네트워크 안정성 측정을 위한 가중치 기반의 계산알고리즘 연구

이원혁* · 노민기* · 조부승* · 김태연** · 김현철***

요 약

오늘날에는 기후, 전파천문, 고성능 컴퓨팅 자원 공동 활용 등의 첨단 기술을 요구하는 그룹별 연구 활동이 강화되고 있다. 이에 따라 이와 같은 오늘날의 첨단 응용 연구를 뒷받침하기 위해 첨단 연구망 역시 첨단 응용연구를 위한 수단으로서 점차적으로 자리매김하고 있다. 따라서 이러한 추세를 반영해서 첨단 연구망 네트워크의 관리적 측면 역시 집중적으로 조명되어야 할 필요성이 있다. 그렇기 때문에, 본 논문에서는 제시하고자 하는 가장 큰 맥락에서의 목표가 '여러 첨단 응용 연구의 도구적 수단으로서의 네트워크를 안정적으로 관리할 수 있도록 하는 연구'이다.

A weigth-based algorithm for network stability measurement

Wonhyuk Lee* · Minki Noh* · Buseung Cho* · TaeYeon Kim** · Hyuncheol Kim***

ABSTRACT

Today, group research efforts are increasing for cutting edge technologies related to climate, radio astronomy, co-utilization of high performance computing resources and more. Accordingly, networks for cutting-edge researches are also regarded as means for cutting-edge researches nowadays. Therefore, it is necessary for administrating cutting-edge network to be considered, reflecting such a current trend. In this sense, 'Research which can help to administrate the networks which are usually used as means for a lot researches' was set as a goal of this paper.

Key words : Weight, Network Stability, Route Algorithm

접수일(2014년 12월 1일), 게재확정일(2014년 12월 16일)

* 한국과학기술정보연구원 (KISTI)
** 한국전자통신연구원(ETRI)
*** 남서울대학교 컴퓨터학과 교수

1. 서 론

최근 기상기후, 천체우주, 고에너지물리 분야 등 첨단연구 분야에서 그룹 연구활동이 강화되고 있으며, 원격 융합 연구활동이 가속화되고 있는 상황이다. 따라서 최신 연구활동을 위하여 네트워크 기술도 최적의 환경으로 구현되고 있으며, 첨단 연구활동을 위한 서비스 플랫폼까지 설계의 주안점이 이루어지고 있다. 그러므로 응용연구단의 활용을 위한 미들웨어, 네트워크 서비스 스택까지 고려하여 최적의 경로와 환경을 제공할 수 있는 방법론이 필요한 시점이다. 이런 관점에서 많은 연구자들을 위한 도구로서 네트워크 설계 및 서비스 제공이 이루어질 수 있는 안정적인 환경 구현이 본 연구의 핵심이다.

이 목적을 위하여 본 논문에서는 네트워크 안정성을 계산하여 적절한 네트워크 구조관리를 위한 의미를 고찰해 보고자 한다. 또한 연구목적 수행을 위하여 네트워크 안정성 개념을 사용하여 단순 연결지향형 네트워크 기반이 아닌, 구조적 관점에서 필요한 사항을 점검하고, 이를 위하여 점검해야할 기본 정보의 도출을 시도할 것이다. 또한 네트워크 안정성 관련된 기존의 연구활동을 분석하고, 새로운 시도를 진행하고, 이를 확인할 수 있는 네트워크 안정성 계산 알고리즘을 연구하고자 한다.

2. 관련 연구

2.1 단절노드의 인접노드 수를 이용한 연구

이 방법은 네트워크에서 단절 노드와 연결된 인접 노드의 개수를 이용해서 해당 단절 노드의 중요도를 나타내고 있다. 단절 노드에 연결된 노드의 수가 많을수록 그 단절 노드에 문제가 발생할 때 더 많은 연결의 단절이 발생하기 때문에 단절 노드에 인접한 노드의 수가 이 안정도 공식의 주요 변수로 채택되었다.

$$N_{Stability}\begin{cases}\dfrac{n}{\sum_{i=1}^{n} T_i} & (\text{if } n \geq 1)\\ 1 & (\text{if } n = 0)\end{cases}$$

