

<https://doi.org/10.7236/JIIBC.2023.23.6.97>
JIIBC 2023-6-15

개방형 SW를 이용한 IoT 네트워크 성능시험기 개선에 관한 연구

A Study on the Improvement of IoT Network Performance Test Framework using OSS

정영준*, 정이도**, 이성화***, 김진태****

Joung Youngjun*, Jeong Yido**, Lee SungHwa***, Kim JinTae****

요약 본 연구는 최근 다양화, 대규모 되어가는 IoT시스템을 위한 시험기 개선에 관한 것으로 IoT성능시험기의 패킷 처리 성능개선, 트래픽 프로토콜 생성과 동작에 유연성을 확보하는 방안에 대한 것이다. 본 연구의 목표는 고도화되어 가는 대용량 IoT 네트워크 시스템에서 데이터 트래픽 전송의 안정성을 사전에 검증하고 성능을 측정하는 개방형 소프트웨어인 DPDK기반 고속 IoT 네트워크 성능시험 시스템을 설계하는 것이다. DPDK 기반 트래픽 발생기를 이용하여 고속 IoT 성능시험기의 기본 구성을 설계하였으며 실험을 통해 해당 시스템 적용시 트래픽 모델링하고 패킷생성능력의 기대효과를 제시하였다.

Abstract This study is to provide improvement of tester for IoT system, which has recently become diversified and large-scale and It is about a method to improve the packet processing performance of the tester and securing flexibility in traffic protocol creation and operation. The purpose of this study is to design a OSS DPDK-based high-speed IoT network performance test system, which pre-verifies and measures the performance of data traffic transmission in an increasingly sophisticated high-capacity IoT network system. The basic structure of the high-speed IoT performance tester was designed using a DPDK-based traffic generator, the expected effect was suggested to traffic modeling and packet generation capability when the system was applied through experiments

Key Words : IoT Testing, Traffic Generator

1. 서론

IoT 시스템은 홈네트워크, 스마트 팩토리, 헬스케어, 자율주행시스템, 공공관리, 미세먼지, 재난예방 등 적용

분야가 증가하고, IoT네트워크 및 트래픽은 다양화, 대규모화 되어가고 있다. 하지만, 이를 검증하는 시험도구는 성능의 제약으로 시스템의 사전 안정성 검증 및 품질 확보에 어려움을 가지고 있어, 품질비용 증가 및 서비스

*정희원, 에릭슨LG R&D 센터

**정희원, (주)기술법인세종

***중신희원, 제주한라대학교 컴퓨터정보과

****정희원, (주)파이브텍

접수일자 2023년 9월 26일, 수정완료 2023년 11월 9일
게재확정일자 2023년 12월 8일

Received: 26 September, 2023 / Revised: 9 November, 2023 / Accepted: 8 December, 2023

*Corresponding Author: jin85@fivetek.com

Dept. of Convergence Information Technology Graduate School of Information and Telecommunications, Konkuk University, Korea

지연의 문제를 발생시킬 수 있다.

IoT 시스템 시험은 IoT 시스템 구성요소의 안정적인 동작과 서비스의 품질을 보장하기 위해 시스템개발 및 유지보수의 단계에서 기기 자체의 기능, 네트워크 성능 및 보안 등과 이들의 상호작용 등을 확인하는 활동이다^[1]. 이러한 시험과정에서 대량의 IoT 네트워크 구성요소를 실제 구축하는 것은 많은 비용을 발생시킨다. 따라서, 시험의 구성, 반복시험, 확장과 관리의 편리성, 제어 기능을 가진 가상으로 구축된 시험도구 또는 프레임워크가 필요하다.

본 논문은 고도화된 대용량 IoT 네트워크 시스템에서 데이터 트래픽 전송의 안정성을 검증하고 성능을 측정할 수 있는 개방형 소프트웨어인 DPDK(Data Plane Development Kit) 기반 트래픽 생성기를 이용한 고속 IoT 네트워크 성능시험기를 제안한다.

