http://dx.doi.org/10.7236/JIIBC.2013.13.3.111
JIWIT 2013-1-15

3GPP RLC 프로토콜의 검증 및 평가 테스트베드의 구축

Implementation of 3GPP RLC Testbed for Protocol Verification and Evaluation

성정환*, 서효중**

Junghwan Sung, Hyo-Joong Suh

요 약 전송을 보장하는 연결 기반의 TCP 프로토콜은 비연결성의 UDP 프로토콜에 비하여 네트워크 오버헤드가 크다. 3GPP는 제한된 대역폭을 분할하여 사용하는 무선 네트워크 및 이동 통신 환경에서, 오버헤드가 적은 UDP 프로토콜을 이용하면서도 데이터의 신뢰성을 보장하기 위한 상위 계층인 RLC 프로토콜을 제안하였다. 따라서 단말 및 기지국을 개발하는 경우 3GPP 표준인 RLC 프로토콜을 구현하여야 하며, 상호 동작을 보장하기 위한 검증 테스트를 통과하여야 한다. 본 논문은 RLC 프로토콜을 시험하기 위한 검증 테스트 베드를 구현하였으며 하위 네트워크에서 발생할 수 있는 다양한 패킷 역전 및 손실을 모사할 수 있는 환경을 구축하였다. 결과적으로 본 논문에서 구축한 테스트 베드를 RLC 프로토콜에 대한 시험 도구로 제안한다.

Abstract The connection based TCP protocol provides reliable packet delivery, but it takes certain overhead compare to the connection-less UDP protocol. Thus, 3GPP devise the upper-layer RLC protocol which provides reliability based upon the UDP protocol. Consequently, each implement of a base station and/or mobile terminal require the development of the RLC protocol, and it must qualify various interoperable tests. In this paper, we implement a testbed which verifies the RLC protocol under various packet losses/inversion circumstances of networks. Finally, we propose our testbed as the RLC protocol tester for the developments.

Key words: RLC Protocol, Verification, Network Emulation, Testbed

Ⅰ. 서 론

유비쿼터스 환경이 급속히 확장되어감에 따라 최근 무선 이동 통신 기술은 스마트폰, 넷북 등 여러 기기에 수용·보급되어, 음성뿐 아니라 오디오, 비디오 등 다양한 활용성을 요구하게 되었다. TCP/IP(Transmission Control Protocol/Internet Protocol)를 사용하는 무선통신 환경에서는 제한된 대역폭으로 인한 오버헤드의 문제로 UDP(User Datagram Protocol) 프로토콜을 사용한다. 하지만 비연결성의 UDP 프로토콜은 시간만료, 재조립, 재전송 등과 같은 신뢰성 보장을 위한 기능이 없기 때문에 데이터의 역전 및 손실이 발생하게 되어 서비스 품질 저하의 문제점이 있다. 이에 3GPP(3rd Generation Partnership Project)에서는 데이터의 신뢰성 보장을 위한 상위계층인 RLC(Radio Link Control) 프로토콜[1]을 표준화하였고 인증기관을 통해 검증 받도록 체계화 하였다[2].

*준회원, 가톨릭대학교 컴퓨터공학과
**정회원, 가톨릭대학교 컴퓨터정보공학부(교신저자)
접수일자 2013년 3월 18일, 수정완료 2013년 5월 12일
게재확정일자 2013년 6월 14일

Received: 18 March 2013 / Revised: 12 May 2013 / Accepted: 14 June 2013
**Corresponding Author: hjsuh@catholic.ac.kr
Dept. of Computer Science and Information Engineering, the Catholic University, Korea

하지만 이러한 인증기관에서의 인증 절차는 높은 시간당 비용이 부과되며, 절차적인 방법으로 인증의 항목별 통과 또는 실패 여부만을 테스트하므로, 항목 시험과 디버깅이 병행되는 프로토콜의 개발 진행 단계에서의 테스트과정과 큰 차이가 있다. 따라서 관련 기기를 개발하기 위해서는 우선 표준 프로토콜에 따르는 기기를 시험기기를 이용하여 자체 내에서 통신 시험을 진행하여야 되며, 이러한 내부적 시험을 완료한 후에 앞서 이야기한 공인 인증기관에서의 인증 절차를 밟게 된다.

