엔터프라이즈 IP 네트워크 연결 정보 관리 시스템 설계 및 구현*

김은희[0], 최미정[1], 주홍택[2], 홍원기[1]

[0, 1] 포항공과대학교 컴퓨터공학과, [2]계명대학교 컴퓨터공학과

Design and Implementation of Enterprise IP Network Connectivity Information Management System

Eun-Hee Kim[0], Mi-Jung Choi[1], Hong-Taek Ju[2], James W. K. Hong[1]

[0, 1] Dept. of Computer Science and Engineering, POSTECH, [2]Dept. of Computer Engineering, Keimyung University

della@postech.ac.kr, mjchoi@postech.ac.kr, juht@kmu.ac.kr, jwkhong@postech.ac.kr

요 약

현재 대부분의 엔터프라이즈 IP 네트워크의 구성도의 작성 및 유지는 수작업에 의존하고 있어 시간이 지난 후에는 장비의 추가, 삭제가 제대로 반영되지 않아 실제 네트워크 구성과 달라지게 된다. 이는 전체 네트워크 장비와 네트워크 상태를 파악하는 것을 불가능하게 하여 결국은 관리자의 네트워크 관리를 어렵게 하는 원인이 된다. 효율적인 네트워크 관리를 위해서는 관리하는 네트워크 장비간의 물리적인 연결 정보를 추출하여 네트워크 구성도를 자동적으로 생성하는 시스템이 필요하다. 본 논문에서는 SNMP 에이전트를 탑재하고 있는 네트워크 장비의 MIB 정보를 통하여 연결 정보를 파악하고 이것을 바탕으로 네트워크 구성도를 생성하는 시스템을 제안하고자 한다. 본 논문에서는 네트워크 장비 검색 및 장비간 물리적 연결 정보를 알아내는 상세한 알고리즘을 제시하고, 제시한 알고리즘을 기반으로 네트워크 구성도를 생성하는 시스템을 설계, 구현한다. 개발한 시스템을 POSTECH 네트워크의 구성도 생성에 적용하여 알고리즘의 효율성을 검증하였다.

1. 서론

현재는 네트워크 중심의 컴퓨팅 시대로 정보의 공유와 교환이 네트워크를 중심으로 이루어지고 있다. 대부분의 기업, 학교, 공공기관 등은 갈수록 빠른 네트워크를 요구하는 시대에 발맞춰 수백대 또는 수천대의 스위치나 라우터를 연결하여 고속 LAN을 구축하고 다양한 네트워크 서비스를 제공하고 있다. 따라서 네트워크에 문제가 발생하면 서비스를 제공하지 못하는 결과를 야기하므로, 네트워크 문제를 빠르게 파악하고 신속하게 처리하는 것이 필요하다.

네트워크에 문제가 발생했을 때, 네트워크 상태와 구성을 실시간으로 파악하고 있다면 문제가 생긴 지점을 정확하고 신속하게 파악하여 빠른 대처를 할 수 있다. 이런 빠른 대처를 위해서는 관리하는 네트워크의 물리적인 네트워크 연결 상태가 반영된 네트워크 구성도가 필수적이다[1]. 그러나, 대부분의 엔터프라이즈 네트워크 구성도의 작성 및 유지는 수작업에 의존하고 있어 변경된 네트워크의 구성이 네트워크 구성도에 실시간으로 반영되기 어렵다. 실제 네트워크 구성과 수작업으로 작성한 네트워크 구성도의 불일치는 네트워크 관리를 어렵게 하는 주요 원인이 된다[2].

따라서, 본 논문에서는 네트워크의 연결 정보 관리를 효율적으로 수행할 수 있는 시스템 개발을 목표로 한다. 이 시스템은 관리하고자 하는 네트워크 장비를 자동으로 검색하여 장비들간의 포트별 연결 정보를 알아내어 네트워크 구성도를 생성하는 기능을 제공해야 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구를 소개하고, 3장에서는 시스템 개발에 대한 요구사항들을 정의하고, 4장에서는 연결 정보를 찾아내고 네트워크 구성도를 생성하는 알고리즘과 시스템 전체 구조에 대해서 설명한다. 5장에서는 4장에서의 설계를 바탕으로 시스템을 구현하고, 구현한 시스템을 POSTECH 네트워크에

* 본 연구는 2007 년도 두뇌한국 21 사업과 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음 (IITA-2006-(C1090-0603-0045))

적용하여 네트워크 구성도를 생성한 결과를 보여준다. 마지막으로 6장에서는 연구 내용을 정리하고 향후 연구 방향을 제시한다.

