LAN 장애 검출 및 위치 확인 알고리즘에 관한 연구

조규억*, 안성진**, 정진욱*
*성균관대학교 전기전자 및 컴퓨터공학부
**성균관대학교 컴퓨터교육과
e-mail : kojoe@songgang.skku.ac.kr

A Study on Algorithm for Fault Detection and Location on LAN

Kyu-Oak Joe*, Seongjin Ahn**, Jin-Wook Chung*
*Dept. of Electric, Electronic and ComputerEngineering, SungKyunKwan University
**Dept. of Computer Education, SungKyunKwan University

요 약

본 논문에서는 LAN 상에서의 장애 검출과 장애를 유발시킨 호스트에 대한 위치 확인을 통하여 보다 효율적인 네트워크 장애 관리가 이루어질 수 있도록 하기 위한 알고리즘을 제시하고 있다. LAN 상에서의 주된 장애로 충돌율에 의한 네트워크 지연 장애를 들 수 있다. 따라서 패킷 충돌 감지를 통하여 네트워크 지연을 감지하고 이를 통하여 브로드캐스트, 멀티캐스트, 에러 발생 패킷 등을 분석하여 충돌율을 많이 발생시킨 호스트를 찾는다. 패킷을 많이 발생시킨 호스트에 대하여 RMON2를 이용하여 응용 프로토콜의 종류를 파악하고 이에 대한 내용을 관리자에게 통보함으로써 네트워크 장애 관리를 효율적으로 수행할 수 있다.

1. 서론

최근 컴퓨터 통신 기술이 발달함에 따라 네트워크는 점차 방대하고 복잡해지게 되었고 이에 따른 네트워크 장애는 피할 수 없는 대상이 되었다. 그러나 신속한 장애 검출과 진단은 네트워크 장애로 인한 심각한 문제를 해결하고 네트워크의 신뢰성을 향상시키는데 중요한 요소가 될 수 있다[1].

특히 LAN 상의 응용 프로그램들의 사용 증가는 트래픽의 병목 현상을 보이는 WAN 뿐만 아니라, LAN 상의 트래픽 양을 크게 증가시키게 되었고, 네트워크 전체의 지연을 초래하게 된다. 따라서 LAN 상의 장애 원인의 신속한 검출 및 진단 없이 네트워크 장비에 대한 과다한 투자를 통하여 장애를 해결하려고 한다면 효율적인 네트워크 관리를 위한 해결책이 될 수 없다. LAN 상의 효율적인 장애 관리는 장비의 투자 및 유지 보수 비용을 낮추고 성능을 최대로 유지하는 것이기 때문이다[3][4].

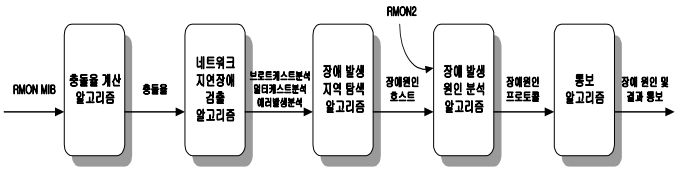
LAN 상의 장애 관리의 필요성이 증대됨에 따라 네트워크 장애에 대한 정확한 정의를 통하여 네트워크 장애에 효율적으로 대처하는 방안이 제안되고 있다. LAN 상의 장애에는 크게 하드웨어 장애와 소프트웨어 장애가 있다. 하드웨어 장애로는 전력 장애, 네트워크 케이블의 물리적인 단절, 라우터나 게이트웨이와 같은 주요 네트워크 구성 요소들의 장애 등이 있다. 이와 같은 하드웨어 장애는 검출하는데 그리 어렵지 않다. 그러나 소프트웨어 장애는 장애에 대한 정의가 명확하지 않기 때문에 장애를 검출하고 진단하는데 어려움이 있다. 일반적으로 알려진 소프트웨어 장애로는 네트워크의 성능 저하나 네트워크 대역폭의 손실 등이 있다[2].

본 논문에서는 네트워크 성능 저하나 대역폭의 손실에 따른 LAN 상의 장애를 검출하고 장애 발생 지역을 확인함으로써 LAN 상의 장애 관리에 대해 보다 효율적으로 대처하고자 한다. 이를 위해 우선 LAN 상의 네트워크 성능 저하나 대역폭의 손실에 직접적인 영향을 미치는 충돌(Collision)율을 파악한다. 충돌율이

임계값을 넘을 경우 장애가 발생한 것으로 파악하고 장애 검출 알고리즘에 따라 장애를 검출하고 장애 발생 지역을 확인하여 관리자에게 통보하는 알고리즘 시스템을 설계하였다.

