JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 22, No. 3: 548~556, Mar. 2018

네트워크 토폴로지 자동 구성 및 원격 장애진단 시스템

심규철 · 황경호*

Network topology automatic configuration and remote fault diagnosis system

Kyou-Chul Shim · Gyung-Ho Hwang*

*Department of Computer Engineering, Hanbat National University, Daejeon 34158, Korea

요 약

NMS(Network Management System)는 네트워크 관리를 목적으로 소규모 또는 대규모 네트워크를 운영하는 곳에서 필수적으로 사용되는 시스템이다. 기존 NMS에서는 네트워크 규모가 커지고 구성정보가 복잡해짐에 따라 네트워크 현황파악이 점점 어려워지고 있으며 네트워크 장비의 장애진단에 시간이 많이 소요된다. 본 논문에서는 NMS의 문제점을 보완하기 위해 JavaScript와 Python, HTML5 기반의 TWaver를 이용하여 자동으로 웹 기반 네트워크 토폴로지를 구현한다. 토폴로지 세부 구현 내용으로는 NMS시스템에 등록된 장비 정보를 기반으로 장비의 연결정보를 자동으로 수집하여 수집정보를 데이터화하고, 웹 기반의 네트워크 토폴로지를 구현하며 구현된 토폴로지에서 원격 장애진단을 수행할 수 있는 기능을 포함한다. 네트워크 토폴로지에서 구성관리, 장애관리, 성능관리 기능을 종합적으로 추가하여 사용자가 네트워크 관리를 함에 있어 체계화된 데이터 관리를 통해 NMS시스템의 품질 향상을 기대할 수 있다.

ABSTRACT

NMS (Network Management System) is a system that is used for a small or large networks management. As the size of network becomes larger and the configuration information become complicated, it becomes more difficult to grasp the network status and it takes much time to diagnose the failure of the network equipment. In this paper, to alleviate the problems of NMS we implement web-based network topology automatically using JavaScript, Python, HTML5 based TWaver. The detailed implementation of the system include the automatic collection of the connection information based on the equipment information registered in the NMS system, the implementation of the web-based network topology and the remote fault diagnosis. In the network topology, we can expect to improve the quality of the NMS system through structured data management by adding the configuration management, fault management and performance management functions in a comprehensive manner.

키워드 : SNMP, MIB, 네트워크 토폴로지, 자동 구성, TWaver, 장애진단

Key word : SNMP, MIB, Network topology, Automatic configuration, TWaver, Fault diagnosis

Received 2 February 2018, **Revised** 22 February 2018, **Accepted** 5 March 2018

*** Corresponding Author** Gyung-Ho Hwang(E-mail:gabriel@hanbat.ac.kr, Tel:+82-42-821-1751)
Department of Computer Engineering, Hanbat National University, Daejeon 34158, Korea

Open Access http://doi.org/10.6109/jkiice.2018.22.3.548 pISSN:2234-4772

Ⅰ. 서 론

최근 인터넷의 확산과 빠른 성장으로 인해 사용자들은 시간과 장소에 관계없이 노트북 또는 모바일기기를 통해 인터넷 등 네트워크에 연결한다. 이를 위해 네트워크의 구성이 필수적으로 요구되며 이후 장비의 장애나 오류로 인한 네트워크 끊김 현상을 방지하기 위해서는 각 장비들의 지속적인 관리가 필요하다. 또한 보안상의 이유로 공공기관 및 군사 관련 기관의 경우 외부 인터넷과는 차단된 별도의 내부 네트워크를 위한 구성이 이루어져야 한다. 이러한 소규모 또는 대규모 네트워크를 관리하기 위한 솔루션이 NMS이다[1].

네트워크 관리를 위해 NMS의 사용이 필수적인 요소가 되었으며, 각각 NMS의 솔루션 별로 제공 기능 또한 상이하다[2]. 하지만 대부분의 NMS는 공통적으로 동일한 문제점을 갖고 있다. 첫 번째 각 장비와 장비사이의 연결정보를 관리자가 직접 입력 및 수정하여 관리한다. 이로 인해 주기적으로 장비 연결 정보 및 기타 정보를 확인하여 버전관리를 해야 하는 문제점이 발생한다. 두 번째 NMS 관리 시스템에는 기본적으로 네트워크 토폴로지를 지원하지만 이러한 토폴로지는 기능이 제한되어 있으며, 일부의 장비 정보만 확인할 수 있도록 되어 있어 토폴로지에서 장비 상태 점검 등 확인이 어려운 불편함이 있다. 세 번째 물리적인 네트워크 장비의 Ping테스트 및 CRC 체크등 기본적인 점검에 대해 직접 Telnet 등으로 접속하고 확인하여 장비의 상태 점검에 많은 시간이 소요된다.