II. 본론

1. IoT(Internet of Things)

사물인터넷(IoT)이란 인터넷을 기반으로 다양한 물리적 사물들을 연결하여 사람의 개입 없이 언제 어디서나 상황에 맞는 상호작용과 지능화를 통해 융합 서비스를 제공하는 기술이다. IoT 시스템의 급속한 발전과 다양한 솔루션들의 적용에 따라 IoT 시스템의 새로운 구성요소 및 서비스의 품질을 평가하고, 시스템의 성능을 최적화하는 방법을 찾기 위해 IoT 아키텍처, 애플리케이션 및 기기 시험 및 인증에 대한 관심이 높아지고 있다.

2. IoT 시스템 시험 및 가상환경 시험도구

IoT 시험(IoT Testing)은 IoT 시스템 구성요소의 안정적인 동작과 서비스의 품질을 보장하기 위해 기기 자체의 기능, 네트워크 및 보안과의 상호작용과 관련된 다양한 측면의 기능 등을 확인하는 것을 말한다.

IoT 네트워크에서는 하드웨어, 소프트웨어 플랫폼, 네트워크 프로토콜, 서비스 제공업체 등의 시스템 구성요소의 다양성, 변동성 때문에 시스템의 검증이 복잡하다^[2]. 따라서, 올바른 동작을 보장하기 위해서는 IoT 시스템을 서비스 환경에 배치하기 전에 기능시험과 성능 평가가 선행되어야 하고, 부합되는 시험 시나리오와 시험기가 필요하다. IoT 시험 유형에는, 구성 별 통합 성능, 적합성, 호환성, 보안등의 시험 등이 있다.

IoT 가상환경 시험도구는 IoT 시스템의 품질과 안정성 확보를 위해 시험 구성용이, 반복성, 관리의 편리성, 확장성, 제어 기능을 가지고 IoT 기기의 행동을 모방하기 위해 가상으로 구축된 시험 프레임워크를 말한다^[3]. IoT 시험에서 실제 환경에서 수백 개의 노드로 구현하려면 많은 시간과 노력 및 자원의 사용을 요구한다. 또한, 실제 시험환경은 다른 요인에 의해 동적으로 변화하기 때문에 반복에 대해 동일한 결과를 재현하기가 어려워서 평가의 오류가 발생할 수 있다. 따라서, 가상환경 시험도구 물리적 구축과 관련된 높은 비용을 줄이고 대규모 IoT 네트워크를 실행하고, 실제 네트워크의 주요 복잡성과 다양성을 모방하여 사용자에게 IoT 성능시험을 위한 방법을 제공한다. IoT 시스템을 시험하기 위한 가상도구에는 IoT 시뮬레이터(Simulator), 테스트 베드(Test bed), 트래픽 생성기 (Traffic Generator) 등이 있다.

표 1. 주요 트래픽 생성기
Table 1. Traffic generator tool

종류	설명
PackETH	이더넷 링크에 패킷 시퀀스를 생성하고 파라미터 조절을 지원하는 간편하고 패킷 생성기
Ostinato	GUI기반 크로스 플랫폼, 오픈 소스 패킷/트래픽 발생기와 분석기
IoTGen	패킷 수준 트래픽 생성기 도구이며 IoT 합성 트래픽 모델 및 생성
IoT-Flock	MQTT 및 CoAP의 정상 및 비정상 IoT 보안 시험 트래픽을 생성할 수 있는 오픈 소스 프레임워크
Ixia IoT	Wi-Fi 시험의 기반. 사용자가 간섭, 로딩, 거리 등 현실적인 조건에서 IoT 기기 검증

네트워크 트래픽 생성기는 인터넷 패킷을 생성하여 네트워크에 주입하는 데 사용되는 도구이다. 네트워크상의 기계와 장치에 의해 생성되는 통신 패킷과 페이로드를 모방하여 네트워크의 보안평가, 성능평가, 네트워크기능 벤치마킹, 문제해결을 위한 도구이다. 표1 은 주요 IoT 관련 트래픽 생성기의 설명이다.

