

효율적인 가상 네트워크 대응 방안

우미애*

Efficient Virtual Network Mapping Method

Miae Woo*

요약

네트워크의 가상화는 최근에 등장한 클라우드 서비스, 데이터 센터 네트워크 등 새로운 서비스를 기존의 네트워크 환경에서 제공하기 위한 하나의 방편으로 인식되고 있다. 본 논문에서는 네트워크 가상화를 위한 가상 네트워크 대응 방안을 가상 노드의 위치, 자원, 가상 링크의 대역폭 요구조건을 고려하면서 효율적인 방안을 제안한다. 제안한 방안은 대응의 우선순위를 대역폭으로 설정하여 대응 가능한 실제 노드들 간의 경로가 존재하는 조합을 발견할 때까지 반복하는 방법을 사용한다. 모의실험 결과 제안한 방안이 가상 네트워크 대응 성공률은 떨어지지 않으면서도 수익률은 높고 실행시간도 빠른 효율적인 방안임을 확인하였다.

Key Words : Network Virtualization, Virtual network, Substrate network, Heuristic

ABSTRACT

Network virtualization is considered as an enabling technology to accommodate recently introduced new services such as cloud services and data center networks on top of the existing network environment. In this paper, an efficient virtual network mapping method is proposed which takes account of the location and resource requirements of virtual nodes and the bandwidth requirements of virtual links. The proposed method sets bandwidth as the priority metric for network mapping, and searches for a set of substrate nodes and paths that houses the virtual nodes and virtual links. Through the simulation experiments, it is verified that the proposed method provides better cost to revenue ratio and fast experiment time without degrading success rate of virtual network mapping.

I. 서론

클라우드 서비스, 데이터 센터 네트워크 등 새로운 서비스들로 인해 인터넷 트래픽은 지속적으로 증가하고 있어^[1], 네트워크 사업자는 기술적으로 제한된 자사의 자원으로 사용자를 수용할 방안을 찾아야 한다. 그러한 방안 중 하나가 물리적 네트워크를 여러 네트워크 사업자들이 공유하며 서비스를 수용할 수 있는 네트워크 가상화(network virtualization)다. 네트워크 가상화의 가장 기본적인 기능은 가상 네트워크에 있는 가상 노드(virtual node)는 실제 네트워크의 실제

노드(substrate node)로, 가상 링크(virtual link)는 실제 네트워크의 노드간의 경로(path)로 대응시키고, 실제 노드와 링크의 자원을 할당하는 것이다^[2]. 이렇게 가상 네트워크의 요구조건을 만족시키면서 가상 네트워크를 실제 네트워크에 대응시키는 문제는 NP-hard 문제다^[3].

지금까지 가상 네트워크를 위한 실제 네트워크의 자원 할당 문제에 대한 여러 가지 연구들이 진행되어 왔다^[2-6]. 이 문제를 효율적으로 해결하는 방법 중 하나가 휴리스틱(heuristic)을 사용하는 방법^[2-4]이다. Zhu가 제안한 방안^[3]은 휴리스틱 기반 방안 중 근간

* First Author : Sejong University Department of Electrical Engineering, mawoo@sejong.ac.kr, 종신회원
논문번호 : KICS2016-08-203, Received August 17, 2016; Revised October 31, 2016; Accepted November 7, 2016

을 제시한 연구로써, 노드와 링크에 가해지는 부하를 분산시키는 것을 목적으로 한다. 그러나 이 연구에서는 실제 네트워크의 노드와 링크의 용량이 무한대이고, 가상 노드와 가상 링크가 단위 처리능력과 단위 대역폭을 요구한다는 비현실적인 가정을 하였다. Nogueira 등이 제안한 방안⁴¹에서는 링크와 노드에 다양한 제약조건을 설정하여 가상 네트워크 요구를 좀 더 현실적으로 변경하고 대응할 수 있는 실제 노드가 적은 가상 노드에게 대응 우선순위를 부여하였다. 이 방안의 단점은 시간 복잡도가 높다는 것이다. Woo가 제안한 방안²¹에서는 가상 노드를 대응시킬 때, 위치를 고려하지 않고 인접 가상 노드들이 실제 네트워크에서도 가능하면 인접한 노드들로 클러스터를 형성할 수 있도록 탐색과정을 진행하도록 하였다. 이를 위하여 가상 노드의 CPU 요구량을 대응 우선순위로 삼고, 탐색 시 우선순위 높은 노드를 대상으로 먼저 대응 가능성을 타진하였다. 대부분의 경우 각 방안의 성능을 가상 네트워크 대응 성공률로 측정하는 데, 비용 대비 수익성에 대하여 비교한 결과도 있다⁴¹.

