LAN 상의 장애 검출 및 위치 확인을 위한 규칙 기반 장애 진단 에이전트 시스템

조 강 홍[†] · 안 성 진[††] · 정 진 욱[†††]

요 약

이 논문에서는 LAN 상의 장애 탐지와 위치 확인을 위한 에이전트 구조와 규칙을 제시하고 있다. LAN 상에서 발생하는 치명적인 장애의 원인을 찾기 위해 충돌 감지 규칙, 에러 감지 규칙, 브로드케스트 감지 규칙, 시스템 위치 탐지 규칙 그리고, 인터넷 응용 서비스 탐지 규칙 등을 제안하고 있다 또한, 이들 규칙의 집합이 서로 연관성을 갖도록 멀티 에이전트 시스템의 구조와 상태 천이를 기술하고 있다 제시된 규칙의 적용성을 검증하기 위하여 실제로 발생했던 LAN 장애 정황 정보를 제시된 규칙에 따라 모니터링하고 분석하여 실제 장애 시스템을 찾는 과정을 보여준다 이러한 규칙 기반 에이전트 시스템은 인터넷 관리자에게 망 정보 수집에서 장애 판단 및 원인 해결까지 큰 도움을 줄 것으로 기대된다

Rule-based Fault Detection Agent System for Fault Detection and Location on LAN

Kang-Hong Cho[†] · Seong-Jin Ahn[††] · Jin Wook Chung[†††]

ABSTRACT

This paper proposes the structure of an agent and rules for fault detection and location on LAN To find out a reason of critical fault incurred LAN, collision detection rule, error detection rule, broadcast detection rule, system location rule, and Internet application location rule are shown. Also, the structure of multi-agent system and state transition diagram is portrayed to have connectivity with the set of rules. To verify availability of proposed rules, the process to find a faulty system is shown by monitoring and analyzing the LAN fault occurrences from the proposed set of rules. Such an rule based agent system is helpful to an Internet manager to solve a reason of fault and make a decision from gathering management information

1. 서 론

컴퓨터 통신 기술이 발전함에 따라 네트워크는 점차 방대해지게 되었고 이에 상응하여 인터넷은 급속하게 성장되었다. 인터넷의 성장과 함께 LAN 상의 응용 프로그램들의 사용 증가는 트레픽의 병목 현상을 보이는 WAN 뿐만 아니라, LAN 상의 트래픽 양을 크게 증가시키게 되었고, 이로 인한 대부분의 장애 요소는 LAN 내부에서 원인이 발생된다. 이와 같은 장애 요인에 의한 네트워크의 마비는 엄청난 비용 손실을 가져오고 작업 효율을 떨어뜨리게 된다. 뿐만 아니라, 어떤 경우에는 장애 원인의 발견 및 복구 없이 LAN 장비에 과다한 투자를 수행하기도 한다. 따라서, LAN상의 장애 관리는 장비의 투자 및 유지 보수 비용을 낮추고 성능을 최대로 유지하기 위한 필수 요소이다[1, 2]

LAN 상의 장애 관리의 필요성이 증대됨에 따라 이와 관련된 많은 네트워크 관리 기술들이 연구되었고,

† 준 회 원 : 성균관대학교 대학원 전기전자 · 컴퓨터공학부
†† 종신회원 : 성균관대학교 컴퓨터교육과 교수
††† 종신회원 : 성균관대학교 전기전자 · 컴퓨터공학부 교수
논문접수 : 1999년 11월 11일, 심사완료 : 2000년 6월 13일

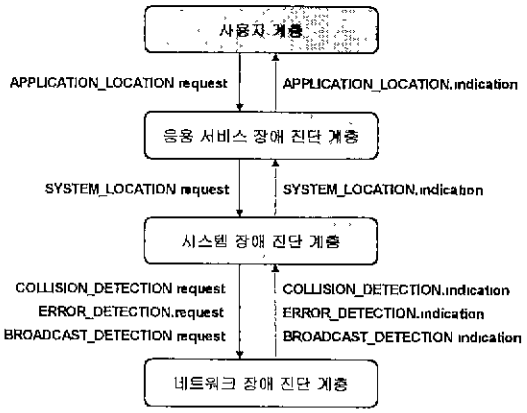
현재는 기존의 stand-alone 방식의 네트워크 관리 형태를 탈피하여 웹 기반 네트워크 관리에 대한 연구가 여러 방면에서 진행중이다[3, 4]. 웹 기반 네트워크 관리는 인터넷 기반 기술을 네트워크나 시스템, 응용 프로그램 관리에 적용시켜 관리 행위의 시간적 공간적 제약을 극복하기 위한 접근 방법으로 이를 위한 표준 관리 기술로 WBEM(Web Based Enterprise Management)와 JMAPI(JavaManagement API) 등이 제안되고 있다[5, 6].

그러나, 이와 같은 관리 기술들이 가지고 있는 근본적인 문제는 인터넷 관리자가 네트워크 및 시스템 관리 기술에 대한 전문 지식과 활용법, 트래픽 분석 능력 등을 숙지하고 있어야 하는데 국내외 네트워크 관리의 현실을 비추어 볼 때, 이는 쉽지 않은 문제이다. 또한, 본 논문의 주제인 장애 관리는 OSI 5대 관리 기능 중 가장 중요하면서도 가장 어려운 기능으로 실제 네트워크 관리 전문가라 하더라도 관리 행위를 수행하는데 많은 시간과 노력이 필요할 것이다. 근래에 들어서 장애 관리를 좀 더 능동적으로 하기 위한 연구들이 많이 진행 중이다. 그러나, 대부분의 연구는 주로 LAN의 장애 항목 자체에 대한 연구나 하위 계층 상의 장애 검출에 그 초점을 두고 있다[6-8].