- $N_{stability}$: 네트워크의 안정도
- n : 단절 노드의 수
- T_i : 각각의 단절 노드에 노드의 수

2.2 깊이 우선 탐색 알고리즘을 이용한 연구

이 연구는 DFS(Depth First Search) 알고리즘을 이용하여 네트워크 내에서 두 지점간에 나타날 수 있는 모든 Unist 경로를 찾아내는 것을 기본으로 한다. 모든 경로를 찾아낸 다음에 각각의 경로에 대하여 살펴보면 하나의 경로는 여러 개의 노드와 노드 사이 구간으로 이루어져 있음을 알 수 있다. 이 연구에서 제시한 방법은 바로 이러한 사실에 기초하고 있다. 다음은 네트워크에서 두 지점 사이의 안정도를 나타내는 식이다.

$$Stability_{AB} = 1 - \prod_{n=1}^{pn}\left(1 - \prod_{m=1}^{nodeN_n}[path_n(m)]\right)$$

- $Stability_{AB}$: A 지점과 B 지점 사이의 안정성
- p_n : A 지점과 B 지점 사이에 존재하는 경로의 수
- $nodeN_n$: X번째 경로에 포함되는 노드의 개수
- $path_n(m)$: X번째 경로에서의 M번째 링크의 안정성

2.3 가중치를 지닌 단절 노드를 이요한 연구

이 선행 연구에서는 각 단절 노드마다 가중치를 부여하고 이를 이용하여 네트워크의 안정성을 산출한다. 여기서 가중치는 해당 단절 노드가 손상되었을 때, 전체 네트워크가 제 기능을 하는 데 얼마나 부정적인 영향을 주는지를 나타내는 지표이다. 이를 반영하기 위해 단절 노드의 가중치를 산출하는 과정에서는 단절 노드 자체의 중요도와 해당 노드로 인해 발생하게 될 트래픽상의 피해를 반영하도록 한다.

$$N_{Stability} \begin{cases} \frac{1}{\sum_{i=1}^{n} W_i} & (\text{if } n \geq 1) \\ 1 & (\text{if } n = 0) \end{cases}$$

- $N_{stability}$: 네트워크의 안정성
- n : 단절 노드의 수
- Wi : 단절 노드의 가중치

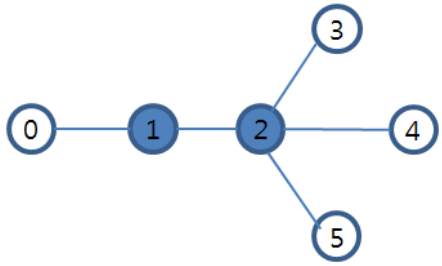
단절 노드의 가중치 산출과정에서 해당 노드의 손상으로 인한 unicast traffic을 반영한다.

3. 제안 알고리즘

본 연구에서는 단절 노드와 링크의 가중치 뿐 아니라, 기존 연구에서 반영되지 못했던 점들을 반영하여 보다 정확한 관리 정보로서의 '네트워크 안정성'을 산출하는 것이 목표라고 할 수 있다. 우선적으로 크게 네트워크 안정성 산출에 반영되는 요소를 전부 검색해 내고 이들 요소들에 대하여 가중치를 산출한다. 그리고 마지막 단계에서 이러한 정보들을 모두 종합하여 네트워크 안정성 산출 공식에 대입하여 해당 네트워크의 구조적 안정성을 측정하도록 한다. 네트워크 안정성을 계산하기 위한 제안된 알고리즘은 다음과 같다.

$$N_{stability} = \begin{cases} 1 - \frac{\sum_{k=1}^{p} W_k + \sum_{l=1}^{q} (\omega_l \times T_l)}{\sum_{i=1}^{m} W_i + \sum_{j=1}^{n} (\omega_j \times T_j)} & (k \geq 1) \\ 1 & (k = 0) \end{cases}$$

- $N_{stability}$: 네트워크 안정성
- m : 네트워크의 노드 개수
- n : 네트워크의 모든 유니캐스트 트래픽의 수
- p : 단절 노드의 수
- q : 단절노드, 링크의 손상 시 끊어지는 유니캐스트 트래픽의 수
- Wi : 노드에 부여되는 가중치
- Wj, Wl : 유니캐스트 트래픽의 가중치
- Tj, Tl : 유니캐스트 트래픽의 양



(그림 1) 예시 토폴로지

(그림 1)의 네트워크 구성을 보면, 단절 노드가 1번과 2번이라는 것을 알 수 있다. 이때 1번 노드에 인접한 노드의 개수가 2개, 2번 노드에 인접한 노드의 개수는 4개임을 알 수 있다. 이를 기존 연구의 가중치를 지닌 단절노드의 선행연구방법으로만 네트워크 안정도를 산출해 보면 다음과 같은 결과가 나타난다.