2. 관련연구

이 장에서는 네트워크 장비의 연결 정보를 파악하기 위한 MIB (Management Information Base)[3]에 대해서 알아보고, 개발하고자 하는 시스템과 비슷한 기능을 제공하는 기존 시스템에 대해서 알아본다.

2.1. MIB

현재 대부분의 네트워크 장비는 SNMP (Simple Network Management Protocol) 에이전트를 탑재하여 관리 기능을 제공한다. 관리 기능은 관리 정보인 MIB을 정의하여 조작함으로써 제공된다. 이 장에서는 본 논문에서 사용하는 관리 정보 MIB을 알아본다.

2.1.1. Bridge MIB

장비의 각 포트별 연결 정보 매칭을 파악하기 위해 Bridge MIB[3]을 사용한다. 전달되는 패킷 헤더의 목적지 MAC address를 보고 다음 노드로 전달해주는 것이 bridge 기능이며, Layer 2 장비들의 네트워크 동작을 나타내 주는 것이 bridge MIB의 역할이다. Bridge MIB은 다섯 개의 그룹으로 구성되어 있으며 본 논문에서 사용되는 두개의 그룹에 대한 설명은 아래와 같다.

- **dot1dBase Group**

포트 수, 포트 타입, 포트 주소와 같은 Bridge의 기본 정보를 제공한다. 이 그룹 중에서 Bridge와 연결된 모든 포트에 관한 일반적인 정보를 가진 dot1dBasePortTable을 이용한다.

- **dot1dTp Group**

Transparent bridge 기능에 관한 관리 객체를 포함한 그룹이다. Transparent bridge 기능은 학습을 통하여 라우팅 테이블을 만들고 이 테이블을 기반으로 프레임을 전달한다. 각 포트에 연결되어 있는 다른 장비의 MAC address를 각 포트로 들어오는 MAC 프레임을 보고 알아내어 학습하고, 그 결과는 전달 테이블로 관리 한다.

2.1.2. MIB-II의 Address Translation 그룹

멀티레이어 스위치인 경우 설정에 따라 Layer 2나 Layer 3를 선택하여 서비스를 제공하게 된다. Layer 2인 경우는 Bridge MIB을 이용하여 포트별 연결 정보를 알아내지만, Layer 3으로 동작하는 경우에는 ifTable의 MAC address가 대부분 동일한 기본 MAC address로 설정되어 MAC address로 포트를 구분할 수 없다. 이 때는 주소 변환 테이블을 이용하여 패킷을 전달하기 때문에 아래의 MIB-II의 at 그룹 중 atTable을 이용한다.

- **at (Address-Translation) Group**

IP address와 MAC address간의 매핑 테이블로 구성되며 장비의 인터페이스 별로 IP 주소와 MAC 주소를 검색할 수 있다.

2.2. 관련 시스템

본 논문에서 구현한 시스템과 비슷한 기능을 제공하는 제품으로 AdventNet의 'OpUtils Switch Port Mapper'[4], CISCO Network Assistant[5], IBM Netcool/Precision[6]이 있다. AdventNet의 제품은 각 스위치의 포트별 상태를 파악하는 것에는 유용하게 쓰이지만 전체 네트워크의 스위치간 연결 정보는 파악할 수 없다. CISCO 제품은 CISCO 장비에 대해서만 네트워크 구성도를 작성한다. IBM 제품은 Layer 2와 3 장비에 대해서 네트워크 구성도를 작성하지만 이들이 연결 정보를 취득하는 방법에 대해서 공개하지 않고 있어 그 방법을 알 수 없다. [1] 논문에서도 표준 SNMP 정보만을 이용하여 2계층의 연결 정보를 찾아 네트워크 토폴로지를 생성하였다. 그러나 하나의 subnet에서만 동작하는 단점이 있다.

따라서, 본 논문에서는 네트워크 장비 검색 및 장비간 연결 정보를 알아내는 방법을 SNMP MIB을 이용한 상세한 알고리즘을 제시하고 제시한 알고리즘을 따라 시스템을 구현한 후, POSTECH 네트워크에 실제 적용하여 알고리즘을 검증하는 것을 목표로 한다.