2. 네트워크 지연 장애 검출 알고리즘

LAN 상의 트래픽 장애를 유발시키는 요인으로 패킷 충돌을 들 수 있다. LAN 상의 충돌율이 높아지면 네트워크 지연이 발생하므로 충돌이 발생하는 원인을 찾아서 이를 해결해야 한다. 따라서 알고리즘은 다음과 같은 순서로 충돌이 많이 발생하는 원인을 검출해 간다. 전체 알고리즘 모델은 [그림 1]과 같다.



[그림 1] 패킷 충돌 장애 검출 알고리즘 모델

이와 같은 패킷 충돌 장애 검출 모델에 RBR(Rule Based Reasoning)을 적용하여 단계적으로 장애의 원인을 찾아내고 이에 대한 결과를 통보한다.

2.1 충돌율 계산 알고리즘($Å_{col}$)

우선 LAN 상의 충돌율을 주기적으로 모니터링 한다.
충돌율은 세그먼트 상의 전체 패킷 수에 대한 패킷 충돌 값을 백분율로 계산한 값으로 RMON probe 에 의해 설정된 특정 인덱스 번호 I_{row} 를 이용하여 폴링한 관리 MIB 변수 etherStatsCollision, etherStatsPkts 을 누적하여 계산한다. 계산 과정은 다음과 같다[5].

과정 1) 초기 MIB 변수 폴링
관련 MIB 변수 집합
V{etherStatsPkts, etherStatsCollision}를 I_{row} 를 이용하여 폴링

과정 2) 비교 값 설정
S_p에 etherStatsPkts, S_{col}에 etherStatsCollision 설정

과정 3) 다음 MIB 변수 폴링
다음 폴링 변수 집합
V′{etherStatsPkts, etherStatsCollision}를 I_{row} 를 이용하여 t_p 시간 간격으로 폴링하여, 폴링 횟수인 n_p 증가

과정 4) 세그먼트 상의 전체 패킷량 누적
i 번째 폴링에 대해 세그먼트 상의 전체 패킷량을 누적

$$\sigma_p = \sum_{i=1}^{n_p} (etherStatsPkts_i - S_p)$$

과정 5) 충돌 패킷량 누적
각 i 번째 폴링에 대해 충돌 패킷량을 누적하고 종료 시에는 과정 6)으로 이동하고 아니면 과정 2)로 이동

$$\sigma_{col} = \sum_{i=1}^{n_p} (etherStatsCollision_i - S_{col})$$

과정 6) 최종 충돌율 계산

$$\mu_{col} = \frac{\sigma_{col}}{\sigma_p \sigma_{col}} \times 100$$

이와 같은 과정을 통하여 주기적으로 충돌율(μ_{col})을 구하고 LAN 관리 시스템이 이를 주기적으로 모니터링 한다.

2.2 네트워크 지연장애 검출 알고리즘($Å_{det}$)

모니터링은 충돌율(μ_{col})에 대한 임계값 T_p 를 가지고 충돌율과 비교하여, 충돌율이 T_p 를 초과할 때에 패킷 충돌 증가에 의한 네트워크 지연 장애를 발견한다. 다음과 같은 규칙을 기반으로 장애를 발견한 후 장애를 유발시킨 원인을 찾는다.

$\mu_{col} > T_p \rightarrow$ 네트워크 지연 장애

패킷 충돌 증가에 의한 네트워크 지연 장애를 유발시킨 가장 큰 원인은 특정 호스트에서 특정 어플리케이션에 대한 트래픽을 과다하게 발생시키는 경우이다. 따라서 장애를 유발시킨 호스트와 그 호스트가 발생시키는 패킷의 유형을 파악해야 한다.
패킷의 유형 중 브로드캐스트나 멀티캐스트 패킷이 세그먼트 상에 지나치게 많이 존재할 경우 대역폭이 낭비되고 수신할 필요가 없는 시스템들이 모두 수신하게 되므로 시스템의 프로세싱 기능을 낭비하게 된다. 또한 에러 패킷의 경우 브로드캐스트나 멀티캐스트 패킷에 속하지 않으므로 이에 대한 트래픽 비율을 분석할 필요가 있다[5]. 분석 과정은 다음과 같다.

