

가상 네트워크 대응 시 노드 선택 기준에 대한 고찰

우 미 애[°]

A Study on Node Selection Strategy for the Virtual Network Embedding

Miae Woo[°]

요 약

현재의 인터넷은 경직성으로 인해 새로운 서비스 요구를 수용하기 어렵다. 이러한 문제를 해결하기 위한 방안 중 하나가 네트워크 가상화다. 본 논문에서는 네트워크 가상화를 위한 휴리스틱 가상 네트워크 대응 방안을 제안한다. 제안하는 방안에서는 가상 네트워크를 실제 네트워크에 대응시킬 때, 실제 노드가 가상 링크 조건을 만족시킬 수 있는 가능성이 있는 지 확인하고 가상 노드와 실제 노드에 우선순위를 부여하여 우선순위에 따라 대응 순서를 정한다. 또한 대응되는 실제 노드가 클러스터를 형성할 수 있도록 한다. 제안하는 방안의 성능을 시간 복잡도와 가상 네트워크 수락율로 평가한다.

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

ABSTRACT

Due to the ossification of current Internet, it is hard to accommodate new service requirements. One of the solutions to this problem is network virtualization. In this paper, we propose a heuristic virtual network embedding method for network virtualization. The proposed method checks whether the candidate substrate nodes in the substrate network have the possibility of satisfying virtual link requirements. It gives priority to the virtual nodes and the substrate nodes, and embeds the node with higher priority first. Also, the proposed method tries to cluster the mapped substrate nodes if possible. We evaluate the performance of the proposed method in terms of time complexity and virtual network acceptance rate.

I. 서 론

미래 인터넷워킹 패러다임 중 가장 중요한 것은 새로운 서비스를 수용할 수 있는 다양성을 제공하는 것이다. 그러나 현재의 인터넷은 기술적인 면뿐만 아니라 사업자들 간의 이해관계 때문에 새로운 서비스를 유연하게 수용하지 못하고 있다. 이러한 문제를 해결하기 위한 방안 중 하나가 네트워크 가상화(network virtualization)다^{1,2)}. 네트워크를 가상화하면, 가상 네

트워크(virtual network) 상에서 여러 가지 프로토콜을 사용하여 다양한 통신 서비스를 제공할 수 있다. 이렇게 함으로써 지금의 네트워크 구조를 기술적으로 개량하거나 개선하지 않아도 새로운 서비스를 제공할 수 있다³⁾.

이러한 장점 때문에 가상 네트워크에 관심을 갖는지는 상당히 되었다. 하지만 실제 네트워크(substrate network)에 네트워크 가상화를 적용하려면 아직 기술적으로 해결해야 할 여러 가지 문제들이 있다. 네트워

[°] First and Corresponding Author : Sejong University Department of Information and Communication Engineering, mawoo@sejong.ac.kr, 중신회원
 논문번호 : KICS2014-05-003, Received May 23, 2014; Revised August 2, 2014; Accepted August 11, 2014

크 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 경로로 대응시키고, 실제 노드와 링크의 자원을 할당하는 것이다. 가상 네트워크의 요구조건을 만족시키면서 가상 네트워크를 실제 네트워크에 대응시키는 문제는 NP-hard 문제다⁴⁾.

지금까지 가상 네트워크를 위한 실제 네트워크의 자원 할당 문제에 대한 여러 가지 연구들이 진행되어 왔다. 이 문제를 효율적으로 해결하는 방법 중 하나가 휴리스틱(heuristic) 방안⁴⁻⁸⁾이다. Zhu가 제안한 방안⁴⁾은 휴리스틱 기반 방안 중 근간을 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 방안⁶⁾에서는 현실적으로 변경하여 링크와 노드에 다양한 제약조건을 설정하였다. 실제 노드들 중 노드의 제약조건을 수용할 수 있는 후보 노드군을 구하고, 후보 노드의 노드 잠재력을 선택 기준으로 삼았다. 그러나 선택 과정에서 실질적인 자원할당을 고려하지 않아, 선택 후 대응 시 실패하는 경우도 발생한다. Razzaq 등이 제안한 방안⁷⁾에서는 가상 네트워크 대응 시 실제 네트워크에서의 병목 노드를 줄이려 하였다. 이 방안은 가상 네트워크가 조밀하게 연결된 경우에는 성능이 좋지 않았다. Yu 등이 제안한 방안⁸⁾은 트래픽을 분할할 수 있다는 가정 하에 하나의 가상링크를 실제 네트워크에서는 여러 경로에 나누어 할당하여 좀 더 높은 가상 네트워크를 수용하여 이익을 많이 창출하려 했다. Zhu가 제안한 방안⁴⁾과 Yu 등이 제안한 방안⁸⁾을 비교 분석한 연구도 있다⁵⁾.

