

# 메쉬 측정기법을 활용한 네트워크 품질 관리 방법 (The Method of NQM using a Mesh Measurement Technology)

(주)유미테크 기업부설연구소\*, 한국전자통신연구원 융합기술상용화실\*\*  
윤대일\*, 이재기\*\*, 서현용\*, 이진우\*, 황택진\*\*

차 례

1. 서론
2. 시험환경 구성 및 품질측정 방법
3. 유형별 실험 평가
4. 향후 계획 및 기대 효과
5. 결론

■ keyword : | 네트워크 품질 | SLA | TWAMP | MESH | 트래픽 컨트롤 |

## 1. 서론

5G기반의 최신 무무선 통신은 초저지연, 초연결, 초대용량 특성을 요구하고 있다. 이를 만족하기 위하여 통신망을 구성하고 있는 망 장치의 품질을 상시 측정하고, 정확하게 측정할 필요가 있다.

망 장치의 정확한 품질 측정을 하기 위한 방법으로 TWAMP(Two-Way Active Measurement Protocol) 표준 방법이 공개되어 있고, 표준 품질 지표를 제시하고 있다.[1] 하지만 통신망은 시간의 흐름과 트래픽의 흐름에 따라 서로 다른 네트워크 품질을 보이고 있으며, SLA(Servive Level Agreement)를 만족하는 네트워크 품질을 제공하기 위하여 통신망 전체 영역에 대한 상시 측정이 필요하다.

본 논문에서는 TWAMP를 활용하여, 통신망 전체에 상시 측정할 수 있는 방법으로 메쉬 측정기법을 설명하고, 본 방법의 유효성을 증명하기 위하여, 실험환경 구축 및 시험망을 구성하고, 불특정 시간에 네트워크의 품질을 떨어뜨린 후 그 원인을 찾아내는 실험 방법을 제시한다.

또한 실험결과를 통하여 실험 통신망에서 불특정시간에 SLA를 만족하지 못하는 통신 경로를 찾아내는 결과를 제시함으로써 본 방법의 유효성을 입증하였다.

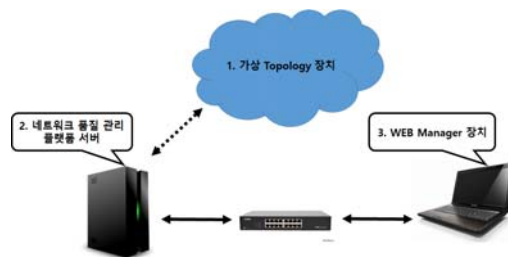
## 2. 시험환경 구성 및 품질측정 방법

### 2.1 실험 환경

실험 환경은 3가지로 구성이 되며, 각각은 다음과 같다. 먼저 전체 환경 구성에 대해 설명하고 가상 토폴로지 장치 환경 구성, 네트워크 품질 관리 플랫폼 환경 구성, 그리고 WEB Manager 환경 구성에 대하여 순서대로 설명하기로 한다.

#### 2.1.1 전체 환경 구성

전체 시험환경은 그림 1과 같이 가상 토폴로지 장치, 네트워크 품질 관리 플랫폼 서버, 그리고 WEB Manager 장치로 구성된다.



▶▶ 그림 1. 전체 시험 환경 구성

전체 시험환경에 참여한 실험 장비와 실험 SW의 명세는 아래의 표 1, 2와 같으며, 측정 프로그램의 구성은 그림 2와 같다.

표 1. 실험 장비

장비 명	규격
네트워크 품질관리 플랫폼 서버(1EA)	Intel(R) Xeon(R) CPU E5-2660 v2 @2.60GHz x 2EA 1TB/256GB RAM Gigabit Ethernet
가상 토폴로지 장치 (1EA)	Intel(R) Xeon(R) CPU E5-2660 v2 @2.60GHz x 2EA 1TB/128GB RAM Gigabit Ethernet
WEB Manager 장치(1EA)	Intel Core i5-5200U core 2.2GHz 256GB/8GB RAM

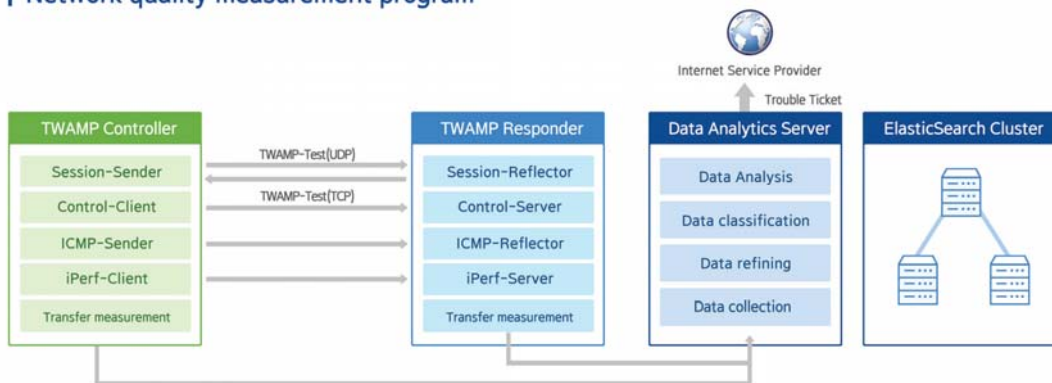
네트워크 품질관리 측정 프로그램은 크게 TWAMP Controller 및 Responder와 데이터 분석 서버로 구성되어 TWAMP