3. 데이터 플레인 가속화 기술과 트래픽 생성기

데이터플레인 가속화(Data Plane Acceleration) 기술은 패킷 폴링(Polling), 직접 버퍼 접근, 메모리 사전할당 등을 통해 기존 리눅스 커널의 네트워크 스택에서 처리 지연 문제를 개선하여 패킷 처리의 속도를 높이는 기

술이다^[4].

DPDK(Data Plane Development Kit)는 패킷 처리를 사용자 계층에서 처리하여 가속화시키는 고속 패킷 IO API이다^[5]. 그림 1은 DPDK 네트워크 스택과 라이브러리 구성이다. 메모리, 버퍼, 큐, 플로우관리자, 폴모드 드라이버 사용한다. DPDK환경에서 네트워크카드에 수신된 패킷이 커널 영역을 거치지 않고 어플리케이션으로 전달되므로 사용자 공간과 커널 공간 간의 인터럽트와 메모리 복사에 따른 오버헤드가 없어 지연시간이 작고 높은 전송속도를 제공할 수 있다.

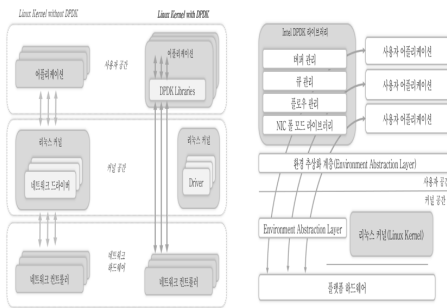


그림 1. DPDK 네트워크 스택 및 라이브러리 구성
 Fig. 1. DPDK network stack and library configuration

T-Rex는 DPDK 기반 오픈 소스의 저비용 트래픽 생성기다^[6]. 서버 하나로 최대 200Gbps까지 확장할 수 있으며 상태 비저장 및 상태 저장 트래픽용 TCP 스택과 L7 응용 프로그램을 시뮬레이션 하는 에뮬레이션을 지원한다. 성능에서 리눅스 기반 iperf 및 pack ETH을 능가한다.

4. IoT 네트워크 성능시험기 개선사항

IoT 네트워크 성능시험기는 IoT 시스템의 안정성시험과 분석을 위해 제어 가능한 방식으로 실제 네트워크 트래픽을 모방하여 프로토콜 트래픽 패킷을 공급하는 도구 또는 프레임워크이다. 새로운 IoT 장치와 애플리케이션(DUT 또는 SUT)의 성능을 분석하고 패킷 처리량(Packet Throughput), 지연 시간(Packet Delay), 패킷 손실(Packet Loss) 등의 각 인터페이스에 대한 중요 네트워크 통계를 파악할 수 있다. 이를 통해 시험자는 기기 또는 애플리케이션에 대한 벤치마크를 생성하고 성능 측면에서 한계를 정의하며 설계 결함을 식별할 수 있다^[7].

IoT 개별 장비는 저속데이터를 효과적으로 전송하도록 설계되었다. 하지만 전체 IoT 네트워크시스템은 많은

개별 IoT 장비가 연결되고, 전체 시스템을 시험하기 위해서는 고속 대용량의 시험기가 필요하다.

고속 시험기는 소프트웨어 기반으로 가능한 최소의 CPU, 메모리 등의 자원으로 10Gbps 또는 40Gbps NIC의 패킷 처리 용량에 빠른 시간에 도달할 수 있어야 한다. 또한, 트래픽 프로토콜과 동작에 유연성을 제공하여야 한다. 다양한 시험 시나리오에 따라 패킷 페이로드 크기, 패킷 시간 간격, 트래픽 플로우 수와 시간 간격 등의 구성이 쉽고 혼합된 IoT 트래픽 모델의 구현이 가능해야 한다. 마지막으로, 상용시험도구는 가격이 고가이고 새로운 기능과 문제점 분석에 대하여 사용자의 요구를 즉각적으로 반영하기 어렵다. 오픈소스 소프트웨어를 이용한 프레임워크 제공은 기능의 확장과 개선을 용이하게 한다.