본 논문에서는 NMS 관리 시스템에서 제공하는 네트워크 토폴로지를 재구성하여 DB에 저장된 장비를 원격으로 접속하여 자동으로 각 인터페이스별로 연결정보를 수집하여 연결정보를 최신화하며, 네트워크 토폴로지에 표현된 장비의 기본적인 Ping테스트 및 CRC체크, 트래픽 확인을 원격으로 요청하여 결과를 바로 확인 할 수 있도록 구현한다[3].

본 논문의 구성은 1장의 서론에 이어 2장에서 네트워크 토폴로지 자동구성을 위한 관련 기술, 네트워크 연결정보 수집 관련 기술, JAVA 및 JavaScript 언어의 주요 기술 및 토폴로지 UI 구현을 위한 JavaScript 기반 twaver 주요 기술과 원격 장애 진단 및 연결정보 수집을 위한 서버프로그램에 대해 기술한다. 3장에서는 관련기술들을 활용하여 네트워크 토폴로지 자동 구성 및 원격 장애진단 기능을 구현하기 위한 시스템 설계 방안을 제시하며, 4장에서는 설계된 내용을 기반으로 네트워크 토폴로지 자동 구성 및 원격장애 진단 시스템을 구현한다. 5장에서는 네트워크 토폴로지 자동 구성 및 원격장애 진단 시스템의 성능 확인 및 분석을 수행한다. 마지막으로 6장에서는 결론을 맺는다.

Ⅱ. 관련 연구

본 장에서는 네트워크 관리를 위한 주요 기술과 네트워크 토폴로지 구성 및 원격장애 진단 시스템을 구현하기 위한 주요 기능과 이를 활용한 구현 방법을 기술한다.

2.1. Simple Network Management Protocol(SNMP)

네트워크 관리 및 전송을 위한 UDP/IP 상에서 동작하는 단순한 형태의 네트워크 프로토콜이며 망 관리를 위해 IETF에서 표준화 하였다[4].

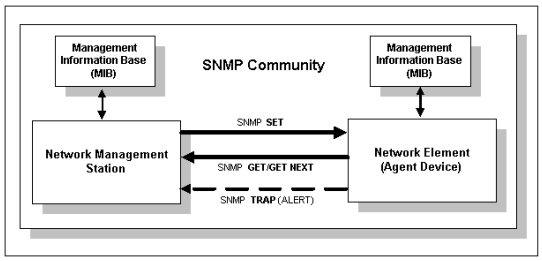

Fig. 1 SNMP messages exchange

그림 1은 SNMP의 메시지 교환 구성이다. SNMP station의 관리 application은 SNMP 관리자를 통해 3가지 형태의 SNMP 메시지를 생성하여 agent로 전송하며 agent는 GetResponse메시지로 SNMP station에 응답한다[3]. 이벤트를 알리기 위해 Trap메시지를 만들어 SNMP station에 알리게 된다[5].

표 1 은 SNMP 메시지 전송 기능을 나타낸다.

Table. 1 SNMP functions

division	function
GET	SNMP Station retrieves object value of this agent
SET	SNMP Station sets the value of this agent's object
TRAP	Agent notifies critical events to SNMP station

2.2. Management Information Base(MIB)

MIB는 네트워크에 연결된 리소스의 데이터 저장소다. MIB-Ⅰ, MIB-Ⅱ, 확장 MIB의 세 가지 종류가 있으며 의미 있는 연결고리를 갖도록 그림 2와 같이 트리구조의 데이터베이스로 저장된다[6].

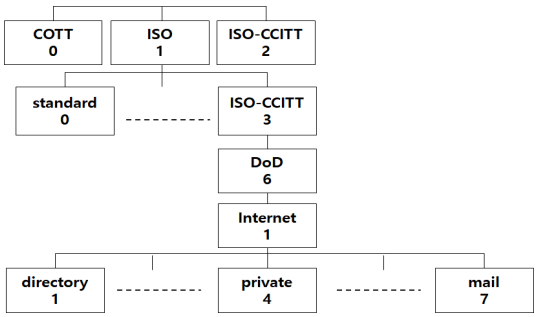

Fig. 2 SNMP MIB-Tree structure

망 관리용 프로토콜인 SNMP 등에 읽혀지며 망 관리 자원 정보를 구조화시킨 관리객체들의 집합이다. 망에서 관리되는 장치들의 정적 또는 동적 정보로서 NMS 시스템은 MIB의 내용을 기반으로 관리하게 된다[6].