내부적 시험에서는 구현한 프로토콜의 극단적이거나 특이한 상황에서의 동작 검증과 성능을 확인하기 위하여, 다양한 네트워크 상황에 대한 시험이 요구된다. 이를 위해 발생 확률이 극도로 낮은 네트워크 상황을 반영한 환경을 구축하여야 하지만 실제 네트워크 환경에서 이러한 조건의 발생은 임의적으로 조절하기 대단히 어렵다. 따라서 이러한 네트워크 환경을 모사하여 개발 기기의 실제 동작에 적용 할 수 있는 도구가 필요하다[3][4][5].

본 연구는 이러한 필요성에 입각하여 단말 또는 기지국 개발 과정에서 RLC 프로토콜을 준수하여 동작하는지 시험할 수 있는 테스트베드를 구현하고 또한 앞서 언급한 다양한 네트워크 상황을 반영할 수 있도록 함으로써 UMTS(Universal Mobile Telecommunications System)[6] 또는 LTE(Long-Term Evolution)[7] 단말이나 기지국을 개발하는 데 사용하여, 프로토콜의 동작성 시험과 검증 기간을 단축하고 인증 비용을 절감할 수 있도록 하는 것이다.

본 논문에서 제시하는 테스트베드는 단말 또는 기지국 역할을 하여 상대편의 프로토콜 동작을 검증할 수 있는 RLC 프로토콜 모듈과 다양한 네트워크에서 발생할 수 있는 패킷의 역전 및 손실을 모사하는 네트워크 에뮬레이터로 구성하였고, 테스트베드의 RLC 프로토콜 모듈과 상대편의 시험 대상 기기 간의 패킷 전송은 손실과 지연 및 역전을 구현하는 네트워크 에뮬레이터를 통해 이루어지도록 하였다. 네트워크 에뮬레이터는 사용자가 입력한 역전 확률 및 손실 확률에 따라 제어되고, 송신 패킷과 수신 패킷을 비교하여 정상적인 동작이 이루어졌는지 확인한다.

본 논문의 구성은 다음과 같다. 우선, 2장에서 RLC 프로토콜 동작 메커니즘과 구현을 기술한다. 3장에서는 RLC 프로토콜 검증 테스트베드의 구성과 네트워크 에뮬레이터의 구현을 기술하고, 4장에서는 테스트베드의 성능을 평가한다. 마지막으로 5장에서는 결론을 맺는다.

II. RLC 프로토콜

RLC 프로토콜은 radio link 상에서 data의 신뢰성 보장을 제공한다. RLC 프로토콜은 이를 위해 여러 가지 RLC entity를 가지고 있으며, TM(Transparent Mode), UM(Unacknowledged Mode), AM(Acknowledged Mode) 등 3가지 data 전송 모드에 따라, TM RLC entity, UM RLC entity, AM RLC entity로 분류된다. 그림1은 data 전송 모드에 따른 RLC entity가 표현된 프로토콜 모델이다.

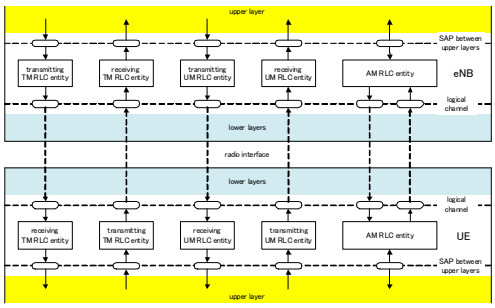


그림 1. RLC sub later의 개괄적 모델[1]
Fig. 1. Overview model of the RLC sub layer[1]

RLC 프로토콜에서 신뢰성 있는 데이터 전송을 담당하는 것은 가장 다양한 기능이 구현되어 있는 AM entity이며 기타 entity는 AM entity의 부분적인 기능을 가지고 있다.

