네트워크 토폴로지 기반의 관리 정보 수집 기법 연구

정회원 한 정 수*, 안 성 진**, 정 진 욱***

A Study on management information gathering technique based on network topology

Jeong-Sooil Han*, Seong-Jin Ahn**, Jin-Wook Chung*** *Regular Members*

요 약

효율적인 네트워크 관리를 위해서는 먼저 관리 네트워크 상에 발생된 장애 발생 시스템의 파악을 통한 관리 대상 선별이 선행되어야 하며, 이를 통해 관리 네트워크 토폴로지 상에서 실제 관리 가능한 시스템과 관리 불가능한 시스템을 구분하여 관리할 수 있어야 한다. 이는 관리자로 하여금 네트워크 관리를 위한 불필요한 관리 트래픽의 발생을 방지하고자 하는 것이다. 따라서 이 논문에서는 관리 네트워크 상에 한 개 이상의 노드들에 의해 장애가 발생했을 때 실제 폴링 가능한 노드와 폴링 불가능한 노드를 찾아내는 알고리즘을 도출하고, 이를 통해 관리 네트워크의 토폴로지 인지 여부에 따라 전체 관리 대상에 대한 폴링 횟수와 폴링 응답 시간에 대한 효율성을 평가하였다. 또한, 이를 실제 네트워크 환경 하에서 실험을 통해 검증하였다.

ABSTRACT

In order to gather management information efficiently in the TCP/IP network, we, the manager, should have a firm grip on the network topology to find out in which system(s) failure occurs and to determine which to manage. If we are fully aware of how the network is formed and which system is failed, we can avoid generating unnecessary management traffic toward the systems which become unmanageable due to failure in itself or themselves, or their link to the failed one(s). Thus we can enhance efficiency in gathering management information. As we all know, by the way, the efficiency can go up or down according to the locations and the number of the systems to be managed. In this paper we would like to present significantly effective algorithms which allow the manager who already knows failure in one system or more in the managed network to choose which to manage. Using this information, we are going to compare and analyze management efficiency affected by whether we are fully aware of the network topology or not, and make an experiment so as to generalize the efficiency according to the location(s) of failed system(s).

Ⅰ. 서 론

네트워크 관리자는 네트워크 사용자에게 보다 안정적이고, 사용자의 요구에 부합되는 네트워크 서비스를 제공하기 위해 네트워크를 항시 감시하고 관리하여야 하며 이를 위해 네트워크 관리를 위한 관리 트래픽이 발생하게 된다[1]. 네트워크 관리를 위한 관리 트래픽을 전송하는 방법으로는 여러 가지 방법이 있지만 여기서는 요청과 응답 기법을 이용하는 폴링 방식을 사용한다. 이러한 폴링 방식은 첫째, 장애에 대한 신뢰성 있는 감지가 가능하고, 두 번째 네트워크 관리를 위한 간단한 프로토콜을 제공한다는 점, 그리고 피관리 시스템에 대한 부하가 적다라는 점 때문에 많은 응용에서 사용된다[2]. 하

* 성균관대학교 전기전자및컴퓨터공학부 정보통신연구실(jshan@songgang.skku.ac.kr)
** 성균관대학교 사범대 컴퓨터교육과 (sjahn@comedu.skku.ac.kr)
*** 성균관대학교 전기,전자 및 컴퓨터 공학부 교수(jwchung@songgang.skku.ac.kr)
논문번호 : 00218-0619, 접수일자 : 2000년 6월 19일

지만 이러한 방식은 네트워크 상에 관리 트래픽을 과도하게 발생시킬 수 있다는 단점을 가지고 있고, 이러한 상황은 관리 대상들 중에 장애 시스템이 있을 경우 더욱 그러하다. 따라서 이러한 단점을 해결하기 위해서는 관리 대상들의 장애 상태를 파악하여 실제 관리 가능한 관리 대상과 관리 불가능한 관리 대상을 구별하여 관리함으로써 네트워크 상에 과도한 관리 트래픽 발생을 줄이는 것이 중요하며[2,3], 이러한 작업은 네트워크 관리자들에 의해 관리 네트워크의 토폴로지 정보를 알 때 더욱 효과적인 관리 트래픽 감소와 네트워크 관리가 가능할 것이다. 이를 위해서는 먼저 관리 네트워크 상에서 장애 발생 시스템을 판별하고 이를 통해 관리 가능한 시스템과 관리 불가능한 시스템을 구별하는 작업이 선행되어야 하며[4-8], 이는 특히 장애가 발생하지 않아도 구성된 네트워크 토폴로지 때문에 장애가 발생된 네트워크 노드에 의해 실제로 관리 불가능한 관리 대상을 찾아내는 일이 수행되어져야 한다. 따라서 본 논문에서는 관리되는 네트워크 토폴로지 상의 관리 대상들에 장애가 발생했을 때 실제 폴링 가능한 관리 대상과 폴링 불가능한 관리 대상을 파악하는 알고리즘을 제시하였고, 이를 토대로 관리 네트워크 토폴로지에 대한 인지 여부에 따른 전체 관리 대상에 대한 폴링 응답 시간과 폴링 횟수에 대한 성능 효율을 평가함으로써 효율적인 네트워크 관리 수행을 도모코자 한다. 이를 위해서 먼저 관리 네트워크 상에 한 개의 관리 대상에 장애가 발생했을 경우를 살펴보고 이를 기반으로 다수개의 관리 대상에서 장애가 발생했을 경우에 대한 알고리즘과 그 성능 효율을 살펴보기로 한다. 또한, 이를 실제 네트워크 환경 상에 다양한 시나리오를 기반으로 검증하기로 한다. 따라서 본 논문은 II장에서 관리 네트워크 상에서 관리 대상을 선별할 수 있는 알고리즘을 제시하고 이에 대한 성능 효율을 살펴보며, 이를 기반으로 III장에서는 알고리즘을 검증하기 위해 실제 네트워크 환경 상에서 실험을 통해 결론을 제시한다. 또한, IV장에서는 관련 연구를 제시하고 V에서 결론을 내리기로 한다.