본 논문에서는 가상 네트워크를 실제 네트워크로 대응할 때, 노드 선택 기준에 대하여 두 가지 관점에서 고찰한다. 먼저, 가상 노드를 대응할 수 있는 실제 노드에 대한 고찰이다. 지금까지 제안된 휴리스틱 방안들은 가상 네트워크 대응 시 노드 선택과 링크를 위한 경로 설정을 완전히 분리하여 두 단계로 진행하였다. 본 논문에서는 고려하는 실제 노드에 연결된 링크가 가상링크의 요구조건을 만족시킬 수 있는지의 여부도 같이 확인함으로써 가상 네트워크 대응 시 고려하는 실제 노드를 줄임으로써 탐색 공간을 제한시키는 효과를 볼 수 있다. 두 번째는 가상 네트워크를 실제 네트워크에 대응할 때, 어느 노드부터 대응시킬 것인가, 즉 대응의 우선순위 설정에 관한 점이다. 본 논

문에서는 대응할 노드의 우선순위를 가상 노드와 실제 노드 모두 고려하여, 가상 노드의 우선순위로 CPU 요구량을, 실제 노드의 우선순위는 잔여 CPU 용량으로 설정한다. 가상 노드를 대응시킬 때, 인접 가상 노드들이 실제 네트워크에서도 가능하면 인접한 노드들로 클러스터를 형성할 수 있도록 탐색과정을 진행한다. 제안한 방안의 성능은 시간 복잡도와 가상 네트워크 수락율을 사용하여 분석한다.

II. 가상 네트워크 대응 방안

이 절에서는 일반적인 가상 네트워크 대응 문제를 먼저 기술하고, 본 논문에서 제안하는 가상 네트워크 대응 방안을 소개한다.

2.1 일반적인 가상 네트워크 대응 문제

실제 네트워크 토폴로지는 무방향 그래프 $G_S = (N_S, L_S)$ 로 표시한다. 여기서 N_S 는 실제 노드 집합을, L_S 는 실제 링크 집합을 뜻한다. 실제 노드 집합 N_S 와 관련된 특성인 A_S^N 는 실제 노드에서 제공 가능한 자원을 나타내며, 실제 링크 집합 L_S 와 관련된 특성인 A_S^L 는 실제 링크에서 제공 가능한 자원을 나타낸다. 또한 P_S 는 실제 네트워크에 존재하는 모든 경로(path)의 집합을 나타낸다⁸⁾.

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 $R_{V_i}^N$, 가상 링크의 요구조건은 $R_{V_i}^L$ 로 표시한다.

가상 네트워크 대응 문제는 G_{V_i} 를 G_S 의 부분집합에 다음과 같은 조건을 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 대응되

고, 그 경로 상에 있는 모든 실제 링크는 가상 링크의 요구조건을 식 (2)와 같이 만족시켜야 한다.

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

식 (2)에서 가상 링크 $\overline{nm'}$ 는 가상 노드 n 에 대응되는 실제 노드 $f_{N_i}(n)$ 과 가상노드 n' 에 대응되는 실제 노드 $f_{N_i}(n')$ 사이의 경로 $P_S(f_{N_i}(n), f_{N_i}(n'))$ 로 대응됨을 나타낸다.

가상 네트워크 요청이 들어오면, 실제 네트워크가 식 (1)과 (2)에서 기술한 노드와 링크의 대응을 결정하여 선택된 실제 네트워크의 노드와 경로에 가상 네트워크가 요청한 자원을 할당하고, 할당된 자원량을 실제 노드와 실제 링크의 관련 특성에서 차감한다. 즉, 가상 네트워크 V_i 요청이 도착해서 실제 네트워크에 대응되면, 실제 네트워크의 자원은 식 (3)과 같이 차감된다.

$$\begin{aligned}
 A_S^N(f_{N_i}(n)) &= A_S^N(f_{N_i}(n)) - R_{V_i}^N(n), \\
 & \quad \forall f_{N_i}(n), n \in N_{V_i} \\
 A_S^L(l') &= A_S^L(l') - R_{V_i}^L(\overline{nm'}), \\
 & \quad \forall l' \in P_S(f_{N_i}(n), f_{N_i}(n')), \overline{nm'} \in L_{V_i}
 \end{aligned} \quad (3)$$

또한 가상 네트워크 V_i 의 유효기간이 만료되면, 할당했던 자원을 실제 네트워크로 반환하며, 그 양 만큼 실제 네트워크의 특성을 증가시킨다.