본 논문에서는 이런 장애 항목들을 계층으로 구분하여 체계화 하였고, 각 분석 항목을 규칙화하여 그 계층간의 연관성에 관해 연구하였다. 본 논문에서는 LAN 상의 장애 관리의 중요성을 인지하고, 장애 관리를 체계적으로 수행하기 위해 LAN 관리에 적합한 RMON MIB을 기반으로 장애 진단 계층 모델을 확립하였고, 현재 차세대 기술로 주목받고 있지만 아직까지 네트워크 장애 진단 분야에 적용이 이루어지지 않은 에이전트 개념을 장애 진단 계층 모델에 적용시켜 장애 진단 멀티 에이전트 시스템의 모델을 설계하였다. 또한, 각 에이전트들이 포함하는 장애 진단 계층상의 장애 진단 규칙들을 제시하고 이를 활용하는 예를 보이기 위해 실제 환경에서의 진단 결과를 보였다.

2. 장애 진단 모델

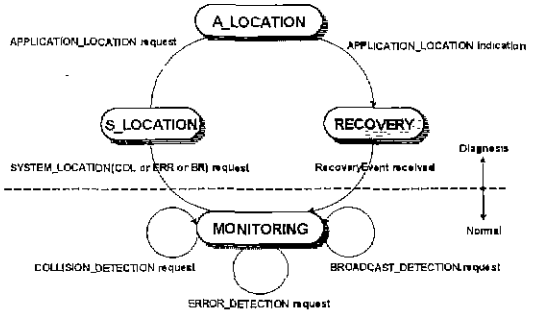
네트워크 상의 장애 관리는 아직까지 분석 항목조차 체계적으로 정의되어 있지 않기 때문에 본 논문에서는 먼저 장애 관리를 효과적으로 수행하기 위해서 사용자의 장애 관점으로부터 그 장애 계층을 구분하고 이에 관련된 분석 항목들을 정의하여 이를 유기적으로 연결하는 장애 진단 계층과 프리미티브들의 모델을 확립하였다. (그림 1)은 LAN 장애 관리를 위해 적합한 RMON MIB을 기반으로 장애 감지와 장애 위치 탐지를 체계적으로 수행할 수 있도록 확립된 장애 진단 계층 모델을 나타낸다. 그림에서 보는 바와 같이 장애 진단 모델은 맨 상위의 사용자 계층을 제외하고 실제 관리 행위가 발생되는 3가지 계층으로 구분하였다. 맨 하위 계층으로 구분된 네트워크 장애 진단 계층은 관리 도메인 상의 여러 개의 네트워크 세그먼트 중에서 장애가 발생된 세그먼트를 구분하는 계층이다. 즉, 각 세그먼트를 모니터링하여 충돌율, 에러율, 브로드캐스트 율 등의 장애 파라미터들을 계산하여 이를 허용 가능한 범위의 임계치와 감지 규칙에 의해 장애 발생 여부를 감지한다. 시스템 장애 진단 계층은 네트워크 장애 진단 계층에 의해 장애가 발생했다고 판단된 세그먼트들에 대해 그 세그먼트 상의 장애 발생 원인이 되는 장애 시스템의 위치를 탐지하는 역할을 수행한다. 네트워크 장애 진단 계층의 장애 파라미터들을 근거로 이 트래픽을 가장 많이 발생시키는 호스트들을 측정하여 그 호스트의 위치를 확인할 수 있다. 가장 상위의 응용 서비스 장애 진단 계층은 시스템 장애 진단 계층에서 탐지된 장애 발생 시스템들이 어떤 응용 서비스로 하여금 장애를 발생하게 하는지 또는 어떤 응용 프로그램이 오동작을 하고 있는지 그 원인을 진단하는 역할을 수행한다.



(그림 1) 장애 진단 계층의 service primitive

또한, (그림 1)에서는 장애 진단 계층 사이의 서비스 프리미티브(service primitive)들을 보여주고 있다. 네트

워크 상의 장애를 진단하고 그 시스템의 위치와 또한 그 시스템의 응용 서비스의 위치를 판단하기 위해 먼저 장애를 발생시킨 시스템의 응용 서비스를 탐지하기 위해 APPLICATION_LOCATION.request를 하위 계층인 응용 서비스 장애 진단 계층으로 보낸다. 응용 서비스 장애 진단 계층에서는 응용 서비스 탐지를 위해 먼저 선행되어야 하는 시스템 장애 진단 계층의 장애 시스템 탐지를 위해 SYSTEM_LOCATION.request를 하위 계층인 시스템 장애 진단 계층으로 보낸다. 시스템 장애 진단 계층에서는 충돌 장애 탐지를 위한 COLLISION_DETECTION.request, 에러 장애 탐지를 위한 ERROR_DETECTION.request 그리고, 브로드캐스트 장애 탐지를 위한 BROADCAST_DETECTION.request를 네트워크 장애 진단 계층을 전송한다. 맨 하위의 네트워크 진단 계층에서는 감지된 장애 요소에 따라 COLLISION_DETECTION.indication, ERROR_DETECTION.request 그리고, BROADCAST_DETECTION.indication을 시스템 장애 진단 계층으로 전송한다. 시스템 장애 진단 계층에서는 이를 근거로 장애를 발생시키는 시스템을 탐지하여 SYSTEM_LOCATION.indication을 응용 서비스 장애 진단 계층으로 전송한다. 마지막으로 응용 서비스 장애 진단 계층에서는 탐지된 장애 시스템 상의 응용 서비스를 진단하여 그 결과를 사용자 계층에게 APPLICATION_LOCATION.indication을 통해 전달하게 된다.