II. 관리 대상 선별 알고리즘

관리 대상을 선별하기 위한 알고리즘은 단일 시스템에 장애가 발생한 경우인 단일 시스템 장애 즉, SSF(Single-System Failure)과, 다수개의 시스템에서 장애가 발생한 경우인 다중 시스템 장애 즉, MSF(Multiple-System Failure)로 나누어 제안하며, 이를 기반으로 관리자의 네트워크 토폴로지 인지 여부에 따라서 전체 관리 대상에 대한 폴링 응답시간과 폴링 횟수의 성능 효율을 평가한다. 제안하는 알고리즘들은 몇 가지 가정을 기반으로 하는데 그 중에서 관리 네트워크 상에 관리 노드들에 대한 인접 리스트(Adjacency list)에 대한 정보가 존재하여야 하는데 이 정보는 각 연결 라우터에 존재하는 MIB-II에 대한 Interface 그룹과 IP 그룹을 폴링함으로써 각 라우터에 있는 라우팅 테이블 정보를 통해 얻을 수 있으며, 이에 대한 연구는 [10]과 같이 MIB-II에서 제공하는 정보를 사용하여 자동적으로 네트워크 토폴로지를 구성하는 방법으로 연구되고 있다. 단, 이러한 알고리즘의 결과로써 나오는 값들은 관리자의 위치에 따라 달라질 수 있으며 여기서는 관리자 시스템이 임의의 위치에 존재함을 가정한다.

1. 단일 시스템 장애 (SSF : Single- System Failure)

여기서는 관리 네트워크 상에 단일 시스템에서 장애가 발생했을 때 관리가 가능한 피관리 시스템과 관리가 불가능한 피관리 시스템을 파악하는 알고리즘을 제시하고, 이를 기반으로 관리자의 관리 네트워크 토폴로지에 대한 인지 여부에 따른 전체 폴링 응답 시간과 폴링 횟수에 대한 효율을 비교한다.

1.1 SSF 알고리즘

SSF상에서, 폴링 가능한 피관리 시스템과 폴링 불가능한 피관리 시스템을 찾아내는 알고리즘은 다음과 같은 정의들과 가정들을 기반으로 7 단계로 나누어진다.

정의)

1. 피관리 시스템들의 집합 N과 그 원소들의 무방향 링크들의 집합 A로 구성된 무방향 그래프 G = (N,A)로 정의한다.($N=\{n_1, n_2, \ldots\ldots n_n\}$)
2. $|Y|$는 집합 Y 원소들의 총 개수를 의미한다.(cardinality)

가정)

1. 관리 네트워크 토폴로지를 기반으로 Adjacency list가 존재한다[10].
2. 관리 네트워크 상에 발생된 한 개의 장애 시스템(SSF)을 fn_i 로 정한다. ($fn_i \in N$)

3.임의의 관리자 시스템에서 장애가 발생한 fn_i 까지 깊이(depth)를 d_i로 정한다.

단계1)

집합 N 상에서 d_i와 같거나 작은 depth의 피관리 시스템들의 집합을 R로 정한다.

R={r_1, r_2, ...r_m} (단 R⊆N, m ≤ n)

단계2)

집합 R에서 SSF를 제외한 피관리 시스템들의 집합 T를 다음과 같이 정의한다.

T=R-{ fn_i } = {t_1, t_2, ...t_t} (단, T⊂N, t=m-1)

단계3)

집합 T의 각 원소들에 대한 인접 리스트들의 집합에 대해 다음과 같이 집합 S를 정의한다. (단, 해당 원소의 depth보다 큰 피관리 시스템들만 포함한다.)

$$\delta(fn_i) = \delta_{s(t_1)}(fn_i) + \delta_{s(t_2)}(fn_i) + \cdots + \delta_{s(t_t)}(fn_i) = \sum_{k=1}^{t} \delta_{s(t_k)}(fn_i)$$

또한, 집합 S에서 장애 발생 시스템을 제외하면 다음과 같이 정의할 수 있다.

$$S = \bigcup_{1 \le k \le t} S(t_k) - \{fn_i\}$$

(단, $S(t_k)$는 k번째 원소에 대한 인접 리스트들의 집합이며, $\delta_{s(t_k)}(fn_i)$은 $S(t_k)$에서 fn_i의 포함 개수이다)

단계4)

fn_i에 대한 인접 리스트 집합과 fn_i을 포함한 것을 집합 F로 정한다.

(단, d_i보다 큰 시스템들만 포함한다.)