2.2 제안하는 가상 네트워크 대응 방안

여기서는 본 논문에서 제안하는 가상 네트워크 대응 방안에 대하여 먼저 노드 선택 기준을 설명하고, 제안 알고리즘의 의사코드를 제시한다.

2.2.1 노드 선택 기준

본 논문에서는 가상 노드의 요구조건인 $R_{V_i}^N$ 에는 CPU 용량을, 가상 링크의 요구조건인 $R_{V_i}^L$ 에는 대역폭을 고려한다. 따라서 실제 노드의 특성은 잔여 CPU 용량, 실제 링크의 특성은 잔여 대역폭으로 설정한다.

예를 들어 그림 1과 같은 토폴로지를 갖는 i 번째 가상 네트워크 요청 V_i 가 왔다고 하자. 이 요청의 가

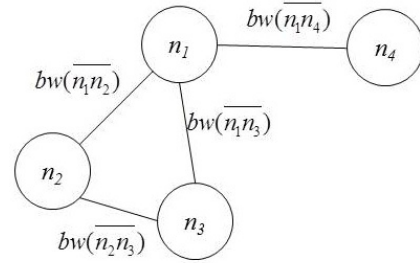


그림 1. 가상 네트워크 토폴로지
Fig. 1. A virtual network topology

상 노드 집합은 $N_{V_i} = \{n_1, n_2, n_3, n_4\}$ 로, 가상 링크 집합은 $L_{V_i} = \{\overline{n_1n_2}, \overline{n_1n_3}, \overline{n_1n_4}, \overline{n_2n_3}\}$ 로 표시할 수 있다. 그림 1에서 가상 링크의 요구조건 $R_{V_i}^L$ 인 가상 링크의 대역폭 요구조건을 $bw(\overline{n_i n_j})$ 로 표시한다.

가상 노드 n_1 에 대응될 수 있는, 식 (1)의 조건을 만족시키는 실제 노드 후보 집합에 노드 m_1 이 있다고 하자. 실제 네트워크 토폴로지에서 실제 노드 m_1 과 직접 연결된 인접 노드들이 그림 2와 같이 있다고 하자. 즉, 그림 2는 실제 네트워크 G_S 의 일부분만을 그림으로 나타낸 것이다. 노드 m_1 과 인접 노드들을 연결하는 링크의 잔여 대역폭이 $bw(\overline{m_1 m_j})$, $j=2, 3, 4$ 로 그림 2에 표시되어 있다.

실제 노드 m_1 이 가상 노드 n_1 에 대응되려면 가상 노드 n_2, n_3, n_4 에 대응할 수 있는 실제 노드 $m'_2 = f_{N_i}(n_2)$, $m'_3 = f_{N_i}(n_3)$, $m'_4 = f_{N_i}(n_4)$ 을 구하고 m_1 과 m'_2, m'_3, m'_4 들 간에 적합한 경로가 있어야 한다. 그러나 주목할 사항은 m_1 과 $m'_2, m'_3,$

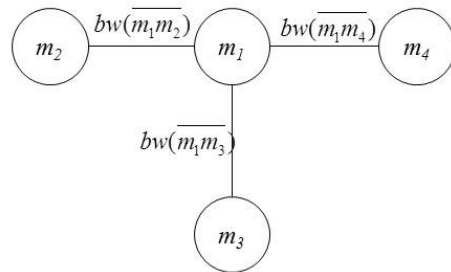


그림 2. 실제 네트워크에서 실제 노드 m_1 에 인접한 노드들의 구성도
Fig. 2. A configuration figure of substrate nodes who are adjacent to substrate node m_1 in the substrate network.

m'_4 들 간에 적합한 경로가 있으려면 적어도 m_1 에 연결된 실제 링크들의 잔여 대역폭은 가상 링크 $\overline{n_1n_2}$, $\overline{n_1n_3}$, $\overline{n_1n_4}$ 의 대역폭 요구사항 $bw(\overline{n_1n_2})$, $bw(\overline{n_1n_3})$, $bw(\overline{n_1n_4})$ 을 수용할 수 있어야 한다는 점이다. 가상 링크의 대역폭 요구조건을 만족시키기 위한 최소요건으로 식 (5)의 조건을 만족시켜야 한다.

$$\sum_{m_1m_j \in L_S} bw(\overline{m_1m_j}) \geq \sum_{n_1n_i \in L_{V_i}} bw(\overline{n_1n_i}) \quad (5)$$

또한 본 논문에서는 경로 분리 (path splitting)^[8]을 고려하지 않는다. 따라서 가상 노드에 연결된 여러 가상 링크 중 가장 큰 대역폭을 요구하는 가상 링크를, 고려하는 실제 노드에 연결된 링크가 수용할 수 있어야 한다. 이러한 제약조건을 식 (6)으로 정리한다.