본 논문에서는 가상 네트워크를 실제 네트워크로 대응할 때, Woo가 제안한 방안²¹에서는 고려하지 않은 가상 노드들의 위치도 요구조건에 하나로 포함하는 경우에 대하여 연구하였다. 제안하는 방안에서는 대응 가능한 실제 노드의 선택 기준을 위치, 가상 노드의 CPU 요구량, 연결된 가상 링크의 대역폭 요구량을 모두 고려한다. 가상 노드의 대응 우선순위를 실제 네트워크의 경로를 최적화 하고자 대역폭을 우선순위로 설정한다는 점이 본 논문과 기존 연구와 다른 점이다. 제안한 방안의 성능은 가상 네트워크 대응 성공률, 비용 대비 수익성, 실제 대응 시간을 사용하여 분석한다. 모의실험 결과, 제안한 방안은 가상 네트워크 대응 성공률을 악화시키지 않으면서 수익성은 높이고 대응 시간은 빠르게 하는 효율적인 방안임이 확인되었다.

II. 일반 네트워크 모델

실제 네트워크 토폴로지는 무방향 그래프 $G_S = (N_S, L_S)$ 로 표시한다²¹. 여기서 N_S 는 실제 노드 집합, L_S 는 실제 링크 집합이다. 실제 노드 집합 N_S 과 관련된 특성인 A_S^N 는 실제 노드에서 제공 가능한 자원을 나타낸다. 실제 링크 집합 L_S 과 관련된 특성인 A_S^L 는 실제 링크에서 제공 가능한 자원을 나타낸다. 또한 P_S 는 실제 네트워크에 존재하는 모든 경로의 집합을 나타낸다.

가상 네트워크 요청은 실시간으로 도착한다고 가정한다. i 번째 도착한 가상 네트워크 요청, V_i 의 토폴로지도 무방향 그래프 $G_{V_i} = (N_{V_i}, L_{V_i})$ 로 나타낸다. N_{V_i} 와 L_{V_i} 는 각각 i 번째 가상 네트워크 요청의 가상 노드 집합과 가상 링크 집합이다. 가상 노드의 요구조건은 $R_{V_i}^N$ 으로 표시하고, 가상 링크의 요구조건은 $R_{V_i}^L$ 로 표시한다.

가상 네트워크 대응 문제는 G_{V_i} 를 G_S 의 부분집합에 가상 노드와 가상 링크의 요구조건인 $R_{V_i}^N$ 과 $R_{V_i}^L$ 을 만족하도록 대응시키는 것이다. 먼저, 가상 네트워크 안의 각 가상 노드는 실제 네트워크의 서로 다른 실제 노드로 대응되고, 실제 노드는 가상 노드의 요구조건을 식 (1)과 같이 만족시켜야 한다.

$$\begin{aligned} f_{N_i} : N_{V_i} &\rightarrow N_S \\ f_{N_i}(n) &\in N_S, \forall n \in N_{V_i} \\ A_S^N(f_{N_i}(n)) &\geq R_{V_i}^N(n), \forall n \in N_{V_i} \end{aligned} \quad (1)$$

가상 노드들을 연결하는 가상 링크는 해당 가상 노드에 대응된 실제 노드들을 연결하는 경로로 대응되고, 그 경로 상에 있는 모든 실제 링크는 가상 링크의 요구조건을 식 (2)와 같이 만족시켜야 한다.

$$\begin{aligned} f_{L_i} : L_{V_i} &\rightarrow P_S \\ f_{L_i}(\overline{nn'}) &\in P_S(f_{N_i}(n), f_{N_i}(n')), \forall \overline{nn'} \in L_{V_i} \\ A_S^L(l') &\geq R_{V_i}^L(\overline{nn'}), \\ &\forall l' \in P_S(f_{N_i}(n), f_{N_i}(n')), \overline{nn'} \in L_{V_i} \end{aligned} \quad (2)$$

식 (2)에서 $P_S(f_{N_i}(n), f_{N_i}(n'))$ 는 가상 링크 $\overline{nn'}$ 는 가상 노드 n 에 대응되는 실제 노드 $f_{N_i}(n)$ 과 가상 노드 n' 에 대응되는 실제 노드 $f_{N_i}(n')$ 사이의 경로를 나타낸다.

가상 네트워크 요청이 들어오면, 실제 네트워크가 식 (1)과 (2)에서 기술한 노드와 링크의 요구조건 충족 여부를 결정하여 선택된 실제 노드와 경로에 가상 네트워크가 요청한 자원을 할당하고, 할당된 자원량을 실제 노드와 실제 링크의 관련 특성에서 차감한다. 또한 가상 네트워크 V_i 의 유효기간이 만료되면, 할당했던 자원을 실제 네트워크로 반환하며, 그 양 만큼 실제 네트워크의 특성을 증가시킨다.

III. 관련연구

이 장에서는 본 논문에서 제안하는 방안과 비교 검토할 Nogueira 등이 제안한 방안^[4]에 대하여 설명한다. 이 방안에서는 노드 스트레스 요인으로 CPU 부하, 코어 수, 주파수, 가용 RAM 용량을, 링크 스트레스로는 할당된 대역폭 합을 고려한다.

