

신기술 해설

홈 네트워크의 위상 검출 및 관리

고 의 열* 엄 익 준**

목 차

1. 서 론
2. 관련 연구
3. 위상 검출 시스템
4. 실험 및 성능분석
5. 결 론

1. 서 론

임베디드 시스템이란 특정한 목적을 수행하도록 만든 컴퓨터 하드웨어와 소프트웨어의 결합체다. 즉, 임베디드 시스템에서 컴퓨터는 특정 목적을 위해 설계된 시스템의 한 구성요소다. 공업용 기계들이나, 자동차, 의학용 기계, 카메라, 세탁기나 냉장고 같은 집안용 기계들, 비행기, 자판기, 그리고 장난감들의 내부에 임베디드 시스템이 포함될 수 있다. 임베디드 시스템의 종류는 앞에서 언급한 것 외에도 너무나 다양하여, 단순하게는 세탁기에서, 복잡한 로켓이나 인공위성 같은 것들에 이르기까지 거의 대부분의 기계들을 임베디드 시스템이라 할 수 있다. 우리가 집안에서 사용하고 있는 대부분의 가전제품 (TV, 냉장고, 세탁기, 컴퓨터 등)들도 임베디드 시스템이라 할 수 있다. 본 논문에서는 임베디드 시스템의 중요 응용 분야 중 하나인 홈 네트워크에 대해서 다루고자 한다.

홈 네트워크란 네트워크 연결 능력이 다양한 정

보가전 기기에 포함되고 이런 것들이 가정 내로 파급되면서 이들 제품들 간에 정보를 교환하고 공유할 필요성이 제기되면서 탄생했다. 일반적으로 홈 네트워크 개념은 외부 접근 망에 대한 접점을 이루는 홈 게이트웨이 혹은 셋톱 등의 기기를 중심으로 가정 내에 위치하는 네트워크 구조 및 네트워크 장비와 정보가전기기 간의 통합 및 운영을 위한 소프트웨어, 서비스 등을 말한다. 홈 네트워크는 연결되는 정보가전 기기의 특성에 따라 데이터 네트워크(인터넷 접속, 파일 공유, 주변기기 공유 등), 엔터테인먼트 네트워크(DVD, D-TV, 셋톱박스, 게임기 등), 제어 네트워크(조명 및 온도 제어, 에너지 관리 등)로 나뉜다.

홈 네트워크 내의 각각의 네트워크는 서로 다른 특징과 목적을 갖고 있기 때문에 이들이 연결되어 있는 네트워크 미디어도 여러 유형으로 나타난다. 예를 들어 엔터테인먼트 네트워크에 속해 있는 것들은 대용량의 데이터를 주고받기 때문에, 많은 대역폭과 실시간으로 전송할 수 있는 기능이 필요하다. 따라서 이에 적합한 AV네트워크에 연결되어야 하므로 이를 연결시켜 주는 브릿지가 필요하다. 다른 예로 제어 네트워크 같은 경우는 제어 명령이 단

* 한국과학기술원 전산학과 석사

** 한국과학기술원 전산학과 조교수

<본 연구는 삼성전자와 대학 IT연구센터 육성지원사업의 연구결과로 수행되었음>

순하므로 주고받는 데이터양이 많지 않다. 하지만 PLC제품들을 제어할 수 있는 프로토콜을 이해할 수 있는 프록시 기능을 갖춘 브릿지를 필요로 한다.

전통적인 컴퓨팅 환경이나 네트워킹 환경은 사무용에 알맞게 되어있으므로 이를 그대로 홈네트워킹에 적용할 수 없다. 현재는 기존의 사무용 네트워크 기술을 홈 네트워크 환경에 적용시킬 수 있도록 보완하려는 노력이 계속되고 있지만, 홈 네트워킹 고유의 특성으로 인해 새로 개발되어야 하는 기술들도 많다. 예를 들어 홈 네트워크 내에 존재하는 다양한 네트워크 미디어 및 여기에 연결되어 있는 장비들의 정확한 위상을 파악하는 기술도 이러한 부류에 속한다.