$$\max\{bw(\overline{m_1m_j})\} \geq \max_{m_1m_j \in L_S, n_1n_i \in L_{V_i}} \{bw(\overline{n_1n_i})\}, \quad (6)$$

이에, 본 논문에서는 가상 노드 n 을 대응시키기 위해 고려하는 실제 노드 m 은 조건 식 (7)을 만족시켜야 하는 것으로 식 (1)의 조건을 확장한다.

$$\begin{aligned} A_S^N(m) &\geq R_{V_i}^N(n) \\ \sum_{mm' \in L_S} bw(\overline{mm'}) &\geq \sum_{nm' \in L_{V_i}} bw(\overline{nm'}) \\ \max\{bw(\overline{mm'})\} &\geq \max_{for mm' \in L_S, nm' \in L_{V_i}} \{bw(\overline{nm'})\}, \end{aligned} \quad (7)$$

이렇게 함으로써 본 논문에서 제안하는 방안은 대응시킬 가상 노드의 요구조건 뿐만 아니라, 그 가상 노드와 연결된 가상 링크의 요구조건을 만족시킬 수 있는 링크의 대역폭을 확보한 실제 노드만 대응 고려 대상으로 택한다.

2.2.2 제안 알고리즘

지금까지 진행된 연구들에서는 대응할 우선순위를 가상 노드에 대해서만 적용하거나, 실제 노드에만 적용하였다. 제안하는 방안에서는 가상 네트워크의 가상 노드와 실제 네트워크의 실제 노드를 특정한 메트릭에 따라 정렬한다. 가장 큰 메트릭을 갖는 가상 노드부터 시작하여 가장 큰 메트릭을 갖는 실제 노드에 대응 가능한지 탐색한다. 대응 가능여부가 결정되면, 대응된 가상 노드의 인접 가상 노드에 대하여 대응될 실

제 노드의 인접 노드부터 탐색하는 방법을 채택한다. 이렇게 하면 고려하는 가상 네트워크의 노드를 실제 네트워크에 클러스터링 시킴으로써 실제 네트워크에서 할당하는 비용이 절감될 수 있고, 유지보수 오버헤드를 줄일 수 있다.

그림 3은 제안한 가상 네트워크 대응 방안에 대한 알고리즘을 의사코드로 정리한 것이다. 먼저 가상 네트워크 요청이 들어오면 대응 우선순위 메트릭에 따라 가상 노드와 실제 노드를 우선순위로 정렬한다. 가상 노드의 수는 상대적으로 적으므로 정렬 리스트 (sorted list)를 사용하고(1 단계) 실제 노드 수는 가상 노드보다 많으므로 우선순위 큐 (Priority Queue: PQ)를 사용한다(2 단계). 우선순위가 높은 가상 노드부터 (3 단계) 우선순위 큐에 있는 실제 노드가 조건이 맞으면 대응시키고(4.3 단계) 가상 네트워크상의 다른 가상 노드들은 너비 우선 탐색 (Breath First Search)을 사용하여 인접노드부터 차례로 실제 노드에 대응시킨다(4.6 단계). 우선순위가 가장 높은 가상 노드

```

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. Make a priority queue PQ of substrate nodes  $N_{<S>}$ 
3. Delete  $n_{<1>}$  from  $N_{<V_i>}$ 
4. While PQ is not empty
    4.1 Dequeue  $m_{<1>}$  from PQ
    4.2 If  $m_{<1>}$  does not satisfy Eq. (7) for  $n_{<1>}$ ,
        go to 4.
    4.3 Map  $n_{<1>}$  to  $m_{<1>}$ 
    4.4 Set  $n_{cur} = n_{<1>}$ ; Set  $m_{cur} = f_{N_i}(n_{cur})$ 
    4.5 Save  $A_S^N, A_S^L$ .
    4.6 Do Breadth_First_Search
        4.6.1 For all neighbors of  $n_{cur}$ , say  $n'$ ,
            find a substrate node  $m'$  which satisfy Eq. (7)
            for  $n'$  starting from the adjacent node of  $m_{cur}$ 
        4.6.2 Find a shortest path between  $m_{cur}$  and  $m'$  that
            satisfies Eq. (2)
        4.6.3 If no path found, go to 4.6.1
        4.6.4 Map  $n'$  to  $m'$ ; Set  $n_{cur} = n'$ 
        4.6.5 Update  $A_S^N, A_S^L$  using Eq. (3) for  $m_{cur}, m'$ ,
            and the path between two nodes.
    4.7 If Breadth_First_Search succeeded,
        return SUCCEEDED.
    4.8 Else restore  $A_S^N, A_S^L$  and go to 4.
5. Return FAILED.
    
```