3. 시스템 요구사항

본 논문에서 제안하는 시스템은 네트워크 장비의 연결 정보를 알아내어 네트워크 구성도를 생성한다. 네트워크 연결 정보 관리를 위한 요구사항은 다음과 같다.

- **Discovery 기능**

관리 대상인 장비를 자동으로 검색하여 IP address별 장비 목록을 작성해야 한다. 또한, 하나의 장비에 여러 개의 IP address가 할당 되어 있을 수 있으므로 하나의 대표 IP로 그룹핑해서 장비 목록이 작성되어야 한다.

- **Map을 통한 연결 정보 관리 기능**

각 포트별 장비간의 연결 정보를 확인 할 수 있어야하며, 장비의 포트별 사용 여부(enable/ disable)를 색으로 구분하여 관리자가 효율적으로 장비를 관리 할 수 있도록 한다.

- **장비 구성 관리 기능**

장비의 추가, 삭제 및 장비의 구성 정보를 변경하는 기능이 있어야 하며, 포트의 연결 정보를 변경 할 수 있는 기능이 제공되어야 한다.

- **데이터베이스 관리 및 백업기능**

주기적으로 장비의 존재여부 및 구성 정보에 대한 값을 가져와 관리 정보 데이터베이스에 최신정보를 유지하여 정확한 정보를 제공해야 한다. 관리자에 의해 변경된 구성 정보가 데이터베이스에 저장됨과 동시에 실제 장비의 구성 정보에도 즉시 적용되어야 한다.

- **쉬운 사용자 인터페이스 기능**

네트워크 구성 요소들간의 연결 정보 및 구성 정보를 언제 어디서나 쉽게 접근하여 관리하도록 웹 기반의 사용자 인터페이스를 제공해야 한다. 또한, 시스템이 전반적으로 사용하기 쉽고 단순하게 구성되어야 한다.

4. 시스템 설계

이 장에서는 3장의 요구 사항을 바탕으로 네트워크 연결 구성 정보 관리 시스템을 설계한다. 먼저 연결 구성도를 생성하기 위한 포트별 IP 매칭 알고리즘을 설명하고, 이 알고리즘에 근거한 전체 시스템 구조를 제시한다.

4.1. 포트별 IP 매칭 알고리즘

네트워크의 연결 구성도를 생성하기 위한 포트별 IP 매칭 알고리즘의 전체 순서도는 그림 1과 같다.

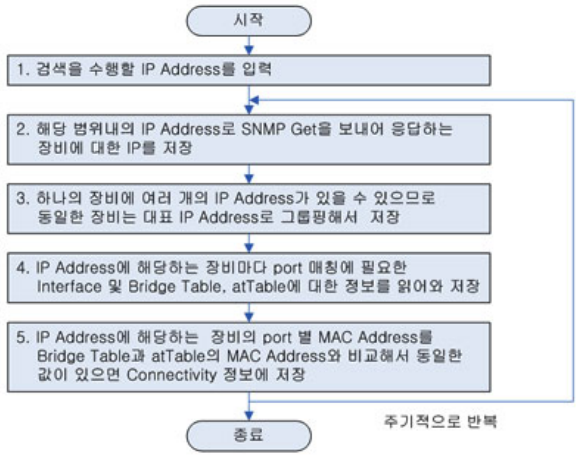

그림 1. 포트별 IP 매칭 알고리즘

단계 1: 관리 장비를 자동으로 검색하기 위해 엔터프라이즈 네트워크의 관리 대역 IP address 범위를 외부로부터 입력 받는다.

단계 2: 입력 받은 모든 IP address로 SNMP 에이전트의 탑재 여부 및 관리 대상 장비에 해당되는지 알아내기 위하여 각 장비의 sysServices 정보를 요청하는 SNMP 메시지를 보내어 응답하는 장비에 대해서만 관리 장비 목록에 추가한다. sysServices는 7비트 코드로 해석되는 값으로 각 비트는 OSI의 각 레이어를 의미하며, 이 값을 기반으로 관리 대상 장비로 분류하게 된다. 응답 받은 sysServices가 '2', '3'의 값을 가지면 'L2 스위치'며, '6'이면 'L3 스위치', '78'이면 '멀티레이어 스위치'로 판단하며 이에 해당되는 IP address 및 sysServices정보를 저장하여 결과 목록을 작성한다.