홈 네트워크의 외부 접근망은 네트워크 사업자들이 관리를 해주므로 사용자의 관리가 필요 없지만, 내부의 경우 사용자의 취향 등 다양한 요소에 의해 결정되고 네트워크 별로 갖는 특징이 다르므로 사용자의 관리가 필요하다. 하지만 사용자들은 네트워크 환경의 관리에 대한 지식이나 경험이 부족하므로 이런 문제를 해결하기 위해 홈 네트워크를 구성하는 제품들이 상호 협조하여 사용자가 개입하지 않는 상태에서 자동으로 전체 네트워크 위상을 구성하고 관리하는 기술이 필요하다. 이런 기술의 개발 시 응용 분야는 다음과 같다. 일단, 전체 네트워크의 위상을 파악하고 있으므로, QoS 관리에 도움을 줄 수 있다. 또한 홈 네트워크에 연결된 장비들의 물리적 위치에 따른 세그먼트 관리 및 응용프로그램 서비스를 할 수 있으며, 홈 네트워크 기기들에 대한 통합 제어가 필요한 경우 현재 연결되어 관리할 수 있는 기기들에 대한 정보를 수집할 수 있다. 또한 검출된 위상을 활용한 여러 기능의 응용 서비스들이 가능하게 된다.

본 논문에서는 홈 네트워크에서의 위상 검출 시스템을 제안한다. 제안하는 위상 검출 시스템은 사용자의 개입을 최소로 한 상태에서 전체 홈 네트워

크를 구성하는 제품들이 상호 협조하여 자동으로 전체 네트워크의 위상을 검출하고 변화가 없는 한 검출된 위상을 유지한다. 또한 메시지 손실에 의해 일어나는 노드간의 유지하고 있는 위상의 불일치는 히스토리를 이용하여 알아내고 위상 정보의 재전송을 통해 이를 해결한다. 또한 제안하는 위상 검출 시스템은 네트워크를 최대한 적게 사용하면서 전체 위상을 검출하고 유지 하므로 대역폭이 크지 않은 네트워크에서도 잘 동작할 수 있으며 시스템의 자원 역시 가능한 적게 사용한다.

본 논문은 다음과 같은 순서로 구성된다. 먼저 2장에서는 네트워크의 위상 검출과 관련된 다른 연구들에 대해 소개한다. 3장에서는 본 논문에서 제안하는 위상 검출 시스템을 소개하고 관련된 알고리즘을 기술하도록 한다. 4장에서는 실험을 통해 제안하는 위상 검출 시스템의 성능을 분석한다. 마지막으로 5장에서는 본 논문의 결론을 맺도록 한다.

2. 관련 연구

네트워크의 위상 검출의 목적은 크게 두 가지로 나눌 수 있다. 첫 번째로는 네트워크의 라우팅을 위함이고, 두 번째로는 네트워크의 관리를 위해서이다. 현재 네트워크의 위상 검출은 위 두 개의 목적을 가지고 연구되고 있지만, 홈 네트워크 상에서의 위상 검출과 직접적으로 관련되는 연구는 아직까지는 없다. 따라서 본 장에서는 일반 인터넷 환경에서의 위상 검출 시스템에 대한 관련 연구를 소개한다.

먼저 라우팅을 위한 위상 검출 방식으로 소개하도록 한다. 현재 사용되는 혹은 연구되고 있는 대부분의 라우팅 프로토콜은 라우팅을 위해 주변 라우터들의 위상을 검출한다. 대표적인 라우팅 프로토콜인 RIP[1], OSPF[2]는 라우팅을 위해 주변 라우터들의 위상을 검출하고, 이를 라우팅에 사용한다. 또한 최근에 개발되어지고 있는 애드 혹 네트워크에서의 라우팅 프로토콜들인 DSR[3],

AODV[4] 역시 애드 혹 네트워크에서의 라우팅을 위해 주변 노드들의 위상 정보를 검출한다. 하지만 위에서 제시된 연구들은 기본적으로 네트워크의 위상 검출이 목적이 아닌 라우팅을 위한 것들로, 라우팅 패스를 찾기 위해 라우터들을 찾는 것이 주목적이 된다. 또한 많은 라우팅 프로토콜들은 전체 네트워크의 위상 정보를 가지고 있기에는 너무 그 정보의 양이 크므로 제한된 정보만을 가지고 있다. 마지막으로 이런 연구들은 전체 네트워크가 어떤 물리적인 연결로 되어 있는지를 구별할 수 없다. 즉, 이 연결이 유선랜인지, 무선랜인지, 혹은 다른 미디어로 연결된 것인지에 대해서는 관심이 없다.

