ATM PNNI에서 셔플넷을 이용한 링크 상태 정보 집단화 방법

(Link State Aggregation using a Shufflenet in ATM PNNI Networks)

유 영 환 † 안 상 현 †† 김 종 상 †††

(Younghwan Yoo) (Sanghyun Ahn) (Chong Sang Kim)

요 약 본 논문에서는 ATM PNNI(Private Network-Network Interface) 망에서 경로 배정을 위해 필요로 하는 링크 상태 정보(link state information)를 효율적으로 집단화(aggregation)하는 방법을 제시한다. 이 방법은 집단화할 동료 그룹(peer group)의 경계 노드(border node)들을 셔플넷(shufflenet)의 노드들로 사상시킴으로써 표현해야 할 링크의 수를 완전 그물망 방법(full-mesh approach)의 N^2에서 pN(p는 정수, N은 경계 노드 수)으로 줄인다. 이는 공간 복잡도가 O(N)인 신장 트리(spanning tree) 방법에서 필요로 하는 링크의 수와 비슷하지만, 신장 트리 방법과는 달리 비대칭망(asymmetric network)에서 사용할 수 있다는 것이 큰 장점이다. 모의 실험 결과 셔플넷 방법은 pN개의 링크만을 표현하면서도 상태 정보의 정확성은 완전 그물망 방법에 근접함을 알 수 있었다.

Abstract In this paper, we present an algorithm for efficiently aggregating link state information needed for source routing in PNNI networks. In this algorithm, each border node in a peer group is mapped to a node of a shufflenet. By this mapping, the number of links for which state information is maintained becomes pN (p is an integer, N is the number of border nodes) which is significantly smaller than N^2 in the full-mesh approach.

Another novel aspect of our algorithm is that it can be applied to asymmetric networks, while many previous algorithms such as the spanning tree approach can be applied only to symmetric networks.

Experimental results show that our shufflenet algorithm performs as good as the full-mesh approach, with only a much smaller amount of information.

1. 서 론

경로 배정(routing) 방법은 크게 근원지 경로 배정(source routing)과 홉 단위 경로 배정(hop-by-hop routing)으로 분류된다. 근원지 경로 배정은 호를 설정하려는 노드가 자신의 데이타베이스를 기반으로 목적지까지의 전체 경로를 결정하는 방법이고, 홉 단위 경로 배정은 중간의 노드들이 자신의 데이타베이스에 근거하여 연결 요청 메시지가 목적지에 이를 수 있도록 다음 노드를 각자 결정하는 방법이다.

그러나, 홉 단위 경로 배정 방법은 몇 가지 단점을 지닌다. 먼저, 각각의 시스템이 서로 다른 경로 배정 알고리즘을 사용하거나 데이타베이스가 일치하지 않을 경우 루프(loop)가 생길 수 있고, 다음 노드를 선택하기 위한 알고리즘이 중간의 모든 노드에서 수행되어야 하므로 시간과 비용면에서 비효율적이다. 특히, ATM 망에서는 각 연결마다 별도로 요구되는 대역폭과 QoS를 지원해야 하므로, 선택 알고리즘이 더욱 복잡해질 수 있다. 이 경우 홉 단위 경로 배정 알고리즘의 비효율성은 더욱 부각된다.

· 이 논문은 2000년도 두뇌한국21 사업에 의하여 지원되었음

† 학생회원 : 서울대학교 컴퓨터공학부
yhyoo@archi.snu.ac.kr
†† 종신회원 : 서울시립대학교 전산통계학과 교수
(corresponding author임)
ahn@venus.uos.ac.kr
††† 종신회원 : 서울대학교 컴퓨터공학부 교수
cskim@archi.snu.ac.kr

논문접수 : 2000년 3월 17일
심사완료 : 2000년 10월 11일

홉 단위 경로 배정의 이러한 단점들 때문에 PNNI(Private Network-Network Interface) 표준에서는 근원지 경로 배정을 채택하였다. 연결을 설정하고자 하는 노드는 자신이 가진 망 상태 정보(network state information)에 기반하여 목적지까지의 경로를 스스로 결정하고, 이를 DTL(Designated Transit List)에 담아 중간 노드들에게 전달한다. 선택된 경로상의 모든 노드들은 이 리스트에 따라 연결 요청 메시지를 전달할 뿐이므로 루프가 생길 염려가 없고, 중간 노드에서 다음 노드 선택 알고리즘을 별도로 수행할 필요도 없다. 이는 망 내의 모든 노드들이 반드시 같은 경로 배정 알고리즘을 사용할 필요가 없음을 의미한다. 각 노드들이 어떤 정책을 사용하든 간에 효율적인 경로를 선택하도록 하고, 선택한 경로에 루프가 없게 하는 것은 모두 근원지 노드의 책임이기 때문이다. 따라서, 경로 배정 알고리즘은 PNNI 표준의 범위에 포함하지 않고, 여러가지 다양한 정책의 사용을 허용하고 있다. 단, 이러한 모든 경로 배정 알고리즘에서 연결이 DTL에 나타난 경로 그대로 항상 설립되는 것은 아니다. 연결 요청 메시지에 요구된 QoS를 지원할 수 없을 경우에는 DTL을 구성한 노드에게 요청 메시지를 되돌려 보냄으로써 다시 다른 경로를 선택하도록 하기도 하는데, 이런 경우를 크랭크백(crankback)이라고 한다.