$N_{stability}$ = 2 / (2 + 4) = 1 / 3 = 0.3333

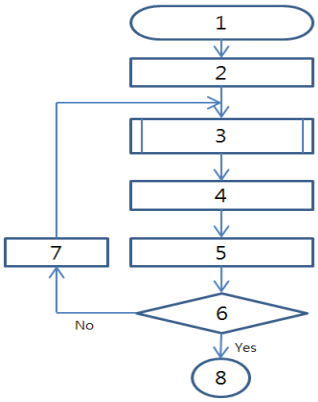
그러나 이 방법에서는 보완되어야 할 점이 있으며 다음과 같다. 우선, '단절 노드의 수'가 반영되지 않는 점이다. 두 번째로, 단절 노드에 가중치(Weight)가 부여되지 않았다는 점이다. 이 방법에는 단절 노드에 가중치가 부여되어 있지 않는데 이 말은 모든 노드가 같은 가치를 가진다는 말이다. 다시 말해서, 각각의 노드가 손상되었을 때를 생각해보면 이때 손상된 노드가 다르더라도 전체 네트워크에 미치는 피해의 정도가 같다는 것을 의미한다. 하지만 이러한 가정은 현실적이지 않다. 왜냐하면, 네트워크를 구성하고 있는 여러 노드는 각자의 특성, 처리하고 있는 데이터에 따라 중요도가 다르기 때문이다.

그리고 세 번째로는 안정도 산출 요소로서 '인접 노드의 수'의 부적절하다는 점을 들 수 있다. 단절 노드의 중요도를 바탕으로 안정성을 산출하는 것이 기존 연구에서 제시한 방법이다. 그런데 단절 노드의 중요도를 산출하기 위해서 이 방법에서는 '단절 노드에 인접한 노드의 수'를 사용하고 있다. 그 이유는 단절 노드의 손상으로 인해 생길 주변 연결 단절의 피해를 측정하기 위한 것이다. 하지만, 인접 노드만으로는 주변 연결 단절의 피해를 정확하게 측정한다고 보기 어렵다. 물론, 대략적인 관점에서 볼 때는 인접 노드의 개

수를 사용해도 그다지 문제가 되지 않는다. 하지만, 이 방법은 정확하다고는 할 수 없다. 왜냐하면 연결 단절의 피해 정도를 측정하기 위해서는 인접 노드가 아니라, 단절 노드의 손상으로 인해 나타나는 네트워크의 부분 조각인 여러 개의 Subgraph를 중심으로 생각해야 한다. 이 'Subgraph 사이의 연결'을 반영하는 것이 더 정확한 측정이 될 수 있기 때문이다.

따라서 본 연구에서는 네트워크의 안정성을 산출하는 방식을 다음과 같이 여러 단계에 걸쳐서 제시하였다. 아래와 같은 Flow Chart를 통해서 네트워크 안정성 산출 알고리즘을 나타낼 수 있는데 가장 위에 제시된 Flow Chart가 제일 거시적인 관점에서 접근한 것이고 아래로 갈수록 좀 더 세부적인 과정을 나타내는 것이다.

1 : Calculation of damage according to link failure
2 : Selection of specific disconnected node
3. Calculation of expected potential damage
4. Calculation of generating probability linked node
5. Selection of damage size
6. Every disconnected node?
7. Selection of next disconnected node
8. Return



(그림 2) 제안 알고리즘 절차

상기의 과정을 따라 구현되는 알고리즘은 다음과 같다.

```
FindSubgraph( ); // Insert sub graph into FindSubgraph[MAX][][] if AN[i] is damaged

for(j=0; j<MAX; j++) {
if(Network[AN[i]][j]==1) {
Adjacent[adjptr]=j; adjptr++  } }
for(j=0; j<adjptr; j++) {
for(k=j+1; k<adjptr; k++)
{
if(*FindSubgraph[MAX][Adjacent[j]]==
*FindSubgraph[MAX][Adjacent[k]]) {
*FindSubgraph[MAX][Adjacent[j]]=0;
*FindSubgraph[MAX][Adjacent[k]]=0;  } } }
for(j=0; j<adjptr; j++) {
if(*FindSubgraph[MAX][Adjacent[j]]!= {0 }) {
Link[linkptr][0]=AN[i]; Link[linkptr][1]=Adjacent[j];
linkptr++;  }  }
}
}
CalcLinkWeight( ){ // Function to calculate a link weight

int i, j, k, l, m;
for(i=0; i<linkptr; i++) {
*Copy_network=*Network; *FindSubgraph= {0 };
Copy_network[Link[i][0]][Link[i][1]]=0;
Copy_network[Link[i][1]][Link[i][0]]=0;

FindSubgraph( ){
// Insert sub graph into FindSubgraph[MAX][ ][ ] if link is damaged

for(j=0; j<MAX; j++) {
for(k=j+1; k<MAX; k++) {
if(*FindSubgraph[MAX][j]==*FindSubgraph[MAX][k])
*FindSubgraph[MAX][k]=0;
} }
for(j=0; j<MAX; j++) {
if(*FindSubgraph[MAX][j]!= {0 }) {
for(k=0; k<MAX; k++) {
if(*FindSubgraph[MAX][k]!= {0 } && k!=j) {
for(l=0; l<MAX; l++) {
if(FindSubgraph[MAX][j][l]==1) {
for(m=0; m<MAX; m++) {
if(FindSubgraph[MAX][k][m]==1)
LinkWeight[i]+=Weight[l]*Traffic[l][m];
}  }  }  }  }  }
}
}
```