2.3. JAVA & JavaScript

그림 3 은 JAVA Architecture를 보여주는 그림이다. JavaScript는 코어 자바스크립트 언어를 표준화하는 ECMA-262 스펙 외에도 ECMA-357이 있는데 자바스크립트 확장 기능인 E4X(ECMAScript for XML)을 표준화 하였다. 클라이언트 JavaScript는 웹 브라우저에 내장된 JavaScript 인터프리터에 의해 실행되며 문서 객체 모델 DOM(Document Object Model)과 결합되어 작동한다.

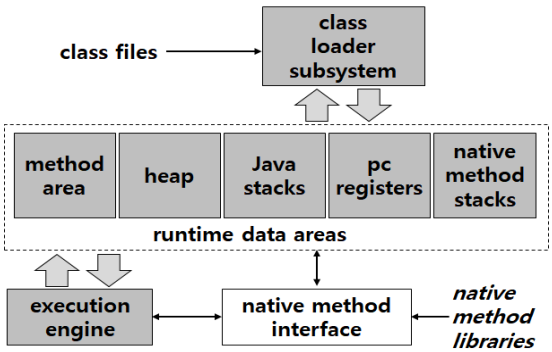

Fig. 3 JAVA architecture

JAVA는 국내 웹 어플리케이션 개발에서 가장 많이 쓰이는 언어로 MVC를 위한 Spring Framework와 ORM framework인 myBatis를 적용하여 쉽게 데이터를 조회 및 수정할 수 있다. 전자정부 표준 프레임워크에서도 적용하는 것처럼 검증된 언어 및 Framework이다.

2.4. TWaver

TWaver는 그림 4와 같이 HTML5기반으로 사용자 시각화를 위한 UI 전용 라이브러리로 광범위한 2D/3D 디스플레이 기능을 제공하고 다양한 기술 플랫폼을 지원한다. 통신, 전력 및 금융업계에서 널리 사용되고 2D 또는 3D로 복잡한 네트워크 토폴로지 데이터, 맵 및 장비 구조를 시연할 수 있다[7].

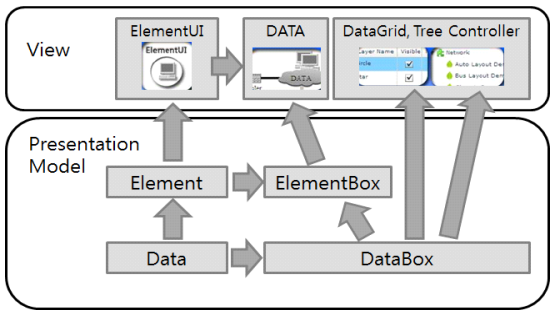

Fig. 4 TWaver HTML5 design model

2.4.1. TWaver UI 구현 및 이벤트 처리과정

본 논문에서는 TWaver의 twaver.Element 중에서 Topology Element로 토폴로지 UI를 구현한다. 그림 5와 같이 Topology Element는 Dummy, Node 및 Link를 포함하고 있다[7].

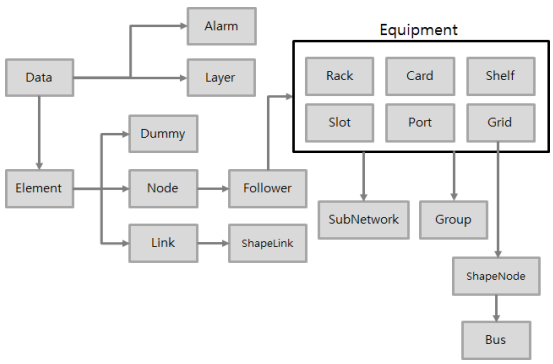

Fig. 5 TWaver topology element data structure

Ⅲ. 시스템 설계

3.1. 연결정보 수집 절차 설계

네트워크 토폴로지를 자동으로 구성하기 위해서 네트워크 장비의 연결정보는 필수적으로 관리해야 한다. 장비의 물리적인 연결정보를 Python을 통해 물리적인 장비에 Telnet접속하여 인터페이스의 Config정보를 파일로 저장하며, 저장된 Config내용에서 필요한 정보를 추출하여 자동으로 각 포트별 연결정보를 수집한다[8].

그림 6의 수집절차를 통해 수집된 정보를 취합하여 최종적으로 DB에 저장하여 네트워크 토폴로지를 자동으로 구성 할 수 있도록 설계한다.