P 규격 기술 및 확장기술, ICMP/iPerf의 확장기술 등을 구현, 스트리밍 데이터에 대한 전송처리 기능과 신뢰성 제어, 다중 Reflector에 대한 품질측정을 수행한다.[5]

표 2. 실험 S/W

S/W	기능 모듈
네트워크 품질관리 플랫폼 S/W	품질 측정 요청 모듈
	품질 측정 결과 분석 모듈
	장애 관리 모듈
품질 측정 S/W	품질 측정 WEB
	TWAMP Sender 모듈
	TWAMP Reflector 모듈
	TWAMP Control Server 모듈
	TWAMP Control Client 모듈

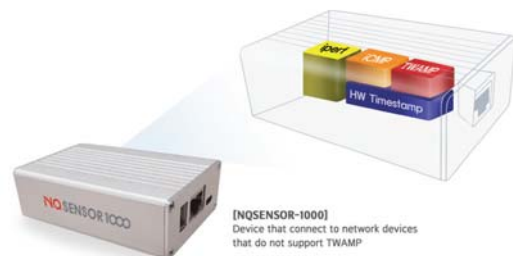
| Network quality measurement program



▶▶ 그림 2. 네트워크 품질 측정 프로그램 구성도

또한 통신사업자 및 자가망 사업자가 제공하는 망장치가 TWAMP를 지원하지 않는 경우도 고려하여 주변 망 검색기능과 확장성을 갖춘 포터블 기능을 가지는 네트워크 품질 측정 보조 장치(NQSENSOR-1000)도 제공하여 5G와 같은 실시간, 대용량, 초연결을 특징으로 하는 네트워크서비스를 제공하도록 하였다.

본 장치의 주요 특징은 TWAMP 표준을 준수하고 최대 1 Gbps를 지원, ICMP(Internet Control Message Protocol) 및 iPerf를 함께 제공하며, NPT(Number Translation Processor)를 기본으로 제공한다.



▶▶ 그림 3. 네트워크 품질 측정 보조 장치(NQSENSOR-1000)

세부 규격은 아래와 같으며, 표준준수 및 호환성을 제공, 쉽게 망장치에 연결, 빅데이터 서버를 탐색, 연동할 수 있으며, 기본적인 네트워크 품질측정 프로그램을 임베디드 OS

및 네트워크 드라이버와 함께 포팅하여 소형화된 이동장치로 품질관리 편리성을 제공하였다.[3]

네트워크 품질 측정의 주요지표가 되는 Loss, Latency/Delay, TTL(Time to Live), Duplicate Packets & Out of Order Packets, PDV(Jitter), IPDV(IP Delay Variation), Inter Delay 등을 측정 가능하도록 프로그램을 구현, 측정이 가능하도록 하였다.[3],[6]

### Features

- It complies with the TWAMP standard.
- Supports up to 1Gps.
- Supports ICMP and IPERF.
- It is based on NTP.
- It is convenient to use with MQMP.

### Hardware Specifications

- CPU : Quad-Core ARM
- Storage : 16GB
- RAM (SDRAM) : 1 GB
- Network : 1GB Eth Port x 1
- OS : Linux
- Power : 5V 2A

[주요 특징 및 하드웨어 사양]

### Standard compliance

- **RFC2898** PKCS #5: Password-Based Cryptography Specification Version 2.0
- **RFC4656** A One-way Active Measurement Protocol (OWAMP)
- **RFC5357** A Two-Way Active Measurement Protocol (TWAMP)
- **RFC5618** Mixed Security Mode for the Two-Way Active Measurement Protocol (TWAMP)
- **RFC5938** Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)
- **RFC6038** Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features
- **RFC5481** Packet Delay Variation Applicability Statement

[표준 준수 및 호환성 제공]

#### 2.1.1.1 가상 토폴로지 장치

Firecracker 기반 네트워크 품질 측정 도구(Sender/Reflector)가 설치되어 있는 VM과 오픈소스 기반 네트워크 운영 체제인 VyOS가 설치된 가상 Router로 구성되어 있는 장치이다.[9]

#### 2.1.1.2 네트워크 품질 관리 플랫폼 서버

네트워크 품질 측정 요청 기능, 네트워크 품질 측정 결과를 분석하여 SLA를 만족하지 못하는 통신 경로를 탐지하는 기능, 장애 관리 기능을 갖는 네트워크 품질 관리 서버이다.

