# Network Simulator 2 에서의 새로운 Network Component 추가에 관한 연구

안혜환*, 손재기**
*성균관대학교 정보통신공학부
**전자부품연구원 유비쿼터스 연구 센터
e-mail : *hyehwan@ece.skku.ac.kr
**jgson @keti.re.kr

# A Research about Adding a New Network Component in Network Simulator 2

*Hye Hwan Ahn, **Jae Gi Son
* School of Information and Communication
SungKyunKwan University
** Ubiquitous Research Center, KETI (Korea Electronics Technology Institute)

**요 약**

본 논문은 NS-2 에서의 새로운 네트워크 요소 모듈 개발에 관한 연구를 목적으로 한다. 현재 대부분의 새로운 네트워크 요소들은 경제적인 이유와 네트워크의 환경적인 제약으로 인해서 시뮬레이터 환경에서 실험 되어지고 컴퓨터 공학에서 가장 많이 알려진 시뮬레이션툴 중의 하나로 NS-2 가 많이 사용되어 지고 있다. 이러한 NS-2 는 Otcl 이라는 MIT 에서 개발한 언어와 C++ 가 상호 연동 되어져서 이벤트 처리 부분은 OTcl 이 담당하고 패킷 처리 부분은 C++ 가 담당 하고 있지만 새로운 네트워크 요소를 추가 하는 부분을 이해 하기에는 현재 공개된 문서 만으로는 이해하기 어려운 부분이 많다. 본 논문에서는 NS-2 에 새로운 네트워크 요소를 추가 하는대 있어서 필요한 NS-2 의 기술적인 부분을 설명함을 목적으로 하였다.

**Keywords :** Network Simulator 2 , C++, OTcl, 이벤트, 패킷

## 1. 서 론

컴퓨터 시뮬레이션은 시간적으로나 경제적으로 매우 효과적인 실험을 할 수 있는 중요한 수단으로서 자리매김 해오고 있고 수없이 많은 새로운 시스템이나 아이디어는 이러한 컴퓨터 시뮬레이션을 통해서 충분히 검증된 다음에 실제로 적용되어 지고 있다. 이러한 시뮬레이션 툴 중에서 컴퓨터 공학 학계에서 대표적으로 이용되어 지고 있는 Network Simulator 2 (NS-2) 는 유선, 무선 네트워크를 비롯해서 각종 알고리즘을 성능평가 할 수 있을 뿐만 아니라 최근에는 센서 네트워크 까지 시뮬레이션 할 수 있도록 설계된 대표적인 시뮬레이션 툴의 하나 이다.[1]

이러한 NS-2 는 UC Berkely 에서 C++과 MIT 대학에서 개발한 OTcl 로 쓴 객체지향적인 이산사건 구동 시뮬레이터로서 두가지 언어가 공존해서 작동하는 구조를 가지고 있다. NS-2 는 다양한 IP 네트워크를 시뮬레이션 하고 TCP 와 UDP 와 같은 네트워크 프로토콜, FTP, Telnet, Web, CBR 과 VBR 그리고 Drop Tail, RED 와 CBQ 와같은 라우터 큐 관리 메카니즘, Dijkstra 등과 같은 라우팅 알고리즘을 구현하여 시뮬레이션 할 수 있다. 시뮬레이션은 C++로 구현된 네트워크 모듈들을 OTcl 이 사용하는 구조를 가지고 있고 필요한 모듈을 연동하여 시뮬레이션 하면 그 결과로 텍스트 구조의 트레이스 파일과 동작을 비쥬얼하게 보여주는 애니메이션 파일을 제공한다.

현재 NS-2 와 관련된 문서는 네트워크의 특정 부분만을 깊이 있게 설명한것이 대부분이다. 이러한 NS-2 문서 특성상 초보자가 NS-2 에 새로운것을 구현하여 시뮬레이션 할때 염두해 두어야 할 필수적인 상호연관성에 대해서 알아야 할 부분들이 빠져 있기 때문에 많은 사람들이 NS-2 에서 시뮬레이션을 하더라도 NS-2 가 제공해 주는 기초적인 시뮬레이션 스크립트 파일을 이용하여 초보적인 수준의 시뮬레이션을 하던가 아니면 대부분 부족한 지식으로 인해서 NS-2 에서 시뮬레이션 하는것을 도중에 포기하되 된다.

본 논문은 NS-2 에서 새로운 요소를 구현하는대 있어서 필요한 C++와 OTcl 간의 상호연관성에 관한 기술적인 부분을 설명 할것이고 아래와 같이 구성 되어진다. 2 장에서는 Network Simulator 2 내부에서 C++와 OTcl 의 역할에 대해서 기술 3 장에서는 결론에 대해 기술한다.