다음으로 네트워크의 관리를 위하여 네트워크의 위상을 검출하는 연구들을 소개한다. 이런 연구로는 [5], [6], [7]과 같은 연구들이 있으며, 네트워크의 위상을 검출하는 것이 주 목적으로 네트워크 내의 노드들의 위상을 검출하게 된다. 하지만 이 역시 라우터나 스위치를 대상으로 연구 되었으며, 물리적인 연결의 종류에 대해서는 관심이 없다.

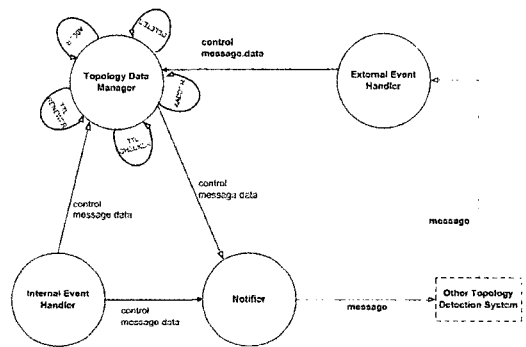
따라서 현재까지의 연구들은 소규모며, 다양한 노드와 네트워크로 구성된 홈 네트워크에서 효율적으로 전체 위상 검출을 할 수가 없다. 따라서 홈 네트워크와 같은 환경에서 효율적으로 전체 위상을 검출할 수 있는 새로운 시스템이 필요로 된다.

3. 위상 검출 시스템

본 논문에서는 홈 네트워크에 알맞은 위상 검출 시스템을 제안한다. 제안하는 위상 검출 시스템은 홈 네트워크에서 각 노드의 위상을 검출하고, 노드가 어떤 물리적인 미디어로 연결되어 있는지를 검출한다. 또한 QoS 관리를 위해 각 네트워크의 사용 가능한 대역폭을 주기적으로 계산한다. 또한 제안하는 위상 검출 시스템은 사용자가 원하는 시간 안에 전체 네트워크에서의 위상 변화를 감지할 수 있도록 디자인 되었다. 마지막으로, 메시지 드랩에 의

한 노드 간의 위상 정보의 불일치를 없애고, 검출한 위상의 신뢰성을 위해 히스토리를 사용한 재전송을 사용하였다.

위상 검출 시스템은 (그림 1)과 같이 구성되어 있다. 전체 시스템은 크게 External Event Handler (EEH), Topology Data Manager(TDM), Internal Event Handler(IEH), Notifier의 4개의 모듈로 구성된다. 각 모듈은 시스템 내부에서 메시지를 주고받으며 동작하고, 전체 시스템은 외부의 다른 노드의 위상 검출 시스템과 메시지를 주고받으면서 동작한다.



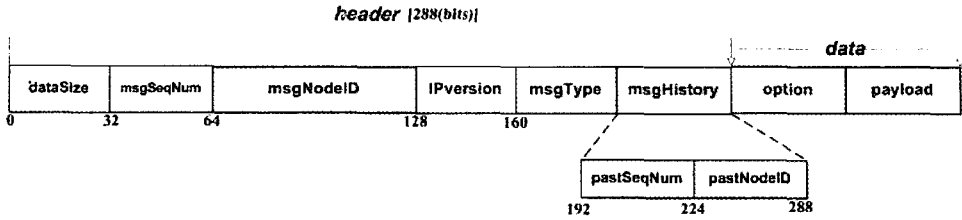
(그림 1) 위상 검출 시스템 구조도

위상 검출 시스템은 사용자에게 자신이 가진 위상 정보를 보여주기 위해 XML(eXtensible Markup Language)[8]을 이용하여 표현한다. 이렇게 표현된 XML 파일은 임베디드 시스템 내부에 저장되고, 이렇게 저장된 XML 파일은 간단하게는 인터넷 익스플로어나 넷스케이프같은 웹 브라우저를 이용하여 볼 수 있으며, 본 시스템에서는 Java를 이용하여 그림으로 보여준다.