1. RLC 프로토콜의 재전송 메커니즘

RLC 프로토콜은 전송된 data에 대한 결과를 알려주는 status report가 있어, 이 status report에 재전송 여부를 판단할 수 있는 정보가 포함된다. 송신측에서는 전송된 패킷에 대한 결과를 알기위해서 poll bit이 설정된 패킷을 전송하여 status report 전송을 요청하고, 수신측에서는 기다리는 순서번호가 수신한 패킷의 순서번호와 다르면 reordering timer를 동작한다. 타이머 만료 전까지 해당 패킷이 수신되지 않으면 패킷 손실로 판단하고, 손실로 판단한 패킷의 재전송을 요청하게 된다. status report를 요청한 쪽에서는 요청 후 pollretx timer를 동작하고 만료

전까지 응답이 없으면 다시 요청하게 된다. 이러하게 송수신측 상호 간에 status report의 교환이 이루어지게 되며 결과적으로 성공적인 패킷 전송을 유도한다.

2. RLC 프로토콜의 구현

RLC 프로토콜의 핵심은 UDP 프로토콜 상에서 신뢰성 있는 데이터 전송을 보장하여야 하므로, 패킷 전송의 확인 및 재전송이 그 주요한 부분이 된다. RLC 프로토콜의 하위에서는 MAC 계층이 구성되어야 하고, 상위에서는 응용 계층이 구성되므로 RLC 프로토콜은 이러한 상하위의 계층과 연계하여 동작한다. 그림 2는 3GPP에서 정의한 AM entity의 개념적 아키텍처를 본 논문에서 구현한 모듈을 도시한 것이다.

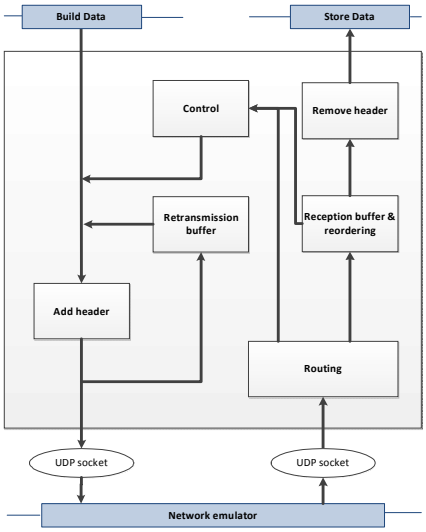


그림 2. 구현한 RLC 모듈
Fig. 2. Implemented RLC module

기본적으로 RLC 프로토콜에서 다루는 패킷은 전송할 데이터가 담긴 데이터 패킷과 status report 절차를 통해 데이터 전송 결과를 알려주는 콘트롤 패킷으로 나눌 수 있다. 전송되는 패킷은 D/C 필드의 값에 따라 데이터 패킷과 콘트롤 패킷으로 구분된다. 데이터 패킷은 순서번호를 나타내는 SN 필드와 status report를 요청하기 위한 poll bit인 P 필드가 추가적으로 구성되며 그림 3은 구현한 패킷 구조를 도시한 것이다.

콘트롤 패킷의 추가적인 구성 필드에는 수신 완료된 패킷의 순서번호를 나타내는 ACK_SN 필드와 수신하지 못한 패킷의 순서번호를 나타내는 NACK_SN 필드 그리고 E1 필드가 있다. 콘트롤 패킷의 NACK_SN 필드는 수신하지 못한 패킷이 발생할 경우에만 나타나므로, ACK_SN 필드 또는 NACK_SN 필드의 존재 여부를 구분하는 E1 필드를 부가한다. 그림 4와 그림 5는 이러한 두 가지 경우의 패킷 구조를 나타낸 것이다.