그림 3. 제안 가상 네트워크 대응 알고리즘 의사코드
Fig. 3. Virtual network embedding algorithm pseudo code

$n_{<1}$ 을 기준으로 인접 노드부터 시작하여 대응 시킬 실제 노드가 식 (7)을 만족시키는 지를 확인하고 (4.6.1 단계), 대응 시키려는 가상 노드들 사이에 존재하는 가상 링크 조건을 만족시키는 실제 경로가 있어야 대응이 가능하다(4.6.2 단계). 너비 우선 탐색 과정에서는 대응이 결정되면 실제 네트워크의 잔여 자원량을 대응에 필요한 만큼 감소시키고(4.6.5 단계), 다음 대응 작업을 수행한다. 따라서 대응과정에서 발생하는 자원 감소분을 기준으로 후속 대응이 이루어진다.

III. 성능 분석

이 절에서는 II.2절에서 제안한 가상 네트워크 대응 방안에 대하여 최악의 경우에 대한 시간 복잡도 (worst case time complexity)를 알아본다. 또한 성능 분석 지표로 가상 네트워크 수락율을 사용하여 제안한 방안의 성능을 모의실험을 통하여 분석한다. 가상 네트워크 수락율은 모의실험 시간 동안 들어오는 모든 가상 네트워크 요청 중 실제 네트워크에 대응에 성공한 가상 네트워크 요청의 비율이다.

성능 비교 대상으로는 기존 연구 중 Nogueira 등이 제안한 방안⁶⁾을 선택하였다. 선택 이유는 가상 네트워크에서 가상 노드에서의 CPU 요구조건과 가상 링크에서의 대역폭 요구조건을 고려하고, 대응 시 실제 네트워크에서 경로 분리를 고려하지 않는다는 공통점이 있기 때문이다.

3.1 시간 복잡도 분석

가상 네트워크의 가상 노드의 수를 $|N_v|$, 가상 링크의 수를 $|L_v|$ 라고 하자. 또한 실제 네트워크의 실제 노드의 수를 $|N_s|$, 실제 링크의 수를 $|L_s|$ 라고 하자. 모든 가상 네트워크 대응 방안에서는 대응되는 두 실제 노드간의 경로를 찾아야 한다. 경로 탐색에 드는 시간 복잡도는 최단 경로 우선 알고리즘을 사용하는 경우 $O(|N_s|\log|N_s|+|L_s|)$ 다.

제안한 방안에서는 가상 네트워크를 대응시킬 때 그림 3의 4.6 단계에서 너비 우선 탐색을 한다. 너비 우선 탐색의 기본적인 시간 복잡도가 $O(|N_v|+|L_v|)$ 이다. 너비 우선 탐색과정 중 실제 네트워크에서의 경로를 찾는 과정이 반복적으로 수행하고, 너비 우선 탐색이 실패하는 경우 우선순위 큐에 있는 실제 노드에 대하여 반복적으로 너비 우선 탐색을 하므로 제안하는 방안의 시간 복잡도는 $O(|N_s|(|N_v|+|L_v|)(|N_s|\log|N_s|+|L_s|))$ 다.

Nogueira 등이 제안한 방안에서는 각각의 가상 노드에 대하여 대응할 수 있는 실제 노드의 후보군을 구하고, 가상 링크로 연결된 두 가상 노드에 대응될 수 있는 실제 노드 후보군에 있는 모든 실제 노드에 대하여 노드 잠재력을 구한다. 노드 잠재력에는 실제 노드들 간의 경로 비용이 포함되므로 가상 노드 하나에 대한 노드 잠재력을 구하는 데 드는 시간 복잡도는 $O(|N_s|^2(|N_s|\log|N_s|+|L_s|))$ 다. 노드 잠재력을 가상 네트워크에 있는 모든 가상 노드들에 대하여 가상 링크에 연결된 가상 노드 쌍을 대상으로 구하므로 시간 복잡도는 $O(|N_v||L_v||N_s|^2(|N_s|\log|N_s|+|L_s|))$ 다. 따라서 Nogueira 등이 제안한 방안은 제안한 방안보다 $O(|N_s||N_v|)$ 혹은 $O(|N_s||L_v|)$ 의 시간 복잡도가 추가로 발생하여 가상 네트워크 대응을 시키는 경우 상당한 시간이 걸린다.