본 장에서는 각 모듈과, XML 파일의 구조, 그리고 Java를 이용한 프로그램에 대해 간단하게 소개한다.

3.1 External Event Handler(EEH)

EEH는 같은 네트워크를 구성하고 있는 다른 노드들로부터, 위상 검출을 위한 메시지를 받아서 처리하는 모듈이다. 위상 검출을 위해 노드들이 주고



(그림 2) 위상 검출 시스템의 메시지 포맷

받는 메시지는 (그림 2)와 같은 포맷을 갖는다. 메시지는 ADD, AADD, DEL, KEEP_ALIVE 4개이며, 각각은 다음과 같은 역할을 한다.

SEND_AADD, DEL, TTL_CHECK, KEEP_ALIVE로 총 6가지의 연산을 수행한다. 각각의 컨트롤을 받았을 때, 수행하는 작업은 <표 2>와 같다.

<표 1> 메시지의 종류와 역할

| 종 류 | 역 할 |
|------------|--|
| ADD | 위상 정보가 추가되었음을 알리는 역할을 한다. data 부분에는 추가할 위상 정보를 가진다. |
| AADD | 노드가 가진 위상 정보 모두를 data 필드에 가진 메시지로 이 메시지를 받은 노드는 자신의 위상 정보 리스트에 data 필드의 정보를 추가한다. (중복되는 것은 무시한다) AADD 메시지는 error recovery를 위한 것으로, 노드들 간의 history가 다른 것이 발견되었을 때, 노드들은 AADD 메시지를 발생시켜 history가 다른 것을 없애고, 자신의 위상 정보 리스트를 다른 노드들과 같게 한다 |
| DEL | 위상 정보가 삭제되었음을 알리는 역할을 한다. data 부분에는 삭제될 위상 정보를 가진다. |
| KEEP_ALIVE | 노드가 살아 있음을 주기적으로 알린다. |

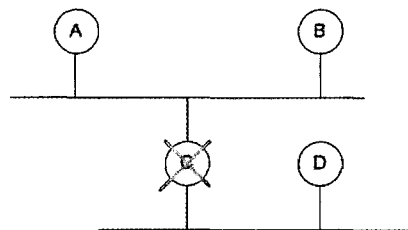
<표 2> TDM의 수행 작업

| 컨트롤 | 수행 작업 |
|------------|--|
| ADD | EEH로부터 오며, 추가할 위상 정보를 같이 받아 노드의 위상 정보 리스트에 추가한다. 추가한 후에는 다른 노드들에게 알리기 위하여 ADD 컨트롤과 추가된 위상 정보를 notifier에게 전달한다. |
| AADD | EEH로부터 오며, 추가할 위상 정보를 같이 받아 노드의 위상 정보 리스트에 추가한다. 추가한 후에는 AADD 메시지를 보낸 네트워크를 제외한 다른 네트워크의 노드들에게 알리기 위해 ADD컨트롤과 추가된 위상 정보, 그리고 제외할 네트워크를 notifier에게 전달한다. |
| SEND_AADD | EEH로부터 오며, 자신의 모든 위상 정보를 모아 AADD컨트롤과 함께 보낸다. 또한 AADD메시지를 보낼 네트워크를 같이 전달하여, AADD 메시지가 모든 네트워크로 브로드캐스트 되는 것을 막는다. |
| DEL | EEH와 TDM으로부터 오며, 삭제될 위상 정보를 같이 받아 노드의 위상 정보 리스트에서 삭제한다. 삭제 한 후에는 (그림 3)과 같은 네트워크 파티션이 일어났는지를 찾아내어 파티션이 되었을 경우, 자신이 속하지 않은 파티션의 모든 노드를 지운다. 그리고 난 후, 다른 노드들에게도 알리기 위해 DEL컨트롤과 삭제된 위상 정보를 notifier에게 전달한다. |
| TTL_CHECK | EEH로부터 오며, 노드의 위상 정보 리스트에 있는 neighbor들의 TTL을 검사한다. 저장되어 있는 TTL이 현재 시각보다 10초 이상 차이나면, neighbor노드가 죽었다고 판단하고, DEL컨트롤과 삭제할 위상 정보를 TDM에게 보낸다. |
| KEEP_ALIVE | EEH로부터 오며, 위상 정보 리스트에서 KEEP_ALIVE를 보낸 노드를 찾아 노드의 TTL값은 현재 시각으로 업데이트 한다. |