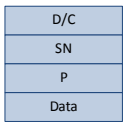


그림 3. 데이터 패킷 구조
Fig. 3. Data packet Structure

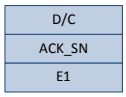


그림 4. 미수신 패킷이 있는 콘트롤 패킷
Fig. 4. Control packet with an unreceived packet

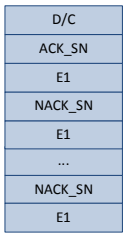


그림 5. 미수신 패킷이 여럿인 콘트롤 패킷
Fig. 5. Control packet with several unreceived packets

구현한 RLC 프로토콜 모듈은 메인 프로세스와 두 개의 쓰레드로 구성된다. RLC 프로토콜 모듈이 실행되면 메인 프로세스에 의해 CLI 쓰레드와 timer 쓰레드가 생성된다. CLI 쓰레드는 전송할 데이터의 발생 빈도와 네트워크 에뮬레이터에 대한 지연 및 손실 조건을 입력 받아 RLC 처리부와 네트워크 에뮬레이터에 전달하는 역할을 수행하는 쓰레드이며, 타이머 쓰레드는 네트워크 에뮬레이터의 패킷 손실 판단을 위한 reordering timer 루틴과 status report 요청을 위한 pollretx timer 루틴을 포함하고 있다. 그림 6은 이러한 프로그램 동작 절차이다.

프로그램에서 데이터 전송의 핵심이 되는 메인 프로세스는 패킷 전송을 담당하는 Tx부과 패킷 수신을 담당하는 Rx부로 구성되어 있다. Tx부에는 스펙에서 정한

전송 우선순위에 따라, 콘트롤 패킷을 가장 우선으로 전송하고 재전송 패킷 전송, 새로운 패킷 전송의 순으로 처리한다. Rx부에서는 데이터 패킷과 콘트롤 패킷으로 판별하여, 데이터 패킷일 경우 수신 패킷의 순서번호가 순차적인지를 검사하는 reordering 과정을 수행하고, 콘트롤 패킷일 경우에는 NACK_SN 필드의 유무에 따라 해당하는 순서번호의 패킷을 재전송한다. 그림 7은 이러한 과정을 나타낸다.

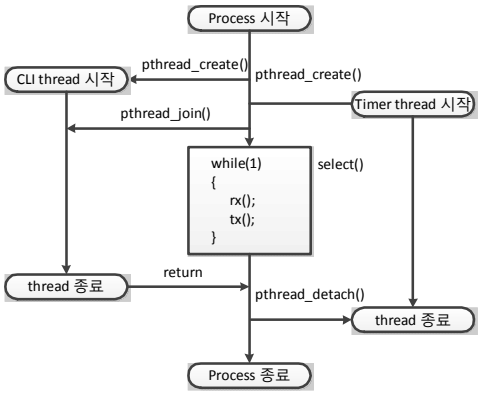


그림 6. 프로그램의 동작 절차
Fig. 6. Operation steps of the program

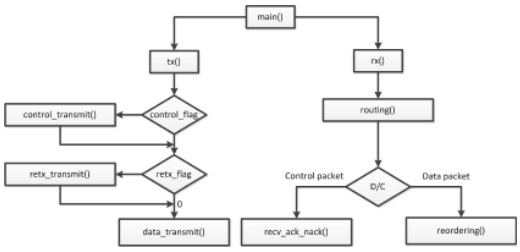


그림 7. Tx 및 Rx에 따른 수행 절차
Fig. 7. Execution steps of the Tx and Rx

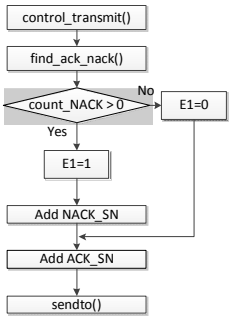


그림 8. 콘트롤 패킷의 생성 과정
Fig. 8. Generation steps of a control packet

만일 Rx에서 기다리는 순서번호와 다른 번호의 패킷이 전송되어 올 경우, reordering 과정을 수행하게 된다. reordering 과정은 reordering timer가 만료될 때까지 기다리는 순서번호의 패킷을 받지 못할 경우, 마지막으로 수신 완료한 패킷의 순서번호와 수신하지 못한 패킷의 순서번호를 명기하여 콘트롤 패킷에 채워 전송한다. 또한 수신한 데이터 패킷에 poll bit이 설정되어 있을 경우에도 마찬가지로 진행 중인 순서번호에 대한 내용을 콘트롤 패킷에 담아 전송한다. 그림 8는 콘트롤 패킷의 생성 과정을 도시한 것이다.

