **port key region:** 포트는 상위페이지와 하위 페이지를 연관시키는 플레이스다. 본 예에서는 모든 플레이스들이 하위페이지와 연관되어 있기 때문에 모두 포트이다. 포트 플레이스에는 [그림 6]에 보이는 바와 같이 포트임을 나타내는 P라는 글자가 쓰인 상자와 입·출력을 표시하는 상자가 나타나는데 이 상자들을 port key region이라 한다.



[그림 6] port 플레이스의 예

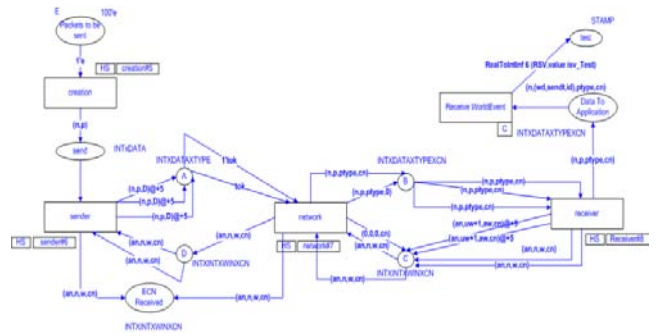
④ **Hierarchy Page :** Hierarchy Page는 [그림 7]처럼 상위 페이지와 하위 페이지와의 관계를 간단하게 보여준다.



[그림 7] Hierarchy Page

3. TCP의 계층화

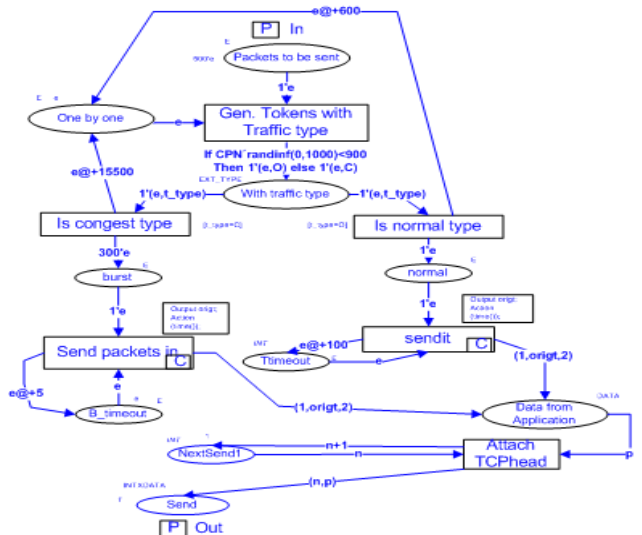
이제까지 간단한 예제를 통해서 계층화하는 방법에 대해 알아보았다. 이제 본격적으로 TCP의 페트리 넷 모형, [그림 1]을 계층화해보자. 계층화 과정을 보이기 전에 계층화 결과를 먼저 보이면 [그림 8]과 같다. [그림 8]에는 다음과 같은 4 개의 대치 트랜지션(substitution transition)을 포함한다: 생성부(creation), 송신부(sender), 네트워크부(network), 수신부(receiver).



[그림 8] 계층화된 TCP의 페트리넷

1) 생성부(creation)

[그림 8]의 트랜지션 creation은 creation#5라는 하위페이지로 대치되며, creation#5의 내용은 [그림 9]와 같다. [그림 9]는 트랜지션 creation이 어떻게 세그먼트를 생성하여 플레이스 send로 내보내는지 상세한 과정을 보인다.



[그림 9] creation#5에 구조화된 모형

2) 송신부(sender)

[그림 8]의 'sender'는 sender#6라는 하위페이지로 대치되며, sender#6의 내용은 [그림 10]과 같다. [그림 10]은 트랜지션 sender가 다음과 같은 주요 트랜지션들을 포함함을 상세히 나타낸다: Send Packet, ReSend, Receive Acknow, Send Probe.

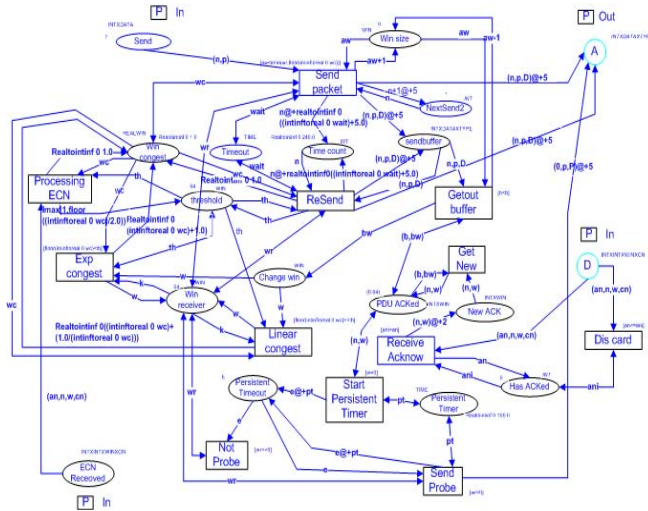
3) 네트워크부(network)

[그림 8]의 트랜지션 network는 network#7라는 하위페이지로 대치되며, network#7의 내용은 [그림 11]과 같다. [그림 11]은 트랜지션 Network이 어떻게 패킷을 전송하고 통신 혼잡을 판단하는지 상세히 보인다.