F={f_1, f_2, ... f_l}∪{ fn_i } (단, F⊂N, i<m)

단계5)

만약 R∪S∪F ≠ N이면 관리하는 피관리 시스템 모두를 포함하지 않은 것이므로, R∪S∪F=N 될 때 까지 다음 단계를 반복한다.

i) 집합 S의 각 원소들에 대한 인접 리스트들의 집합 A를 통해 S를 다음과 같이 재정의한다.

$$A = \bigcup_{1 \le t \le t} A(S_k)$$

(단, $A(S_k)$는 S의 k번째 원소의 인접 리스트 집합)

따라서, S = S∪A로 정의한다.

ii) 집합 F의 각 원소들에 대한 인접 리스트들의 집합 B를 통해 F를 다음과 같이 재정의한다.

$$B = \bigcup_{1 \le k \le i} B(f_k)$$

(단, $B(f_k)$는 F의 k번째 원소의 인접 리스트 집합)

따라서, F = F∪B로 정의한다.

단계6)

실제로 폴링 가능한 피관리 시스템들의 집합을 P로 하면, 다음과 같이 정의한다. P = S

단계 7)

폴링 가능한 시스템들의 집합을 C로 하면, 다음과 같이 정의한다. C = F - P

위의 알고리즘의 결과들에 대해 살펴보면, 단계3)의 결과인 집합 S는 fn_i의 depth보다 같거나 작은 피관리 시스템들은 폴링 가능한 피관리 시스템으로 간주하는 집합이다. 또한 단계4)의 결과인 집합 F는 fn_i와 이의 링크와 연결된 폴링 불가능한 시스템들에 대한 집합이며, 단계5)는 모든 피관리 시스템들을 포함하기 위한 단계로써 집합 S와 집합 F의 결과들에 대한 인접 리스트들을 사용하여 모든 피관리 시스템들을 포함할 때 까지 계속 반복하게 된다. 따라서, 단계6)과 단계7)은 SSF상에서 모든 피관리 시스템들을 대상으로 실제로 폴링 가능한 시스템과 폴링 불가능한 시스템들을 각각 집합 P와 집합 C로 얻을 수 있다.

1.2 SSF 성능 효율

SSF 상에서 위의 알고리즘 결과들을 바탕으로 관리자가 관리 네트워크의 토폴로지에 대한 인지 여부에 따른 전체 피관리 시스템들에 대해서 전체 폴링 응답시간과 폴링 횟수에 대한 성능 효율을 비교한다.

1) 관리 네트워크 토폴로지를 인지하고 있을 때 전체 피관리시스템 폴링 횟수는

$Single\_TP_{yes_{num}} = |P| + |fn_i| * R_t$ (단, R_t는 재전송횟수)

로 정의할 수 있으며, 전체 피관리 시스템 폴링 응답시간은 $Single_TP_{yes_{time}} = \sum_{j=1}^{|P|} \rho_{p_j} + \varpi_{fn_i}$로 정의할 수 있다. 여기서, ρ_{p_i}는 집합 P상의 j번째 피관리

시스템의 폴링 응답시간으로서 응답시간($2T(d_{1,}d_{p_j})$)과 j번째 피관리 시스템의 처리시간($\alpha(p_j)$)을 더한 값으로 다음과 같이 정의할 수 있다. $\rho_{p_i} = 2T(d_{1,}d_{p_j}) + \alpha(p_j)$

또한, ϖ_{fn_i}는 SSF(fn_i)의 재전송 폴링 응답시간으로써 다음과 같이 정의할 수 있다.

$\varpi_{fn_i} = R_t(2T(d_{1,}d_{fn_i}) + \alpha(fn_i))$

전체 피관리 시스템의 폴링 횟수는 실제로 폴링 가능한 시스템인 집합 P와 SSF에 대한 재전송 횟수를 포함한 것이고, 폴링 응답 시간은 각 집합의 원소들의 depth에 따른 응답 시간과 피관리 시스템의 처리시간을 포함한 것이다.

2) 관리 네트워크 토폴로지를 인지하지 못할 때

전체 피관리 시스템 폴링 횟수는

$Single\_TP_{no_{num}} = |P| + |C| * R_t$ (단, R_t는 재전송횟수)

로 정의할 수 있으며, 전체 피관리 시스템 폴링 응답시간은 (|N| = |P| + |C|)

$Single\_TP_{no_{time}} = \sum_{i=1}^{|P|}\rho_{p_i} + \sum_{j=1}^{|C|}\omega_{c_j}$ 로 정의할 수 있다. 여기서, ω_{c_j}는 집합 C상의 j번째 피관리 시스템의 재전송 폴링 응답시간이며, ρ_{p_i}는 집합 P상의 I번째 피관리 시스템의 폴링 응답 시간으로써 각각 다음과 같이 정의할 수 있다.

$\rho_{p_i} = 2T(d_{1,}d_{p_j}) + \alpha(p_j)$

$\varpi_{c_j} = R_t(2T(d_{1,}d_{c_j}) + \alpha(c_j))$

관리 네트워크의 토폴로지를 인지하지 못했을 때의 전체 피관리 시스템의 폴링 횟수와 폴링 응답시간은 단계6)의 결과인 실제 폴링 가능한 시스템인 집합 P와 단계7)의 결과인 폴링 불가능한 시스템인 집합 C에 대한 재전송을 포함한 것이고, 폴링 응답 시간은 각 집합의 원소들의 depth에 따른 응답시간과 처리시간을 포함한 것이다.