단계 3: 하나의 장비에 여러 개의 IP address가 할당될 수 있으므로 동일한 장비는 대표 IP address로 그룹핑해서 관리 대상 목록을 작성한다. 장비의 시스템 정보 중 sysDescr, sysObjectID, sysName, sysLocation이 모두 같으면 동일한 장비로 판단한다.

단계 4: 각 장비 포트별 연결 정보를 알아내기 위해 포트 매칭에 필요한 장비의 Interface 및 Bridge MIB, atTable의 정보를 가져와 저장한다.

단계 5: 장비 목록에 저장된 IP address 순서대로 장비의 포트별 MAC address를 읽어와 다른 장비의 Bridge MIB과 atTable의 정보를 비교해서 동일한 address가 검출되면 물리적으로 연결되어 있는 것으로 판단하고 연결 정보를 저장한다.

이 알고리즘은 관리자의 필요해 의해 즉시 수행되거나, 관리자가 설정한 주기에 의해 반복 수행됨으로 네트워크 장비의 연결 정보를 최신 정보로 유지한다.

4.2. 네트워크 장비별 정보 수집 알고리즘

포트별 연결 정보 매칭을 위해 장비별로 필요한 정보를 수집한다. 장비 리스트에서 IP address를 순서대로 추출해서 추출한 IP address에 해당되는 장비에서 필요한 정보를 가져와 저장한다.

먼저, 장비의 sysServices가 '78'인 경우, Layer 3에서 동작하는 장비로 설정되어 동작 할 수 있으므로 추가적으로 장비의 MIB-II 정보 중 atTable을 저장한다. 다음으로, 장비의 MIB-II 정보 중 ifTable의 인터페이스 값이 현재 동작중인 포트만 의미있으므로 ifAdminStatus와 ifOperStatus의 값이 모두 up(1)인 상태의 장비에 대해서만 ifDescr이 VLAN[6]으로 시작되는 String을 읽어와 앞의 VLAN을 제외한 값만 저장한다.

다음으로, Bridge-MIB에 속하는 2개의 테이블인 dot1dBasePortTable과 dot1dTpFdbTable 값을 읽어와서 저장한다. Bridge MIB의 테이블에서 특정 index의 값을 읽기 위해서 Community String Indexing[7]이라는 문법을 사용한다. VLAN 별로 장비의 포트에 관한 정보를 읽어오는 이유는 VLAN 별로 Bridge 기능이 제공, 관리되기 때문이다.

dot1dTpFdbAddress	Dot1dTpFdbPort	dot1dTpFdbStatus
00.00.0C.07.AC.01	12	learned(3)
00.0B.60.AC.3A.8A	11	learned(3)
00.0B.60.AC.3E.4A	12	learned(3)

표 1. dot1dTpFdbTable을 'public@1'로 읽은 결과

표 1은 community string을 'public@1'으로 하여 얻은 dot1TpFdTable값이고 이 결과로 11, 12번 포트가 'VLAN1'으로 그룹되어 있는 것을 알 수 있다. 새로운 패킷 헤더의 목적지 address가 표 1의 '00.00.0C.07.AC.01'와 같다면 12번 포트로 전달된다. 12번 포트로 MAC address가 '00.00.0C.07.AC.01'와 '00.0B.60.AC.3E.4A'을 가진 패킷이 전달되어 학습되었음을 알 수 있다.

4.3. 네트워크 장비의 포트별 IP 매핑 알고리즘

그림 2는 4.2장에서 추출한 정보를 바탕으로 각 장비의 포트별 연결 정보를 생성하는 알고리즘이다. 먼저, Device LIST의 각 장비 IP address의 ifTable에서 enable되어 있는 포트에 대해서만 MAC address를

읽어온다. 다음으로 읽어온 MAC address와 Bridge Table 또는 atTable의 MAC address에 동일한 값이 있는지 비교한다. 동일한 값이 있으면 ifTable에서 읽어온 IP address(Source IP address)와 포트 번호에 매칭되는 IP address(Destination IP address)와 포트 번호를 데이터베이스의 Connectivity 테이블에 저장한다. 모든 장비에 대해서 이 알고리즘을 수행하면, 모든 장비의 연결 정보가 검출되어 저장된다.