과정 1) 초기 RMON MIB 변수 폴링
관련 RMON MIB 변수 집합
V{ hostOutBroadcastPkts, hostOutMulticastPkts, hostOutErrors}를 I_{row} 와 호스트 MAC 주소인 h_{mac} 을 이용하여 폴링
h_{bp} = hostOutBroadcastPkts
h_{mp} = hostOutMulticastPkts
h_{ep} = hostOutErrors 로 설정

과정 2) 브로드캐스트 순위별 분석
호스트가 발생하는 브로드캐스트 패킷량을 누적, 즉 i 번째 폴링에 대해 호스트가 발생하는 브로드캐스트 패킷량 누적

$$b_p = \sum_{i=1}^{n_p} (hostOutBroadcastPkts_i - h_{bp})$$

상위 n 개 호스트의 브로드캐스트 패킷량
$= \{(h_{b1}, b_1), (h_{b2}, b_2),\ldots, (h_{b3}, b_n)\}$

과정 3) 멀티캐스트 순위별 분석

호스트가 발생하는 멀티캐스트 패킷량을 누적, 즉 i 번째 폴링에 대해 호스트가 발생하는 멀티캐스트 패킷량 누적

$$m_p = \sum_{i=1}^{n_p} (hostOutMulticastPkts_i - h_{mp})$$

상위 n 개 호스트의 멀티캐스트 패킷량
$= \{(h_{m1}, m_1), (h_{m2}, m_2),\ldots, (h_{m3}, m_n)\}$

과정 4) 에러 발생 순위별 분석

호스트가 발생하는 에러 발생 패킷량을 누적, 즉 i 번째 폴링에 대해 호스트가 발생하는 에러 발생 패킷량을 누적

$$e_p = \sum_{i=1}^{n_p} (hostOutErrors_i - h_{ep})$$

상위 n 개 호스트의 에러 발생 패킷량
$= \{(h_{e1},e_1), (h_{e2} ,e_2),\ldots, (h_{e3} ,e_n)\}$

이와 같이 브로드캐스트, 멀티캐스트, 에러발생 패킷에 대한 순위별 분석을 통하여 충돌율을 증가시키는 호스트를 찾기 위한 기반을 마련한다.

2.3 장애 발생 지역 탐색 알고리즘($Å_{src}$)

브로드캐스트, 멀티캐스트, 에러발생 패킷에 대한 순위별 분석을 기반으로 패킷을 많이 발생하는 각 호스트가 LAN 세그먼트 상의 전체 패킷에 미치는 영향을 파악한다. 그리고 LAN 상에 심각한 영향을 미치는 값에 대해서 그 값을 유발시킨 호스트를 찾아낸다. LAN 세그먼트 상의 전체 패킷을 나타내는 RMON MIB 변수는 etherStatsPkts 이다.

ehterStatsPkts = T_N 라고 하면

각 패킷 종류에 대한 상위 n 개의 호스트가 발생하는 패킷이 LAN 상의 전체 패킷에서 차지하는 비율을 계산한다.

과정 1) 브로드캐스트 패킷율

상위 n 개의 호스트에 대하여 LAN 세그먼트 상의 전체 패킷량에 대한 각 호스트의 브로드캐스트 패킷량의 비를 구한다.

$B_{p1} = b_1 / T_N$
$B_{p2} = b_2 / T_N$
:
$B_{pn} = b_n / T_N$

호스트 h_n 에 대한 브로드캐스트 패킷율 중 상위 n 개을 선택한다.

상위 n 개의 호스트에 대한 브로드캐스트 패킷율
$= \{(h_{b1}, b_1, B_{p1}), (h_{b2}, b_2, B_{p2}),\ldots, (h_{bn}, b_n, B_{pn})\}$

과정 2) 멀티캐스트 패킷율

상위 n 개의 호스트에 대하여 LAN 세그먼트 상의 전체 패킷량에 대한 각 호스트의 멀티캐스트 패킷량의 비를 구한다.

$M_{p1} = m_1 / T_N$
$M_{p2} = m_2 / T_N$
:
$M_{pn} = m_n / T_N$

호스트 h_n 에 대한 멀티캐스트 패킷율 중 상위 n 개을 선택한다.

상위 n 개의 호스트에 대한 멀티캐스트 패킷율
$= \{(h_{m1}, m_1, M_{p1}), (h_{m2}, m_2, M_{p2}),\ldots, (h_{mn}, m_n, M_{pn})\}$

과정 3) 에러 발생 패킷율

상위 n 개의 호스트에 대하여 LAN 세그먼트 상의 전체 패킷량에 대한 각 호스트의 에러 발생 패킷량의 비를 구한다.