3.2 모의실험 환경

모의실험을 위해서 먼저 실제 네트워크 토폴로지를 생성하고, 토폴로지 안에 있는 실제 노드에 대해서 노드 특성인 CPU 용량을 할당하며, 실제 노드들을 연결하는 링크의 대역폭을 정해야 한다. 네트워크 토폴로지는 Waxman 랜덤 토폴로지⁹⁾ 생성 방법($\alpha=0.4, \beta=0.4$)⁶⁾을 사용하였다. 실제 노드의 수는 30개로 고정하였고, 실제 노드의 가용 CPU 용량은 균등분포 (uniform distribution)를 사용하여 50에서 100 사이에서 랜덤하게 생성하였다. 또한 실제 네트워크의 링크 대역폭도 1Gbps에서 2Gbps 사이에서 균등분포로 랜덤하게 생성하였다. 이렇게 함으로써 실제 노드와 실제 링크에서 제공 가능한 자원 양을 모의실험 시작부터 다르게 설정할 수 있다.

가상 네트워크 요청은 랜덤하게 도착하는 데, 모델은 포아송(Poisson) 프로세스 ($\lambda=0.1$)를 사용하였다¹⁴⁾. 요청되는 가상 네트워크는 일정 시간동안 실제 네트워크의 자원을 점유하여 서비스를 받고자 하는 데, 그 시간을 가상 네트워크의 수명이라고 한다. 가상 네트워크의 수명은 지수(Exponential) 랜덤 변수 ($\mu=250$)로 랜덤하게 생성하였다. 가상 네트워크의 가상 노드의 수는 최소 가상 노드 수와 최대 가상 노드 수 사이에서 균등분포를 사용하여 정하였다. 최소 가상 노드의 수는 2로 고정시켰고, 최대 가상 노드의 수는 4에서 10까지 변화를 주면서 모의실험을 실행하였다. 가상 노드들을 연결시키는 가상 링크는 연결 확률은 0.5로 설정하여 생성하였다. 가상 노드에서 요구

하는 CPU 요구량과 가상 링크의 대역폭 요구량도 균 등분포를 사용하여 랜덤하게 생성하였다. 모의실험 시 가상 노드의 CPU 요구량은 10과 20 사이를 고려하였다. 가상 링크의 대역폭 요구량은 100Mbps에서 200Mbps, 300Mbps에서 400Mbps 두 가지 경우를 고려하였다. 총 모의실험 시간은 10,000으로 설정하였다.

3.3 모의실험 결과

3.3.1 대응 우선순위 메트릭 평가

기존 연구에서 사용한 우선순위 설정 방법에는 가상 노드의 우선순위를 대응 가능한 실제 노드 수⁶⁾나 CPU 요구량⁷⁾으로 하거나, 실제 노드의 우선순위를 노드의 잔여 CPU 용량과 링크의 잔여 대역폭의 곱⁸⁾을 사용하였다. 본 논문에서는 우선순위 메트릭으로 CPU 용량, CPU 용량과 대역폭의 곱을 가상 노드와 실제 노드의 정렬에 사용하여 제안한 방안의 모의실험 결과를 비교 검토해 보았다. 그림 4는 가상 노드의 CPU 요구량을 uni[10, 20]으로 설정하고, 가상 링크

의 대역폭 요구량을 uni[100 Mbps, 200 Mbps]와 uni[300 Mbps, 400 Mbps]로 변화를 주어 실행한 모의실험 결과이다.

그림 4의 (a)와 (b)를 비교해보면 가상 링크의 대역폭 요구량이 변화해도 제안한 방안의 가상 네트워크 수락율의 변화는 미미함을 알 수 있다. 동일한 우선순위 메트릭을 사용하는 경우 가상 네트워크 수락율의 차이는 대부분 0.1%도 나지 않았고, 최대 0.15% 정도의 차이만 기록했다.

우선순위 메트릭으로 CPU 용량을 사용한 경우의 성능이 CPU 용량과 대역폭의 곱을 사용한 경우보다 좋았다. 최대 가상 노드 수가 7로 증가하면서 부터 가상 네트워크 수락율의 차이가 조금씩 더 커지고, 최대 가상 노드 수가 10일 때, CPU와 대역폭의 곱을 우선순위 메트릭으로 사용한 경우는 CPU를 우선순위 메트릭으로 사용한 경우보다 약 3% 낮은 성능을 보였다. 최대 가상 노드 수가 증가함에 따른 가상 네트워크 수락율의 하락도 CPU를 우선순위 메트릭으로 사용한 경우가 적어서, 최대 가상 노드 수가 4인 경우와 비교하여 최대 가상 노드 수가 10일 때 약 23%의 성능저하를 보였다. CPU와 대역폭의 곱의 경우에는 성능저하가 25%되었다.

따라서, 좋은 결과를 보인 CPU 용량을 본 논문에서 사용하는 메트릭으로 정하여 노드의 정렬 메트릭을 가상 노드는 CPU 요구량으로, 실제 노드는 잔여 CPU 용량으로 삼아 이후의 모의실험에 사용하였다.