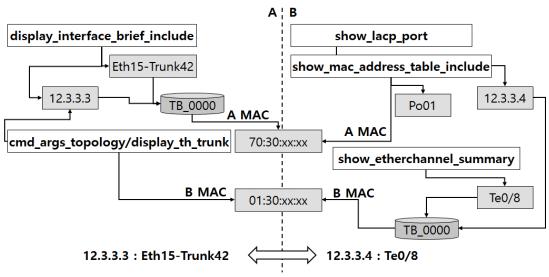

Fig. 6 Collection of connections information between equipment A and B

3.2. 연결정보 수집 데이터베이스 설계

추출된 연결정보 수집 데이터와 연결정보에 맞는 장비 리스트(Tree data)를 저장하기 위해 기존 NMS 시스템에서 관리되고 있는 장비 정보 테이블과 인터페이스 정보 테이블을 그대로 활용하여 연결 정보만 별도의 테이블에 저장하도록 연결 정보 테이블을 표 2와 표 3과 같이 추가로 생성한다.

Table. 2 Network topology tree menu table

COLUMN	TYPE	PK	NULL
REG_NO	VARCHAR(20)	1	N
MNG_NO	NUMBER(10)		
MENU_LV_NO	NUMBER(3)		
MENU_NODE_NM	VARCHAR(200)		
DEV_NAME	VARCHAR(100)		
UP_MNG_NO	NUMBER(10)		
NW_TP_CD	VARCHAR(10)		

표 2는 네트워크 토폴로지를 구성하기 위한 장비 메뉴 리스트를 관리하는 테이블 구조로 연결정보 수집으로 추출된 장비들을 연결 계위에 맞게 트리구조로 데이터를 관리하기 위해 장비관리 번호(MNG_NO)를 기반으로 계층형 구조로 데이터를 저장 한다.

표 3은 장비의 실제 연결정보 데이터를 관리하기 위한 테이블 구조로 장비관리 번호를 기준으로 해당 장비에 연결된 각각 인터페이스 정보까지 관리하여 토폴로지를 구현하기 위한 정보를 모두 관리한다.

Table. 3 Connection information table

COLUMN	TYPE	PK	NULL
REG_NO	VARCHAR(20)	1	N
MNG_NO	NUMBER(10)		
NE_LV_CD	NUMBER(5)		
IF_NM_CD	VARCHAR(10)		
RMT_MNG_NO	NUMBER(10)		
RMT_NE_LV_CD	NUMBER(5)		
RMT_IF_NM_CD	VARCHAR(10)		
NW_TP_CD	VARCHAR(10)		

3.3. 장비계위별 토폴로지 UI설계

네트워크 토폴로지를 구현하기 위해 토폴로지의 UI와 장비를 선택 할 수 있는 Tree메뉴 UI를 설계 한다.

3.3.1. 장비선택 Tree 메뉴 설계

토폴로지를 UI로 표현하기 위해서 기준 장비를 선택하여 토폴로지를 표현할 수 있도록 장비의 Navigation이 필요하다. 이를 위해 그림 7과 같이 장비 전체의 Tree 형태의 메뉴 데이터를 생성하여 UI로 표시한다.

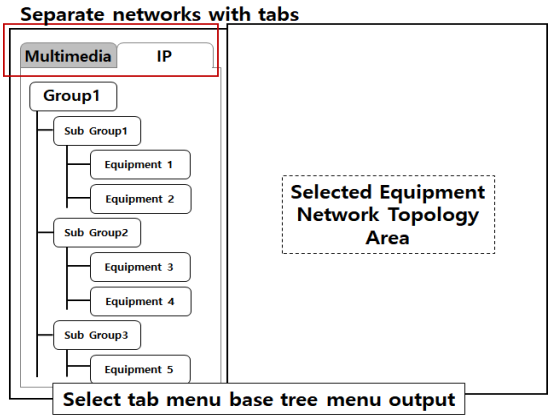

Fig. 7 UI design on equipment selection tree menu

3.3.2. 네트워크 토폴로지 UI 설계

네트워크의 특성에 따라 IP망(일반 네트워크)과 멀티미디어망(IPTV, Voice, IoT등) 두 가지 항목으로 분류한다. 즉, 인터넷망과 LTE망을 분류하여 각각의 네트워크 연결정보를 한눈에 확인할 수 있도록 하며 기준 장비에서 상위 계위의 연결정보는 모두 나타내며 하위의 계위는 1 depth까지만 표시하여 토폴로지 UI의 복잡함을 최소화하여 그림 8과 같이 토폴로지 UI를 설계 한다.