이러한 SSF 성능 효율은 표1과 같이 정리할 수 있으며, 결과적으로 관리 네트워크의 토폴로지를 인지하지 못했을 때는 토폴로지 상에서 장애가 발생된 피관리 시스템에 의해 실제로 폴링 불가능한 시스템을 관리자는 파악할 수 없어서 불필요한 폴링

표 1. SSF상에서 관리 네트워크 토폴로지 인지여부에 따른 성능 효율

토폴로지 인지여부	총 폴링 횟수	총 폴링 응답시간
Yes	$Single\_TP_{yes_{num}} = \|P\| + \|fn_i\| * R_t$	$Single\_TP_{yes_{time}} = \sum_{j=1}^{|P|}\rho_{p_j} + \varpi_{fn_i}$
No	$Single\_TP_{no_{num}} = \|P\| + \|C\| * R_t$	$Single\_TP_{no_{time}} = \sum_{j=1}^{|C|}\omega_{c_j} + \sum_{i=1}^{|P|}\rho_{p_i}$

응답시간과 폴링 횟수가 소요된다는 것을 알 수 있다.

2. 다중 시스템 장애 (MSF : Multiple-System Failure)

여기서는 장애가 다수개의 피관리 시스템에서 발생되었을 때에 대한 폴링 가능한 시스템과 폴링 불가능한 시스템을 찾아내는 알고리즘과 관리자에 의한 관리 네트워크의 토폴로지 인지 여부에 따른 폴링 횟수와 폴링 응답시간에 대한 성능 효율을 살펴보도록 한다.

2.1 MSF 알고리즘

MSF상에서 사용하는 정의는 SSF의 정의를 그대로 사용하고 가정은 다음을 추가하기로 한다.

가정)

1. MSF들의 집합을 FN으로 다음과 같이 정의한다.
FN = {fn1, fn2, ... fni} (단, FN⊆N, i≤n)
2. 각 MSF들에 대한 depth의 집합을 D={d1, d2, ...di}로 정하고, 그 중 가장 큰 depth 값을 d_{max}로, 가장 작은 값은 d_{min}으로 다음과 같이 정의한다.
(d_{max} = max(D), d_{min} = min(D))

단계1)

집합 N상에서 d_{max}와 같거나 작은 depth의 시스템의 집합을 R로 정한다.

R = {r_1, r_2, ...r_m} (단, R⊆N, m≤n)

단계2)

집합 R에서 MSF을 제외한 집합을 T로 정하면

T=R-FN={t_1, t_2, ...t_t} (단, T⊂N, t=m-n)

단계3)

집합 T의 각 원소들에 대한 인접 리스트들의 집합 S를 다음과 같이 정의한다. (단, 각 원소들의 depth 보다 같거나 큰 피관리 시스템들만 포함한다.)

$S = \bigcup_{1 \le k \le t} S(t_k)$ (단, $S(t_k)$는 집합 T의 k번째 원소의 인접 리스트의 집합)

이들 각 집합에서 집합 FN의 각 원소들의 개수를 조사하여 1보다 큰 FN의 원소들에 대해서만 새로운 집합 G를 정의한다. (G는 장애가 발생한 피관리 시스템으로의 다른 링크가 존재함으로 의미한다.)

$\delta(fn_i) = \delta_{s(t_1)}(fn_i) + \delta_{s(t_2)}(fn_i) + \delta_{s(t_t)}(fn_i)$

$= \sum_{k=1}^{t} \delta_{s(t_k)}(fn_i)$ 일 때,

(단, $\delta_{s(t_k)}(fn_i)$는 집합 T의 k번째 원소에 대한 인접 리스트 집합 상에서의 fn_i의 개수)

따라서, 집합 S는 다음과 같이 정의할 수 있다.

$S = \bigcup_{1 \le k \le t} S(t_k)$

단계4)

MSF의 집합 FN의 인접 리스트의 집합 중에서 FN을 포함한 집합을 F로 정한다. (단, 각 원소들의 depth보다 같거나 큰 피관리 시스템들만 포함한다.)

$F = \bigcup_{1 \le k \le i} F(fn_k)$ (단, $F(fn_k)$는 집합 FN의 k번째 원소에 대한 인접 리스트의 집합)

집합 FN중에서 연결관계에 의해 폴링 할 필요가 없는 시스템의 집합을 FC로 하면 다음과 같이 정의할 수 있다.

$FC = \bigcup_{1 \le k \le i} [(F(fn_k) \cap FN)]$

단계5)

만약 R∪S∪F ≠ N이면, 관리하는 피관리 시스템 모두를 포함하지 않은 것이므로, R∪S∪F = N 될 때 까지 다음 단계를 반복한다.

i) 집합 S의 각 원소들에 대한 인접 리스트들의 집합 A를 통해 S를 다음과 같이 재정의한다.

$A = \bigcup_{1 \le k \le t} A(S_k)$

(단, $A(S_k)$는 S의 k번째 원소의 인접 리스트 집합)

따라서, S = S∪A로 정의한다.

ii) 집합 F의 각 원소들에 대한 인접 리스트들의 집합 B를 통해 F를 다음과 같이 재정의한다.

$B = \bigcup_{1 \le k \le i} B(f_k)$

(단, $B(f_k)$는 F의 k번째 원소의 인접 리스트 집합)

따라서, F = F∪B로 정의한다.

단계6)

실제로 폴링 가능한 시스템의 집합을 P로 하면, P는 다음과 같다. P = S

단계7)

폴링 불가능한 시스템의 집합을 C로 하면, 집합 C는 F-P와 같이 정의한다. 또한, 집합 C중에서 연결관계에 의해 폴링 할 필요없는 MSF를 제외한 피관리 시스템들의 집합을 Z = C - FC로 정의할 수 있다.