EEH는 위와 같은 메시지를 다른 노드로부터 받은 후, 이 메시지가 valid한지 아닌지를 판단한 후, Topology Data Manager(TDM)에게 넘겨주게 된다. 메시지가 valid한지를 판단하는 기준은 msg History 필드로, 노드 자신이 가지고 있는 history와 메시지에 있는 history가 같은 경우 valid하다고 판단하게 된다. 이런 history를 노드의 인터페이스별 로 유지하게 된다.

3.2 Topology Data Manager(TDM)

TDM은 노드 자신의 위상 정보 리스트에 대한 관리자로 위상 정보 및 QoS 정보 등을 위상 정보 리스트에 추가 및 삭제 등을 수행한다. TDM이 다른 모듈로부터 받는 컨트롤은 ADD, AADD,



(그림 3) 네트워크 파티션

3.3 Internal Event Handler(IEH)

IEH는 노드 내부적으로 주기적으로 동작해야 하는 작업을 수행한다. 노드는 주기적으로 자신이 살아있음을 다른 노드에 알려야 하며, 자신의 위상 정보 리스트에서 neighbor들의 TTL값을 검사해야 한다. 그리고 마지막으로 네트워크의 QoS 관리를 위해 주기적으로 available bandwidth를 계산하여 노드들에게 알린다.

자신이 살아있음을 알리기 위해서 IEH는 주기적으로 KEEP_ALIVE컨트롤을 notifier에게 알린다. 자신의 위상 정보 리스트를 검사하기 위해서 IEH는 TTL_CHECK컨트롤을 TDM에게 보내어 neighbor들의 TTL을 검사하게 한다. 마지막으로 available bandwidth의 측정을 위해서 노드는 주변의 노드들에게 임의의 packet을 n개 보내어 n-1개가 받아지는 시간을 측정 후, 이 시간을 가지고 available bandwidth를 측정한다. 측정 후에는 ADD 컨트롤과 함께 TDM에게 보내어 위상 정보 리스트의 네트워크 QoS 정보를 업데이트하게 한다.

3.4 Notifier

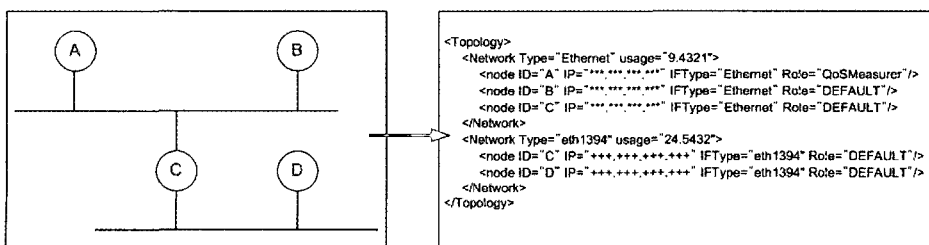
Notifier는 TDM이나 IEH로부터 컨트롤과 데이터를 받아 (그림 2)와 같이 메시지를 만들어 주변의 노드들에게 브로드캐스트하는 역할을 한다. 이때, 대부분의 컨트롤의 경우에는 연결된 모든 네트워크에 브로드캐스트하며, AADD의 경우, 보내야 하는 네트워크에만 브로드캐스트하고, ADD컨트롤

중 제외해야 할 네트워크가 명시되어 온 경우, 그 네트워크만을 제외하고 브로드캐스트하게 된다.