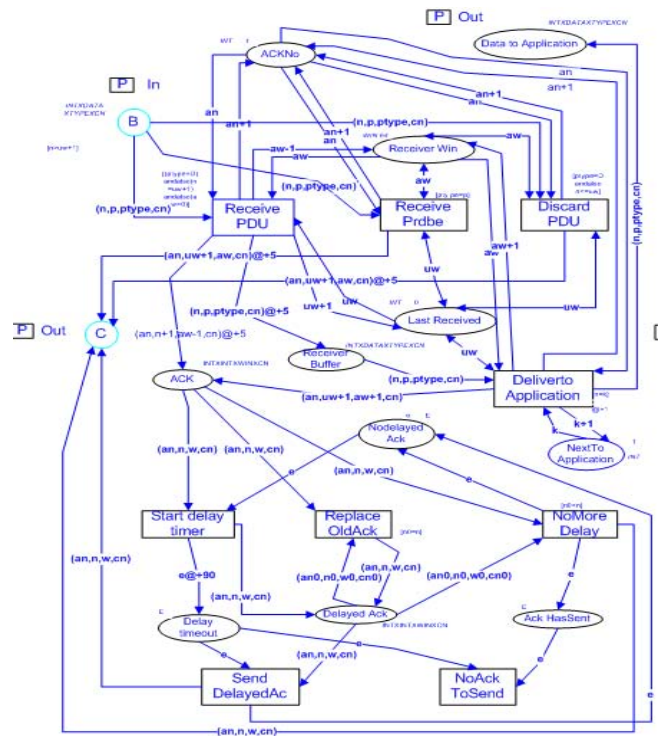
4) 수신부(receiver)

[그림 8]의 트랜지션 receiver는 receiver#8라는

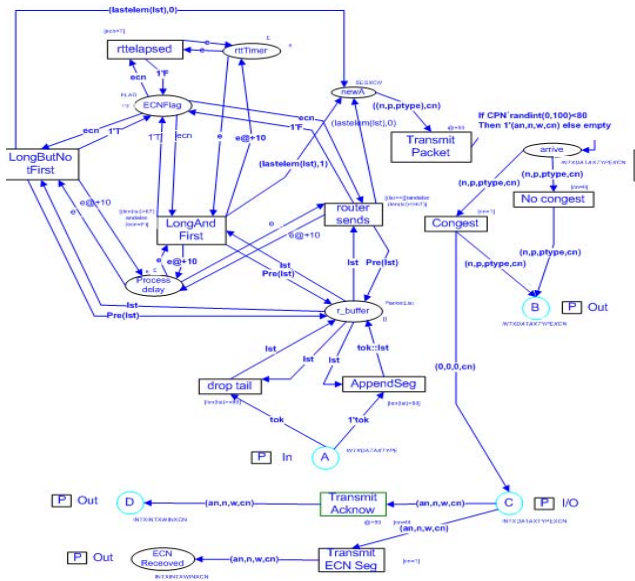
하위페이지로 대치되며, receiver#8의 내용은 [그림 12]와 같다. [그림 12]는 트랜지션 receiver가 어떻게 세그먼트를 수신하여 수신확인 메시지를 전송하는지 상세한 과정을 보인다.



[그림 10] sender#6에 구조화된 모형



[그림 12] receiver#8에 구조화된 모형



[그림 11] network#7에 구조화된 모형

4. 결론

무선통신이 보편화되어감에 따라 무선 환경에서 TCP의 효율성을 개선하기 위한 연구가 활발히 진행되고 있으며, TCP의 페트리 넷 모형이 이러한 연구에 한몫을 담당하고 있다. TCP의 페트리 넷 모형은 너무 복잡하여 한꺼번에 그릴 때 전체 구조를 파악하기 힘들다. 본 논문은 Design/CPN의 계층화 방법을 적용하여 TCP의 페트리 넷 모형을 계층 구조화하는 과정과 결과를 소개하였다. 계층 구조화 결과, TCP의 페트리 넷 모형 전체 구조를 한눈으로 파악할 수 있게 되었으며, 흥미에 따라 지정한 부분

의 자세한 사항도 파악할 수 있게 되었다. 향후 연구 과제로 하위페이지의 트랜지션을 더욱 상세히 표현하여 프로그램 수준까지 계층화하고자 한다.

참고문헌

- [1] The Network Simulator ns-2 [Online]. <http://www.isi.edu/nsnam/ns>.
- [2] T. Murata, "Petri nets: Properties, analysis and applications," Proceedings of the IEEE, Vol. 77. no. 4, pp. 541-580, April 1989.
- [3] K. Jensen, Design/CPN [Online]. Dept. Computer Science, Univ. Aarhus, Denmark. Available: <http://www.daimi.au.dk/designCPN/>.
- [4] Braden, R., and Postel, J., "Requirements for internet gateways," Request for Comments (Standard) RFC 1009, IETF, June 1987, <http://www.ietf.org/rfc.html>.
- [5] Lin, T.Y.Chen, Y.C., "A congestion control approach for LAN/MAN interconnection via ATM," Proceedings of IEEE 13th Networking for Global Communications, Vol. 2, 06/12 - 06/16, 1994, Toronto, Ont., pp. 892-901.
- [6] Dracinschi, A., Fdida, S., "Congestion avoidance for unicast and multicast traffic," 1st European Conference on Universal Multiservice Networks, 2000, 10/02 - 10/04, 2000, Colmar,, pp. 360-368.
- [7] 임재걸, 이계영, 장익현, 윤준영, "ECN 마킹을 위한 적당한 Threshold," 제31회 춘계학술발표대회 논문집 3, 한국정보과학회, 2004년 10월 22일 - 23일, 서울대학교, pp. 172-174.
- [8] <http://www.redhat.com/>