가상 네트워크 요청이 오면, 먼저 모든 실제 링크와 노드에 대하여 링크 스트레스, 노드 스트레스를 계산한다. 실제 링크 스트레스는

$$S_{LS}(\overline{mm'}) = \sum_i \sum_{\overline{nm'}} R_{V_i}^L(\overline{nm'}),$$

$$\overline{nm'} \in L_{V_i}, \overline{mm'} \in P_S(f_{N_i}(n), f_{N_i}(n'))$$

로 정의한다. 즉, 실제 링크 스트레스는 아직 서비스 중인 가상 네트워크를 위해 설정된 경로 상에 해당 링크가 있는 경우, 할당된 가상 링크의 요구조건의 합이다. 실제 노드 m 의 노드 스트레스는

$$S_N(m) = \frac{\sum_i \sum_{n \in N_{V_i}} A(n, m)}{\xi + RAM \cdot CPU_{Freq} \cdot (N.CPU - Load)}$$

로 정의한다. $A(n, m)$ 은 가상 노드 n 이 실제 노드 m 에 대응되면 1, 아니면 0의 값을 갖는 지시함수다.

가상 네트워크 요청 V_i 에 있는 가상 노드 $n \in N_{V_i}$ 에 대하여, 요청된 제약조건, 즉 위치, CPU 수, CPU 주파수, 가용 RAM 용량 등을 수용할 수 있는 실제 노드들을 n 의 후보군($n.Candidates$)에 추가한다. 모든 가상 노드들을 후보 수가 적은 가상 노드가 먼저 대응되도록 정렬한 후 노드 대응과 경로 선택을 한다. 가상 네트워크상에서 가상 노드 n 의 인접 가상 노드 n' 이 있다고 하자. 가상 노드 n 의 대응 가능 후보인 실제 노드 m 과 가상 노드 n' 의 대응 가능 후보인 실제 노드 m' 사이의 경로 비용을 계산한다. 실제 노드 m 의 노드 잠재력 $\pi(m)$ 는 가상 노드 n 의 인접 가상 노드들의 대응 가능 후보군에 있는 모든 실제 노드들 간의 경로 비용($D(m, m')$)의 합과 실제 노드 m 의 노드 스트레스 $S_N(m)$ 의 곱을 가상 노드 n 의 대응 후보 수($count(n.Candidates)$)로 나눈 값으로 정의한다.

$$\pi(m) = \pi(m) + \frac{\sum_{m' \in V_c} D(m, m') \cdot S_N(m)}{count(n.Candidates)}$$

가상 노드 n 의 대응 가능 후보군의 모든 실제 노드

에 대하여 노드 잠재력을 구한 후, 그 중 가장 작은 노드 잠재력을 갖는 실제 노드를 선택한다. 즉,

$$f_{N_i}(n) = m, \pi(m) = \min \{ \pi(x) | x \in n.Candidates \}.$$

마지막으로 선택된 실제 노드들 간에 가상 링크에 대응하는 경로를 설정한다.

IV. 제안 방안

이 장에서는 먼저 제안하는 방안에서 추가적으로 고려하는 대응 문제를 네트워크 모델 관점에서 기술하고, 제안 방안의 알고리즘을 제시한다.

4.1 제안 방안에서 고려하는 대응 문제

본 논문에서 고려하는 가상 노드의 요구조건은 위치에 대한 요구조건과 노드의 자원에 대한 요구조건이다. 따라서 이 후 식에서는 $R_{V_i}^N$ 은 노드의 자원에 대한 요구조건만을 표시하는 것으로 한다. 위치에 대한 요구조건은 가상 노드 n 의 위치를 (x_n, y_n) 좌표 값으로 지정한다. 가상 노드 n 과 대응 가능한 실제 노드 $f_{N_i}(n)$ 의 거리($dist$)는 가상 노드의 좌표 값 (x_n, y_n) 과 $f_{N_i}(n)$ 의 좌표 값 $(x_{f_{N_i}(n)}, y_{f_{N_i}(n)})$ 간의 거리로 계산하고, 그 거리가 일정 범주(δ) 안에 들어야 한다. 또한 본 논문에서는 가상 링크를 실제 네트워크에서 하나의 경로로 제공하도록 하여 W_{00} 가 제안한 방안^[2]에서와 같이 가상 노드에 연결된 가상 링크의 대역폭 요구사항의 합도 대응할 실제 노드를 선택하는 최소 필요조건으로 삼는다. 가상 링크의 대역폭 요구조건을 $R_{V_i}^L(\overline{nm'})$ 라 하자. n 과 n' 는 가상 링크로 연결되는 인접 가상 노드이다. 가상 노드 n 에서 필요한 대역폭 요구조건 $bw_R(n)$ 은

$$bw_R(n) = \sum_{\overline{nm'} \in L_{V_i}} R_{V_i}^L(\overline{nm'}), \text{ for } n, n' \in N_{V_i} \quad (3)$$