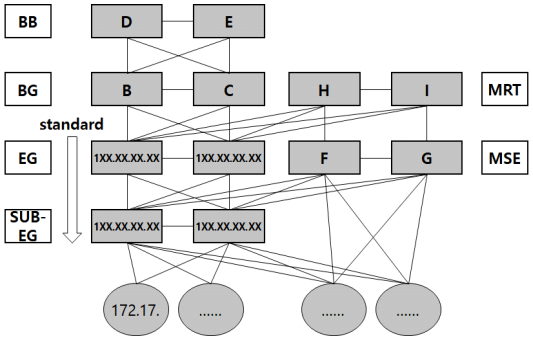

Fig. 8 UI design on network topology

3.4. 원격 장애진단 설계

웹 브라우저에서 장애 진단 요청 시 message queue로 요청 정보를 python으로 구현된 서버프로그램에 보낸다. 서버에서 대상 장비의 장애 진단을 실시하고 진단 결과를 NMS 시스템 데이터베이스에 저장한다.

장애 진단을 시작하면 장애진단의 결과가 데이터베이스에 저장될 때까지 polling한 후 데이터가 저장될 때 사용자 화면에 장애 진단 결과를 표시하여 장애진단을 종료 할 수 있도록 설계한다.

3.4.1. 데이터 송수신 규격

장애 진단의 첫 시작을 알리는 시점으로 웹 브라우저에서 특정 장비의 장애진단 요청 정보를 표 4 의 규격에 맞도록 하여 message queue로 요청 정보를 보낸다.

표 4의 parameter에서 reqno는 요청 정보의 고유 번호로 장애진단 요청 시간과 일련번호로 이루어진다. mng_no는 NMS 시스템에서 관리하는 장비 관리 번호로 장애진단 요청된 장비의 고유 번호이다. 해당 고유번호를 통해 장애 진단 주 처리 프로그램이 장비의 상세 정보 등을 획득하여 장애진단을 실시한다. testgbn항목은 장애진단의 종류에 대한 정보이다. 기본적인 장애진단 항목인 Ping Test, CRC Check, 트래픽 확인 중 어떤 장애진단을 실행할지에 대한 정보이며 각각 장애진단을 따로 할 수 있으며 3가지의 장애진단을 모두 함께 실행 할 수 있도록 설계 한다.

Table. 4 Remote fault diagnosis request data

parameter	parameter description
reqno	request number
mng_no	equipment control number
testgbn	fault diagnosis type

3.4.2. 원격 장애진단 주 처리 프로그램

장애진단 대상 장비의 모든 인터페이스에 Ping Test, CRC 체크, 트래픽 정보를 일괄적으로 실시하는 서버프로그램을 그림 9와 같이 IP망과 멀티미디어 망의 네트워크의 구분에 따라 프로그램을 분리하여 실행되도록 설계 한다.

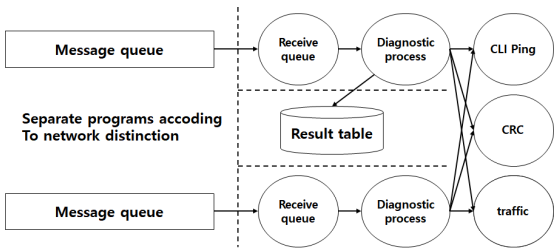

Fig. 9 Remote fault diagnosis main processing program flow chart

Ⅳ. 시스템 구현

4.1. 장비 연결 정보 수집 구현

장비 연결정보 수집을 위해 서버 프로그램은 두 부분으로 나누어 구현하였다. 모든 장비에 프로그램이 telnet 접속을 하여 장비의 설정 내용을 확인하는 show lacp port와 같은 command를 실행하여 결과를 파일로 생성하여 저장하는 프로그램과 설정 정보를 저장한 파일을 로드하여 각각의 설정파일의 내용을 파싱하여 연결 정보 수집에 필요한 최종 정보를 추출하는 프로그램이다. 모든 서버 프로그램은 python으로 구현 하였으며, 설정 파일의 저장 위치는 정해진 폴더 안에 각각 command와 동일한 폴더를 생성하여 해당 장비의 ip를 파일명으로

저장하여 2차 프로그램인 추출 부분에서 장비의 정보 및 command종류를 쉽게 구분 할 수 있도록 하였다.

연결 정보를 모두 추출하면 장비의 인터페이스에 연결된 대상 장비의 mac address를 확인 할 수 있다. 확인된 mac address로 대상 장비의 정보를 생성하고 대상 장비의 인터페이스에 연결된 장비까지 확인하여 양쪽 인터페이스의 정보를 모두 확인하여 최종적으로 연결된 정보를 추출하여 연결정보 테이블에 최종 저장한다.