III. 시스템 설계

1. 시스템 구성

네트워크를 통해서 대용량의 트래픽을 생성할 수 있는 DPDK 기반 고속 IoT 네트워크 성능 시험기를 제안한다. 그림 2는 제안하는 시스템의 구성을 나타낸다. 시스템은 사용자응용계층, T-Rex 코어 계층, 네트워크 인터페이스로 구성한다.

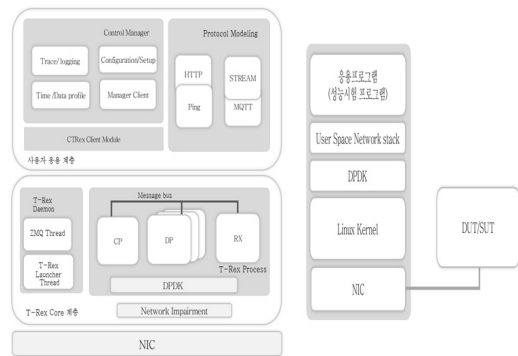


그림 2. 제안된 고속 IoT성능시험기 구성
 Fig. 2. Proposed high-speed IoT performance tester configuration

사용자응용 계층은 제어 관리자 (Control Manager)와 프로토콜 모델링(Protocol Modelling) 블록으로 구성한다. IoT 디바이스의 사용자 프로토콜 트래픽 모델을 생성하여 T-Rex 코어 계층에 전달하고 관리한다. 제어 관리자는 매니저 클라이언트, T-Rex 코어 셋업 구성, 트

래픽 모델 입력 파라미터 관리, 추적로그 기능, 시험 자동화 인터페이스 모듈을 가진다. 프로토콜 모델링은 MQTT, HTTP, PING, STREAM 등의 네트워크 응용 프로토콜의 프로그램 등으로 구성한다. T-Rex 코어 계층은 DPDK 기반 T-Rex 트래픽 생성기 구동영역이다. 사용자 응용 계층과 RPC 인터페이스로 연결한다. 사용자 응용계층으로부터 전달받은 프로파일 정보를 로딩하고 데이터플랜(DP)에서 트래픽 패킷을 생성한다. 마지막으로, 네트워크 인터페이스계층은 생성된 패킷 트래픽을 NIC을 통해 네트워크로 전송하고 DUT로부터 네트워크를 통해 패킷을 수신한다.

2. 시스템 동작

그림 3는 제안시스템의 MQTT 프로토콜 트래픽 모델에서 구성요소들간의 동작흐름을 나타낸 것이다.

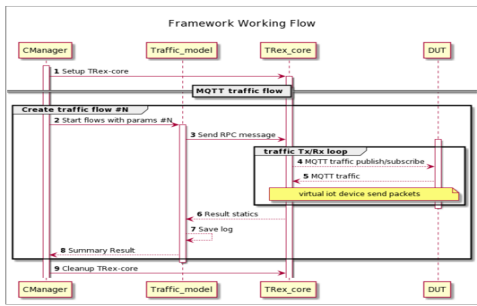


그림 3. MQTT 트래픽 모델 동작
Fig. 3. MQTT traffic model behavior

제어관리자 블록(CManger)은 구성/셋업 모듈에서 각 포트에서 사용할 네트워크 PCI 구성과 포트 IP정보, CPU 코어 설정 등의 구성 값을 기반으로 T-Rex 코어 계층과 함께 각 블록을 셋업 한다. 또한, 제어관리자 블록의 매니저 클라이언트 모듈은 MQTT 트래픽 모델의 패킷 사이즈, 패킷 간격, 전송횟수의 입력 Time/Data 프로파일 파라미터를 MQTT 트래픽 모델링 모듈로 전송하면서 트래픽을 생성을 명령한다.