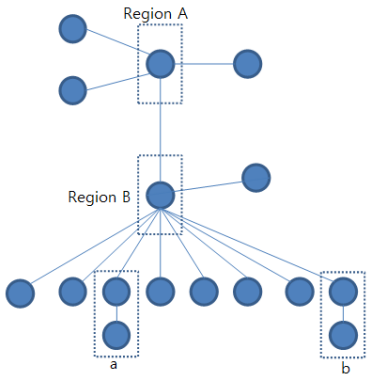
4. 시뮬레이션

본 절에서는 실험을 통해 검증된 본 연구에서의 알고리즘의 실제 환경에 대한 적용성을 확인하기 위해서 실제적으로 망의 토폴로지 관점에서 안정성 확보 전후의 구성을 가지고 적용한다. 네트워크 안정성이라는 측면에서 망 구조의 변화외에 사용하는 트래픽 양이나 노드의 중요 가중치는 같다고 가정하고 전체를 1로 가

정하고 진행하였다.

4.1 망 이중화 전의 구성에 대한 안정성 측정

망 이중화를 구현하기 전의 스타형의 네트워크 구성은 (그림 3)과 같이 32개의 노드와 31개의 링크로 구성되어 있다. 우선 트래픽양의 가중치는 제외하고 모든 노드의 초기 가중치를 1로 두고, 모든 트래픽 양의 가중치 역시 1로 두고 프로그램을 이용하여 안정성을 산출하였다.



(그림 3) 이중화 전의 토폴로지

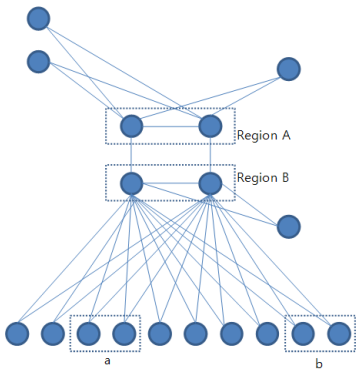
그 결과, 이중화 전의 구성에 대한 네트워크 안정성은 0.000157로 산출되었다.

4.2 망 이중화 후의 구성에 대한 안정성 측정

스타형 네트워크 구조의 단절노드 영향성을 줄이기 위하여 스타형 중심구간에 이중화 구성을 진행하여 (그림 4)와 같이 35개의 노드와 57개의 링크로 구성을 변경한 구성에 대하여 안정성을 측정하였다. 앞서 변경 전 토폴로지의 안정성을 산출했던 방법과 같은 방식으로 진행된 안정성 결과는 0.000683로 기존의 토폴로지에 비하여 4.35배의 안정성 향상을 가져왔다. 즉, 네트워크상에서 일어나는 모든 트래픽의 양과 각 노드의 초기 가중치가 동일하다고 가정할 때 이중화 등의 구조적 변화가 전체 망의 구조적 안정성이 향상된다는 것이다.

5. 결 론

지금까지 여러 예시를 통해서 본 연구 과제에서 제시한 네트워크 안정성 산출 알고리즘과 다르게 여러 가지 망 구조의 변화에 알맞은 안정도 측정을 통하여 망의 구성변경 전에 네트워크 구축후의 안정도의 변화를 예측할 수 있도록 객관적인 수치를 제시할 수 있다. 따라서 이러한 결과를 바탕으로 신뢰성 높은 네트워크 관리정보의 제공이 가능할 것으로 기대된다.