로 정의한다. 실제 노드 m 과 m' 사이의 실제 링크의 가용 대역폭은 $A_S^L(\overline{mm'})$ 라고 하자. 실제 노드 m 에서 제공 가능한 가용 대역폭 $bw_A(m)$ 은

$$bw_A(m) = \sum_{\overline{mm'} \in L_S} A_S^L(\overline{mm'}), \text{ for } m, m' \in N_S \quad (4)$$

로 정의한다. 그러면 대응 가능한 실제 노드는 가상 노드의 요구조건을 식 (5)과 같이 만족시켜야 한다.

$$\begin{aligned}
 & f_{N_i} : N_{V_i} \rightarrow N_S \\
 & f_{N_i}(n) \in N_S, \forall n \in N_{V_i} \\
 & dist(f_{N_i}(n), n) \leq \delta, \forall f_{N_i}(n) \in N_S, n \in N_{V_i} \quad (5) \\
 & A_S^N(f_{N_i}(n)) \geq R_{V_i}^N(n) \\
 & bw_A(f_{N_i}(n)) \geq bw_R(n)
 \end{aligned}$$

4.2 제안 알고리즘

본 논문에서 제안하는 가상 네트워크 대응 방안에서는 가상 노드의 대응 우선순위를 실제 네트워크의 경로를 최적화 하고자 대역폭을 우선순위로 설정한다. 따라서 가상 네트워크 요청 V_i 가 들어오면, 가상 노드들을 식 (3)의 대역폭 요구조건에 따라 정렬하여 정렬 리스트 $N_{<V_i>}$ 를 구성한다. 각 가상 노드에 대하여 식 (5)를 만족시키는 실제 노드들의 집합을 구한 후, 집합 내의 실제 노드들도 식 (4)의 가용 대역폭에 따라 정렬 리스트를 구성한다.

가상 노드와 대응 가능한 실제 노드 집합을 모두 정렬한 후, 각 가상 노드별로 대응 가능한 실제 노드 정렬 리스트의 제일 앞에 있는 실제 노드들을 대상으로 가상 링크에 대응하는 경로를 설정할 수 있는지 확인한다. 경로 설정이 가능하면, 해당 가상 네트워크의 대응이 성공하는 것이다. 만일 이 조합에 대한 경로설정이 불가능하면, 정렬된 가상 노드 리스트의 제일 끝에 있는 노드에 대하여 다른 실제 노드를 선택하여 경로설정 여부를 확인한다. 이러한 과정은 경로설정이 가능할 때까지, 정렬리스트에서 순차적으로 다른 실제 노드를 고려하며 반복적으로 확인한다.

그림 1은 대역폭 조건에 따라 정렬한 가상 노드의 리스트와 각 가상 노드의 요구조건을 만족시키는 대응 가능한 실제 노드들의 정렬된 리스트의 예를 보여 준다. 가상 네트워크 요청에 가상 노드가 4개가 있다면, 가상 노드들은 $bw_R(<n_1>) \geq bw_R(<n_2>) \geq$

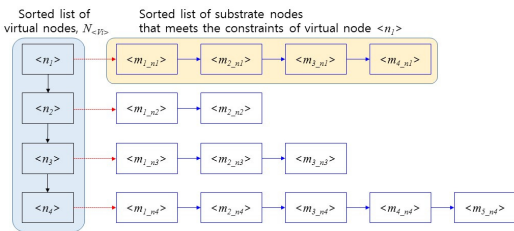


그림 1. 우선순위로 정렬된 가상 노드의 정렬 리스트와 각 가상 노드에 대응 가능한 실제 노드의 정렬 리스트
Fig. 1. An example of sorted list of virtual nodes and the sorted lists of mappable substrate nodes for each virtual node

$bw_R(<n_3>) \geq bw_R(<n_4>)$ 순으로 정렬하여 리스트에 저장한다. 가상 노드 $<n_1>$ 에 대하여 식 (5)를 만족시키는 실제 노드가 4개가 있다면, 이들 실제 노드들을 $bw_A(<m_{1-n_1}>) \geq bw_A(<m_{2-n_1}>) \geq bw_A(<m_{3-n_1}>) \geq bw_A(<m_{4-n_1}>)$ 의 조건으로 정렬하여 리스트를 구성한다. 그림 1에서는 가상 노드 $<n_i>$ 에 대응 가능한 실제 노드들을 구별하기 위하여 $<m_{j-n_i}>$ 의 표현을 사용하고, 실제로 다른 가상 노드에 대응 가능한 실제 노드 $<m_{j-n_i}>$ 와 $<m_{l-n_k}>$ 은 같은 노드일 수 있다.