(그림 2) 장애 진단 계층의 상태 천이도

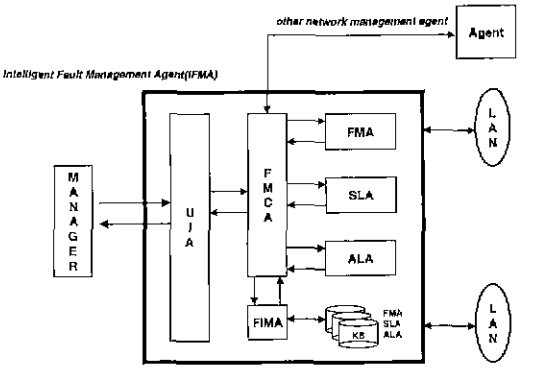
(그림 2)는 장애 진단 계층 상의 상태 천이를 나타낸다. MONITORING 상태는 네트워크의 장애 여부를 감지하는 상태로 장애가 발생하지 않은 상태이며, 네트워크 상의 충돌, 에러, 브로드캐스트 등의 장애가 발생했을 경우에는 S_LOCATION 상태로 천이하게 된다. S_LOCATION 상태에서는 세그먼트 상의 시스템들의 트래픽 통계를 통해 가장 많은 장애 트래픽을 발생시키는 시스템을 탐지하게 되고, 탐지한 후에는 A_LOCATION 상태로 천이한다. A_LOCATION 상태는 장애 시스템의 트래픽 통계를 분석하여 장애로 판단되는 응용 서비스를 발견한다. 그 다음의 상태는 RECOVERY 상태로 관리자로 하여금 장애 원인이 고립되거나 제거된 상태를 의미하여 RecoveryEvent를 통해 다시 해당 네트워크의 장애를 모니터링하는 정상 상태로 돌아오게 된다.

3. 장애 진단 에이전트 시스템

3.1 장애 진단 에이전트 구조

에이전트는 큰 의미에서 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결하여 주는 소프트웨어라고 할 수 있고[9, 10], 이런 관점에서 관리자를 대신해서 네트워크 장애 관리의 전문가 역할을 수행하는 장애 진단 에이전트 시스템을 설계할 수 있다.

(그림 3)은 장애 진단 모델 계층을 기반으로 설계된 장애 진단 에이전트 시스템인 IFMA(Intelligent Fault Management Agent)의 모델을 보여주고 있고 각 에이전트의 기능은 다음과 같다.



(그림 3) 장애 진단 에이전트 시스템 모델

3.1.1 사용자 인터페이스 에이전트(User Interface Agent : UIA)

사용자 인터페이스 에이전트는 사용자와의 상호작용을 위한 인터페이스 역할을 수행하는 에이전트이다. 관리 도메인 상의 에이전트로부터 장애 감지와 장애 위치 확인 결과를 수신하여 사용자에게 전달하는 기능

을 가진다. 인터넷 관리자의 지적 대리인으로서의 기능을 수행하기 위해 관리자의 네트워크 관리 행위에 대해 관찰하고 관리자로부터 받은 피드백을 기반으로 네트워크 장애 트래픽 패턴을 학습하는 메커니즘을 포함한다.

3.1.2 장애 관리 제이 에이전트(Fault Management Control Agent : FMCA)

장애 진단 에이전트를 구성하는 각 에이전트를 유기적으로 연결하고 제어하며, 전체 장애 관리 흐름을 총괄하는 에이전트로 새로운 관리 정책이 추가되거나 또는 변경되었을 경우, 그리고 또 다른 장애 에이전트 구성 요소가 추가되었을 경우 이 에이전트의 제어를 통해 쉽게 적용할 수 있다. 또한, 에이전트 기반의 또 다른 관리 도메인의 네트워크 관리 에이전트와의 정보 공유 및 교류를 수행하는 기능을 포함하고 있다.

3.1.3 장애 모니터링 에이전트(Fault Monitoring Agent : FMA)

인터넷 관리자를 대신하여 관리 도메인 상의 각 네트워크 세그먼트들을 모니터링하여 장애 발생 여부를 감지하는 에이전트로 이를 위해 다음 절에서 제시하는 충돌 진단 규칙, 에러 진단 규칙 그리고, 브로드캐스트 장애 진단 규칙 등을 포함하고 있다 관리 세그먼트 상의 장애를 감지했을 경우, 에이전트간 통신을 위한 메시지 형태로 변환하여 장애 발생 여부를 사용자 인터페이스 에이전트와 시스템 위치 탐지 에이전트에게 전달하는 역할을 수행한다.

3.1.4 시스템 위치 진단 에이전트(System Location Agent : SLA)

시스템 위치 진단 에이전트는 관리 세그먼트 상의 장애 시스템을 진단하는 에이전트로 시스템 위치 탐지 규칙을 포함하고 있다. 장애 모니터링 에이전트로부터 장애 발생 감지 메시지를 수신하였을 경우, 감지된 장애 종류를 기반으로 세그먼트 전체에 장애 영향을 미치는 시스템을 분석한다 분석 결과는 다시 응용 서비스 진단 에이전트로 전달된다.

3.1.5 응용 서비스 탐지 에이전트(Application service Location Agent : ALA)

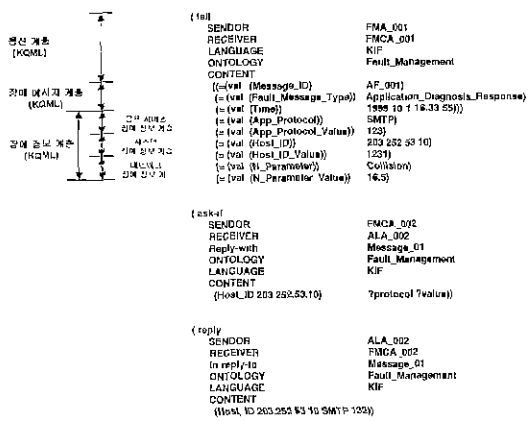
응용 서비스 탐지 에이전트는 장애 시스템이 발생시키는 응용 서비스 트래픽 중 장애를 일으키는 응용 서비스를 진단하는 에이전트로 응용 서비스 탐지 규칙을 포함한다. 시스템 위치 탐지 에이전트로부터 장애 유발 시스템을 통보 받았을 경우, 해당 시스템이 발생시키는 응용 프로토콜을 분석하는 기능을 가진다. 분석 결과는 사용자 인터페이스 에이전트를 통해 사용자에게 전달한다

3.1.6 장애 정보 관리 에이전트(Fault Information Management Agent)

장애 진단 에이전트에서 각 에이전트 구성 요소들이 동작하는데 필요로 하는 네트워크 구성 정보와 장애 관련 지식 등을 관리하는 에이전트로 장애 정보를 기반으로 학습하는 메커니즘을 포함한다.