$\epsilon_{p1} = e_1 / T_N$
$\epsilon_{p2} = e_2 / T_N$
:
$\epsilon_{pn} = e_n / T_N$

호스트 h_n 에 대한 멀티캐스트 패킷율 중 상위 n 개을 선택한다.

상위 3 개의 호스트에 대한 에러 발생 패킷율
$= \{(h_{e1}, e_1, \epsilon_{p1}), (h_{e2}, e_2, \epsilon_{p2}),\ldots, (h_{en}, e_n, \epsilon_{pn})\}$

위와 같이 각 패킷에서 상위 n 개의 호스트가 발생하는 패킷이 LAN 상의 전체 패킷에서 차지하는 비율을 비교하여 특히 많은 패킷을 발생하는 호스트를 찾아낸다. LAN 세그먼트 상의 전체 패킷에서 특정 호스트가 발생하는 패킷이 차지하는 비율이 ρ_c(계산상수) 이상인 호스트를 찾아낸다.

2.4 장애발생 원인분석 알고리즘($Å_{anal}$)

장애 발생의 원인인 호스트에 대하여 특히 어떤 종류의 응용 프로토콜이 패킷 트래픽을 많이 발생시키는지를 찾는다. 이를 위해 여기서는 RMON2 에서 정의된 MIB 을 사용한다.

과정 1) 프로토콜 패킷의 종류 파악

Protocol Directory Group 의 관리 MIB 변수인 protocolDirID 를 구하여 현재 호스트에서 발생되는 프로토콜 패킷의 종류를 구한다. 그리고 같은 그룹의 관리 MIB 변수인 protocolDirHostConfig 을 *SupportedOn(3)*으로 설정함으로써 Network-Layer Host Group 과 Application-Layer Host Group 을 활성화시킨다. 여기서 검사하고자 하는 것은 장애가 발생한 호스트의 출력 패킷이므로 Network-Layer 와 Application-Layer Host Group 의 출력 패킷을 조사한다.

과정 2) Network-Layer 상의 프로토콜

Network-Layer Host Group 에서는 관리 MIB 변수인 nlHostOutPackets 을 조사한다. nlHostOutPackets 은 Network Layer 상의 프로토콜 중 이 호스트로부터

전송되는 에러 없는 프로토콜 패킷의 수(N_n)를 의미한다.

과정 3) Application-Layer 상의 프로토콜

Application-Layer Host Group 에서는 관리 MIB 변수인 alHostOutPkts 를 조사한다. alHostOutPkts 은 Application Layer 상의 프로토콜 중 이 호스트로부터 전송되는 에러 없는 프로토콜 패킷의 수(A_n)를 의미한다.

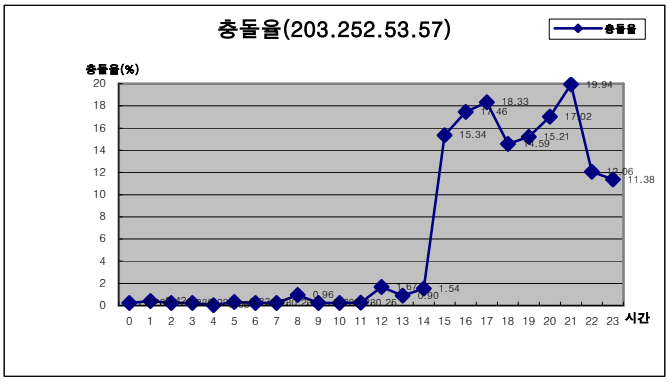
과정 4) 에러 발생 패킷

여기서 N_n 와 A_n 는 에러 패킷을 포함하고 있지 않기 때문에 RMON1 에서 정의된 Host Group 의 관리 MIB 변수인 hostOutErrors 를 조사하여 에러 패킷 발생 수(E_n)를 조사한다.

이와 같은 네트워크 지연 장애의 원인을 찾고 관리자 시스템에게 트래픽을 많이 발생시키는 프로토콜에 대한 정보를 그 호스트의 IP 주소와 함께 통보한다.