3.3.2 기존 연구와의 성능 비교

여기에서는 기존 연구 중 Nogueira 등이 제안한 방안과 모의실험 결과를 비교함으로써 제안 방안의 타당성 검토한다. 모의실험 설정은 대응 우선순위 메트릭 평가 때와 동일하게 설정하였다.

모의실험 결과는 그림 5와 같다. 대역폭 요구량이 uni[100 Mbps, 200 Mbps]인 경우에는 Nogueira 등이 제안한 방안이 본 논문에서 제안한 방안보다 최대 가상 노드 수가 증가함에 따라 더 나은 성능을 보임을 그림 5 (a)를 통해 확인할 수 있다. 최대 가상 노드 수가 10인 경우 제안한 방안이 Nogueira 등이 제안한 방안보다 약 7.9% 낮은 성능을 보였다. 그러나 그림 5 (b)에서 보여주는 바와 같이 가상 링크의 대역폭 요구량이 uni[300 Mbps, 400 Mbps] 늘어난 경우 Nogueira 등이 제안한 방안의 성능은 급격히 나빠짐을 알 수 있다. 최대 가상 노드 수가 4인 경우 제안한 방안이 10% 높은 가상 네트워크 수락율을 보였다. 최대 가상 노드 수가 10으로 증가한 경우 제안한 방안은

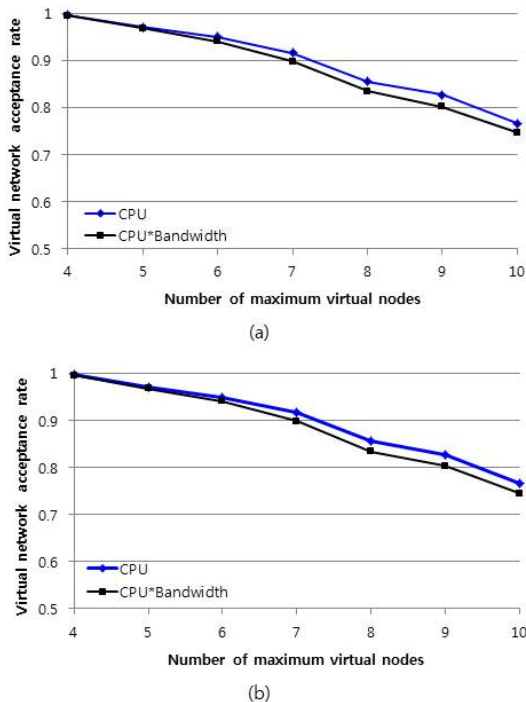


그림 4. 우선순위 메트릭에 따른 제안 방안의 가상 네트워크 수락율 (a) 가상 링크 대역폭 요구량 : uni[100Mbps, 200Mbps] (b) 가상 링크 대역폭 요구량 : uni[300Mbps, 400Mbps]

Fig. 4. Virtual network acceptance rate of the proposed method according to sorting rule (a) bandwidth requirement of virtual links: uni[100Mbps, 200Mbps] (b) bandwidth requirement of virtual links: uni[300Mbps, 400Mbps]

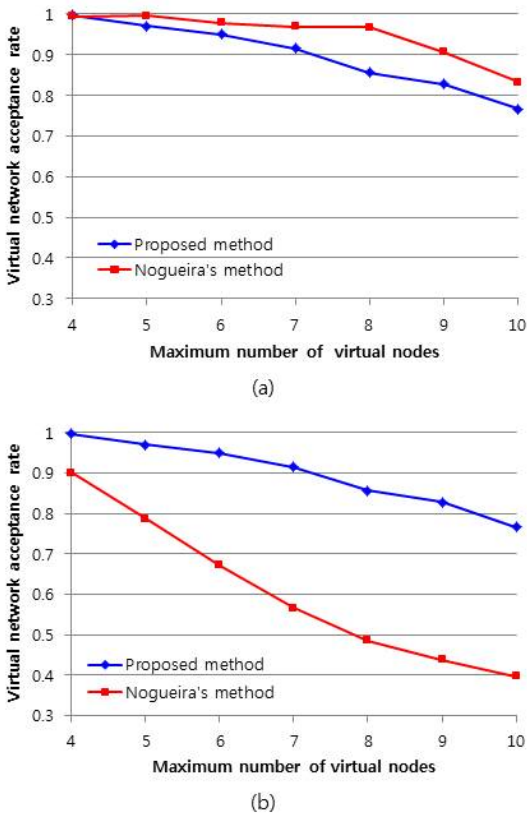


그림 5. 제안 방법과 Nogueira 등이 제안한 방법의 가상 네트워크 수락율 비교 (a) 가상 링크 대역폭 요구량: uni[100M, 200M] (b) 가상 링크 대역폭 요구량: uni[300M, 400M]
 Fig. 5. Comparison of virtual network acceptance rates between the proposed method and the Nogueira's method (a) bandwidth requirement of virtual links: uni[100M, 200M] (b) bandwidth requirement of virtual links: uni[300M, 400M]

Nogueira 등이 제안한 방안보다 거의 2배의 가상 네트워크 수락율을 기록하였다.