MQTT 트래픽 모델링 모듈은 입력 파라미터를 바탕으로 프로그래밍 된 프로파일 정보를 JSON 포맷으로 생성하여 T-Rex 코어로 전송한다. T-Rex 코어는 프로파일을 로딩 후 패킷을 생성하고 DPDK 라이브러리를 이용하여 패킷을 직접 NIC로 전달하며, 물리계층을 이용하여 연결된 DUT로 트래픽을 전송한다.

패킷은 네트워크 경로를 거쳐 시험 대상인 DUT의 계

층을 흐르고 다시 T-Rex 코어로 회신 되어 돌아온다. 이때 DUT는 시험 패킷을 수용하고 패킷 부하에 따른 IOT DUT 기능을 확인하기 위해 별도의 기능블록 또는 시험 모드 프로그램을 가질 수 있다. MQTT 트래픽의 플로우는 가상의 IoT 디바이스의 동작으로 매핑할 수 있다.

트래픽 모델링 블록은 Tx/Rx 패킷 수, 바이트 수, 트래픽 에러, 지연시간 같은 통계정보를 T-Rex 코어로부터 전달받는다. 제어관리자 블록의 추적로깅 모듈은 동작과정에서의 주요 정보나 오류를 로그에 저장하거나 패킷 캡처를 제어한다. 매니저 클라이언트에게 각 트래픽 플로우의 전송한 메시지 전달정보 등의 전송 결과값을 요약하여 전달받는다.

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-09-24 21:09:54.86	100.0.0.24	210.162.0.72	TCP	74	43658 -> 1003 [SYN] Seq=204276118 Win=0
2	2021-09-24 21:09:54.89	210.162.0.72	100.0.0.24	TCP	74	1003 -> 43658 [SYN, ACK] Seq=204276166 Win=0
3	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	MQTT	111	Connect Command
4	2021-09-24 21:09:54.89	210.162.0.72	100.0.0.24	TCP	68	1003 -> 43658 [ACK] Seq=204276167 Win=0
5	2021-09-24 21:09:54.89	210.162.0.72	100.0.0.24	MQTT	78	Connect Ack
6	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	TCP	68	43658 -> 1003 [ACK] Seq=204276166 Win=0
7	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	MQTT	117	Publish Message (id=1001) [monitor]!
8	2021-09-24 21:09:54.89	210.162.0.72	100.0.0.24	MQTT	78	Publish Ack (id=1001)
9	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	TCP	68	43658 -> 1003 [ACK] Seq=204276170 Win=0
10	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	MQTT	2042	Publish Message (id=1002) [monitor]!
11	2021-09-24 21:09:54.89	210.162.0.72	100.0.0.24	MQTT	78	Publish Ack (id=1002)
12	2021-09-24 21:09:54.89	100.0.0.24	210.162.0.72	TCP	68	43658 -> 1003 [ACK] Seq=204276183 Win=0
13	2021-09-24 21:10:00.35	100.0.0.24	210.162.0.72	MQTT	1411	Publish Message (id=1003) [monitor]!
14	2021-09-24 21:10:00.35	210.162.0.72	100.0.0.24	MQTT	78	Publish Ack (id=1003)
15	2021-09-24 21:10:00.35	100.0.0.24	210.162.0.72	MQTT	68	Disconnect Req
16	2021-09-24 21:10:00.35	210.162.0.72	100.0.0.24	TCP	68	1003 -> 43658 [FIN, ACK] Seq=204276183
17	2021-09-24 21:10:00.35	100.0.0.24	210.162.0.72	TCP	68	43658 -> 1003 [FIN, ACK] Seq=204276186
18	2021-09-24 21:10:00.35	210.162.0.72	100.0.0.24	TCP	68	1003 -> 43658 [ACK] Seq=204276186 Win=0