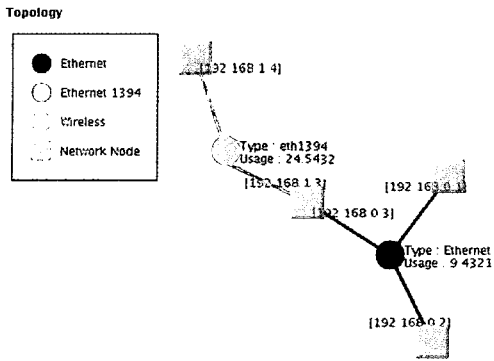
3.5 Representation

앞에서 언급한 대로, 위상 검출 시스템은 사용자에게 검출된 위상을 보여주기 위해 XML을 사용하고, 만들어진 XML을 이용하여 Java프로그램을 통해 네트워크 그림을 만들어 보여준다. 전체 네트워크는 네트워크 밑에 노드가 있는 계층적인 구조를 가지므로 XML을 이용하여 표현할 때도 이런 계층적인 구조로 표현한다. 각 노드의 위상 검출 시스템은 전체 네트워크의 위상 정보를 계층적인 구조로 가지고 있다. 예를 들어 (그림 4)의 왼쪽의 네트워크가 있을 때, 각 노드는 "전체 네트워크는 A, B 네트워크로 구성되어 있으며, 네트워크 A에는 a, b, c 3개의 노드가 있으며, 네트워크 B에는 c, d 2개의 노드가 있다."라는 전체 위상을 가지고 있게 된다. 위상 검출 시스템은 이를 (그림 4)의 오른쪽에 있는 XML로 바꾸어 저장하고 있게 된다.

이런 XML 파일은 위상 검출 시스템의 위상 정보 리스트가 업데이트될 때마다 새로 생성되며, Java프로그램을 이용하여 전체 네트워크를 그림으로 볼 수 있다. Java프로그램은 Java에서 기본적으로 지원하는 DOM[9] 패키지를 이용하여, 위상 검출 시스템이 만든 XML 파일을 파싱하고, 이를 적절한 데이터 구조를 이용하여 저장한 후, Java의 스윙을 이용하여 표현해 준다. Java프로그램을 이용하여 (그림 4)의 네트워크를 그려보면 (그림 5)와 같다.



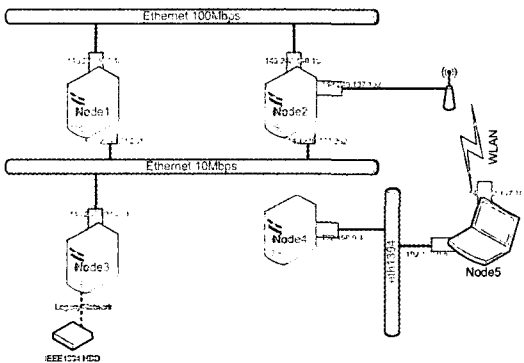
(그림 4) 위상 정보의 XML 표현의 예



(그림 5) Java를 이용한 네트워크 위상 정보의 표현

4. 실험 및 성능 분석

4.1 실험 환경



(그림 6) 실험 환경

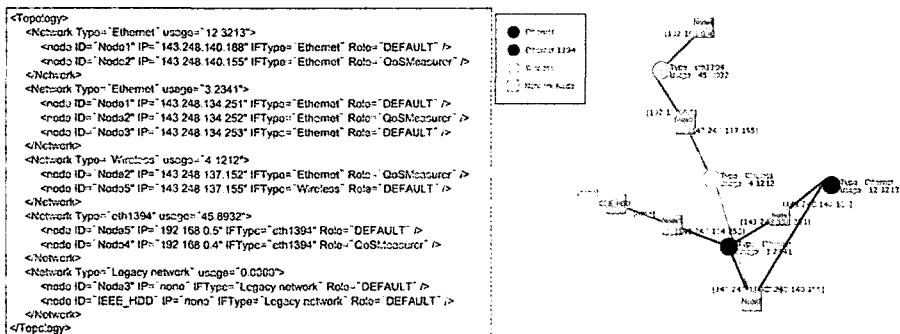
실험 환경은 (그림 6)과 같다. 실험에 사용된 네트워크는 100Mbps Ethernet, 10Mbps Ethernet,

IEEE 802.11b WLAN, 그리고 IP over IEEE 1394 4가지다. 각 노드에는 위상 검출 시스템이 설치되어 있으며, QoS 정보의 수집을 위해 available bandwidth를 측정하는 노드는 Node2와 Node4이다. 최근의 많은 임베디드 시스템이 Linux를 기반으로 동작하므로, 각 노드에는 Linux를 설치하여 실험하였다.