네트워크 품질관리플랫폼은 표준기술기반 품질측정 기술과 통신사업자들의 보유하고 있는 신형 및 구형 망 장치를 모두 지원이 가능하며, 비정상 망장치 탐색, 동적 임계치 설정에 따른 진단 기능과 실시간 품질 검사, 상시품질검사, Trouble Ticket 발생 등 차별성을 제공한다. 특히, 안정된 고품질 인

터넷 서비스를 제공하고 Loss Rate 및 Latency가 낮고 BW (Bandwidth)가 넓은 일반적인 개념의 좋은 품질보다는 네트워크의 정확한 품질측정 방법의 도입이 필요하며, 이에 대한 대안으로 OWAMP(One-Way Active Measurement Protocol)나 TWAMP 등을 활용한 네트워크 품질측정 방법을 제공, 네트워크 품질 측정의 필요성 및 편리성을 만족할 수 있는 플랫폼을 개발, 적용하여 검증하였다.[2],[4]

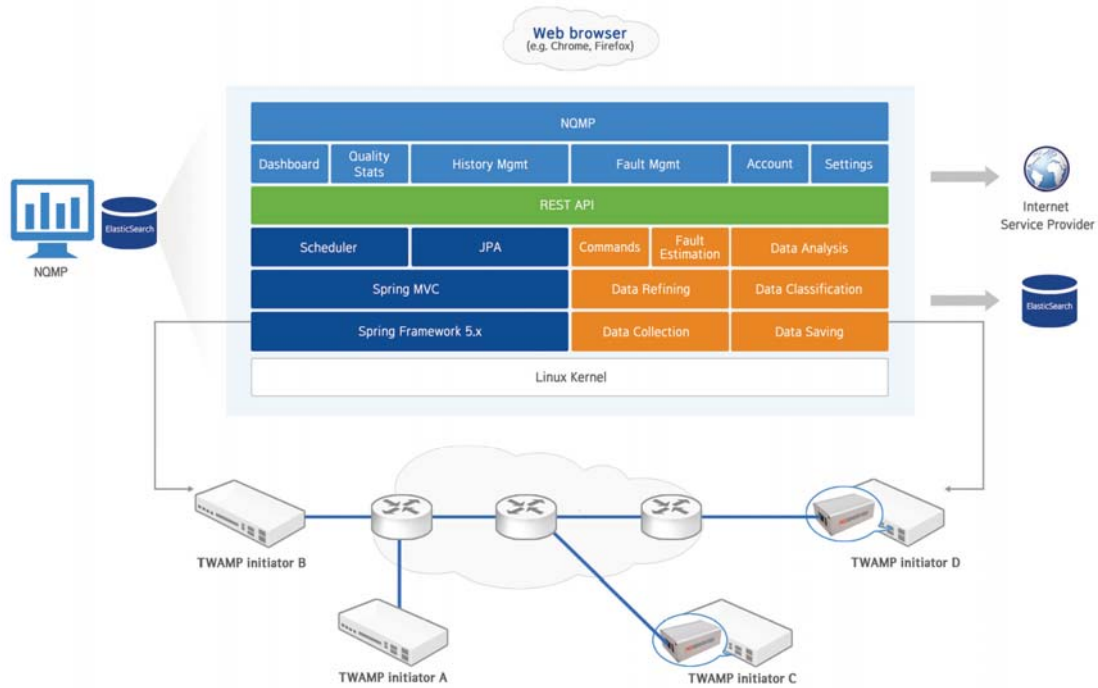
#### Introducing Network Quality Management Platform



즉, 품질 측정에 적용한 네트워크 품질관리플랫폼(NQMP : Network Quality Management Platform)의 전체 구조는 그림 4와 같으며, 크롬, FireFox 등 웹 브라우저를 사용하여

고품질 네트워크 유지 및 능동적 네트워크 관리, 효율적인 인적/물적 자원 운영과 측정결과 데이터 기반의 대응 및 조치가 가능한 네트워크 품질관리플랫폼을 개발, 적용하였다.

**| Configuration Of Network quality management platform**



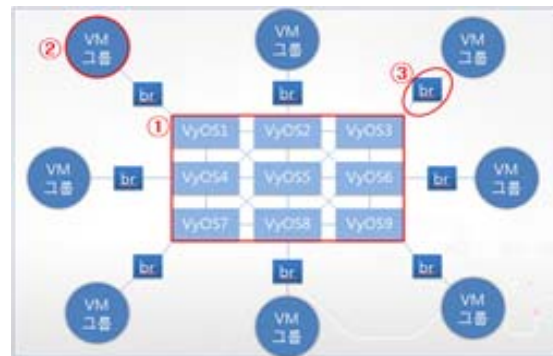
▶▶ 그림 4. 네트워크 품질관리플랫폼(NQMP)의 구조

2.1.1.3 WEB Manager 장치

네트워크 품질 측정 요청 및 품질 측정 결과, 장애를 확인할 수 있는 기능을 제공하는 WEB Manager가 설치되어 있는 장치이다.

2.1.2 가상 토폴로지 장치 환경 구성

가상 토폴로지 장치 환경은 아래 그림 5와 같이 9개의 VyOS 네트워크 운영체제로 구성되어 있는 부분과 8개의 VM 그룹으로 구성되어 있는 부분, 그리고 VyOS와 VM(Virtual Machine) 그룹간의 네트워크 연결 설정 부분 등 총 3개 부분으로 구성되어 있다.