위의 알고리즘 결과들에 대해 살펴보면, 단계3)의 결과인 집합 S는 MSF의 depth보다 같거나 작은 피관리 시스템들은 폴링 가능한 피관리 시스템으로 간주하는 집합이다. 또한 단계4)인 결과인 집합 F는 MSF와 이들의 링크와 연결된 폴링 불가능한 시스템들에 대한 집합이며 또한, 집합 FC는 MSF 중에서 이들의 토폴로지 연결에 의해 실제로 폴링 할 필요가 없는 MSF들에 대한 집합이다. 단계5)는 모든 피관리 시스템들을 포함하기 위한 단계로써 집합 S와 집합 F의 결과들에 대한 인접 리스트들을 사용하여 모든 피관리 시스템들을 포함할 때까지 계속 반복하게 된다. 따라서 단계6)과 단계7)은 MSF 상에서 모든 피관리 시스템들을 대상으로 실제로 폴링 가능한 시스템과 폴링 불가능한 시스템들을 각각 집합 P와 집합 C로 얻을 수 있으며, 더욱이 집합 C중에서 이들의 토폴로지 연결에 의해 실제 폴링 할 필요없는 집합 FC를 제외한 집합 Z를 정의할 수 있다.

2.2 MSF 성능 효율

MSF 상에서 위의 알고리즘 결과들을 바탕으로 관리자가 관리 네트워크의 토폴로지에 대한 인지 여부에 따른 전체 피관리 시스템들에 대한 전체 폴링 응답시간과 폴링 횟수에 대한 성능 효율을 평가한다.

1) 관리 네트워크 토폴로지를 인지하고 있을 때 전체 피관리 시스템 폴링 횟수는

$Multi\_TP_{yes_{num}} = |P| + |FN-Z| * R_t$ (단, R_t은 재전송 횟수)로 정의할 수 있으며, 전체 피관리 시스템 폴링

응답시간은

$Multi\_TP_{yes_{time}} = \sum_{k=1}^{|P|} \rho_{P_k} + \sum_{j=1}^{|FN-Z|} \omega_{(FN-Z)_j}$ 로 정의할 수 있다. 여기서, ρ_{P_k}는 집합 P상의 k번째 피관리 시스템의 폴링 응답시간으로써 응답시간 ($2T(d_{1,}d_{p_k})$)과 피관리 시스템의 처리시간 ($\alpha(p_k)$)을 포함한 시간이며, $\varpi_{(FN-Z)_j}$는 집합 (FN-Z)상의 j번째 MSF의 재전송 폴링 응답시간으로써 각각 다음과 같이 정의할 수 있다.

$\rho_{P_k} = 2T(d_{1,}d_{p_k}) + \alpha(p_k)$

$\varpi_{(FN-Z)_j} = R_t(2T(d_{1,}d_{(FN-Z)_j}) + \alpha((FN-Z)_j))$

전체 피관리 시스템의 폴링 횟수는 실제로 폴링 가능한 시스템인 집합 P와 단계7)의 결과인 MSF상에서 그들의 토폴로지에 의해 실제 폴링 할 필요없는 피관리 시스템들(Z)을 제외한 집합들에 대한 재전송 횟수를 포함한 것이며, 폴링 응답시간은 각 집합의 원소들의 depth에 따른 응답시간과 각 원소들의 처리시간을 포함한 것이다.

2) 관리 네트워크 토폴로지를 인지하지 못할 때

전체 피관리 시스템 폴링 횟수는

$Multi\_TP_{no_{num}} = |P|+|C|*R_t$ (단, R_t는 재전송 횟수)로 정의할 수 있으며, 전체 피관리 시스템 폴링 시간은

($|N|=|P|+|C|$) $Multi\_TP_{no_{time}} = \sum_{k=1}^{|P|} \rho_{P_k} + \sum_{j=1}^{|C|} \omega_{C_j}$ 로 정의할 수 있다. 여기서, ρ_{P_k}는 집합 P상의 k번째 시스템의 폴링시간을, ω_{C_j}는 집합 C상의 j번째 MSF의 재전송 폴링 응답시간을 말한다.

$\rho_{P_k} = 2T(d_{1,}d_{p_k}) + \alpha(p_k)$

$\omega_{C_j} = R_t(2T(d_{1,}d_{c_j}) + \alpha(c_j))$

관리 네트워크의 토폴로지를 인지하지 못했을 때 MSF상에서 전체 피관리 시스템의 폴링 횟수와 폴링 응답시간은 단계6)의 결과인 실제 폴링 가능한 시스템인 집합 P와 단계7)의 결과인 폴링 불가능한 시스템인 집합 C에 대한 재전송 폴링 응답시간을 포함한 것이고, 폴링 응답시간은 각 집합의 원소들의 depth에 따른 응답시간과 각 원소들의 처리시간을 포함한 것이다.

표 2. MSF상에서 관리 네트워크 토폴로지 인지여부에 따른 성능 효율

토폴로지 인지여부	총 폴링 횟수	총 폴링 응답시간
Yes	$Multi\_TP_{yes_{num}} = \|P\|+\|FN-Z\|*R_t$	$Multi\_TP_{yes_{time}} = \sum_{k=1}^{\|P\|} \rho_{P_k} + \sum_{j=1}^{\|FN-Z\|} \omega_{(FN-Z)_j}$
No	$Multi\_TP_{no_{num}} = \|P\|+\|C\|*R_t$	$Multi\_TP_{no_{time}} = \sum_{k=1}^{\|P\|} \rho_{P_k} + \sum_{j=1}^{\|C\|} \omega_{C_j}$

이러한 성능 평가는 표2와 같이 정리할 수 있으며, 관리 네트워크의 토폴로지를 인지하지 못했을 때는 SSF와 같이 토폴로지 상에서 장애가 발생된 피관리 시스템에 의해 실제로 폴링 불가능한 시스템을 관리자는 파악할 수 없어서 이들에 대한 불필요한 폴링 응답시간과 폴링 횟수가 소요되는 것을 알 수 있다.