## 2. Network Simulator 2 내부에서 C++와 OTcl 의 역할

이 장에서는 NS-2 내부에서 동작하는 두 가지 언어의 상호 연관성의 이해를 돕기 위해 각각의 언어의 특징과 상호동작에 대해서 설명을 하였다.

### 2.1 C++ 의 역할

NS-2 내부에서 C++ 의 역할은 패킷처리를 담당하는 것으로 일축 할 수 있다. NS-2 를 사용하여 새로운 컴퓨터 시스템이나 데이터 처리 알고리즘을 구현하고자 한다면 패킷을 처리 하여야 하고 이러한 것을 위해서 C++ 의 구현 및 수정은 매우 중요한 기술이다. 한가지 주의 해야 할 것은 패킷을 처리 하는 C++ 파일과 패킷을 처리 하지 않는 C++ 파일의 내부 구조가 다르다는 것이다.

```
static class TCPHeaderClass : public
PacketHeaderClass {
public:
        TCPHeaderClass():
PacketHeaderClass
("PacketHeader/TCP",sizeof(hdr_tcp))
          {
          offset(&hdr_tcp::offset_);
          }
} class_tcphdr;
```

그림 1. C++ 에서 패킷헤더 등록

NS-2 내부에 C++ 파일을 수정하거나 새로이 구현하기 위해 사용자가 주의해야 할 것은 2 가지로서 그 첫번째는 패킷을 처리 하는 C++ 파일은 반드시 패킷해더을 등록하는 부분을 그림 1 와 같이 C++ 파일 내부에 코드를 해주어야 한다는 것이고 두번째로는 OTcl 에서 객체를 불러들여서 사용 할 수 있도록 해쉬테이블에 등록하는 부분을 그림 2 와 같이 코드해 주어야 한다는 것이다. 만약에 패킷처리하는 기능을 가지지 않은 모듈을 구현한다면 패킷해더 등록하는 부분의 코드 작업은 하지 않아도 된다는 것이다. 이 두가지 코드 방법은 구현상의 난이도에 있어서 많은 차이를 두고 있다.

```
static class TcpClass : public TclClass
{
public:
        TcpClass()                    :
TclClass("Agent/TCP") {}
        TclObject*
    create(int , const char*const*) {
    return (new TcpAgent());
        }
} class_tcp;
```

그림 2. C++ 에서 해쉬 테이블에 등록

패킷처리 하는 부분이 없는 C++ 는 매우 간단하게 구현할수 있지만 패킷을 처리해야 하는 C++ 는 고도의 코드 기술을 요구 하기 때문이다. 이러한 난이도의 차이는가 발생하는 이유는 패킷을 처리하는 코드가 삽입되게 되면 그림 3 와 같이 네트워크 상위 레이어와 하위 레이어에 연결되는 멤버변수를 사용하여서 자신의 레이어 에서의 동작이 완료해서 상위나 하위 레이어로 패킷을 전송하는 부분까지 고려 하여 코드를 하여야 하기 때문이다.

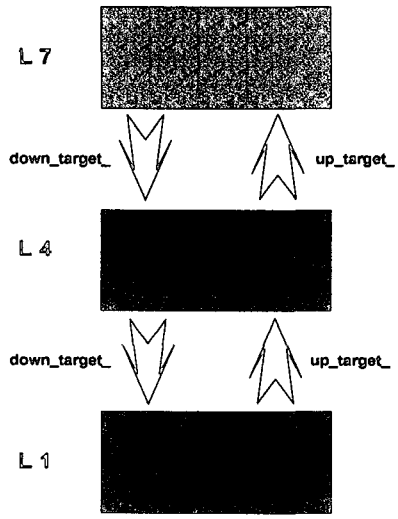


그림 3. C++ 에서 각 레이어간 상하위 통신

### 2.2 OTcl 의 역할

NS-2 내부에서 OTcl 은 여러가지 용도로 사용되어지고 정리해 보면 아래와 같다

- C++ 객체간 연관성 구현
- 토폴로지 구성
- 특정시간 이벤트 구현
- 실시간 이벤트 처리

이 4 가지 중에 첫번째를 제외한 나머지 3 가지는 매우 간단한 처리방식을 가지고 있기 때문에 초보자가 쉽게 지식을 얻을 수 있지만 처음에 제시된 C++ 객체간 연관성 구현은 새로운 시스템이나 아이디어를 NS-2 에 구현하기 위해서 필수적으로 이해하고 있어야 하는 부분이고 현재 공개된 문서에는 언급되어 지지 않고 있는 부분이라 할 수 있다.