본 논문에서 제안한 방안의 시간 복잡도가 Nogueira 등이 제안한 방안의 시간 복잡도보다 상당히 낮은 점을 고려한다면, 본 논문에서 제안한 방안은 가상 링크의 대역폭의 요구조건의 변화에 민감하지 않으면서 안정적인 성능을 제공하는 면에서 훌륭한 대안이 될 수 있다.

가상 링크의 대역폭 요구조건이 증가함에 따라 Nogueira 등이 제안한 방안의 성능이 나빠지는 원인은, 노드 잠재력을 계산하는 과정에서 경로의 비용을 계산하는 데 이 때 자원할당이 고려되지 않기 때문이다. 노드 잠재력이 높은 실제 노드로 선택한 후 실제로 가상 링크에 대응할 경로를 선정하는 과정에서 기

설정된 경로와 겹치게 되면 실제 경로를 확보하지 못하게 된다. 그에 비하여 제안한 방안에서는 가상 노드에 대응할 실제 노드를 선정하는 경우 경로 설정 가능 여부를 타진하고, 다른 노드를 대응할 때 기 대응된 경로에 대한 자원이 고려되기 때문에 상대적으로 안정적인 성능을 보여준다.

IV. 결론

본 논문에서는 기존의 휴리스틱 방안들과는 달리 가상 네트워크 대응 시 고려하는 실제 노드가 가상 노드와 가상 링크의 조건을 만족시킬 수 있는 최소한의 요건을 검토한 후 대응 대상 노드로 선정하도록 하였다. 또한 대응 시 노드의 우선순위를 가상 노드의 경우 CPU 요구량, 실제 노드의 경우 잔여 CPU 용량으로 정하여 우선순위가 높은 노드 먼저 대응하게 하는 방안을 제안하였다. 제안하는 방안은 인접한 노드들에 대한 대응을 순차적으로 하여 가능한 한 실제 네트워크에 대응되는 실제 노드들도 인접한 영역에 클러스터를 형성할 수 있도록 하였다. 기존 연구와의 비교를 통하여 제안한 방안은 비교적 낮은 시간 복잡도를 갖으며 가상 대역폭의 요구조건의 변화에 무관하게 안정적인 성능을 보여줌을 알 수 있었다.

본 논문에서는 가상 노드의 요구조건으로 CPU 용량을, 가상 링크 요구조건으로 대역폭을 고려하였다. 앞으로의 연구에서는 가상 노드에 대한 위치 조건이나 메모리 용량, 가상 링크의 지연에 대한 조건도 고려하여 현재 제안한 방안을 확장할 계획이다.

References

- [1] D. Park, S. Lee, and C. Yoo, "Bandwidth isolation with priority based bandwidth control in network virtualization," in *Proc. Symp. KCC*, Jun. 2012.
- [2] M. Woo, "A survey on the virtual network embedding," in *Proc. KICS*, pp. 489-490, Nov. 2013.
- [3] N.M.M.K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *Proc. INFOCOM*, pp. 783-791, 2009.
- [4] Y. Zhu and M. Ammar, "Algorithms for assigning substrate network resources to virtual network components," in *Proc.*

INFOCOM, pp. 1-12, 2006.

- [5] C. K. Pyoung and S. J. Baek, "A comparison of virtual network mapping methods," in *Proc. KICS*, pp. 186-187, Nov. 2011.
- [6] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, "Virtual network mapping into heterogeneous substrate networks," in *Proc. ISCC*, pp. 438-444, 2011.
- [7] A. Razzaq, P. Sjodin, and M. Hidell, "Minimizing bottleneck nodes of a substrate in virtual network embedding," in *Proc. Network of the Future (NOF)*, pp. 35-40, Paris, Nov. 2011.
- [8] M. Yu, Y. Yi, J. Rexford, and M. Chian, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Computer Comm. Rev.*, vol. 38, no. 2, pp. 17-29, Apr. 2008.
- [9] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Selected Areas in Commun. (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 정보통신공학과 교수

<관심분야> 네트워크 구조 및 프로토콜, 이동 네트워크