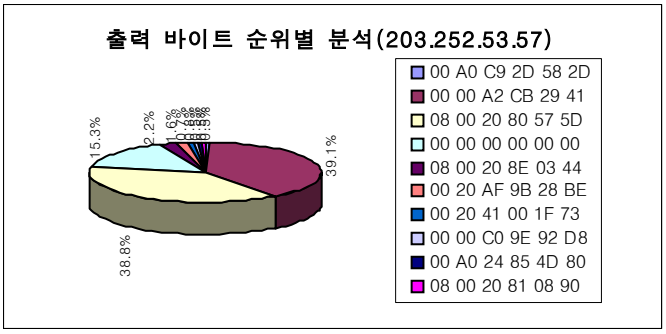
3. 실험 및 고찰

본 논문에서 제시한 장애 진단 알고리즘 모델을 실제 네트워크에 적용시켜 장애를 감지하고 그 원인을 분석한 예를 나타내었다.



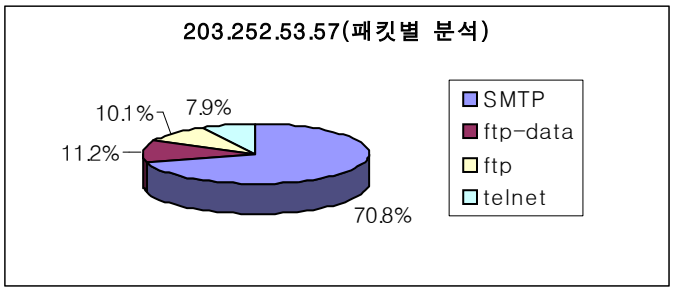
[그림 2] 장애 진단을 위한 충돌율 적용의 예

[그림 2]는 네트워크 장애 진단을 위한 충돌율을 성균관 대학교 내부 망인 203.252.53.0 의 각 시간대별로 트래픽의 유형을 모니터링 한 결과를 나타낸다.



[그림 3] 장애 발생 탐색 적용의 예

[그림 3]은 장애를 유발시킨 원인이 된 호스트를 찾기 위한 출력 바이트 순위별 분석을 나타내고 있다. 이를 통해 패킷을 많이 발생시킨 호스트들을 찾아낸다.



[그림 4] 응용 프로토콜 분석의 예

[그림 4]은 충돌율을 많이 유발시키는 응용 프로토콜에 대한 검출 결과를 나타낸다.

4. 결론

본 논문에서는 LAN 상의 트래픽 장애 관리를 효율적으로 하기 위한 네트워크 지연 장애 검출 알고리즘 모델을 설계하였다. 네트워크 지연 장애 검출을 위해 RMON MIB 을 이용하여 충돌율을 구하고, 이를 이용하여 장애를 감지하고 브로드캐스트, 멀티캐스트 및 에러 발생 패킷을 분석하여 장애 유발의 원인이 된 호스트의 위치를 확인하였다. 그리고 RMON2 를 이용하여 확인된 각 호스트들이 발생시키는 응용 프로토콜이 무엇인지를 파악하여 관리자에게 통보하는 전체 알고리즘을 설계하였다. 이러한 네트워크 지연 장애 검출 알고리즘은 최근의 복잡하고 방대해진 네트워크에서 발생할 수 있는 장애에 대하여 능동적으로 대처할 수 있도록 네트워크 정보 수집에서 장애 판단 및 원인 해결까지 체계적인 과정을 나타내었다. 따라서 이러한 네트워크 지연 장애 검출 알고리즘은 LAN 상의 트래픽 장애 발생시 효율적으로 대처할 수 있는 충분한 해결책을 제시해 줄 것으로 기대된다.

차후 본 논문에서 제시한 알고리즘에 RBR(Rule-Based Reasoning) 기반의 네트워크 장애 검출 규칙을 적용시켜 보다 체계적이고 지능적인 네트워크 지연 장애 관리 시스템을 설계할 것이고 이는 한 단계 높은 수준의 네트워크 장애 관리 시스템이 될 것이다.

참고문헌

[1] I. Rouvellou and G.W. Hart. “Automatic alarm correlation for fault identification”, In *Proc. IEEE INFOCOM*, pages 553-561, 1995

[2] R. Maxion. “A case study of Ethernet anomalies in a distributed computing environment”, IEEE transactions on Reliability, 39(4), Oct 1990

[3] Allan Leinwand, “Accomplishing Performance Management with SNMP”, INET’93, pages CEA-1~CEA-5, 1993

[4] William Stallings, “SNMP, SNMPv2, SNMPv3 and RMON 1 and 2”, Addison-Wesley, 1999

[5] 안성진, 정진욱, “SNMP MIB-II 를 이용한 인터네트 분석 파라미터 계산 알고리즘에 관한 연구”, 정보처리학회, 제 5 권 제 8 호, page 2102-2116, 1998