▶▶ 그림 5. 가상 토폴로지 장치 환경 구성

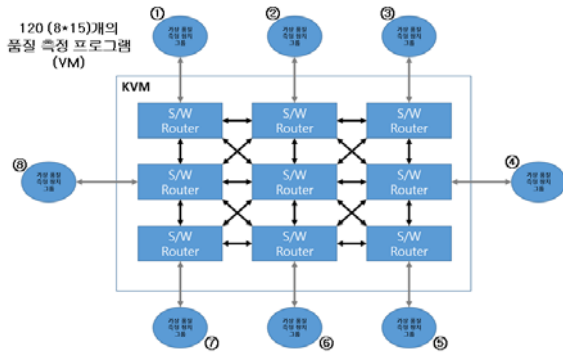
2.1.2.1 9개의 VyOS 네트워크 운영체제 환경 구성

가상 네트워크 구성을 위하여 네트워크 운영체제인 VyOS를 이용하여 9개의 S/W 라우터를 구성하며, 구성 방법은 다음과 같다.[7],[8]

먼저 VyOS 이미지를 다운로드 한 후 VMM(Virtual Machine Manager)을 이용하여 미리 준비된 9개의 가상머신에

VyOS를 설치하고 9개의 VyOS간의 네트워크 연결을 설정한다.[9] 이 때 VyOS에서 OSPF(Open Shortest Path First) 프로토콜을 이용하여 설정한다. VyOS간의 네트워크 연결에 대한 부연설명으로 OSPF 프로토콜 동작원리는 다음과 같다. [10]

OSPF 프로토콜은 IP 망 토폴로지 정보를 서로 간에 주고 받는데, 이때 각각의 VyOS는 자신의 링크 정보를 다른 OSPF가 설정된 VyOS에게 전파하는데, 전파를 받은 VyOS는 링크 정보 중 Cost값을 통하여 해당 VyOS 목적지까지의 최단 경로 토폴로지(SPT : Short Path Topology)를 구성하는 방식으로 동작한다.



[네트워크 품질측정을 위한 Topology 구성 예]

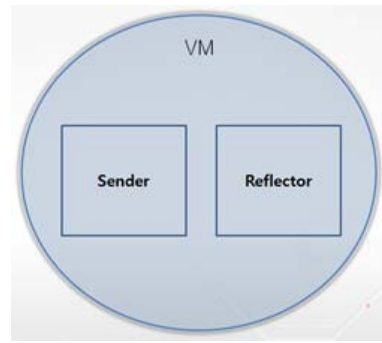
2.1.2.2 8개 가상머신 그룹 환경 구성

2.1.2.2.1 가상머신 그룹 구성

가상머신(VM) 그룹이란 네트워크 품질측정을 위한 가상 호스트 VM의 그룹으로서 본 환경에서는 Amazon에서 제공하는 VMM인 Firecracker를 통해 각 그룹별로 15개, 전체 120개의 VM을 구성하였으며, 각각의 그룹은 하나의 S/W 라우터에 연결되어 있는 형태를 갖는다. (위 네트워크 품질측정을 위한 토폴로지 구성 예시 그림 참조)

2.1.2.2.2 가상머신(VM) 구성

네트워크 품질측정을 위한 가상 호스트 VM으로 TWAMP 측정을 위한 Sender와 Reflector가 설치되어 있다. 이 때 Sender는 TWAMP 측정을 위하여 측정패킷을 전송하는 기능을 담당하며, Reflector는 Sender로부터 수신한 측정패킷에 품질측정에 필요한 정보를 추가하여 다시 Sender로 패킷을 전송하는 기능을 담당한다.



▶▶ 그림 6. VM 구성

2.1.3 네트워크 품질관리플랫폼 서버 환경 구성

앞에서 설명한 가상 토폴로지 장치를 관리하기 위한 서버로서 상시 측정을 위하여 30초 간격으로 Sender를 실행하도록 설정하여 메쉬 형태로 동작하는 기능, 품질 측정 기능, 그리고 메쉬와 품질 측정 기능 동작 시 필요한 Reflector들의 정보 관리 기능으로 구현 되어 있다.

2.1.3.1 메쉬(Mesh) 동작 기능

VM이 네트워크 품질 관리 플랫폼 서버에게 Sender를 실행하는 명령을 받으면 VM에서는 네트워크 품질 관리 플랫폼 서버에게 품질 측정을 실행할 목적지 IP 정보를 포함하고 있는 Reflector 정보를 요청한다.

요청 결과인 Reflector 정보를 수신한 VM은 해당 Reflector IP 에게 순차적으로 30초 주기마다 Sender를 실행 한다. Sender는 측정 후 결과 Data를 네트워크 품질 관리 플랫폼에 저장 한다.

2.1.3.2 품질 측정 기능

품질 측정 기능이란 네트워크 품질 측정 관리 플랫폼 서버가 WEB Manager 에게 품질 측정 요청 Data를 수신하면 해당 Data를 기반으로 Reflector IP에게 품질 측정 요청을 송신하는 기능을 말한다.

2.1.3.3 Reflector 정보 관리 기능

Reflector 정보 관리 기능이란 목적지 IP를 포함하고 있는 Reflector 정보에 대하여 추가, 삭제, 변경과 같은 작업들을 수행 할 수 있는 기능을 말한다. 기능 중에 이전에 메쉬 동작 기능에서 설명 했듯이 Reflector 정보를 송신하는 기능도 Reflector 정보 관리 기능에 포함 된다.