제안하는 방안은 정렬 리스트 $N_{<V_i>}$ 에 있는 가상 노드 순서대로 실제 노드에 대응 가능한지 탐색한다. 그림 1의 예에서 만일 $<m_{1-n_1}>, <m_{1-n_2}>, <m_{1-n_3}>, <m_{1-n_4}>$ 가 서로 다른 실제 노드라면, 이 노드들 간 가상 링크를 대응시킬 실제 경로가 설정 가능한지 확인한다. 만일 경로가 제공될 수 있으면 해당 가상 네트워크 요청을 수락한다. 만일 실제 네트워크 상 경로를 제공할 수 없다면 가장 우선순위가 낮은 가상 노드를 다른 실제 노드로 대응 가능한지 살펴보게 된다. 즉, $<m_{1-n_1}>, <m_{1-n_2}>, <m_{1-n_3}>, <m_{1-n_4}>$ 의 조합으로 대응이 불가능 할 경우 $<m_{1-n_1}>, <m_{1-n_2}>, <m_{1-n_3}>$,

```

Location_constrained_virtual_network_embedding
( $G_S, A_S^N, A_S^L, G_{V_i}, R_{V_i}^N, R_{V_i}^L$ )
1. Make a sorted list of virtual nodes  $N_{<V_i>}$ 
2. For each  $<n_i>$  in  $N_{<V_i>}$ ,
    2.1 Make a sorted list of substrate nodes that meet constraints given in Eq. (5).
    2.2 Return FAILED if list in 2.1 is empty.
3.  $i = 1$ 
4. While (there is  $<m_{i-n_i}>$  in the list) do
    4.1 If (Check_mappability( $G_S, A_S^L, G_{V_i}, R_{V_i}^L, N_{<V_i>}, 2$ )),
        return SUCCEEDED.
    4.2 Else, increase  $i$  and continue
5. Return FAILED.

Check_mappability( $G_S, A_S^L, G_{V_i}, R_{V_i}^L, N_{<V_i>}, j$ )
1.  $k = 1$ 
2. While (there is  $<m_{k-n_i}>$  in the list) do
    2.1 If  $<m_{k-n_i}>$  is used for  $<n_x>, 1 \leq x < j$ ,
        increase  $k$  and continue.
    2.2 If  $<n_j>$  is last node in  $N_{<V_i>}$ 
        2.2.1 If (Check_Path( $G_S, A_S^L, G_{V_i}, R_{V_i}^L, N_{<V_i>}, j$ )),
            return SUCCEEDED.
        2.2.2 Else if (Check_mappability
            ( $G_S, A_S^L, G_{V_i}, R_{V_i}^L, N_{V_i}, j+1$ )),
            return SUCCEEDED.
        2.2.3 Else increase  $k$  and continue.
3. Return FAILED.
    
```

그림 2. 제안 가상 네트워크 대응 알고리즘 의사코드
Fig. 2. The proposed virtual network embedding algorithm pseudo code

$\langle m_{2_n_4} \rangle$ 의 조합에 대한 경로 설정이 가능한지 보고, 경로 설정이 가능할 때 까지, $\langle n_4 \rangle$ 에 대응 가능한 실제 노드 $\langle m_{j_n_4} \rangle$ 를 대상으로 순차적으로 알아본다. 만일 $\langle n_4 \rangle$ 에 대한 탐색이 실패하면 다음 단계에서는 $\langle n_3 \rangle$ 에 대한 실제 노드를 $\langle m_{2_n_3} \rangle$ 로 변경하고 $\langle m_{1_n_4} \rangle$ 부터 시작하여 대응 가능성을 확인한다. 이러한 절차를 대응이 가능할 때까지, 혹은 $\langle n_1 \rangle$ 에 대응 가능한 실제 노드의 리스트가 끝날 때 까지 반복적으로 확인한다.

그림 2는 제안한 가상 네트워크 대응 방안에 대한 알고리즘을 의사코드로 정리한 것이다. 각 가상 노드에 대한 대응 가능한 실제 노드와 경로 설정을 알아보는 과정은 재귀적으로 처리한다.

V. 성능분석

제안한 가상 네트워크 대응 방안의 성능은 모의실험을 통하여 평가하였다. 평가 지표로는 가상 네트워크 대응 성공률, 비용 대 수익 비율, 평균 실행시간을 채택하였다. 비교 대상으로는 기존 연구 중 Nogueira 등이 제안한 방안^[4]를 선정하였다.

5.1 모의실험 환경

모의실험을 위해서 실제 네트워크와 가상 네트워크를 생성하였다. 실제 네트워크 토폴로지 생성에는 100×100 공간 안에서 Waxman 랜덤 토폴로지^[7] ($\alpha = 0.4, \beta = 0.4$)^[4] 생성 방법을 사용하였다. 가상 네트워크 요청은 포아송(Poisson) 프로세스 ($\lambda = 0.1$)

표 1. 실제와 가상 네트워크 요소 생성에 사용한 값
Table 1. Values used to generate network elements for both substrate network and virtual network

Network elements		Value
Substrate nodes	Number	30 (fixed)
	CPU capacity	UNI[50, 100]
	Location	x-axis: UNI[0, 100], y-axis: UNI[0, 100]
Substrate links	Bandwidth	UNI[1G, 2G]
Virtual nodes	Number	UNI[2, MaxNodeNo] MaxNodeNo: 4~8.
	CPU capacity requirement	UNI[10, 20], UNI[20, 30]
	Location requirement	x-axis: UNI[0, 100], y-axis: UNI[0, 100]
Virtual links	Bandwidth	UNI[100M, 200M], UNI[300M, 400M]