이 외에도 status report 송수신, pollretx timer 등 다양한 조건에 대한 진행 절차가 테스트베드 프로그램에 구현되었다.

III. 테스트베드 구성

1. 테스트베드 동작 형태

본 논문에서 구현한 테스트베드는, 다양한 네트워크 상황에 따른 RLC 프로토콜 모듈의 동작성 검증을 확인하는데 그 목표가 있다. 따라서 이러한 소프트웨어 계층을 테스트하기 위해 실제 기지국과 무선 통신을 담당하는 물리 계층을 필요로 하지 않으며, 물리 계층은 TCP/IP 프로토콜을 이용하여 동작할 수 있도록 하였다. 따라서 테스트베드는 3GPP와 무관하게 테스트베드 자체의 독립적인 유무선 네트워크로 연결된 PC를 이용하여 테스트할 단말의 기능을 수행하도록 하거나, 한 대의 PC에 복수 개의 테스트할 단말을 연결시키고 TCP/IP 통신을 이용한 프로토콜 동작을 시험하도록 구현할 수 있다. 이러한 방법은 물리적인 무선 계층 환경과 무관하게 상위 계층 프로토콜 부분만을 독립적으로 테스트할 수 있는 장점을 가지며, 특히 단말 개발에 있어서 단말을 직접 PC와 USB 또는 이더넷 등으로 연결하여 TCP/IP 연결을 경유할 수 있도록 하여 물리적 네트워크 계층과 분리하여 테스트할 수 있으므로, 개발 단말에서 물리 계층의 영향 없이 프로토콜 검증과 성능 측정을 시험할 수 있게 한다.

RLC 프로토콜 모듈 간 통신은 네트워크 에뮬레이터를 통해 이루어진다. 그림 9는 한 대의 PC를 이용하여 리눅스 상에 구축한 일례로, 두 개의 RLC 모듈을 VMWare 가상 머신에서 동작시켜 각자 RLC 통신을 하는 단말 역

할을 하도록 하고, 네트워크 에뮬레이터를 경유하여 RLC 모듈이 메시지를 주고받는 형태로 테스트베드를 구성한 것이다. 각 RLC 프로토콜 모듈은 송수신자 역할 모두를 수행할 수 있으며 네트워크 에뮬레이터는 리눅스 가상머신이나 네이티브 수행 등 다양한 형태로 구성할 수 있다.

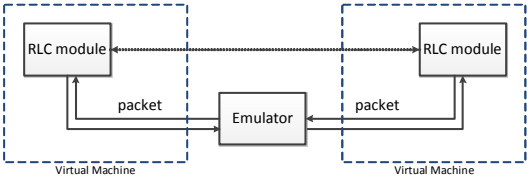


그림 9. 테스트베드 수행 모델
Fig. 9. Execution model of the testbed

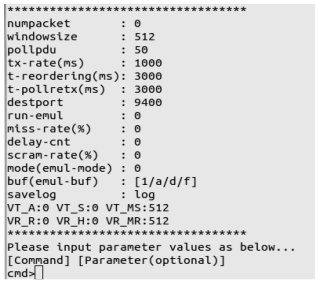


그림 10. 명령 입력
Fig. 10. Command input

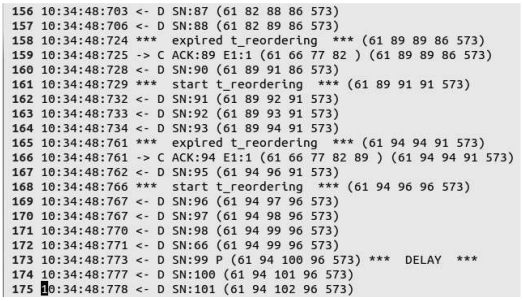


그림 11. 로그 화면
Fig. 11. Log screen

RLC 프로토콜 모듈과 네트워크 에뮬레이터는 일반적이거나 극단적인 네트워크 상황에 대한 시험을 위해 각종 환경 조건을 설정할 수 있다. RLC 프로토콜 모듈에서는 패킷 생성주기, 타이머 시간값, 윈도우 사이즈 등을 설정 할 수 있고, 실제적인 네트워크 경로를 모사하는 중심 역할인 네트워크 에뮬레이터에서는 네트워크에서 발생할 수 있는 패킷 지연과 손실, 역전 등을 설정 할 수 있다. 또한, RLC 프로토콜 모듈과 네트워크 에뮬레이터 모두 각각 로그파일을 생성하여 패킷의 송수신마다 기록하고 패킷의 손실률 및 역전률, 재전송 요청 횟수와 재전송 횟수를 확인할 수 있게 한다.