2.1.4 WEB Manager 환경 구성

WEB Manager 환경이란 이전에 설명한 네트워크 품질 관리 플랫폼 서버가 수행하는 작업을 WEB 환경에서 편리하게 이용하도록 하계끔 구현한 환경을 말한다.

WEB Manager는 주로 품질 측정, 품질 이력, Reflector 관리 페이지로 이루어져 있다.

## 2.2 실험 조건 및 방법

### 2.2.1 사전 준비 조건

환경 구성이 완료된 상태에서 전체 120개 호스트 VM간의 네트워크 품질 측정이 메쉬 형태로 동작하는 상태이어야 한다.

### 2.2.2 실험 기준

실험기준은 아래와 같이 가정하여 진행하였으며, 이를 통하여 SLA를 만족하지 못하는 이상 경로를 탐지하는 실험을 진행하였다.

표 3. SLA 비만족 실험 기준

SLA 비만족 실험 기준
Lost Packet 발생
Duplicate Packet 발생
Out of Order Packet 발생
PDV -10ms 이하, 10ms 이상 소요
IPDV -5ms 이하, 5ms 이상 소요

### 2.2.3 Lost Packet 경로 탐지 실험

#### 2.2.3.1 Traffic Control 설정

구성된 토폴로지 상에서 Lost Packet을 발생시키기 위하여 특정 호스트 VM에 Traffic Control을 설정하였으며, 이에 대한 설정 방법은 다음과 같다.

```
$ tc qdisc add dev eth0 root netem loss 10%
```

Lost Packet이 발생 될 확률을 10%로 설정하였다. 이후 정상적으로 설정이 되었는지 확인 한다.

```
$ tc qdisc list
```

이러한 Traffic Control 설정을 불특정 시간에 주기적으로 활성화/비활성화 하기 위하여 Shell Script 파일을 만든 후, 특정 시각에 명령어가 수행 되도록 등록이 가능한 리눅

스용 작업 스케줄러인 Cron을 사용한다. 먼저 Shell Script 파일 내용은 다음과 같다.

```
tc qdisc del dev eth0
tc qdisc add dev eth0 root netem loss 10%
sleep 30
tc qdisc del dev eth0 root
```

그리고 앞에서 만든 Shell Script 파일을 Cron 스케줄에 등록하는 방법은 다음과 같다.

```
$ crontab -e
```

그 후 Shell Script 반복 주기 입력 및 Shell Script 파일을 실행하는 Command를 입력한다.

```
* * * * * /root/tc_lost_packet.sh -p 10 -s 30 >> /root/tc.log 2>&1
```

1분에 한번씩 Shell Script를 반복하여 실행하고 Packet Loss 확률은 10%, Traffic Control 유지시간은 30초로 설정 하였다. 그리고 해당 Command의 실행 결과로 보여 지는 출력 문자를 로그파일로 저장하기로 하였다. 그 후 Cron 서비스를 재시작하고 서비스가 정상적으로 동작하는지 확인 한다.

```
$ service cron restart
$ service cron status
```

### 2.2.4 Duplicate Packet 경로 탐지 실험

#### 2.2.4.1 Traffic Control 설정

구성된 토폴로지 상에서 Duplicate Packet을 발생시키기 위하여 특정 호스트 VM에 Traffic Control을 설정하였으며 이에 대한 설정 방법은 다음과 같다.

```
$ tc qdisc add dev eth0 root netem duplicate 10%
```

Duplicate Packet이 발생 될 확률을 10%로 설정하였다. 이후 정상적으로 설정이 되었는지 확인 한다.

```
$ tc qdisc list
```

이러한 Traffic Control 설정을 불특정 시간에 주기적으로 활성화/비활성화 하기 위하여 Shell Script 파일을 만든 후, 특정 시각에 명령어가 수행 되도록 등록이 가능한 리눅

스용 작업 스케줄러인 Cron을 사용한다. 먼저 Shell Script 파일 내용은 다음과 같다.

```
tc qdisc del dev eth0
tc qdisc add dev eth0 root netem duplicate 10%
sleep 30
tc qdisc del dev eth0 root
```

그리고 앞에서 만든 Shell Script 파일을 Cron 스케줄에 등록하는 방법은 다음과 같다.

```
$ crontab -e
```

그 후 Shell Script 반복 주기 입력 및 Shell Script 파일을 실행하는 Command를 입력한다.

```
***** /root/tc_duplicate_packet.sh -p 10 -s 30 >>
/root/tc.log 2)&1
```

1분에 한번씩 Shell Script를 반복하여 실행하고 Duplicate Packet 확률은 10%, Traffic Control 유지시간은 30초로 설정 하였다. 그리고 해당 Command의 실행 결과로 보여 지는 출력 문자를 로그파일로 저장하기로 하였다. 그 후 Cron 서비스를 재시작하고 서비스가 정상적으로 동작하는지 확인한다.

```
$ service cron restart
$ service cron status
```

## 2.2.5 Out of Order Packet 경로 탐지 실험

### 2.2.5.1 Traffic Control 설정

구성된 토폴로지 상에서 Out of Order Packet을 발생시키기 위하여 특정 호스트 VM에 Traffic Control을 설정하였으며 이에 대한 설정 방법은 다음과 같다.

```
$ tc qdisc add dev eth0 root netem delay 1ms
reorder 10%
```