모델을 사용하여 랜덤하게 도착하도록 하였다^[3]. 각각의 가상 네트워크가 신청하는 서비스 기간은 지수 (Exponential) 랜덤 변수 ($\mu = 250$)를 사용하고, 가상 노드들을 연결시키는 가상 링크는 연결 확률은 0.5로 설정하여 생성하였다. 실제 네트워크를 구성하는 실제 노드와 링크의 특성, 가상 네트워크의 가상 노드와 가상 링크의 요구조건 및 특성을 지정하는 방법을 표 1에 정리하였다. 가상 네트워크를 대응시킬 때, 가상 노드의 지정 위치의 특정 변경 안에 있는 실제 노드에 대응시키는 데, 반경은 5로 지정하였다. 총 모의실험 시간은 10,000으로 설정하였다.

5.2 모의실험 결과

모의실험 결과를 평가하는 데 있어 가상 네트워크 대응 성공률, 비용 대 수익 비율, 모의실험의 수행시간을 평가 지표로 선택하였다.

먼저, 가상 네트워크 대응 성공률을 가상 노드의 CPU 요구 조건에 따른 결과가 그림 3과 그림 4에 있다. 그림 3에서 볼 수 있듯이 가상 노드의 CPU 요구 조건이 UNI[10, 20]인 경우, 가상 링크의 대역폭 요구 조건이 UNI[100M, 200M]이면 본 논문에서 제안한 방안이나 Nogueira 등이 제안한 방안의 성공률은 거의 비슷하였다. 가상 링크의 대역폭 요구조건이 UNI[300M, 400M]로 증가하더라도 제안한 방안의 가상 네트워크 대응 성공률은 가상 링크의 대역폭 요구

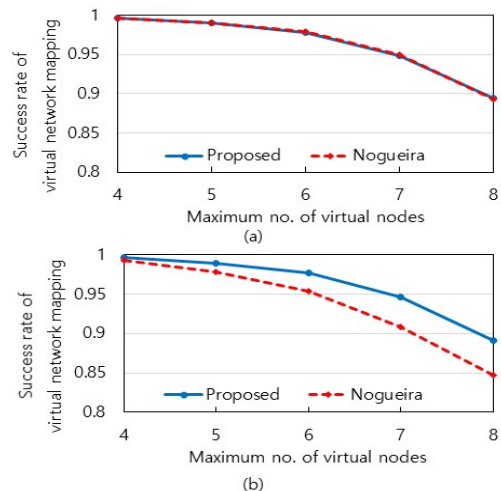


그림 3. CPU 요구조건이 UNI[10, 20]인 경우 가상 네트워크 대응 성공률 (a) 대역폭 요구조건: UNI[100M, 200M] (b) 대역폭 요구조건: UNI[300M, 400M]
Fig. 3. Success rate of virtual network mapping when the CPU requirement is UNI[10, 20] (a) bandwidth requirement: UNI[100M, 200M] (b) bandwidth requirement: UNI[300M, 400M]

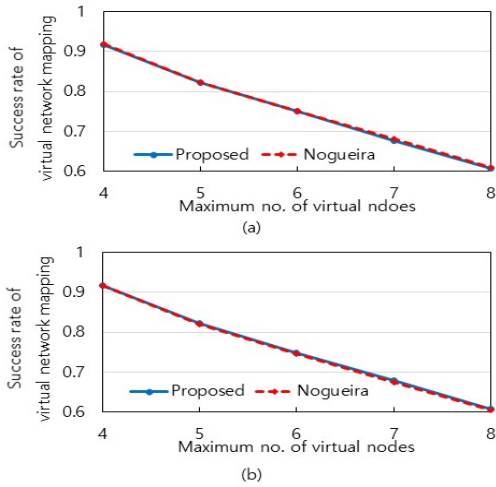


그림 4. CPU 요구조건이 UNI[20, 30]인 경우 가상 네트워크 대응 성공률 (a) 대역폭 요구조건: UNI[100M, 200M] (b) 대역폭 요구조건: UNI[300M, 400M]
 Fig. 4. Success rate of virtual network mapping when the CPU requirement is UNI[20, 30] (a) bandwidth requirement: UNI[100M, 200M] (b) bandwidth requirement: UNI[300M, 400M]

조건이 UNI[100M, 200M]인 경우와 최대 가상 노드 수에 따라 거의 비슷하였다. 그에 반하여 Nogueira 등이 제안한 방안의 경우 가상 링크의 대역폭 요구조건의 증가가 가상 네트워크 대응 성공률을 떨어뜨렸다. 이는 CPU와 대역폭 요구조건 중 대역폭 요구조건이 가상 네트워크 대응에 좀 더 제한요건으로 동작하는 경우 대역폭을 대응 우선순위로 하는 제안 방안의 장점이 나타나는 것이다. 가상 노드의 CPU 요구조건이 UNI[20, 30]으로 증가하면 그림 4에서 볼 수 있듯이 가상 링크의 대역폭 요구조건이 증가하더라도 두 방안 모두 비슷한 가상 네트워크 대응 성공률을 기록하였다. 이는 실제 노드의 CPU 용량을 UNI[50, 100]으로 설정하였기 때문에 CPU 요구조건이 가상 네트워크 대응 성공에 더 많은 영향을 미치는 것으로 해석할 수 있다.