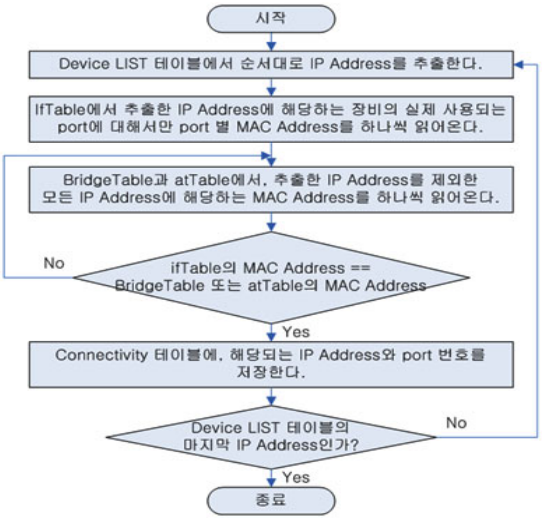

그림 2. 장비 포트별 연결 정보 생성 순서도

저장된 연결 정보를 순서대로 읽어와 맵을 생성할 때, 먼저 Source IP address 장비의 포트 번호와 Destination IP address 장비의 포트 번호를 읽어오고, 반대로 읽어온 Destination IP address 장비의 포트 번호에 Source IP address 장비의 포트 번호가 연결되어 있는지 검색한다. 두 값이 동일할 때에만 물리적으로 서로 연결되었다고 한다. 이와 같이 두 장비간의 포트별 연결 정보는 각 장비가 상대 장비에 대해 포트별 연결 정보가 서로 일치한 것에 대해서만 물리적인 맵을 생성하게 된다.

4.4. 시스템 구조

그림 3은 웹 기반의 포트별 연결 정보 관리 시스템의 구조이다. 각 주요 모듈의 기능은 아래와 같다. 이 외에도 연결 정보 저장을 위해 데이터베이스가 사용되며 주요한 모듈 중 하나이다.

- **장비 스캐너(Device Scanner)**

외부로부터 입력 받은 IP address 범위 내에 있는 SNMP 에이전트를 탑재한 장비를 검색하는 모듈로 수행한 후에 실제 관리 대상이 되는 장비의 IP 목록을 얻게 된다.

- **장비 그룹퍼(Device Grouper)**

라우팅 기능을 제공하는 장비인 경우 하나의 장비에 여러 개의 IP address가 할당 되어 있을 수 있으므로 중복되는 장비를 하나의 대표 IP로 그룹핑한다. 이 모듈을 수행한 후에 하나의 장비에 하나의 대표 IP가 매핑된 실제 관리 대상 장비 목록을 얻게 된다.

- **장비 정보 수집기(Device Information Collector)**

장비 그룹퍼에 의해 작성된 IP address 목록을 순서대로 입력 받아 포트별 IP 매칭을 위해 필요한 MIB 정보인 MIB-II의 ifTable과 Bridge-MIB의 dot1dBasePortTable, dot1dTpFdbTable을 각각의 장비에 요청하여 저장한다. 또한, Layer 3로 동작하는 스위치에 대해서 MIB-II의 atTable을 가져와 저장한다.

- **연결 정보 작성자(Connectivity Mapper)**

장비별 연결 정보를 생성하기 위한 모듈로 장비간 실제 물리적인 연결 정보를 파악하는 기능을 수행한다. 이 모듈의 실행 후, 최종적으로 장비간 포트별 연결 정보 목록을 얻게 된다.

- **구성 관리자(Configuration Manager)**

실제 관리 장비 목록에서 장비를 선택하게 되면 선택된 장비에 대해 IP address를 입력 받아, 선택된 장비의 구체적인 정보와 포트의 사용여부 상태 등을 확인할 수 있으며 커뮤니티 변경 및 포트의 사용여부의 변경, 대표 IP address의 변경을 하는 기능을 제공한다.