Out of Order Packet이 발생 될 확률을 10%로 설정하였다. 이후 정상적으로 설정이 되었는지 확인 한다.

```
$ tc qdisc list
```

이러한 Traffic Control 설정을 불특정 시간에 주기적으로 활성화/비활성화 하기 위하여 Shell Script 파일을 만든 후, 특정 시각에 명령어가 수행 되도록 등록이 가능한 리눅

스용 작업 스케줄러인 Cron을 사용한다. 먼저 Shell Script 파일 내용은 다음과 같다.

```
tc qdisc del dev eth0
tc qdisc add dev eth0 root netem delay 1ms reorder
10%
sleep 30
tc qdisc del dev eth0 root
```

그리고 앞에서 만든 Shell Script 파일을 Cron 스케줄에 등록하는 방법은 다음과 같다.

```
$ crontab -e
```

그 후 Shell Script 반복 주기 입력 및 Shell Script 파일을 실행하는 Command를 입력한다.

```
***** /root/tc_outoforder_packet.sh -p 10 -s 30 >>
/root/tc.log 2)&1
```

1분에 한번씩 Shell Script를 반복하여 실행하고 Out of Order Packet 확률은 10%, Traffic Control 유지시간은 30초로 설정 하였다. 그리고 해당 Command의 실행 결과로 보여 지는 출력 문자를 로그파일로 저장하기로 하였다. 그 후 Cron 서비스를 재시작하고 서비스가 정상적으로 동작하는지 확인한다.

```
$ service cron restart
$ service cron status
```

## 2.2.6 PDV(Packet Delay Variation) 경로 탐지 실험

### 2.2.6.1 Traffic Control 설정

구성된 토폴로지 상에서 기준치 이상 값의 PDV를 발생시키기 위하여 특정 호스트 VM에 Traffic Control을 설정하였으며 이에 대한 설정 방법은 다음과 같다.

```
$ tc qdisc add dev eth0 root netem delay 200ms
```

기준치 이상 값의 PDV를 발생시키기 위하여 최소 패킷 Delay값을 200ms로 설정하였다. 이후 정상적으로 설정이 되었는지 확인 한다.

```
$ tc qdisc list
```

이러한 Traffic Control 설정을 불특정 시간에 주기적으로

로 활성화/비활성화 하기 위하여 Shell Script 파일을 만든 후, 특정 시각에 명령어가 수행 되도록 등록이 가능한 리눅스용 작업 스케줄러인 Cron을 사용한다. 먼저 Shell Script 파일 내용은 다음과 같다.

```
tc qdisc del dev eth0
tc qdisc add dev eth0 root netem delay 200ms
sleep 30
tc qdisc del dev eth0 root
```

그리고 앞에서 만든 Shell Script 파일을 Cron 스케줄에 등록하는 방법은 다음과 같다.

```
$ crontab -e
```

그 후 Shell Script 반복 주기 입력 및 Shell Script 파일을 실행하는 Command를 입력한다.

```
**** */root/tc_pdv_packet.sh -d 200 10 -s 30 >>
/root/tc.log 2>&1
```

1분에 한번씩 Shell Script를 반복하여 실행하고 최소 패킷 Delay 값은 200ms, Traffic Control 유지시간은 30초로 설정 하였다. 그리고 해당 Command의 실행 결과로 보여 지는 출력 문자를 로그파일로 저장하기로 하였다. 그 후 Cron 서비스를 재시작하고 서비스가 정상적으로 동작하는지 확인한다.

```
$ service cron restart
$ service cron status
```

## 2.2.7 IPDV(IP Delay Variation) 경로 탐지 실험

### 2.2.7.1 Traffic Control 설정

구성된 토폴로지 상에서 기준치 이상 값의 IPDV를 발생시키기 위하여 특정 호스트 VM에 Traffic Control을 설정하였으며 이에 대한 설정 방법은 다음과 같다.

```
$ tc qdisc add dev eth0 root netem delay 200ms
```

기준치 이상 값의 IPDV를 발생시키기 위하여 최소 패킷 Delay값을 200ms로 설정하였다. 이후 정상적으로 설정이 되었는지 확인 한다.

```
$ tc qdisc list
```

이러한 Traffic Control 설정을 불특정 시간에 주기적으로 활성화/비활성화 하기 위하여 Shell Script 파일을 만든 후, 특정 시각에 명령어가 수행 되도록 등록이 가능한 리눅스용 작업 스케줄러인 Cron을 사용한다. 먼저 Shell Script 파일 내용은 다음과 같다.

```
tc qdisc del dev eth0
tc qdisc add dev eth0 root netem delay 200ms
sleep 30
tc qdisc del dev eth0 root
```

그리고 앞에서 만든 Shell Script 파일을 Cron 스케줄에 등록하는 방법은 다음과 같다.

```
$ crontab -e
```

그 후 Shell Script 반복 주기 입력 및 Shell Script 파일을 실행하는 Command를 입력한다.

```
**** */root/tc_ipdv_packet.sh -d 200 10 -s 30 >>
/root/tc.log 2>&1
```

1분에 한번씩 Shell Script를 반복하여 실행하고 최소 패킷 Delay 값은 200ms, Traffic Control 유지시간은 30초로 설정 하였다. 그리고 해당 Command의 실행 결과로 보여 지는 출력 문자를 로그파일로 저장하기로 하였다. 그 후 Cron 서비스를 재시작하고 서비스가 정상적으로 동작하는지 확인한다.

```
$ service cron restart
$ service cron status
```

## 2.3 실험 결과

### 2.3.1 Lost Packet 경로 탐지 실험 결과

실험 방법 2.2.3의 Lost Packet 경로 탐지 실험을 통하여 측정된 실험 결과는 표 4와 같다.