다음으로는 가상 네트워크 대응이 성공한 경우 비용 대 수익 비율을 알아본다. 가상 네트워크 요청을 수용하여 발생하는 수익은 가상 네트워크에서 필요로 하는 대역폭의 합으로 정의한다. 가상 네트워크 요청을 수용하는 데 발생하는 비용은 가상 네트워크 요청안의 모든 가상 링크를 대응하기 위하여 사용되는 실제 네트워크의 경로에서 필요한 대역폭의 합으로 정의한다⁶⁾. 따라서 발생 비용은 실제 네트워크상의 경로의 길이가 길어지면 비용도 증가하게 된다.

비용 대 수익의 비교 결과는 그림 5와 그림 6에 있다. 두 그림에서 볼 수 있듯이 본 논문에서 제안한 방안이 Nogueira 등이 제안한 방안보다 항상 더 높은 비용 대 수익률을 기록하였다. 이는 제안한 방안이 대응 우선순위로 대역폭을 선택했기 때문에, 설정된 경로 길이가 상대적으로 짧기 때문이다. 두 방안 공통적으

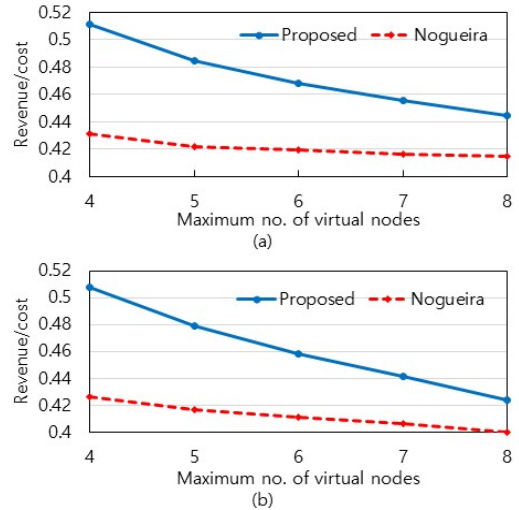


그림 5. 가상 노드의 CPU 요구조건이 UNI[10, 20]인 경우 비용 대 수익 비율 (a) 대역폭 요구조건: UNI[100M, 200M] (b) 대역폭 요구조건: UNI[300M, 400M]
 Fig. 5. Cost to revenue ratio when the CPU requirement is UNI[10, 20] (a) bandwidth requirement: UNI[100M, 200M] (b) bandwidth requirement: UNI[300M, 400M]

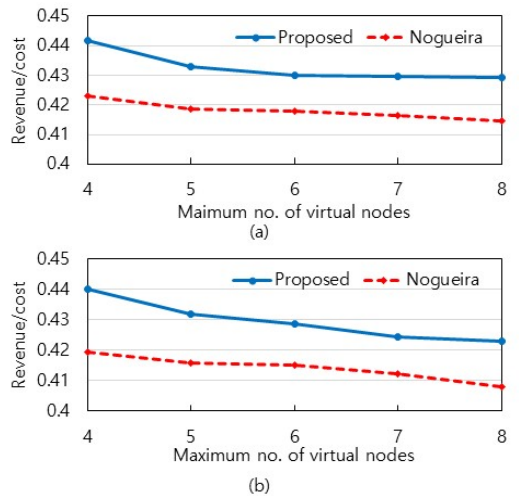


그림 6. CPU 요구조건이 UNI[20, 30]인 경우 비용 대 수익 비율 (a) 대역폭 요구조건: UNI[100M, 200M] (b) 대역폭 요구조건: UNI[300M, 400M]
 Fig. 6. Cost to revenue ratio when the CPU requirement is UNI[20, 30] (a) bandwidth requirement: UNI[100M, 200M] (b) bandwidth requirement: UNI[300M, 400M]

VI. 결 론

본 논문에서는 가상 노드의 위치, CPU 요구조건, 가상 링크의 대역폭 요구조건을 만족시키는 가상 네트워크 대응 방안을 제안하였다. 기존의 휴리스틱 방안들과의 차이점은 대응할 때 요청된 가상 네트워크를 구성하는 가상 노드의 우선순위를 해당 가상 노드에 연결된 가상 링크의 대역폭 요구조건의 합으로 정하여 우선순위가 높은 노드 먼저 대응하는 실제 노드를 선택하고, 선택된 실제 노드들 간에 가상링크를 연결할 경로를 탐색하는 점이다. 만일 해당 경로가 없는 경우 우선순위가 낮은 가상 노드를 대응 가능한 다른 실제 노드로 바꾸어 경로가 있는 노드 집합이 형성될 때 까지 반복적으로 탐색을 한다. 이러한 제안 방안의 성능을 가상 네트워크 대응 성공률, 비용 대 수익 비율, 모의실험의 수행시간으로 비교 분석한 결과, 제안한 방안이 가상 네트워크 대응 성공률은 떨어지지 않으면서도 수익률은 높고 실행시간도 빠른 효율적인 방안임을 확인하였다.