3.2 통신 프로토콜

에이전트 상에서 이기종의 에이전트 들과의 정보 교환을 위한 통신 프로토콜은 필수적이다. 장애 진단 에이전트 시스템에서는 최근 통신 언어(ACL : Agent Communication Language)의 표준으로 인정받고 있는 KQML(Knowledge Query Manipulation Language)와 KIF(Knowledge Interchange Format)로 사용하였고, 그 메시지의 예는 (그림 4)와 같다



(그림 4) 장애 진단 에이전트 통신 메시지 형식

(그림 4)에서 보는 바와 같이 에이전트 사이에 장애 정보는 앞에서 설명한 장애 진단 계층에 따라서, 네트워크 장애 정보, 시스템 장애 정보, 응용 서비스 장애 정보 계층으로 구분되어 표현되어지고, 가장 상위에는

KQML로 구성된 통신 계층으로 구성된다.

에이전트가 특정 도메인에 관한 질의를 하거나 통신하고자 할 때, 특정 도메인에 대한 공통적 어휘 기반을 제공해야 하며 이를 온톨로지(Ontology)라고 한다. 즉, 이기종 시스템과의 정확하고 분명한 정보 교환을 위해서는 온톨로지에 대한 개발이 선행되어야 한다. 장애 진단 에이전트 시스템과 관련해서도 먼저 장애 항목과 장애 정보들에 대한 정의가 먼저 표준화되어야 하며 이와 같은 장애 관련 온톨로지를 개발하는 것은 매우 쉽지 않은 일이다. 따라서, 현재 본 시스템에서 정의된 장애 관련 온톨로지는 다른 많은 사람들과의 협력하에 개발되어야 하며 이를 표준화하는 것이 요구된다.

4. 장애 진단 규칙

4.1 충돌 감지 규칙

LAN에서 가장 많이 사용되는 이더넷(ethernet)과 같은 CSMA/CD(Carrier Sense Media Access/Collision Detection)방식으로 패킷을 상호 전송하는 네트워크에서는 네트워크 상의 노드 수와 전송 패킷이 증가함에 따라 충돌이 급격히 증가한다. 충돌이 발생했을 경우, truncated binary exponential back-off 랜덤화 알고리즘에 의해 재전송이 일어나는데, 이는 전송 지연 시간을 발생시킴으로써 사용자의 응답 시간을 증가시키는 결과를 낳는다[11]. 따라서, 충돌율은 LAN 상의 트래픽 장애 발생을 일으키는 중요한 원인중의 하나이며, 네트워크의 장애 여부를 감지하는 장애 진단 파라미터로 사용될 수 있다. 즉, 충돌율을 산출함으로써 관리 네트워크 장애 여부를 감지할 수 있으며 이미 논문[12]에서 RMON MIB을 이용한 충돌율 산출 알고리즘을 보이었다. 아래의 (그림 5)에서는 이를 기반으로 충돌율을 계산하고 이를 임계치 값과 비교함으로써 장애 발생 여부를 진단하는 충돌 감지 규칙을 제시하였다.

4.2 에러 감지 규칙

충돌과 함께 중요한 장애 진단 파라미터로 이용될 수 있는 것이 에러율이다. RMON MIB을 이용하여 에

```
while(t  T)
{
    i = t;
    while(i  n)
    {
            σp(i)=etherStatsPkts(i)-etherStatsPkts(i-Δ),
            σcol(i)=etherStatsCollision(i)-etherStatsCollision(i-Δ),
            μcol(i)=( σcol(i)/( σp(i)- σcol(i)) )×100,

            if( μcol(i) ≥ colth) then γcol(t)= γcol(i)+1;
    I = i +Δ;
    }
    if ((( γcol(t+n) - γcol(t)) / n) ≥ colp)
            then SYSTEM_LOCATION.request;
    else
             then COLLISION_DETECTION(COL).request;
    t=t+1;
}
```

$\sigma_p(i)$: 시간[i , i-Δ] 폴링 사이의 전체 패킷 수
$\sigma_{col}(i)$: 시간[i , i-Δ] 폴링 사이의 충돌 패킷 수
$\mu_{col}(i)$: 시간[i , i-Δ] 사이의 충돌율
$\gamma_{col}(i)$: 진단 시간 동안에 충돌율이 임계값을 넘어선 횟수
col_{th} : 충돌율의 임계값
col_p : 진단 시간 동안 임계값을 넘어선 허용 가능한 확률
T : 전체 시간
n : 진단 시간
σ 진단 간격