이러한 장비의 연결정보 수집 과정을 장비의 구성이 변경되는 것에 대비하여 1일 1회 실행하여 연결정보를 항상 최신으로 유지하도록 구현하였다.

4.2. 네트워크 토폴로지 자동구성 구현

수집된 연결정보를 활용하여 TWaver와 JavaScript 및 JAVA를 사용하여 그림 10과 같이 네트워크 토폴로지 UI를 구현하였다. 특정 장비 1대를 선택하여 해당 장비의 네트워크 토폴로지를 구성하는 화면으로 연결정보를 검색하여 연결된 모든 장비를 보여준다.

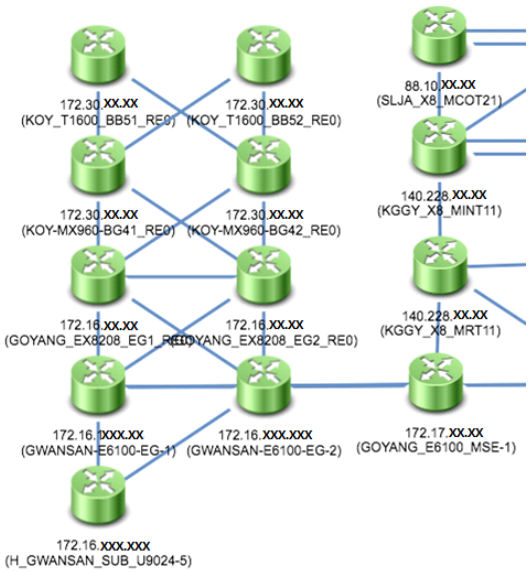

Fig. 10 IP network topology implementation

그림 11 은 네트워크 토폴로지 UI에서 특정 장비를 선택하면 장비에 연결된 모든 인터페이스 정보를 사용자에게 좀 더 상세하게 보여주는 화면으로 하단에 있는 선택장비의 인터페이스에 연결된 장비 정보를 상세하게 보여준다.

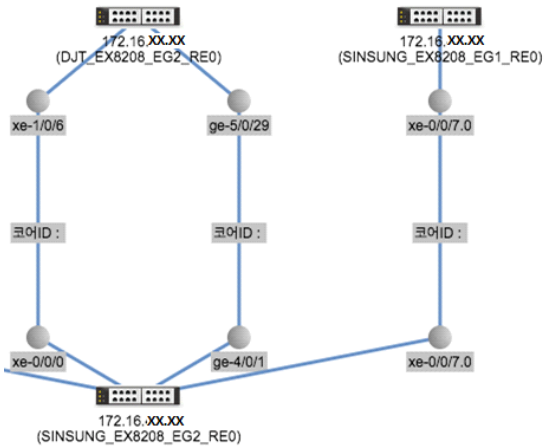

Fig. 11 Interface connection Information

4.3. 원격 장애진단 시스템 구현

네트워크 토폴로지에서 특정 장비의 장애 진단 메뉴를 선택하여 원격장애진단을 실시 할 수 있도록 구현하였다.

원격 장애 진단이 종료되면 장애 진단 주 처리 프로그램에서 진단결과를 데이터베이스에 저장하며, 웹 브라우저는 저장된 진단결과를 사용하여 진단결과표를 작성하여 표시한다. 진단결과표 내용에는 장비의 모든 인터페이스의 트래픽의 IN, OUT정보와 연결된 Peer장비의 IP, 인터페이스 이름, Ping Loss, CRC, Peer 장비의 트래픽 정보를 표시 한다.

Ⅴ. 성능 확인

5.1. 네트워크 토폴로지 페이지 로딩 속도 측정

웹 어플리케이션의 성능 프로파일링은 웹 어플리케이션이 실행하는 주기 동안 사용하는 총 자원을 측정한 결과이다[9]. 로딩속도, 메모리 사용량 등 다양한 항목들에 대한 분석을 하여 그림 12와 같이 한 화면에서 볼 수 있다.

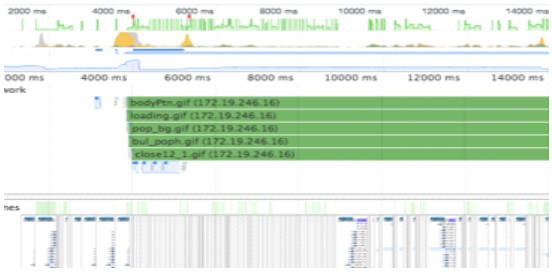

Fig. 12 Total descriptions on performance analysis

특정 네트워크 장비를 선택하여 자동으로 수집된 연결 정보를 조회하여 네트워크 토폴로지를 UI로 구현 하는 과정의 소요시간은 그림 13에서 확인 할 수 있다.