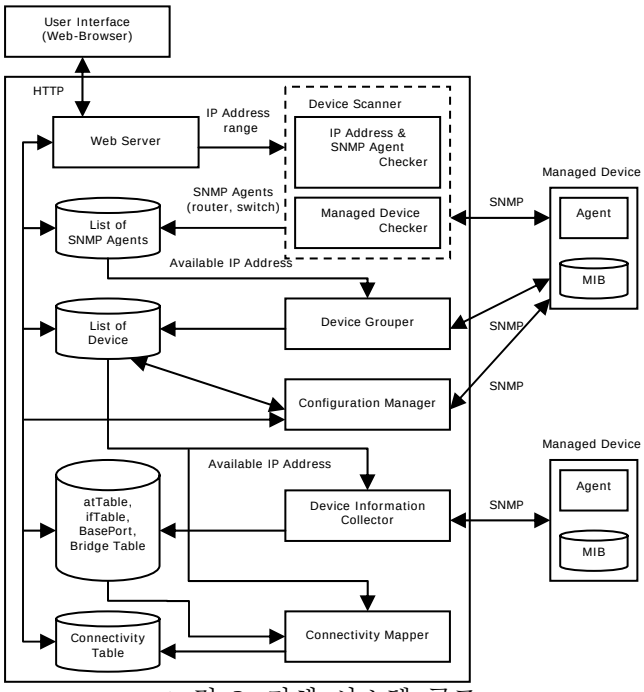

그림 3. 전체 시스템 구조

5. 구현

이 장에서는 4장의 설계 구조를 바탕으로 구현한 시스템을 POSTECH 네트워크에 적용하여 네트워크 구성도를 생성한 것에 대하여 살펴본다.

개발환경으로 하드웨어는 CPU 2.4GHz, 1GB 메모리를 사용했으며, 소프트웨어 환경은 윈도우 2000 운영체제하에, 톰캣 서버와 오라클 데이터베이스를 이용하여 자바 언어로 구현했으며, AdventNet에서 제공하는 SNMP API[9]를 사용했다.

POSTECH 네트워크 대역인 '141.223.1.1'부터 '141.223.255.255'에 대해서 스캐닝을 하여 네트워크

장비만 추출하여 동일한 장비를 하나의 IP address로 묶기 위한 기능을 수행한다. 그림 4의 화면은 장비 스캐닝 이후, 하나의 대표 IP 로 그룹핑 했을 때의 결과이다.

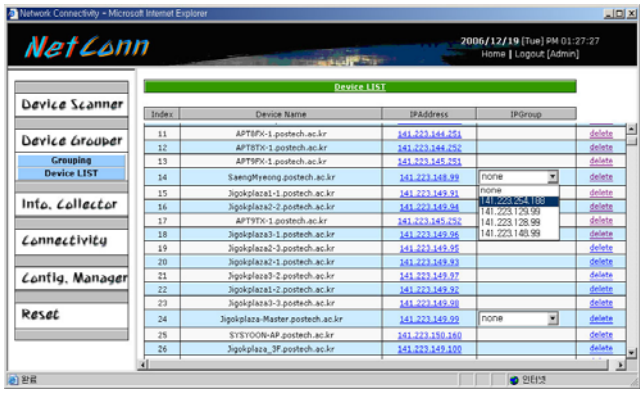

그림 4. 장비 그룹핑 수행 후 결과화면

여러 개의 IP address가 할당된 장비에 대해서만 IPGROUP의 펼침 목록 메뉴가 나타나며 이 목록에서 장비에 할당된 IP address를 확인할 수 있다. 관리자가 원하는 IP address로 설정해서 장비 관리를 수행할 수 있다. 이 리스트에서 장비 삭제도 가능하다.