표 4. Lost Packet 경로 탐지 실험 결과

	측정 패킷 수	Lost Packet
Traffic Control 설정 전	50	0
Traffic Control 설정 후	50	4



### 2.3.2 Duplicate Packet 경로 탐지 실험 결과

실험 방법 2.2.4의 Duplicate Packet 경로 탐지 실험을 통하여 측정된 실험 결과는 다음과 같다.

표 5. Duplicate Packet 경로 탐지 실험 결과

	측정 패킷 수	Duplicate Packet
Traffic Control 설정 전	50	0
Traffic Control 설정 후	50	5

### 2.3.3 Out of Order Packet 경로 탐지 실험 결과

실험 방법 2.2.5의 Out of Order Packet 경로 탐지 실험을 통하여 측정된 실험 결과는 표 6과 같다.

표 6. Out of Order Packet 경로 탐지 실험 결과

	측정 패킷 수	Out of Order Packet
Traffic Control 설정 전	50	0
Traffic Control 설정 후	50	4

### 2.3.4 PDV 경로 탐지 실험 결과

실험 방법 2.2.6의 PDV 경로 탐지 실험을 통하여 측정된 실험 결과는 다음과 같다.

표 7. PDV 경로 탐지 실험 결과

	측정 패킷 수	PDV
Traffic Control 설정 전	50	0.66963387
Traffic Control 설정 후	50	203.1592034

### 2.3.5 IPDV 경로 탐지 실험 결과

실험 방법 2.2.7의 IPDV 경로 탐지 실험을 통하여 측정된 실험 결과는 다음과 같다.

표 8. IPDV 경로 탐지 실험 결과

	측정 패킷 수	PDV
Traffic Control 설정 전	50	0.23399034
Traffic Control 설정 후	50	201.7205715

## 3. 유형별 실험 평가

### 3.1 LP, DP, OOP 실험 평가

#### 3.1.1 Lost Packet 경로 탐지 실험 평가

Traffic Control을 통해 Packet Loss율을 10%로 설정한 후 해당 경로에 대해 TWAMP를 활용하여 네트워크 품질 측정을 수행한 결과 전체 TWAMP 측정 패킷 50개 중 4개가 유실되는 것을 탐지할 수 있었다. 이에 따라 해당 경로에 패킷이 유실되는 경우 이를 TWAMP 측정을 통해 찾아낼 수 있음이 확인되었다.

#### 3.1.2 Duplicate Packet 경로 탐지 실험 평가

Traffic Control을 통해 Packet Duplicate율을 10%로 설정한 후 해당 경로에 대해 TWAMP를 활용하여 네트워크 품질 측정을 수행한 결과 전체 TWAMP 측정 패킷 50개 중 5개가 중복되는 것을 탐지할 수 있었다. 이에 따라 해당 경로에 패킷이 중복되는 경우 이를 TWAMP 측정을 통해 찾아낼 수 있음이 확인되었다.

#### 3.1.3 Out of Order Packet 경로 탐지 실험 평가

Traffic Control을 통해 Packet Reorder율을 10%로 설정한 후 해당 경로에 대해 TWAMP를 활용하여 네트워크 품질 측정을 수행한 결과 전체 TWAMP 측정 패킷 50개 중 4개가 순서가 바뀐 것을 탐지할 수 있었으며, 이에 따라 해당 경로에 패킷이 순서가 변경 되는 경우 이를 TWAMP 측정을 통해 찾아낼 수 있음이 확인되었다.

### 3.2 PDV 및 IPDV 실험 평가

#### 3.2.1 PDV 경로 탐지 실험 평가

Traffic Control을 통해 Packet Delay를 200ms로 설정한 후 해당 경로에 대해 TWAMP를 활용, 네트워크 품질 측정을 수행한 결과 전체 TWAMP 측정 패킷 50개에 대한 PDV 값이 증가하는 것을 탐지할 수 있었고 이에 따라 해당 경로에 지연이 발생하는 경우 이를 TWAMP 측정을 통해 찾

아낼 수 있음이 확인되었다.

### 3.2.2 IPDV 경로 탐지 실험 평가

Traffic Control을 통해 Packet Delay를 200ms로 설정한 후 해당 경로에 대해 TWAMP를 활용하여 네트워크 품질 측정을 수행한 결과 전체 TWAMP 측정 패킷 50개에 대한 IPDV 값이 증가하는 것을 탐지할 수 있었다. 이에 따라 해당 경로에 지연이 발생하는 경우 이를 TWAMP 측정을 통해 찾아낼 수 있음이 확인되었다.