그림 4. MQTT Publish 트래픽 캡처
Fig. 4. Capture MQTT Publish traffic

그림 4는 패킷 분석을 위한 와이어샤크 프로그램을 이용하여 캡처한 것으로 제안 시스템에서 생성한 가상 IoT 디바이스에서 서버로 전송하는 MQTT 모델의 발행(Publish) 트래픽이다. MQTT 는 연결을 위한 Connect/Ack, 클라이언트와 브로커 서버간 패킷을 전송하는 발행(Publish) 메시지, 연결종료하는 Dis connect Req 구성된다. 발행 메시지가 3번이 반복하고 각 메시지에 Ack 패킷을 수신 받는다. MQTT는 다수의 발행 메시지 전송 과정에서 TCP의 SYN 연결이 없이 여러 메시지를 전송하므로 SYN 오버 헤더를 줄일 수 있다.

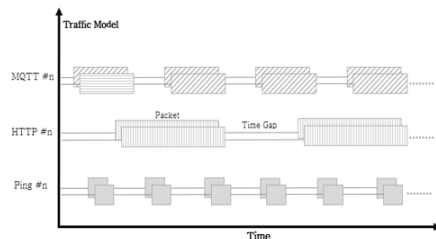


그림 5. 혼합 트래픽(Mixed traffic) 모델 예시
Fig. 5. Mixed traffic model example

제안한 시스템은 사용자 응용계층을 통해 실제 트래픽을 모방하는 혼합된 IoT트래픽 생성이 가능하며 DUT 네트워크 인터페이스의 안정성을 사전 확인한다. 그림 5는 MQTT, PING, HTTP의 혼합 트래픽 모델을 나타낸 것이다. 시험시나리오와 트래픽 모델링을 통하여 각 프로토콜별로 패킷 크기와 간격을 조절할 수 있다. 제안된 고속 IoT시험기를 통해 IoT 패킷 생성뿐만 아니라 다수의 혼합 응용 프로토콜 생성으로 IoT 네트워크의 실제 트래픽을 모방할 수 있을 확인하였다.

3. 시스템 성능측정 및 결과

본 논문에서 제안한 고속 IoT 네트워크 성능시험기의 트래픽 발생 성능 분석을 위해 패킷 및 비트 처리율(Throughput)과 CPU 사용율을 확인하였다. 처리율은 장치에 의해 프레임이 손실되지 않는 단위 시간당 전송 데이터 처리하는 양이다. PPS(Packet Per Second)는 초당 전송된 패킷의 수이고 BPS(Bit Per Second)는 초당 전송된 비트(bit) 수이다. CPU 사용율은 전송데이터 처리시 CPU 코어의 사용비율을 %로 표시한 값이다.

그림 6은 제안 시스템을 시험하기 위한 구성환경이다. 시험은 Ubuntu 18.04가 설치된 Dell 서버에서 하였다. 10G 가상NIC #0과 #1을 루프-백(loop-back) 연결을 하였고, CPU 코어(Core) 1개를 사용하여 TCP 패킷을 30초간 생성하여 패킷 처리율과 비트 처리율을 T-Rex 코어 통계를 이용하여 측정하였다.

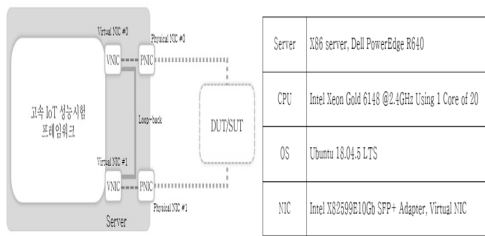


그림 6. 제안 시스템 시험환경 구성
 Fig. 6. Proposal system test environment configuration

표 2. 제안 시스템의 TCP 처리성능
 Table 2. TCP processing performance in system

Packet size (in Bytes)	처리율		CPU 사용률 (%)
	Gbps	MPPS	
64	0.81	3.18	100
128	1.59	3.12	100
256	3.09	3.02	100
512	6.06	2.96	100
1024	9.11	2.22	81
1408	11.51	2.05	75
2048	12.00	1.46	60
4096	14.31	0.88	45
6144	14.84	0.60	40
8950	15.19	0.43	36