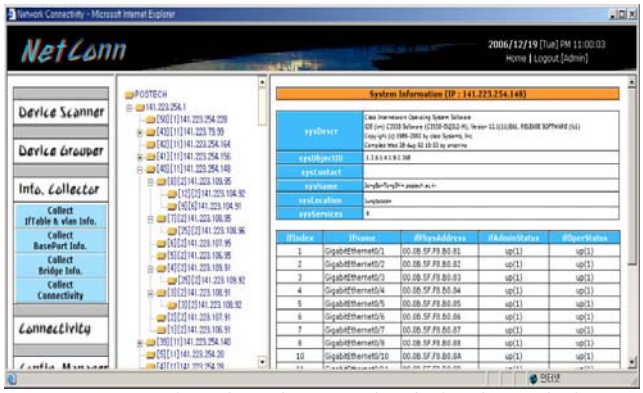

그림 5. 네트워크의 포트별 연결 정보 화면

그림 5는 장비간 포트별 연결 정보를 확인 할 수 있는 화면으로 연결 정보 화면은 윈도우 탐색기 형식으로 구성되어 각 장비의 대표 IP address를 기준으로 포트별 연결 정보를 트리 구조로 보여준다. 그림 5는 141.223.254.148에 연결된 모든 장비 리스트를 보여주는 것으로, 화면에서 첫번째 대괄호안의 숫자는 연결된 상대 장비의 포트를 의미하고 두번째 대괄호의 숫자는 상대 장비와 연결된 자신의 장비의 포트를 의미한다. 즉, '[40][11]141.223.254.148'의 의미는 '141.223.254.1'의 40번 포트와 '141.223.254.148' 장비의 11번 포트가 연결되었음을 나타낸다. 이 포트 매칭 정보가 실제 POSTECH 전산소에서 제공하는 정보와 일치한다.
또한 트리의 특정 장비의 IP address를 선택했을 때, 장비의 상세 정보가 오른쪽 화면에 나타난다. 그림 5는 141.223.254.148의 장비에 관한 상세 정보인 시스템 정보 및 각 포트별 장비의 상태 및 MAC address 정보를 보여준다.

POSTECH 내의 600여대의 네트워크 장비를 discover하는데 약 14분, 동일 장비를 그룹핑하는데 약 3분, 필요한 MIB 정보를 수집하는데 약 82분, 장비간 연결 정보를 생성하는데 약 61분의 시간이 소요된다.

6. 결론 및 향후 과제

본 논문에서는 네트워크 연결 정보 구성 관리를 효율적으로 수행할 수 있도록 네트워크 장비간의 물리적인 연결 정보를 알아내는 시스템을 개발하였다. 본 논문에서 우리는 포트별 IP 매칭 알고리즘을 제안하고 그에 따른 요구사항을 분석하고, 시스템 설계, 구현하여 네트워크 구성도를 생성하는 방법을 제시하였다. 네트워크 장비 검색 및 장비간 연결 정보를 알아내는 알고리즘을 제시했다는 점과 제시한 알고리즘을 기반으로 시스템을 구현한 후, POSTECH 네트워크에 적용하여 알고리즘을 검증함으로써 실제 네트워크 관리에 도움을 주는 시스템을 설계 및 구현했다는 점에 의의를 둔다.

향후 과제로 네트워크의 규모가 커져 장비의 수가 늘어나도 적정 시간내에 장비를 검색할 수 있는 scalability 보장에 관한 연구가 이루어져야 한다. 또한, 장비를 검색함에 있어서 전체 네트워크에 대한 탐색이므로 빠른 시간 내에 장비 검색이 이루어지도록 performance를 향상시키는 연구가 이루어져야 한다.

참고 문헌

[1] Bruce Lowekamp, David R. O'Hallaron, Thomas R. Gross, "Topology Discovery for Large Ethernet Networks", SIGCOMM, pp. 237~248, Aug. 2001.
[2] 박윤규, "웹 기반의 네트워크 트래픽 모니터링 시스템의 자동 구성 및 재구성 기법", Masters Thesis, POSTECH GSIT, 2000.
[3] E. Decker, "Definitions of Managed Objects for Bridges", IETF, RFC 1493, Jul. 1993.
[4] AdventNet Inc.,"Switch Port Mapper Tool", http://manageengine.adventnet.com/products/oputils/switch-port-mapper.html, Refer Jan. 2007.
[5] Cisco, "Network Assistant Feature", http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cna/v4_0/gsg4_1/gsg_en/feature.htm, Refer Jan. 2007.
[6] IBM, "Netcool/Precision for IP Networks", http://www-306.ibm.com/software/tivoli/products/netcool-precision-ip/index.html, Refer Jan. 2007.
[7] Cisco, "SNMP Community String Indexing", http://www.cisco.com/warp/public/477/SNMP/camsnmp40367.html, Refer Jan. 2007.
[8] Cisco, "How To Get Dynamic CAM Entries (CAM Table) for Catalyst Switches Using SNMP", http://www.cisco.com/warp/public/477/SNMP/cam_snmp.html, Refer Jan. 2007.
[9] AdventNet Inc, "AdventNet API", http://snmp.adventnet.com/help/snmpapi/snmpv1/, Refer Jan. 2007.