결과를 보면 약 2~3초 후 모두 화면에 표시되는 것을 확인 할 수 있었다. 모든 네트워크 장비의 네트워크 토폴로지를 화면에 구현하는 시간은 최대 3초 이내의 소요시간이 소요됨에 따라 네트워크 토폴로지 UI를 표현하는 로딩속도 및 rendering 속도가 빠르다는 것을 확인할 수 있었다.

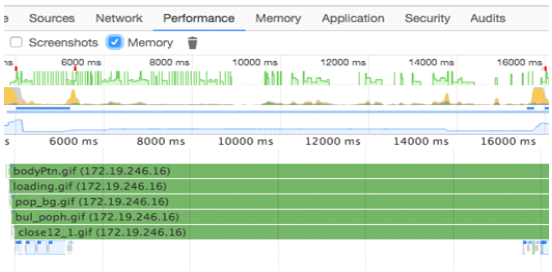

Fig. 13 Measuring network topology loading speed

그림 14의 성능 결과에서 네트워크 토폴로지의 페이지 데이터 로딩부터 네트워크 장비의 토폴로지 UI를 구성하는 과정까지 전체 약 3~5초의 시간이 걸렸으며 이외의 렌더링 시간과 UI페인팅 시간 등은 모두 1초미만의 시간이 소요된 것을 확인 할 수 있었다.

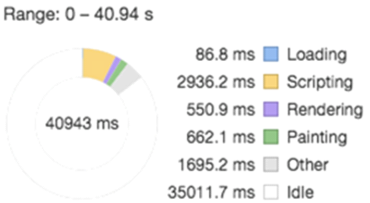

Fig. 14 Performance on total elapsed time for network topology creation

5.2. 메모리 사용 성능 확인

그림 15는 데이터 로딩 후 UI구현을 위해 사용되는 메모리 사용량을 나타내며 JS Heap 메모리 사용량이 증가하는 것을 확인 할 수 있다. 이는 UI를 구현하는 TWaver가 JavaScript 기반의 웹 어플리케이션이기 때문에 JS Heap메모리를 사용하여 UI를 구현하고 있다는 것을 알 수 있다.

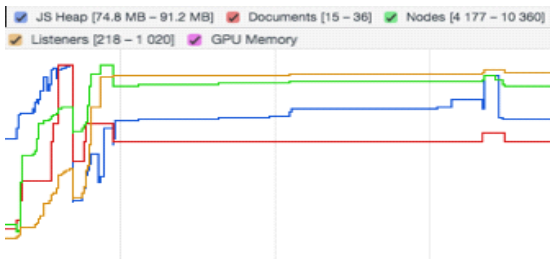

Fig. 15 Memory usage at network topology creation

5.3. Resource 성능 분석 및 상태 정보 확인

네트워크 토폴로지의 로딩 속도와 메모리 사용 현황을 제외한 UI를 구성하기 위해 필요한 모든 정보의 데이터 사용량 및 소요 시간을 분석한 결과 TWaver 관련 부분, 즉 JavaScript 영역에서 사용하는 용량과 메모리 사용량 등의 사용빈도가 가장 많으며 처리 시간 또한 가장 오래 걸린 점을 그림 16에서 확인 할 수 있다.

JavaScript는 접속 PC에서 실행되는 프로그램으로 네트워크 토폴로지의 UI 성능은 서버성능이 아닌 client PC의 성능에 따라 차이가 있을 수 있다[10].

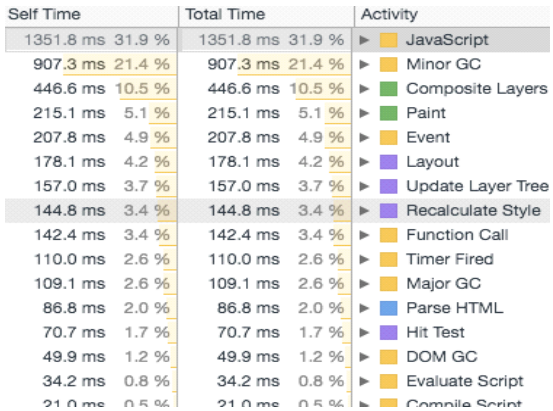

Fig. 16 Performance on data usage and elapsed time for network topology creation

Ⅵ. 결 론

본 논문에서 구현한 네트워크 토폴로지 자동 구성 및 원격 장애진단 시스템에서는 장비와 장비의 물리적인 연결정보를 python으로 구현된 프로그램으로 자동 수집하여 토폴로지를 자동 구성하고 웹으로 구현된 토폴로지 상에서 원격 장애진단을 실시하여 자동 점검을 수행한다. 이를 통해 네트워크 관리에서 연결정보 수집과 장애진단에 소요되는 시간을 단축할 수 있고 연결정보 수집 시 함께 수집하는 회선 정보 및 트래픽 정보를 통해 네트워크의 성능 정보를 한눈에 확인할 수 있다.