(그림 5) 충돌 감지 규칙

```
While(t ≤ T)
{
    i = t;
    while(i ≤ n)
    {
            σp(t)=etherStatsPkts(t)-etherStatsPkts(t-Δ);
            σcc(t)=etherStatsCRCAlignErrors(t)-etherStatsCRCAlignErrors(t-Δ);
            σue(i)=etherStatsUndersizePkts(t)-etherStatsUndersizePkts(t-Δ),
            σoe(i)=etherStatsOversizePkts(t)-etherStatsOversizePkts(t-Δ);
            σje(i)=etherStatsJabbers(t)-etherStatsJabbers(t-Δ),
            σfe(i)=etherStatsFragments(t)-etherStatsFragments(t-Δ);
            μerr(i)=(( σcc(t)+ σue(t)+ σoe(t)+ σje(t)+ σfe(t))/( σp(t)) ×100,

            if( μerr(i) ≥ errth) then γer(i)= γcol(t)-1;

            i = i + Δ,
    }
    if ((( γerr(t+n) - γerr(t)) / n) ≥ errp)
            then SYSTEM_LOCATION request,
    else
            then ERROR_DETECTION(ERR) request,
    t=t+1;
}
```

$\sigma_p(t)$: 시간[i , i-Δ] 폴링 사이의 전체 패킷 수
$\mu_{err}(i)$: 시간[i , i-Δ] 사이의 에러율
$\gamma_{err}(i)$: 진단 시간 동안에 충돌율이 임계값을 넘어선 횟수
err_{th} 에러율의 임계값
err_p 진단 시간 동안 임계값을 넘어선 허용 가능한 확률
T : 전체 시간
n 진단 시간
Δ : 진단 간격

(그림 6) 에러 감지 규칙

러율을 계산할 수 있으며, 일반적으로 계산된 에러율의 적정성 여부는 NIST(National Institute of Standard and Technology)가 제시한 ANSI X3.102 권고안 기준에 따르면 2%가 넘었을 때는 경고 수준으로, 5%가 넘었을 경우는 위험수준으로 인지하게 된다 에러율에는 CRC에러, 비유효 프레임 크기 에러, 그리고 이 두 가지의 복합적인 에러들을 모두 포함한 결과를 의미하여, 이런 다양한 종류의 프레임 에러는 여러 가지 원인에 의해 발생될 수 있는데, 선로의 품질 상태, 네트워크 하드웨어의 오동작, 케이블 길이 초과 등이 그 주 원인이 된다[13]. 이러한 에러 프레임은 실제적으로 네트워크 전체에 영향을 미칠 수 있으며 또한, 사용자 관점에서는 재전송으로 인한 응답 시간을 지연시키는 결과를 가져온다. RMON MIB에서는 에러 프레임의 원인에 따라 각 오브젝트들을 제공하고 있으며, 다음의 (그림 6)은 이를 기반으로 한 에러 감지 규칙이다.

4.3 브로드캐스트 감지 규칙

브로드캐스트 패킷은 세그먼트 상의 모든 노드들에게 전송되어 시스템의 성능에 크게 영향을 미친다. 따라서, 급격한 브로드캐스트 패킷의 증가는 네트워크와 시스템의 성능을 크게 떨어뜨리게 되며 최악의 경우, 브로드캐스트 패킷의 급격한 증가로 발생되는 브로드캐스트 스톰(broadcast storm)은 네트워크 전체를 다운시키는 영향을 끼치게 된다[14]. 실제로, MAC 계층을 통해 상위 계층으로 전달되는 브로드캐스트 패킷은 CPU성능에 따라 처리 시간에 큰 지연을 초래할 수 있고, 이를 통해 사용자의 응답 시간의 지연은 크게 증가하게 된다. 일반적으로 브로드캐스트 패킷이 초당 20~30개 이상 발생했을 경우에는 네트워크의 성능에 장애를 발생시키게 되며, 이를 측정함으로써 네트워크의 또 다른 종류의 장애 요소를 감지할 수 있다[14]. (그림 7)에서는 브로드캐스트 감지 규칙을 나타내었다.

```
while(t ≤ T)
{
    i = t;
    while (i ≤ n)
    {
            δt (t)=sysUpTime(t)-sysUpTime(t-Δ);
            σbr (i)=etherStatsBroadcastPkts(t)-etherStatsBroadcastPkts(t-Δ);
            μbr (i)=( σbr(i) / δt(t));

            if( μbr (i) ≥ brth) then γbr(t)= γbr(i)+1;
    i = i + Δ;
    }
    if ((( γbr(t-n) - γbr(t)) / n) ≥ brp)
            then SYSTEM_LOCATION.request;
    else
            then BROADCAST_DETECTION(BR) request;
    t=t+1;
}
```

$\delta_t(t)$: 시간[t , $t-\Delta$] 사이의 시간 변화량
$\sigma_{br}(i)$: 시간[i , $i-\Delta$] 사이의 브로드캐스트 패킷 수
$\mu_{br}(i)$: 초당 브로드캐스트 패킷 수
$\gamma_{br}(t)$: 진단 시간 동안에 초당 브로드캐스트 패킷 수가 임계값을 넘어선 횟수
br_{th} 초당 브로드캐스트 패킷 수의 임계값
br_p : 진단 시간 동안 임계값을 넘어선 허용 가능한 확률
T 전체 시간
n : 진단 시간
Δ : 진단 간격

(그림 7) 브로드캐스트 감지 규칙

4.4 시스템 위치 탐지 규칙

시스템 장애 진단 계층은 네트워크 진단 계층의 충돌 감지 규칙, 에러 감지 규칙 그리고, 브로드캐스트 감지 규칙을 통해 감지된 장애의 원인이 되는 호스트의 위치를 확인하는 기능을 수행한다 실제로 네트워크 상의 다양한 시스템들이 복잡하게 연결된 LAN 상에서 이와 같은 장애 시스템의 위치를 파악하는 것은

```
if(primitives==SYSTEM_LOCATION(COL))
    Set hostTopNRateBase = hostTopNOutPkts;
else if(primitives=SYSTEM_LOCATION(ERR))
    Set hostTopNRateBase = hostTopNOutErrors;
else if(primitives=SYSTEM_LOCATION(BR))
    Set hostTopNRateBase = hostTopNOutBroadcastPkts;

while(j ≤ hn)
{
   σhpn(k)= σhpn(k)+hostTopNRate(k);
   k = k + 1;
}

while(k ≤ hn)
{
   μhpn(k)= (hostTopNRate(k) / σhpn(hn))×100;

   if( μhpn(k)>hpnp)
           then APPLICATION_LOCATION.request;
   k = k + 1;
}
```