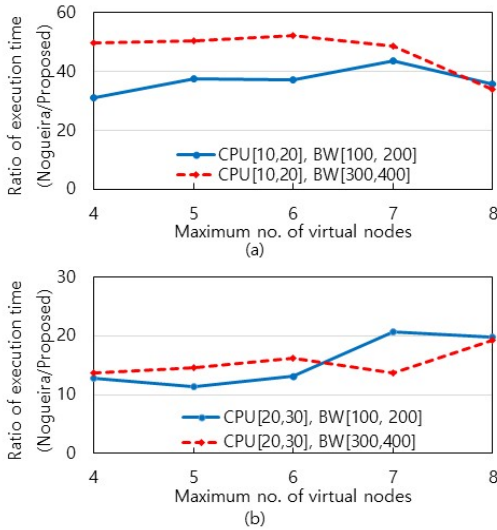


그림 7. 평균 실행시간 비교 (a) CPU 요구조건: UNI[10, 20]인 경우 (b) CPU 요구조건: UNI[20, 30]인 경우
Fig. 7. Comparison of average execution time (a) CPU requirement: UNI[10, 20] (b) CPU requirement: UNI[20, 30]

로 가상 링크의 대역폭 요구량이 높아지면, 해당 가상 링크를 수용하기 위한 실제 네트워크상의 경로가 길어질 수 있어서 수익률은 감소할 수밖에 없다. 또한 가상 네트워크 대응 성공률에서도 언급한 바와 같이 가상 노드의 CPU 요구조건이 높아지면 대응할 수 있는 실제 노드가 줄어들기 때문에 짧은 경로의 선택권이 줄어들게 되기 때문에 수익률이 감소할 수밖에 없다.

마지막으로 실제 모의실험을 실행하는 데 걸리는 시간을 검토해 보았다. 실제 실행 시간 값은 그 자체로는 의미가 크지 않기 때문에, 본 논문에서 제안한 방안과 Nogueira 등이 제안한 방안의 실행시간의 비율을 그림 7에 제시하였다. Nogueira 등이 제안한 방안이 조건에 따라 본 논문에서 제안한 방안보다 10~50배 정도 실행시간이 더 걸렸다. Nogueira 등이 제안한 방안은 가상 노드 하나를 대응시킬 실제 노드를 정하기 위해서 대응 가능 후보군 노드 모두에 대하여 노드 잠재력을 구해야 하고, 노드 잠재력은 인접 노드 후보군 모두에 대한 경로 비용을 계산해야하기 때문에 시간 복잡도가 높다. 제안한 방안이 일관적으로 Nogueira 등이 제안한 방안보다 실행시간이 매우 빠르게 나타났다는 점은 대응 우선순위를 대역폭으로 지정함으로써 가상 링크를 수용할 수 있는 실제 네트워크상의 경로 탐색시간이 평균적으로 빠르다는 것을 의미한다.

References

- [1] B. Chen, J. Zhang, W. Xie, J. Jue, Y. Zhao, and G. Shen, "Cost-effective survivable virtual optical network mapping in flexible bandwidth optical networks," *J. Lightwave Technol.*, vol. 34, no. 10, pp. 2398-2412, 2016.
- [2] M. Woo, "A study on node selection strategy for the virtual network embedding," *J. KICS*, vol. 39b, no. 8 pp. 491-498, Aug. 2014.
- [3] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc. INFOCOM 2006*, pp. 1-12. 2006.
- [4] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," in *Proc. ISCC 2011*, pp. 438 -444, 2011.
- [5] C. Pyoung and S. Baek, "Virtual network mapping algorithm for minimizing piecewise linear cost function," *J. KICS*, vol. 41, no. 6 pp. 672-677, Jun. 2016.
- [6] M. Woo, "A comparison of cost to revenue according to mapping priority for virtual network embedding," in *Proc. KICS Int. Conf.*

Commun., pp. 1513-1514, Jeju-do, Korea, Jun. 2015.

- [7] B. Waxman, "Routing of multipoint connections," *IEEE JSAC*, vol. 6, no. 9, pp. 1617-1622, Dec. 1988.

우 미 애 (Miae Woo)



1985년 2월 : 연세대학교 전자공학과 졸업

1991년 12월 : 미국 Purdue 대학교 전기컴퓨터공학과 석사

1995년 12월 : 미국 Purdue 대학교 전기컴퓨터공학과 박사

1985년~1989년 : DACOM 연구원

1996년~1998년 : 삼성전자(주) 수석연구원

1998년~현재 : 세종대학교 전자정보통신공학과 교수
<관심분야> 네트워크 구조 및 프로토콜, 이동 네트워크