그림 10은 테스트베드에서 구축한 RLC 모듈 및 네트워크 에뮬레이터의 각종 파라미터를 입력 받는 프론트엔드의 동작을 보여주는 것으로, 패킷의 생성과 네트워크 조건을 입력 받고 있다. 그림 11은 본 논문에서 구축한 테스트베드에 대해 각종 파라미터를 설정하고 동작 시킨 결과를 로그 형태로 나타낸 것을 보여준 것이다.

2. 네트워크 에뮬레이터

네트워크 에뮬레이터는 네트워크 경로에서 발생할 수 있는 다양한 조건을 모사할 수 있어야 한다. 그림 12는 테스트베드에서 구현한 네트워크 에뮬레이터에서 이러한 네트워크 조건을 입력 받아 수행하는 형태를 도시한 것으로, UDP 형태의 패킷 전송에서 나타날 수 있는 다양한 조건을 설정 가능하도록 하였다.

데이터를 주고받을 종단의 해당되는 단말의 RLC 프로토콜 모듈로부터 생성된 패킷은 네트워크 에뮬레이터의 큐에 적재된 후 큐의 적재 패킷 수를 제어하여 네트워크에서 발생하는 지연을 모사할 수 있으며, 손실과 역전 여부에 대한 연산을 거쳐 전송된다. 네트워크 에뮬레이터는 이러하게 지연 및 손실과 역전을 구현하여 수신될 RLC 모듈 쪽으로 전송을 시작한다.

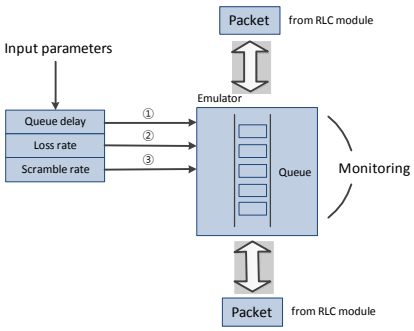


그림 12. 네트워크 에뮬레이터의 패킷 지연, 손실, 역전
Fig. 12. Pack delay, losses, and inversion in the network emulator

패킷 손실과 역전은 네트워크 에뮬레이터 내의 큐에서 구현되는데, 패킷을 송신한 RLC 모듈로부터 패킷을

큐에 적재하고 순차적으로 저장된 패킷은 손실 및 역전 제어를 위해 2단계의 랜덤연산을 거치게 된다. 먼저 패킷 손실률에 해당하는 loss 값을 기준으로 하여 매 패킷마다 난수를 발생하여 확률적으로 패킷의 손실 여부를 결정하여 큐에서 삭제한다. 손실을 거쳐 큐에 남게 된 패킷은 또 다른 난수를 발생하여 확률적으로 scrambling 값을 통해 다음 단계인 역전 조건을 판단하게 되며 역전을 하지 않는 경우 순서를 유지하고, 역전을 하는 경우 큐의 저장된 다음 패킷과 순서를 바꾸게 된다. 이때 만약 큐에 적재된 패킷이 없다면 순서를 바꾸는 과정 없이 전송한다. 그림 13은 2단계 랜덤연산 과정을 도시한 것이다.