$\sigma_{hpn}(i)$: 진단 시간 동안 발생된 호스트의 순위별 패킷 수
$\mu_{hpn}(i)$: 각 호스트가 발생시킨 트래픽 비율
h_n : 순위별 감지 호스트의 수
hpn_p : 허용 가능한 호스트 트래픽 비율

(그림 8) 시스템 위치 탐지 규칙

쉽지 않다. 그렇기 때문에, 세그먼트 상의 충돌, 에러 그리고, 브로드캐스트 패킷을 가장 많이 발생시킨 호스트들을 파악하기 위하여 RMON MIB상의 hostTopN 그룹이 이용할 수 있고, (그림 8)에서는 이를 기반으로 한 시스템 위치 탐지 규칙을 나타냈다.

4.5 응용 서비스 진단 규칙

응용 서비스 장애 진단 계층은 시스템 장애 진단 계층에서 탐지된 장애 발생 시스템이 실제로 어떤 응용 프로토콜을 발생시키는지를 진단하는 계층이다. 이를 통해 특정 호스트 상의 응용 프로그램의 오동작 여부를 판단할 수 있으며, 결과적으로 사용자 계층의 장애 요인을 진단할 수 있게 된다.

특정 시스템이 발생시키는 응용 서비스들을 모니터링하기 위해서는 RMON MIB 상의 filter 그룹과 capture 그룹을 이용하여 그 통계량을 분석할 수 있다[12] (그림 9)의 응용 서비스 탐지 규칙에서는 미리 설정된 응용 서비스들에 대한 패킷 수를 channelMatches 오브젝트를 통해 계산할 수 있고, 이를 근거로 충돌, 에러 또는 브로드캐스트 장애를 가진 호스트가 가장 많이 발생시키는 응용 서비스를 구할 수 있다.

```
While(m ≤ pro_n)
{
    σ_pro(m)= σ_pro(m)-channelMatches(m);
    m = m + 1;
}

while(m < pro_n)
{
    μ_pro(m)= channelMatches(m) / n;

    if(m=1) then
    {
            if( μ_hum(m) ≥ μ_hum(m+1)) then A(m) = μ_hum(m);
            else
                    then A(m) = μ_hum(m-1);
    }
    else then
            if( μ_hom(m) ≥ A(m)) then A(m) = μ_hom(m);

    m = m - 1;
}
AppLocationEvent(A(m));
```

$\sigma_{pro}(m)$: 감지된 프로토콜 별 패킷 수
$\mu_{pro}(m)$: 진단 시간 동안의 프로토콜 별 패킷 수
pro_n : 응용 프로토콜의 개수
n : 진단 시간
$A(m)$: 가장 많은 m번째 프로토콜의 패킷 수

(그림 9) 응용 서비스 탐지 규칙

5. 실험 및 고찰

본 논문에서 제시한 장애 진단 모델의 적합성을 실험 및 평가하기 위해 실제 성균관 대학교 LAN 세그먼트 상에 Bay 상의 RMON 에이전트인 SA NMM을 탑재한 Baystack 허브를 설치하여 관리 정보들을 수집하였고, 이에 따라 정의된 장애 진단 모델 규칙들을 적용시켰다.

[실험 환경]

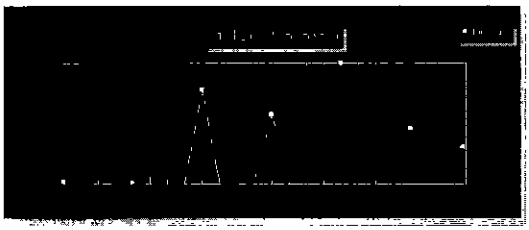
(1) 성균관 대학교 LAN상의 일부 세그먼트(203.252.53.0)를 대상으로 하였고, 그 속도는 10Mbps이다.

(2) 이 세그먼트 상에 존재하는 RMON 에이전트(203.252.53.57)를 탑재한 허브를 이용하여 관리 정보들을 주기적으로 폴링하여 모니터링하였고, 그 기간은 1999년 4월 14일부터 1999년 6월 13일까지 두 달간이다.

(3) 주기적인 모니터링은 Solaris 2.5상의 crontab을 이용하였고, 그 주기는 10간격이었다.

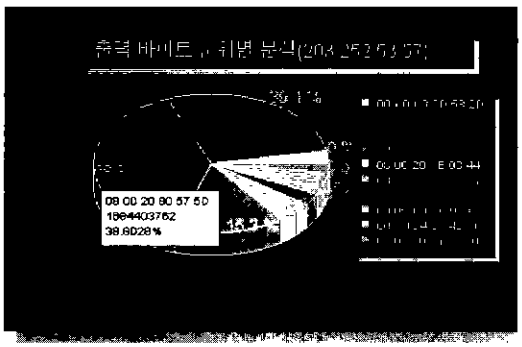
다음은 실험 기간 동안 본 논문에서 제시한 장애 진단 모델을 실험 환경에 적용시켜 장애를 감지하고 그 원인을 분석한 예를 나타내었다 (그림 10)과 (그림 11)은 네트워크 장애 진단 계층의 충돌과 에러 진단 규칙을 토대로 성균관 대학교 내부 망인 203.252.53.0의 각 시간대별로 트래픽의 유형을 모니터링 한 결과를 나타낸다. (그림 10)에서 나타난 바와 같이 충돌율은 14시와 15시 사이에서 급격하게 증가하여 계속 지속되는 것을 볼 수 있으며, 이는 일반적인 LAN 상의 허용 가능한 충돌율 임계값을 훨씬 넘어선 수치임을 알 수 있다 또한, (그림 11)의 에러율의 경우는 8시와 12시 경에 이미 임계치를 넘어섰지만, 14시 이후에 마찬가지로 급격히 증가하여 장애 상태가 계속 유지되는 것을 파악할 수 있다.