3. 알고리즘의 효율성 평가

본 논문에서는 관리 네트워크에 대한 인접 리스트가 존재함으로 가정했는데 이러한 인접 리스트는 관리 네트워크 상의 라우터에 존재하는 라우팅 테이블을 기반으로 생성되므로[10] cyclic graph뿐만 아니라 어떠한 네트워크 토폴로지 상에서도 이 리스트의 생성이 가능하며, 따라서 여기서 제시한 알고리즘을 적용할 수 있다. 본 논문에서 제시하는 알고리즘들(SSF, MSF)에 대한 시간 복잡도(time complexity)를 살펴보면 다음과 같다. 관리 네트워크에 대해서 관리자 시스템의 위치는 임의로 정해질 수 있으므로 이에 따라서 시작되는 관리 노드를 선정하는데 걸리는 시간은 O(n)이다(n은 관리 노드의 수). 또한, 이 시작 노드로부터 인접 리스트를 검색하는 시간은 최대 O(n)이고, 포함해야 하는 관리 노드의 수는 많아야 n이므로 총 시간은 $O(n^2+n)$이다. 이 시간 복잡도는 본 논문에서 제시하는 SSF와 MSF 알고리즘들 모두에 동일하게 적용된다.

Ⅲ. 실험 및 결과

이 논문의 실험은 다음과 같은 실험 환경 하에서 행해졌다.

- 실험 대상 : 본교 네트워크

- 실험 피관리 시스템 수 : 39개 피관리 시스템
- 실험 관리자 시스템 사양 : SunOS Release 5.6 System V Release 4.0, SPARC-20

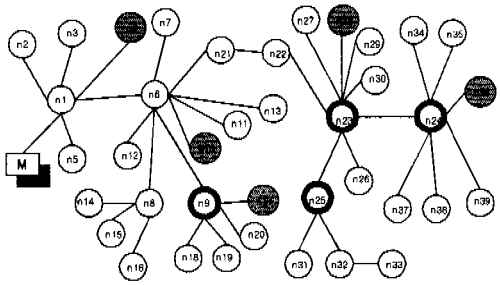


그림 1. 실험 네트워크

그림1은 이 논문의 실험을 위한 본교 네트워크 구성 형태를 보여주고 있다. 그림1에서 보는 바와 같이 하나의 관리자 시스템(M)과 39개의 피관리 시스템으로 구성되어 있음을 알 수 있다. 피관리 시스템들의 집합 N은 {n1, n2, ... n39}이며, 장애가 발생된 시스템들의 집합(FN)과 이에 따른 depth의 집합(D), 그리고 d_{max}에 따라서 실험이 이루어진다. 또한, 각 시스템들을 폴링하는데 사용되는 프로토콜은 SNMP 프로토콜[9]을 사용하며, 폴링 응답시간은 10ms, 장애가 발생한 시스템의 폴링 응답시간은 재전송 횟수(R_t) 3회를 포함한 70ms로 정한다. 단, 장애 발생 시스템들은 본 실험을 위해 무작위로 선정하며, T(i,j)은 동일하다고 가정한다.(단, T(i,j)는 depth i와 그 인접한 depth j 사이의 응답시간이라고 가정한다) 즉, 관리자 시스템으로부터 피관리 시스템에 대한 폴링 응답시간은 해당 피관리 시스템들의 depth와 비례하다고 가정한다.

1. 시나리오 1(SSF 경우)

첫 번째 시나리오는 그림 1의 실험 네트워크 상에서 하나의 시스템에서 장애가 발생했을 때에 대한 시나리오이다. 여기서는 장애가 발생된 위치에 따라 두 가지 경우로 나누어진다. 첫 번째 경우인 마지막(leaf) 노드에 장애가 발생했을 대와 두 번째 경우인 중간 노드에 장애가 발생했을 때이다.

2. 시나리오 2(MSF 경우)

두 번째 시나리오에서는 그림 1의 실험 네트워크에서 표시(색)된 바와 같이 네트워크 토폴로지 상에서 마지막(leaf)시스템에 장애가 발생되었을 때의 시나리오이다.

표 3. 시나리오 1(SSF 경우)의 성능 파라미터 및 실험 결과

성능 파라미터	마지막(leaf)노드에 장애 발생시 ($fn_i = n_{27}, d_i = 6$)	중간노드에 장애발생시 ($fn_6 = n_{27}, d_i = 2$)
$Single\_TP_{no_{num}}$	41회	107회
$Single\_TP_{no_{time}}$	450ms	764ms
$Single\_TP_{yes_{num}}$	41회	8회
$Single\_TP_{yes_{time}}$	450ms	120ms

FN={ $fn_4, fn_{10}, fn_{17}, fn_{28}, fn_{36}$ }, D={2,3,4,6,7}, d_{max} =7일 때

표 4에서 보는 바와 같이 네트워크 토폴로지를 인지하지 못했을 때나 인지했을 때에 전체 피관리 시스템에 대한 폴링 횟수와 응답시간이 모두 동일함으로 알 수 있다. 이것은 장애가 발생된 노드가 마지막(leaf) 시스템이기 때문에 관리자가 각 leaf 시스템까지 폴링해야 장애발생을 알 수 있기 때문에 생기는 결과이다.