(그림 4) 이중화 후의 토폴로지

네트워크 안정성 산출 방안을 제시한 선행 연구에서는 네트워크를 구성하는 요소에 해당하는 노드와 링크를 모두 아우르지 못했다. 이에 따라 본 연구에서는 이 두 요소를 모두 반영한 네트워크 안정성 체계를 수립하고자 한다. 본 연구에서는 네트워크의 모든 요소를 안정성 산출에 반영함으로써 좀 더 실제 네트워크 환경에 가까운 관리 정보를 제공할 수 있을 것으로 기대한다. 또한 네트워크의 각 구성요소에 대한 특징을 고려하여 이를 바탕으로 중요도를 나타내는 척도로서 가중치를 활용할 수 있다.

네트워크를 설계하고 구현하는 입장에서, 단순히 네트워크의 연결성과 전달성만 고려하기 보다는 각 노드가 갖는 중요도를 이해하고, 단절노드의 발생처럼 운영상의 손실을 입을 수 있는 경우를 사전에 분석하여, 가중치가 높은 주요 노드에는 이중화 구성을 통한 장애 발생 가능 지점을 최소화 하는 방안에 활용할 수 있을 것이다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [14-000-05-001, 스마트 네트워킹 핵심기술개발] 또한 본 논문은 한국과학기술정보연구원과 남서울대학교의 연구지원에 의해 연구되었음

참고문헌

[1] A. Bonerjee, J. Drake, J. P. Lang, and B. Turner, "Generalized Multiprotocol Label Switching: an Overview of Routing and Management Enhancements," IEEE Communications Magazine, Vol. 39, No. 1, pp. 144-150, Jan. 2001.

[2] K. Shiomoto, E. Oki, D. Shimazaki, and T. Miyamura, "Multilayer Traffic Engineering Experiments in MPLS/GMPLS Networks," IEEE BROADNETS, 2007.

[3] Ng., A. Y., Zheng, A. X. & Jordan, M. I. Stable algorithms for link analysis. International Conference on Research and Development in Information Retrieval 258 - 266, 2001.

[4] Chien, S., Dwork, R., Simon, D. R. & Sivakumar, D. Link evolution: analysis and algorithms. Internet. 2004.

[5] Callaway, D. S., Newman, M. E. J., Strogatz, S. H. & Watts, D. J. Network robustness and fragility: percolation on random graphs. Phys. Rev. Lett. 85, 5468, 2000.

[6] G. de Veciana, T.-J. Lee, and T. Konstantopoulos, "Stability and performance analysis of networks supporting services with rate control - could the internet be unstable?" in IEEE Infocom, pp. 802 - 810, 1999.

[7] D. Feijer and F. Paganini, "Stability of primal-dual gra dient dynamics and applications to network optimizatio n," Automatica, vol. 46, no. 12, pp. 1974 - 1981, 2010.

[8] Richard Douville, Jean-Louis Le Roux, Stefano Secci, "A Service Plane over the PCE Architecture for Automatic Multidomain Connection-Oriented Services", IEEE Communications Magazine, Vol. 46, No. 6, pp. 94-102, Jun. 2008.

[9] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," IETF RFC 4655, Aug. 2006.

[10] J. Ash and J. L. Le Roux, "PCE Communication Protocol Generic Requirements," IETF RFC 4657, 2006.

[저 자 소 개]

이 원 혁 (Wonhyuk Lee)

2001년 2월 성균관대학교 학사
2003년 2월 성균관대학교 석사
2010년 8월 성균관대학교 박사
2003년 3월 ~ 현재
한국과학기술정보연구원
선임연구원

email : livezone@kisti.re.kr

노 민 기 (Minki Noh)

2009년 2월 성균관대학교 박사
2001년 ~ 현재
한국과학기술정보연구원
선임연구원

email : mknoh@kisti.re.kr

조 부 승 (Buseung Cho)

2000년 2월 성균관대학교 학사
2002년 8월 성균관대학교 석사
2012년 2월 성균관대학교 박사수료
2002년 9월~2005년 6월
고등기술연구원 근무
2005년 ~ 현재 한국과학기술정보
연구원 선임연구원

email : bscho@kisti.re.kr

김 태 연 (TaeYeon Kim)

1990년 2월 중앙대학교 학사
1992년 2월 중앙대학교 석사
2007년 2월 충북대학교 박사
1992년 2월 ~ 현재 한국전자통신
연구원 책임연구원

email : tykim@etri.re.kr

김 현 철 (Hyuncheol Kim)

1990년 2월 성균관대학교 학사
1992년 2월 성균관대학교 석사
2005년 8월 성균관대학교 박사
2006년 9월 ~ 현재 남서울대학교
컴퓨터학과 교수

email : hckim@nsu.ac.kr