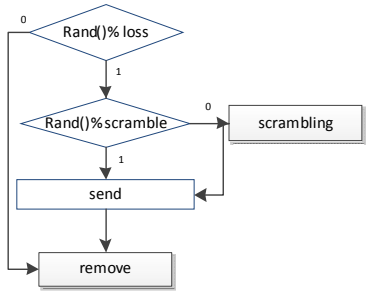


그림 13. 난수에 기반한 패킷의 손실 및 역전
Fig. 13. Randomized packet loss and inversion

IV. 검증 및 성능평가

본 논문에서 제안하는 테스트베드는 UMTS 또는 LTE 기지국이나 단말 개발 시, 개발한 RLC 프로토콜이 데이터의 신뢰성 있는 전송을 보장하는 지에 대한 검증뿐만 아니라 RLC 프로토콜을 이용하는 다양한 조건의 데이터 스트림에 대한 동작성도 검증되어야 한다. 이를 위하여 본 논문에서 구축한 테스트베드에 RLC 프로토콜을 이용할 데이터 스트림을 적용하여, 실시간 비디오, 오디오 등 실제 응용에서 발생하는 조건의 데이터 흐름과 다양한 네트워크 환경에서의 파라미터를 수용하여 시험하였다. 표 1은 본 논문의 구현에서 사용한 테스트베드의 하드웨어 및 운영 환경 사양으로 일반적인 데스크톱 시스템에 사용되는 것이다.

표 1. 구현한 테스트베드의 시스템 사양
Table 1. System specification of the testbed

Processor	AMD Athlon(tm)64 X2 Dual Core 4800+ 2.51GHz
Main memory	2 GByte
Host OS	Windows XP(Service Pack 3)
Virtual machine OS	VMware 9, Ubuntu 12.10

성능 측정은 RLC 프로토콜에 따라서 데이터를 주고받을 양 종단에 해당되는 단말을 설정하고, 양단의 RLC 모듈간에 스트리밍 데이터의 전송이 일어나는 상황을 가정하였으며, 한 대의 기기에 두 RLC 모듈 및 네트워크 에뮬레이터를 모두 실행시키고 각각을 TCP/IP 소켓으로 연결하여 동작시켰다.

데이터는 음성과, 오디오 및 비디오 스트리밍에 자주 사용하는 데이터율을 적용하여 총 3회 반복하였고, 네트워크 에뮬레이터에서 발생시킬 패킷 손실은 국내 인터넷 상에서 패킷 전송 시 나타나는 손실 특성[8]에 따라 패킷의 손실 확률을 입력하여 측정하였다. 스트리밍 데이터의 전송률에 따라서 서로 다른 reordering timer와 pollretx timer를 적용하였는데, 그 이유는 스트리밍 데이터의 요구 전송률에 따라서 패킷의 생성 주기가 달라지

표 2. 데이터 스트리밍 시험 결과 (1Kbytes/packet, 512 window size, 50 pollPDU)
Table 2. Outputs of data streamings (1Kbytes/packet, 512 window size, 50 pollPDU)

	Data Type	voice				audio(MP3)				video(MPEG1)			
Input Parameter	packet interval	600ms				50ms				7ms			
	reordering timer	1800ms				150ms				21ms			
	pollretx timer	1800ms				150ms				21ms			
	transfer rate	13.3kbps				160kbps				1142.8kbps			
	packet loss(%)	0	0	16.11	14	0	0	15.29	12.62	0	0	16.73	13.41
	packet scramble(%)	0	5	0	2.72	0	4.16	0	2.91	0	4.91	0	1.73
Measured Output	Tx/Rx packets	120	120	273	257	120	120	268	206	121	122	245	231
	lost packet	0	0	44	36	0	0	41	26	0	0	41	31
	scrambled packet	0	6	0	7	0	5	0	6	0	6	0	4
	Retransmit	0	0	101	82	0	0	89	55	0	0	77	64
	status report	0	0	52	55	0	0	59	32	0	1	49	47

게 되므로 스트리밍 데이터의 요구 전송률을 유지하는 하에서 패킷의 역전 및 유실을 발생시키기 위한 것이다. 표 2는 이러한 조건에 따른 측정 결과이다.

세 가지 유형의 데이터 스트리밍을 테스트베드에서 시험한 결과에 따르면 본 논문에서 구축한 테스트베드는 패킷의 유실 및 역전에 대해서 RLC 프로토콜에 부합하는 정상적인 동작을 나타내었으며, 네트워크 에뮬레이터에 입력된 패킷 유실 및 역전 등의 파라미터 설정에 따라서 패킷의 재전송 및 status report 패킷의 정상적인 생성과 발생을 확인할 수 있었다.