표 2는 제안 시스템의 TCP 트래픽 (HTTP 프로토콜) 발생 성능을 5회 측정하여 평균한 결과 값이다. 64Byte에서 8950Byte 패킷 크기에 따라 데이터 처리율과 CPU 사용율의 변화를 측정하였다.

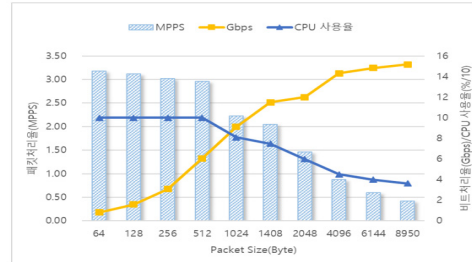


그림 7. 제안 시스템의 TCP 처리성능
 Fig. 7. TCP processing performance of the proposed system

그림 7은 표 2를 이용하여 TCP에서 패킷 크기에 따른 처리율의 변화를 그래프로 나타낸 것이다. 512Byte 이하의 패킷 크기에서 단일 코어의 CPU사용률이 100%로 포화되어 패킷 처리율 평균은 3MPPS, 비트 처리율 평균 2.9Gbps로 전송한계를 가진다. 1024Byte 이상의 패킷 크기에서는 비트 처리율이 NIC 성능인 10Gbps에 수렴하여 패킷 크기가 커질수록 패킷 처리율과 CPU 사용률은 작아진다. 1024Byte 이상에서 패킷 처리율은 평균 1.27MPPS, CPU 사용률은 평균 50%이다. CPU 단일 코어를 사용하여 1024Byte 패킷 크기로 CPU 80%를 사용하여 10Gbps의 비트 처리 성능을 보여주고 있다.

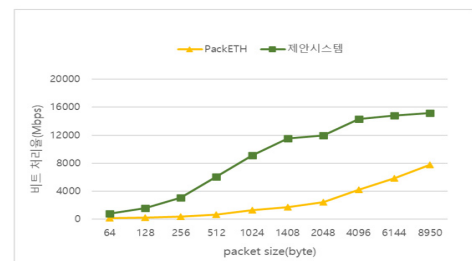


그림 8. 시스템 TCP 트래픽 처리율 비교
 Fig. 8. TCP Traffic Throughput Comparison

그림 8은 TCP 트래픽에서 비트처리율을 기존 시스템에서 측정값과 비교한 것이다^[8]. 102464 Byte 패킷 크기에서 제안시스템은 9113 Mbps의 TCP 비트 처리율을 보이며 기존시스템은 1310 Mbps의 처리율을 보여준다. 전체적으로 1024 Byte 이하의 패킷 크기에서 제안된 시스템은 기존 시스템결과의 코어성능의 고려할 때 최소 평균 3.5 배의 처리율 개선 효과를 보여준다.

IV. 결 론

본 논문은 개방형 소프트웨어인 DPDK기반 트래픽 생성기를 이용한 고속 IoT 성능시험기를 제안하였고 시험기의 동작을 확인하고 성능을 측정하여 기존시스템과 비교하였다.

제안된 고속 IoT시험기는 단일 IoT 패킷 생성뿐만 아니라 다수의 혼합 응용 프로토콜 생성으로 IoT 네트워크의 실제 트래픽을 모방하여 제어가 가능한 트래픽을 모델링할 수 있음을 확인하였다. 그리고 TCP 성능측정에서 초당 전송 트래픽 처리에서 기존 시스템 보다 최소 3배 이상의 처리율 개선을 확인하였다.