## 4. 향후계획 및 기대 효과

향후계획 및 기대 효과는 통신사업자 등 사용자들의 추가 요구사항을 반영한 네트워크 품질관리 장치로써 고품질 네트워크를 유지하고 기존 네트워크서비스 구조를 수용하며, 망 사업자들의 효율적인 CAPEX/OPEX 사용이 가능한 5G 기반의 차세대 네트워크에 적용이 가능한 네트워크 품질관리 플랫폼으로 발전시키는 것이다.

또한 망의 주요 품질 특징인 초저지연, 초연결, 초실감 및 초지능, 초성능 등을 보장할 수 있는 정확한 네트워크 품질 관리에 필수적인 토털 솔루션으로써 측정 장치 및 품질관리 플랫폼으로 적용, 통신사업자 및 자기망사업자 들도 쉽고 편리하게 적용이 가능한 고품질 네트워크 관리 도구 및 플랫폼으로 널리 이용되도록 하는 것이다.

## 5. 결론

본 논문에서는 TWAMP를 활용하여, 통신망 전체에 상시 측정할 수 있는 방법으로 메쉬 측정기법을 제안, 설명하고 본 방법의 유효성을 증명하기 위하여, 실험 통신망을 구성하고 불특정 시간에 네트워크의 품질을 인위적으로 떨어뜨린 후 그 원인을 찾아내는 실험 방법을 제시하였다.

실험결과를 통하여 시험 통신망에서 불특정시간에 SLA를 만족하지 못하는 메쉬 전체 경로 중 문제되는 통신 경로를 찾아내는 결과를 제시함으로써 본 방법의 유효성을 입증하였다.

[3] Network Working Group, “PKCS #5: Password-Based Cryptography Specification Version 2.0”, 2000.  
 [4] IETF, “Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)”, 2010.  
 [5] IETF, “Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features” 2010.  
 [6] Network Working Group, “Packet Delay Variation Applicability Statement”, 2009.  
 [7] VyOS - an Open Source Linux-based Network OS [Website]. (2018, SEP 3), <https://vyos.io>  
 [8] VyOS 위키[Website]. (2019, Jan 1), [https://wiki.vyos.net/wiki/Main\\_Page](https://wiki.vyos.net/wiki/Main_Page)  
 [9] GitHub - firecracker-microvm/firecracker: Secure and fast microVMs serverless computing[Website]. (2019, Jan 1), <https://github.com/firecracker-microvm/firecracker>  
 [10] Open Shortest Path First - Wikipedia[Website]. (2019, Jan 1), [https://en.wikipedia.org/wiki/Open\\_Shortest\\_Path\\_First](https://en.wikipedia.org/wiki/Open_Shortest_Path_First)

## 감사의 글(ACKNOWLEDGMENT)

본 연구결과는 중소벤처기업부 산학연협력 기술개발사업(산연전용과제)의 “빅데이터 기반 네트워크 품질관리시스템 개발”(과제번호: S2635421)에 의해 이루어진 연구결과물임을 밝혀 둡니다.

### 참고문헌

[1] Network Working Group, “A Two-Way Active Measurement Protocol (TWAMP), 2008.  
 [2] Network Working Group, “A One-way Active Measurement Protocol (OWAMP), 2006.

## 저자 소개

## ● 윤 대 일(Dae Il Yune)



·1990년 2월 : 중앙대학교 대학원 졸업 (공학석사, 전자통신공학 전공)  
 ·1996년 6월 : (주)한마이크로텔레콤 연구소장  
 ·1997년 7월 : 드림테크 대표이사  
 ·1999년 1월~현재: (주)유미테크 대

표 이사

〈관심분야〉 : 네트워크 품질관리, 블록체인 네트워크, ID 통신 등

## ● 황 택 진(Taek Jin Hwang)



·2003년 2월 : 한양대학교 전자공학과 학사  
 ·2005년 2월 : KAIST 전기및전자공학 석사  
 ·2005년 1월~현재 : 한국전자통신연구원 융합기술상용화실 선임연구원

〈관심분야〉 : RF Analog IC 회로설계, 초고주파 회로설계, 3D프린팅, SMT(Surface Mounted Technology)

## ● 이 재 기(Jae-Ki Lee)

정회원



·2005년 2월 : 공주대학교 대학원 졸업(공학박사, 정보통신 전공)  
 ·2004년 5월 이공계 대학원 국비장학생  
 ·1983년 3월~현재 : 한국전자통신연구원 융합기술상용화실장(융합센터 총괄)

〈관심분야〉 : S/W플랫폼, EMS, Smart City, 네트워크 품질측정 및 시험인증, 클라우드컴퓨팅(FN/FI/AN), 스마트 제조 등

## ● 서 현 용(Hyun Yong Seo)



·2008년 2월 : 충남대학교 컴퓨터공학과 졸업(공학사)  
 ·2008년 6월~현재 : (주)유미테크 책임연구원

〈관심분야〉 : 네트워크 품질 관리, IoT 플랫폼, 원격 단말 관리 등

## ● 이 진 우(Jin Woo Lee)



·2018년 8월 : 군산대학교 컴퓨터공학과 졸업(공학사)  
 ·2018년 1월~현재 : (주)유미테크 연구원

〈관심분야〉 : 네트워크 품질 관리, IoT 플랫폼