V. 결 론

본 논문은 UMTS/LTE 기지국 또는 단말 개발 시에 사용할 수 있는 RLC 프로토콜 동작성 검증 및 성능 평가를 위한 테스트베드를 구축하였다. 구축한 테스트베드는 다양한 데이터 전송율을 적용할 수 있는 RLC 모듈과 희소한 확률로 발생할 수 있는 극단적인 조건까지를 모두 구현할 수 있는 네트워크 에뮬레이터를 통해, 일반적인 환경에서 쉽게 확인하기 어려운 RLC 프로토콜의 오동작을 탐지할 수 있는 환경을 제공한다. 이러한 테스트베드는 높은 비용과 시간을 필요로 하는 3GPP의 공인 인증 절차 이전에, UMTS/LTE 단말의 내부 개발 과정에서 사용하여 단말 및 프로토콜 계층의 개발 및 검증 기간을 단축하고 결과적으로 인증 시간과 비용을 절감할 수 있는 효과가 있을 것으로 판단된다.

본 논문에서 구축한 RLC 프로토콜의 검증 및 성능 평가 테스트베드는, 다양한 국내의 네트워크 환경을 유형화하여 향후 시나리오별 테스트 환경을 스크립트 형태로 개발하여 빠른 시간에 높은 수준의 RLC 프로토콜 검증을 할 수 있도록 확장할 예정이다.

References

[1] 3GPP TS 36.322, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification V11.0.0", 2012

[2] D.-H. Nam, "Technology Trend of LTE Test and Certification ", TTA Journal, Vol.134, pp. 103-108, 2011

[3] Jang-Hyun Kim, Hyo-Joong Suh, "Implementation of Verification and Evaluation Testbed of WiMax2 PKMv2 Encryption Layer", The Journal of the Institute of Webcasting, Internet and Telecommunication, Vol. 13, Issue 2, pp. 77-82, 2013.

[4] Y.-Doo Kim, Il-Young Moon, "Design of Testbed for Performance Evaluation of Peer-to-Peer Protocols in Mobile Networks", Journal of Korean Institute of Information Technology, Vol. 8, Issue 10, pp. 159-166, Oct. 2010.

[5] Dae-Woo Choi, "Implementation of a Testbed for Wireless Sensor Network", Journal of the Korea Academia-Industrial cooperation Society, Vol.12, No.1, pp. 445-450, 2011.

[6] Harri Holma, Antti Toskala, WCDMA for UMTS: Radio Access for Third Generation Mobile Comm., 3rd ed., John Wiley & Sons, Ltd, 2005

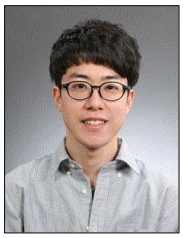
[7] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, S. Parkvall, "LTE: the evolution of mobile broadband", IEEE Communications Magazine, Vol. 47. No. 4, pp. 44-51, 2009.

[8] Jun Sok Park, Min Soo Jung, Dae Sik Ko, " Estimation of the Real-Time Audio Packet Loss over the Korean Internet using UDP", In Proc. Conf. the Korean Institute of Communication and Information Sciences, pp.288-293, 1997

※ 본 연구는 2013년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

저자 소개

성 정 환(준회원)



·2013 ~ 현재 : 가톨릭대학교 컴퓨터공학과 석사과정
·2013년 : 가톨릭대학교 컴퓨터정보공학 학사
<주 관심분야 : 모바일 시스템, 내장형 시스템, 이동 통신>

서 효 중(정회원)

·2003년 ~ 현재 : 가톨릭대학교 컴퓨터정보공학부 부교수
·2000년 ~ 2003년 : 지씨티리서치 연구원
·2000년 : 서울대학교 컴퓨터공학 박사
·1994년 : 서울대학교 컴퓨터공학 석사
·1992년 : 서울대학교 학사
<주 관심분야 : 컴퓨터 구조, 컴퓨터 시스템, 내장형 시스템, 이동 통신>