(그림 10) 충돌 탐지 규칙의 적용 예



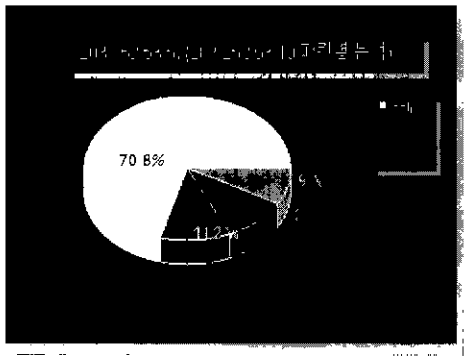
(그림 11) 에러 탐지 규칙 적용 예

네트워크 장애 진단 계층에서 감지된 장애 유형을 볼 때, 세그먼트 상의 트래픽이 굉장히 많이 증가했다는 것을 추측할 수 있다. 따라서, 시스템 장애 진단 계층에서는 트래픽을 가장 많이 발생시킨 시스템들을 통해 그 장애 위치를 파악할 수 있다 (그림 12)는 시스템 위치 탐지 규칙을 통해 203.252.53.0 세그먼트에서 장애가 발생한 시간 동안에 트래픽을 가장 많이 발생시킨 호스트들의 상위 10개를 분석한 결과이다. 그림에서 보는 바와 같이 두개의 탐지된 시스템(00 : A0 : C9 : 2D : 58 : 2D, 00 : 00 : A2 : CB : 29)이 전체 트래픽의 약 80%정도를 차지하고 있는 것을 분석할 수 있다. 실제로 세그먼트 상의 두 시스템은 각각 메일 및 ftp서버(IP주소는 203.252.53.10)와 라우터(IP주소는 203.252.53.1)로 동작하고 있었다. 라우터는 일반적으로 외부 네트워크에서 로컬 네트워크로 사용자의 요구에 따라 많은 트래픽을 발생시키고 있기 때문에 메일 및 ftp 서버에 장애 여부를 의심할 수 있었다.

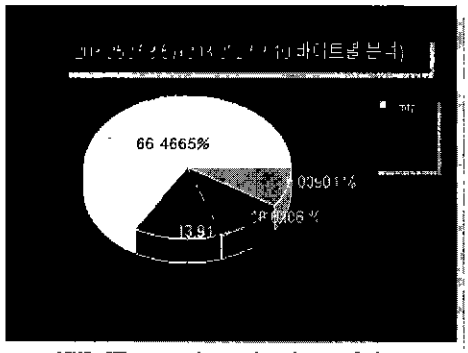


(그림 12) 시스템 위치 탐지 규칙의 적용 예

(그림 13)의 (a)와 (b)는 시스템 장애 진단 계층에서 탐지된 메일 및 ftp 서버를 응용 서비스 진단 계층의 응용 서비스 탐지 규칙을 통해 분석한 바이트 및 패킷 분석의 결과를 나타낸다. 그림에서 보는 바와 같이 바이트별 분석은66% 이상을, 그리고 패킷별 분석은 70% 이상을 SMTP가 차지하고 있는 것을 볼 수 있고 이는 정상적으로 동작할 경우의 SMTP 트래픽을 훨씬 초과하는 수치이다. 이를 통해 203.252.53.0 세그먼트의 메일 및 ftp 서버(IP 주소 203.252.53.10)의 SMTP 프로토콜이 전체 세그먼트에 큰 장애의 원인임을 추측할 수 있었고, 실제 조사해 본 결과, 해당 서버의 sendmail 설정이 잘못되어 비유효 패킷을 계속적으로 발생하고 있는 것으로 확인되었다



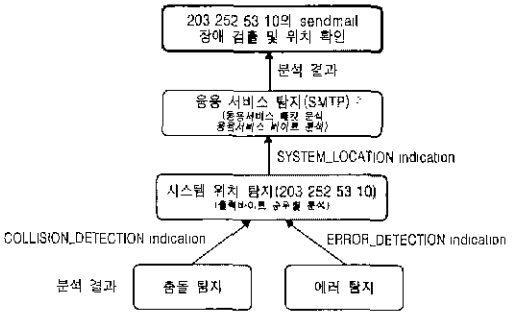
(a) 응용 서비스 패킷 분석 예



(b) 응용 서비스 바이트 분석 예

(그림 13) 응용 서비스 탐지 규칙의 적용 예

결과적으로 (그림 14)와 같은 계층 규칙을 거쳐 실제로 LAN 상에서 감지하기 힘든 LAN 장애 요인을 검출하고 위치 확인함으로써 관리자는 문제점을 쉽게 해결할 수 있었다



(그림 14) 탐지 규칙 적용 단계 및 실험 결과

6. 결 론

본 논문에서는 LAN 상의 장애 관리의 중요성을 인지하고, 장애 관리를 체계적으로 수행하기 위해 LAN 관리에 적합한 RMON MIB을 기반으로 장애 진단 계층 모델을 확립하였고, 현재 차세대 패러디임으로 주목 받고 있는 에이전트 개념을 장애 진단 계층 모델에 적용시켜 장애 진단 에이전트 시스템의 구조를 설계하였다. 또한, 각 에이전트들이 포함하는 장애 진단 계층 상의 장애 진단 규칙들을 제시하고 이를 활용하는 예를 보이기 위해 실제 환경에서의 진단 결과를 보였다.