표 4. 시나리오 2(MSF 경우)의 성능 파라미터 및 실험 결과

성능파라미터	실험결과
$Multi\_TP_{no_{num}}$	49회
$Multi\_TP_{no_{num}}$	690ms
$Multi\_TP_{yes_{num}}$	49회
$Multi\_TP_{yes_{time}}$	690ms

표 5. 시나리오 3(MSF 경우)의 성능 파라미터 및 실험 결과

성능파라미터	실험결과
$Multi\_TP_{no_{num}}$	83회
$Multi\_TP_{no_{num}}$	171ms
$Multi\_TP_{yes_{num}}$	23회
$Multi\_TP_{yes_{time}}$	310ms

3.시나리오 3(MSF 경우)

세 번째 시나리오에서는 그림 1의 실험 네트워크에서 표시(굵은 선)된 바와 같이 네트워크 토폴로지 상에서 장애가 발생한 시스템이 다른 피관리 시스템들을 연결하는 중간 시스템에서 발생했을 때의 시나리오이다.

FN = $\{n_9, n_{23}, n_{24}, n_{25}\}$, D={3,5,6,6}, d_{max} = 6일 때

표 5에서 보는 바와 같이 관리 네트워크의 토폴로지를 인지하지 못했을 경우에는 인지했을 경우보다 훨씬 많은 전체 폴링 횟수와 폴링 응답시간을 소요하고 있음을 알 수 있다. 이러한 결과는 다른 피관리 시스템을 연결하는 중간 시스템에서 장애가 발생했기 때문에 네트워크 토폴로지를 인지하는 경우에는 실제로 폴링 불가능한 피관리 시스템을 파악할 수 있었고, 이러한 이유로 불필요한 폴링 횟수나 응답시간을 줄일 수 있었기 때문이다.

4. 폴링 효율 비교 분석

위의 세 시나리오를 기반으로 다음과 같은 결론을 얻을 수 있다. 첫째, 관리자의 네트워크 토폴로지에 대한 인지 여부는 효율적인 네트워크 관리를 위해서 필요하다. 두 번째, 관리 네트워크 토폴로지 상에서 장애가 발생한 피관리 시스템의 위치에 따라서 전체 피관리 시스템의 폴링 횟수와 폴링 응답시간이 달라질 수 있다. 이는 위의 시나리오들에서 보는 바와 같이 네트워크의 토폴로지를 인지했을 경우 마지막(leaf) 시스템에서 장애가 발생하는 것보다 각 피관리 시스템들을 연결하는 중간 시스템에서 장애가 발생했을 때의 효율이 더 좋으며, 반대로 인지하지 못했을 경우는 각 마지막(leaf) 시스템에서 장애가 발생했을 때가 중간 시스템에서 발생했을 때 보다 더 효율이 좋다는 것을 알 수 있다. 따라서 표 6와 같이 결론을 내릴 수 있다.

표 6. 장애발생 장소와 네트워크 토폴로지 인지여부에 따른 폴링 효율

장애발생위치	네트워크 토폴로지 인지하는 경우	네트워크 토폴로지 인지하지 못한 경우
마지막(leaf) 시스템	중간 시스템에 발생한 것보다 효율이 나쁘다.	중간 시스템에 발생한 것보다 효율이 좋다.
중간 시스템	마지막(leaf)시스템에 발생한 것보다 효율이 좋다.	마지막(leaf) 시스템에 발생한 것보다 효율이 나쁘다.

Ⅳ. 관련 연구

관리 네트워크 상에 관리정보를 수집하기 위해서 일반적으로 사용하는 순차적인 폴링 방식을 개선한 방식이 [1,5]에서 제시되고 있으나, 이들 연구에서는 관리 노드에 대한 장애가 발생했을 때에 대한 경우를 배제한 상태이다. 따라서 장애가 발생한 관리 노드에 대해서 불필요한 관리 트래픽과 관리 시간을 낭비하고 있다.

또한, 관리 네트워크에 대해서 장애가 발생한 관리 노드의 위치를 파악 위한 여러 가지 알고리즘과 방법들이 [6,8]의 연구에서 제시되고 있다. 이들은 관리 노드의 장애 파악을 위해 여러가지 관리 정보를 수집할 뿐 아니라 물리적 장애 뿐만 아니라 트래픽에 대한 임계값 등을 사용하여 논리적인 장애를 파악하는 알고리즘을 제시하고 있다. 하지만 이러한 알고리즘은 물리적이고 논리적인 장애 시스템을 파악하기 위해서 많은 관리 트래픽을 발생시키고 알고리즘이 복잡하다는 단점이 있다.

관리 네트워크에 대한 토폴로지 구성은 [3,7,11]에서 제시되고 있다. 이들 연구에서는 링크계층의 네트워크 토폴로지와 함께 IP 계층에서 구성되는 논리적인 네트워크 토폴로지 검색을 위한 알고리즘을 제시하고 있다. 이러한 구성 방법은 라우터에서 제공하는 라우팅 테이블을 기반으로 구성되는데 이를 위해서 많은 정보를 수집하고 있다.