DPDK를 이용하여 소프트웨어기반의 IoT 성능시험기를 구현할 경우 패킷 처리의 오버 헤더를 효율적으로 처리하고 효과적인 트래픽 모델링으로 대용량, 고속 IoT 시험기로 적용이 가능하며 IoT 시스템의 고품질 서비스 제공에 기여할 수 있다.

References

- [1] Popereshnyak, S., Suprun, O., Suprun, O., Wieckowski, T., "IoT application testing features based on the modelling network", In 2018 XIV-MEMSTECH, 2018. DOI: <https://doi.org/10.1109/MEMSTECH.2018.8365717>
- [2] Esquiagola, J., de Paula Costa, L. C., Calcina, P., Fedrechski, G., Zuffo, M., "Performance Testing of an Internet of Things Platform", In IoTBDS, pp. 309-314, 2017. DOI: <https://doi.org/10.5220/0006304503090314>
- [3] N. D. Patel, B. M. Mehtre, and R. Wankar., "Simulators, Emulators, and Test-beds for Internet of Things: A Comparison", in 2019 Third International conference on I-SMAC, pp. 139-145, 2019. DOI: <https://doi.org/10.1109/I-SMAC47947.2019.9032519>
- [4] Choi, K.I, "Trends on Data Plane Acceleration Technology", Electronics and Telecommunications Trends, Vol. 30, No. 2, pp. 87-94, 2015. DOI: <https://dx.doi.org/10.22648/ETRI.2015.J.300210>
- [5] Intel, "DPDK", <https://www.dpdk.org/>, [Aug. 15, 2021]
- [6] Cisco, "T-Rex's Documentation", <https://trex-tgn.cisco.com/>, [Jul. 28, 2021]
- [7] L. Rodoni, Tobias Bühler, "High-speed Traffic Generation", ETH Zurich, pp. 7-22, 2020.
- [8] S. Srivastava, S. Anmulwar, A. M. Sapkal, T. Batra, A. K. Gupta and V. Kumar, "Comparative study of various traffic generator tools", RA ECS 2014 Recent Advances, pp. 1-6, 2014. DOI: <https://doi.org/10.1109/RAECS.2014.6799557>

저 자 소 개

정 영 준(정회원)



- 2001년 ~ 2005년 : LG전자
- 2005년 ~ 현재 : 에릭슨LG R&D 센터
- 2022년 : 건국대학교 정보통신대학원 융합정보기술학과(공학석사)
- 관심분야 : 5G Test 시뮬레이터, 이동통신 Core System, IoT, AI, Security

정 이 도(정회원)



- 2004년 ~ 2022년 : 대보정보통신(주)
- 2022년 ~ 현재 : (주)기술법인세종 대표
- 2022년 : 건국대학교 정보통신대학원 융합정보기술학과(공학석사)
- 관심분야 : ITS(지능형교통체계), 스마트교통, ICT 및 공공분야 정보통신설비 구축,설계,감리,유지관리

이 성 화(중신회원)



- 1989년 : 건국대학교 공과대학 전자공학과 졸업(공학사)
- 1991년 : 건국대학교 대학원 전자공학과 졸업(공학석사)
- 1998년 : 건국대학교 대학원 전자공학과 졸업(공학박사)
- 1991년 ~ 1993년 : 롯데전자(주) 연구원
- 1999년 ~ 2023.8.31 : 제주한라대학교 컴퓨터정보과 교수
- 주관심분야 : AIoT, Mobile, AI

김 진 태(정회원)



- 1991년 : 건국대학교 공과대학 전자공학과 졸업(공학사)
- 1993년 : 건국대학교 대학원 전자공학과 졸업(공학석사)
- 1999년 : 건국대학교 대학원 전자공학과 졸업(공학박사)
- 2004년 ~ 현재 : (주)파이브텍 대표이사
- 1999년 ~ 현재 : 건국대학교 정보통신 대학원 융합정보기술학과 겸임교수
- 주관심분야 : Real-Time System and Mobile IoT System