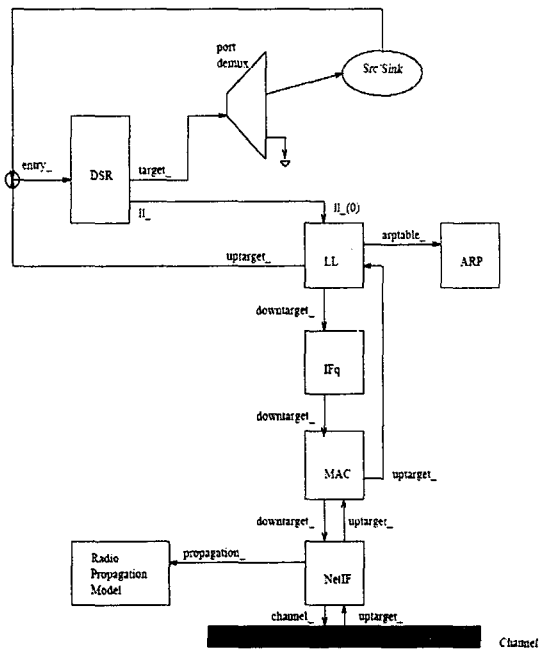


**그림 4. ns-mobilenode.tcl 에서 C++ 객체간 구성**

그림 4 를 보면 각 C++ 객체가 Otcl 언어를 사용해서 Otcl 의 첫번째 기능으로 언급하였던 여러 레이어가 서로 구성 되어져서 하나의 노드가 생성되는 것을 짐작 할 수 있게된다. 이외에도 ns-lib.tcl 파일내부를 보게 되면 ns-config 라는 스크립트 내부에서 여러 C++ 객체들이 어떠한 방식으로 서로 레이어를 설정하고 구성하여 지는지를 확인 할수 있다.

### 2.3 C++ 와 OTcl 의 상호 동작

NS-2 에서 C++ 와 OTcl 두가지 언어가 필요한 이유는 컴퓨터 네트워크를 시뮬레이션 하는대 있어서 패킷을 처리하는 부분과 이벤트를 처리하는 부분이 필요 하기 때문이다. 이러한 패킷과 이벤트는 NS-2 에서 스케쥴링이 되어지기 때문에 새로운 시스템이나 아이디어를 NS-2 에 구현할 때 개발자는 스케쥴러의 동작 방식에 대해서 분석과정은 필요없고 이해만 하고 사용하면 되는것이다.

C++ 에서 패킷을 처리하는 부분을 구현하는대 있어서 주의해야 할 것중 하나는 NS-2 에서 구현된 함수들은 최상위 레이어인 에플리케이션 레이어의 패킷 생성기에 의해서 시작이 되어지고 이러한 패킷생성기는 확률 모델에 의해서 구현되어져 있기 때문에 시뮬레이션이 종료할 때 까지 계속해서 패킷을 생성하게 되고 이에 따라 하위 레이어의 패킷처리 하는 부분들은 해당하는 프로세스가 필요되어질 때마다 반복적으로 호출되어 지는 것이다. 만약 주기적인 시간을 가지고 동작하는 프로세스를 구현하고자 한다면 개발자는 스케쥴러를 이용해서 구현하여야 하고 절대로 C++ 내부에 쓰래드와 같은 것을 구현해서는 안되는것이다.

간단히 정리하면 NS-2 에있는 C++ 객체들은 NS-2 에 있는 스케쥴러만을 사용하여야 하고 쓰래드 등을 이용해서 OS 의 스케쥴러를 사용하는 실수를 해서는 안된다는 것이다. 만약 C++ 구현시 쓰래드와 같은 것을 사용하게 된다면 성능평가 결과로 주어지는 트래이스 파일에 출력된 결과들은 더 이상 신뢰 할 수 없는 데이터가 되는 것 이다.

## 3. 결론

NS-2 는 매우 신뢰 할 수 있는 실험결과를 얻을수 있는 세계적으로 유명한 컴퓨터 네트워크 시뮬레이션 툴로서 이러한 툴을 사용함으로 인해서 많은 연구기관이나 학계에서 새로운 시스템이나 아이디어를 네트워크에 실제로 적용하기 전에 실험을 통해서 양질의 결과를 예측해 낼 수 있고 이러한 것은 경제적으로나 시간적으로 매우 효율적이다. 이러한 네트워크 시뮬레이션툴에서 새로운 네트워크 요소를 추가하는 기술을 표현하는 기술문서화 작업은 연구기관이나 학계에 많은 도움을 줄 것이며 나아가 국가 컴퓨터 공학의 미래를 밝히는대 커다란 도움이 될것이다.

## 참고자료

[1] The ns Manual
(formerly ns Notes and Documentation)
The VINT Project
A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.
Kevin Fall (kfall@ee.lbl.gov). , Kannan Varadhan (kannan@catarina.usc.edu)

[2] Tutorial for The Network Simulator "ns", Marc Greis
http://www.isi.edu/nsnam/ns/tutorial/index.html

[3] WPI Computer Science "Ns by example" Jae Chung, Mark Claypool
http://nile.wpi.edu/NS/

[4] OTcl Tutorial (Version 0.96, September 95)
http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/tutorial.html