따라서 이러한 여러 가지 연구들을 복합적으로 사용하여 효율적인 관리 정보 수집을 위해서 관리 네트워크 토폴로지를 이용하는 연구가 필요하다.

Ⅴ. 결 론

관리 네트워크 상에서 피관리 시스템들의 장애를 파악하고 이를 기반으로 관리 가능한 피관리 시스템과 관리 불가능한 피관리 시스템에 대한 구분은 효율적인 네트워크 관리를 위해서는 반드시 선행되어야 하며, 이는 네트워크 상에 장애 발생 시스템을 인지하지 못해 생기는 불필요한 관리 트래픽의 유출을 방지하고자 함이다. 또한 이러한 점은 관리 네트워크 토폴로지의 인지 여부에 따라서 네트워크 관리 효율이 달라질 수 있다. 따라서, 이 논문은 관리 네트워크 상에서 장애가 발생된 한 개 이상의 피관리 시스템들을 기반으로 관리 가능한 피관리 시스템과 관리 불가능한 피관리 시스템을 파악하는

알고리즘을 제시하고, 이를 기반으로 관리 네트워크 토폴로지 인지 여부에 따라 관리하는 피관리 시스템들에 대한 전체 폴링 응답시간과 폴링 횟수에 대한 성능을 평가하였다. 또한, 실제 환경 하에서 이에 대한 검증을 실시하였다. 결과적으로 네트워크 토폴로지를 인지 하지 못한 경우에는 네트워크 토폴로지를 인지한 경우와 비교해서 장애가 발생된 피관리 시스템 때문에 실제로 폴링 불가능한 피관리 시스템들을 인식하지 못하고 불필요한 폴링 응답시간과 폴링 횟수를 낭비함으로 알 수 있으며, 또한 장애가 발생한 피관리 시스템들의 위치에 따라서도 그 효율이 달라짐을 알 수 있다.

참 고 문 헌

[1] Kang-Hyun Joong, Jin-Wook Chung, "Design and Analysis of the Shuttle Protocol for Gathering Network Management Information", *doctor thesis,* 1995

[2] L.Steinberg, "Techniques for Managing Asynchronously Generated Alerts", *RFC 1224*, May, 1991

[3] Yuri Breitbart, Minos Garofalakis, Cliff Martin, Rajeev Rastogi, S.Seshadri, Avi Silberschatz, "Topology Discovery in Heterogeneous IP Networks", *IEEE INFOCOM* 2000, March, 2000

[4] W. Theilmann, K.Rothermel, Dynamic "Distance Maps of the Internet", *IEEE INFOCOM 2000*, March, 2000

[5] Seong-Jin Ahn, Jin-Wook Chung, "The design of the shuttle protocol with network management data gatheringi" 1996 *IEEE Korea International Symposium on Network Operations and Management,* 1996

[6] Irene Katzela and Mischa Schwarz, "Schemes for Fault Identification in Communication Network", *IEEE/ACM Transaction on Networking,* vol.3, no.6, Dec, 1995

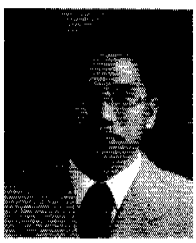
[7] J.Schonwalder, H.Langendorfer, "How to Keep Track of Your Network Configuration", Nov 1-5, 1993, 1993 *LISA*

[8] I.Katzela, A.T.Bouloutas, S.B.Calo, "Centralized vs Distributed Fault Localization", *INM,* 1995

[9] J.Case, M.Feor, M.Schoffstall, and J.Davin, "A Simple Network Management Protocol (SNMP)", *Internet RFC-1157,* May 1990

[10] Glenn Mansfield, M.Ouchi, K.Jayanthi, Y. Kimura, Kohei Ohta, Yoshiaki Nemoto, "Techniques for automated Network Map Generation using SNMP", *IEEE*, INFOCOM 1996

한 정 수(Jeong-Soo Han) 정회원

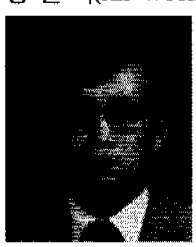


1997년 2월 : 성균관대학교 정보공학과 졸업
1999년 2월 : 성균관대학교 전기전자및컴퓨터공학부 석사 졸업
1999년 3월~현재 : 성균관대학교 전기전자및컴퓨터공학부 박사 과정(수료)
<주관심 분야> 네트워크 관리, 인터넷 관리, 통신 프로토콜

안 성 진(Seong-Jin Ahn) 정회원

1988년 2월 : 성균관대학교 정보공학과 학사
1990년 2월 : 성균관대학교 정보공학과 석사
1990년~1995년 : 한국전자통신연구원 연구전산망 개발실 연구원
1996년 : 정보통신 기술사 자격 취득
1998년 : 성균관대학교 대학원 정보공학과 박사
1999년~현재 : 성균관대학교 컴퓨터교육과 부교수
<주관심 분야> 네트워크 관리, 트래픽분석, 유닉스 네트워킹

정 진 욱(Jin Wook Chung) 정회원



1974년 2월 : 성균관대학교 전기공학과 학사
1979년 2월 : 성균관대학교 전자공학과 석사
1991년 2월 : 서울대학교 계산통계학과 박사

1982년~1985년 : 한국과학기술 연구소 소장
1981년~1982년 : Racal Milgo Co. 객원연구원
1985년~현재 : 성균관대학교 전기전자 및 컴퓨터
공학부 교수
<주관심 분야> 컴퓨터 네트워크, 네트워크 관리, 보안