4.2 결과 및 성능

이와 같이 실험 했을 때, 각 노드들은 전체 위상 정보를 갖게 되고, 이 위상정보가 XML로 표현되어 (그림 7)과 같이 저장된다. 그리고 Java를 이용한 프로그램을 가지고 그림으로 표현하여 (그림 7)과 같이 표현된다.

이렇게 각 노드에서 위상 검출 시스템을 동작시켰을 시, 위상 검출 시스템의 CPU 점유율은 5% 이하였으며, 메모리는 2MBytes미만을 사용하며 동작하였다. 또한 네트워크 사용량을 계산하여 보면, 전체 위상 검출이 끝나고 안정화 되었을 시, 각 노드가 n초 간격으로 KEEP_ALIVE 메시지를 보낸다고 하면, KEEP_ALIVE 메시지의 크기가 94 byte이므로 $752/n$ bits/sec 정도로 매우 작음을 알 수 있다. 또한 메시지의 수를 최소화하기 위해 메시지를 보낼 때, 같은 네트워크의 다른 노드가 같은 메시지를 보내는 지를 확인하고 보내게 되어 있다. 따라서 위상 검출 시스템이 주고받는 메시지에



(그림 7) 실험 네트워크의 XML 표현 및 그림

의한 네트워크의 점유율은 크지 않다고 볼 수 있다.

5. 결 론

본 논문에서는 홈 네트워크에서의 위상 검출 시스템을 제시하였다. 제시한 위상 검출 시스템은 사용자의 개입을 최소로 한 상태에서 네트워크에 연결된 노드들이 자동적으로 서로 메시지를 주고받으면서 전체 네트워크의 위상을 검출한다. 또한 시스템의 자원을 적게 사용함으로써 노드가 자신의 고유의 기능을 수행하면서도 위상 검출 시스템 역시 원활하게 작동할 수 있게 하며, 네트워크의 자원 역시 적게 사용함으로써 홈 네트워크에서 사용될 수 있는 다른 트래픽에 영향을 적게 끼칠 수 있다. 또한 네트워크의 bandwidth가 작은 경우에도 사용이 가능하게 된다. 본 위상 검출 시스템은 검출한 위상 정보를 XML로 표현하여, 사용자에게 쉽게 전달할 수 있으며, 전체 홈 네트워크를 관리하는데도 사용할 수 있게 한다.

참고문헌

[1] C. Hedrick, "Routing Information Protocol," RFC 1058, Network Working Group, June 1988

[2] J. Moy, "OSPF Version 2," RFC 1583, Network Working Group, March 1994.

[3] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad Hoc Networks", Mobile Computing, T. Imielinski and H. Korth, Eds. Kulwer, 1996, pp. 152- 81.

[4] C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Network Working Group, July 2003.

[5] B. Bellur and R.G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks", IEEE INFOCOM'99

[6] Y. Bejerano, Y. Breitbart, M. Garofalakis, and R. Rastogi, "Physical Topology Discovery for Large Multi-Subnet Networks", IEEE INFOCOM'03

[7] Y. Breitbart, M. Garofalakis, C. Martin, R. Rastogi, S. Seshadri, and A. Silberschatz, "Topology Discovery in Heterogeneous IP Networks", IEEE INFOCOM'00

[8] L. Quin, Extensible Markup Language (XML), <http://www.w3.org/XML/>.

[9] P. Le Hegaret, Document Object Model (DOM), <http://www.w3.org/DOM/>.

저자약력



고 의 열

2003년 한국과학기술원 전산학과(학사)
2003년 - 현재 한국과학기술원 전산학과(석사)



염 익 준

1995년 연세대학교 전산공학과(학사)
1998년 Texas A&M University 컴퓨터공학 석사
2001년 Texas A&M University 컴퓨터공학 박사
2000년 Nortel Innovative Lab. 연구원
2001년 경북대학교 연구교수
2002년 - 현재 한국과학기술원 전산학과 조교수