장애 진단 계층 모델은 충돌 감지 규칙, 에러 감지 규칙, 브로드캐스트 감지 규칙 등을 포함하는 네트워크 진단 계층, 시스템 위치 탐지 규칙을 통해 네트워크 상의 장애 발생 시스템 위치를 탐지하는 시스템 장애 진단 계층 그리고, 응용 서비스 탐지 규칙을 통해 장애 시스템 상의 응용 서비스들을 모니터링하여 장애를 유발하는 응용 프로토콜을 진단하는 응용 서비스 장애 진단 계층으로 구분하여 네트워크 상의 장애를 검출하고 그 위치를 확인할 수 체계화 하였다. 또한, 이들 규칙의 집합이 서로 연관성을 갖도록 에이전트 시스템의 구조와 상태 천이를 기술하고 있다 제시된 규칙의 타당성과 적용성을 검증하기 위하여 실제로 발생했던 LAN 장애 상황 정보를 제시된 규칙에 따라 모니터링하고 분석하여 실제 장애 시스템을 찾는 과정을 나타내었다. 이러한 규칙 기반 에이전트 시스템은 인터넷 관리자에게 망 정보 수집에서 장애 판단 및 원인 해결까지 큰 도움을 줄 것으로 기대된다.

차후 본 논문과 관련된 연구로서 본 논문에서 제시한 장애 진단 규칙과 장애 진단 에이전트 모델을 기반으로 장애 진단 에이전트 시스템을 구현할 것이며 이는 인터넷 관리자를 대신하는 지적 능력을 가진 대리인으로서의 한 단계 높은 수준의 관리 행위를 가능하게 할 것이다.

참 고 문 헌

[1] Allan Leinwand, "Accomplishing Performance Management with SNMP," INET'93, pp.CEA-1-CEA-5, 1993.

[2] William Stallings, "SNMP, SNMPv2, and RMON : Practical Network Management," Addison-Wesley Publishing Company, 1996

[3] 유승근, 안성진, 정진욱, "SNMP MIB-II를 이용한 인터넷 관리 시스템의 웹 인터페이스 설계 및 구현", 정보처리학회논문지, 제6권 제3호, pp.699-709, 1999.

[4] Jeong-Soo Han, Seong-Jin Ahn, Jin-Wook Chung, "Web-based Performance Manager System for a Web Server," Network Operations And Management Symposium '98, 1998

[5] http://java.sun.com/products/JavaManagment/

[6] S. Amarnath, Anurag Kumar, "A New for Link Utilization Estimation in Packet Data Networks using SNMP Variables," Globecom '97, Vol.1, 1997.

[7] Cynthia S. Hood, Chuanyi Ji, "Intelligent Agents for Proactive Fault Detection," IEEE Internet Computing, Vol.2, No 2, 1998.

[8] 안성진, 정진욱, "SNMP MIB-II를 이용한 인터네트 분석 파라미터 계산 알고리즘에 관한 연구", 정보처리학회, 제5권 제8호, pp.2102-2116, 1998.

[9] 최중민, "에이전트의 개요와 연구방향", 정보과학회지, 제15권 제3호, pp7-16, 1997.

[10] 이은석, 이진구, 강제연, "인터넷 상에서의 전자상거래를 위한 멀티에이전트 시스템", 정보처리학회지, Vol.4 No.5, 1997

[11] Uyless Black, "Protocols, Standards, and Interfaces." Prentice Hall, 1993

[12] K. H. Cho, "Algorithms to Calculate LAN Perfor-

mance and Fault Analysis Parameters using RMON MIB," The Graduate School of Sung Kyun Kwan University, 1999.

[13] Allan Leinwand, Karen Fang Conroy, "Network Management," Addison-Wesley, 1996.

[14] John Blommers, "Practical Planning for Network Growth," Prentice Hall PTR, 1996.

[15] J. S. Baras, M. Ball, S. Gupta, P Viswanathan, and P Shah, "Automated Network Fault Management," MILCOM 97 Proceedings, Vol.3, 1997.

[16] Irene Katzela, Mischa Schwartz, "Schemes for Fault Identification in Communication Networks," IEEE/ACM Transactions on networking Vol.3, No.6, 1995. 11.

조 강 홍

e-mail : khcho@songgang.skku.ac.kr

1997년 성균관대학교 정보공학과 졸업(학사)

1999년 성균관대학교 전기전자 및 컴퓨터공학부 대학원 졸업(석사)

1999년~현재 성균관대학교 전기전자 및 컴퓨터 공학부 대학원 박사과정

관심분야 : 네트워크 관리, 트래픽 분석

안 성 진

e-mail : sjahn@comedu.skku.ac.kr

1988년 성균관대학교 정보공학과 졸업(학사)

1990년 성균관대학교 대학원 정보공학과 졸업(석사)

1990년~1995년 한국전자통신연구원 연구 전산망 개발실 연구원

1996년 정보통신 기술사 자격 취득

1998년 성균관대학교 대학원 정보공학과 졸업(박사)

1999년~현재 성균관대학교 컴퓨터교육과 전임강사

관심분야 : 네트워크 관리, 트래픽 분석, Unix 네트워킹

정 진 욱

e-mail : jwchung@songgang.skku.ac.kr

1974년 성균관대학교 전기공학과 학사

1979년 성균관대학교 대학원 전자공학과 석사

1991년 서울대학교 대학원 계산통계학과 박사

1982년~1985년 한국과학기술 연구소 실장

1981년~1982년 Racal Milgo Co. 객원연구원

1985년~현재 성균관대학교 전기전자 및 컴퓨터공학부 교수

관심분야 : 컴퓨터 네트워크, 네트워크 관리, 네트워크 보안