구현된 네트워크 토폴로지 자동구성 및 원격 장애진단 시스템을 네트워크 관리자에게 제공하고 테스트에 적용한 결과 장비의 물리적 연결정보의 누락 가능성이 사라졌으며 매일 1회 연결 정보를 자동으로 수집하여 연결정보 또한 최신으로 유지할 수 있었다. 또한, 원격 장애진단 서비스를 통해 Ping Test, CRC 체크, 트래픽 정보 확인 등의 기본적인 점검 항목에 대해 점검시간이 기존 관리자가 직접 TELNET접속을 통해 점검하는 시간보다 약 70% 이상 단축 되었다. 장애진단 서비스는 동시에 여러 장비에 대한 장애진단이 가능해 장애진단 작업에서 시간 단축이 가능하다. 그러나 연결정보 수집을 위해서는 장비의 인터페이스 정보 등을 항상 최신으로 유지해야 하며 Telnet 로그인 정보도 함께 관리해야 하는 문제점이 있다. 또한, 네트워크 장비의 각 모델별로 command가 다르기 때문에 각 모델별 수집 프로그램이 각각 존재해야 한다. 추후 연구에서 사용자 편의성 관점에서 시스템의 자동화 요소를 늘리는 방안을 고려할 것이다.

References

[1] N.S.Kim, H.K.Ryu, J.H.Choe and G.M.Choi, “NMS Trends and The Case of Application of NMS for Effective Network Management,” *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 941-944, Nov. 2003.

[2] K.Urunov, S.Y.Shin, S.H.Park and Y.K.Lim, "Analysis of the Network Management System with Constrained Underwater Devices : Architectural Mechanism of the U-NMS," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 500-501, Jun. 2017.

[3] S.J.Moon, "Server Management Prediction System based on Network Log and SNMP," *Journal of Digital Contents Society*, vol. 18, no. 4, pp. 747-751, Jul. 2017.

[4] K.J.Choi, J.S.Park and M.S.Kim, "A Efficient SNMP MIB Data Gathering Algorithm," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 1783-1786, Jul. 2008.

[5] W.S.Yang, J.H.Kim and J.O.Lee, "A Management for IMS Network Using SDN and SNMP," *Journal of the Korea Academia-Industrial cooperation Society*, vol. 18, no. 4, pp. 694-699, Apr. 2017.

[6] S.I.Byun, B.K.Kim and D.R.Shin, "A Study on the QoS Management Scheme using the MIB," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 1055-1058, Jul. 2001.

[7] TWaver Data Center Visualization, Dec. 2017 [Internet]. Available: http://www.servasoftware.com.

[8] H.C.Park and J.S.Song, "A Study on the Web-based IoT Packet Analysis System Using Python," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 616-617, Nov. 2017.

[9] Y.B.Park, S.Hong and M.Z.Kim, "Performance Bug Detection in Web Applications through Cross-browser Profiling," *Journal of KIISE : Computing Practices and Letters*, vol. 19, no. 11, pp. 559-571, Nov. 2013.

[10] H.J.Kim, J.H.Lee, G.W.Jo and J.J.Lee, "Measuring JavaScript Performance with a Real World Web Application," *Proceedings of the Korean Information Science Society*, vol. 38, no. 2A, pp. 131-134, Nov. 2011.

심규철(Kyou-Chul Shim)

2018년 : 한밭대학교 정보통신전문대학원 컴퓨터공학과 졸업(공학석사)
2010년~현재 (주)커뮤 SI사업본부 차장
※관심분야 : 네트워크 응용 프로그램

황경호(Gyung-Ho Hwang)

1998년 KAIST 전기및전자공학과 졸업(공학사)
2000년 KAIST 전자전산학과 졸업(공학석사)
2005년 KAIST 전자전산학과 졸업(공학박사)
2005년~2007년 삼성전자 무선사업부 책임연구원
2007년~현재 한밭대학교 컴퓨터공학과 부교수
※관심분야 : 이동통신 프로토콜, 무선 센서 네트워크, 모바일 응용 프로그램