ATM 망의 연결은 QoS를 지원해야 하므로, PNNI의 각 노드들이 경로 선택을 하기 위해서는 지연 시간(delay), 가용 대역폭(available bandwidth) 등과 같은 모든 노드와 링크의 자원 상태 정보(state information)를 알고 있어야 한다. 이를 위해서 각 노드들은 주기적으로, 또는 필요한 경우[1] 노드와 링크 상태 정보를 PTSE(PNNI Topology State Elements) 메시지에 담아 서로 교환한다. 그러나, 망이 커서 포함하고 있는 노드와 링크의 수가 많아지면, 망 관리를 위한 상태 정보 자체가 막대한 부담으로 작용한다.

이러한 망 관리 부하를 줄이기 위하여 ATM 망은 계층 트리(hierarchical tree) 형태로 구성된다. 그림 1은 PNNI 계층 구조를 나타낸 것이다. ATM 망의 모든 노드들은 몇 개씩 무리를 지어 최하층의 동료 그룹(peer group)이 되고, 또 다시 이 동료 그룹들은 몇 개씩 묶여 상위 동료 그룹을 이룬다[2]. 상위 레벨의 노드들은 결국 하위 여러 노드들의 묶음인 동료 그룹을 대신하는 논리적 노드(LGN: Logical Group Node)이고, 상위 레벨의 링크들 역시 하위 동료 그룹 사이의 모든 물리적 링크를 합친 논리적 링크이다. 이러한 추상화(abstraction)와 집단화(aggregation) 작업을 통해 하위 레벨의 가용 자원들은 하나의 노드 또는 하나의 링크로 간략하게 표현될 수 있다. 각 동료 그룹에는 PGL(Peer Group Leader)이 있는데, PGL로 선출된 노드는 그룹 내 다른 노드들의 PTSE 메시지로부터 얻은 상태 정보를 집단화하고, 다른 그룹의 PGL에게 전송하는 일을 맡는다. 이어서 다른 그룹들의 PGL은 플러딩 기법을 이용하여 다시 이 정보를 자신의 그룹에 속하는 노드들에게 알린다.

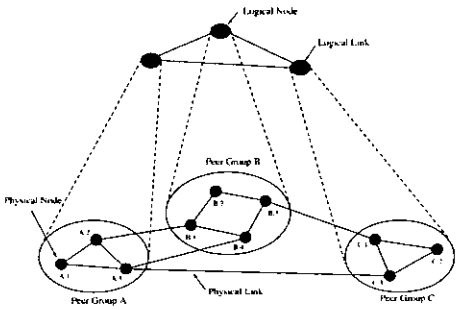


그림 1 PNNI 계층 구조

PNNI의 노드들은 위와 같이 집단화된 상태 정보를 유지하므로, 외부 동료 그룹들은 각각 하나의 논리적 노드들로 간주된다. 따라서, 하나의 근원지 노드가 선택한 경로는 자신이 속한 동료 그룹의 노드들을 제외하고는 실제 물리적 노드들의 연결이 아니라 논리적 노드, 즉 외부 그룹들의 연결이며, 이들 그룹내에서의 지역적인 경로는 각 그룹내에서 알아서 결정하도록 한다.

상태 정보의 부정확성은 망에 여러 가지 부정적인 영향을 미친다. 집단화 방법에 따라 상태 정보가 실제 자원 상태보다 더 좋게 표현될 수도 있고, 더 나쁘게 표현될 수도 있다. 만일 어떤 링크가 실제 가용 자원보다 더 많은 자원을 가진 것으로 표현되면, 많은 노드들이 목적지까지의 QoS를 보장하는 경로를 설정할 때 이 링크를 포함시킬 확률이 높아진다. 이 경우 실제 경로 배정 작업 중에 가용 자원이 알려진 값보다 적다는 것을 알게

1) 특정한 몇몇 링크 속성이 변화하거나 어떤 링크의 자원이 이전 자원의 양을 기준으로 일정 백분율 이상 변화했을 때를 말한다 [3].

2) 각 동료 그룹에 몇 개씩의 노드를 포함시키고 전체 계층을 몇 단계로 구성하는 것이 경로 배정 알고리즘의 효율성을 증대할 수 있는가 하는 문제[1]는 본 논문의 범위를 벗어나므로 다루지 않는다.

돼 지정한 경로를 통한 연결 설정이 실패하고, 다시 다른 경로를 찾기 위해 크랭크백이 일어난다. 크랭크백의 과도한 발생은 경로 배정 완료 시간을 허용치 이상으로 증가시키므로, 결국 연결의 설정 자체가 실패하는 원인이 된다. 반면에 링크 상태 정보가 실제 가용한 자원보다 너무 적은 값으로 표현되면 연결을 설정하고자 하는 노드는 실제로는 충분한 양의 자원이 있음에도 불구하고 연결 설정을 할 수 없다는 판단을 내리게 된다. 비관적인 쪽으로 부정확성이 심화되면 자원의 심각한 낭비를 초래한다.

상태 정보의 부정확성은 경로 배정 알고리즘의 선택에도 크게 영향을 미친다. Guérin은 ATM의 자원 정보 추상화로 인하여 실제 경로 배정에서 발생할 수 있는 문제를 보이고, 집단화된 정보를 이용하는 휴리스틱(heuristic) 경로 배정 알고리즘을 제안했다 [2].

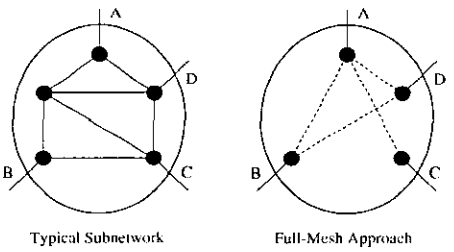


그림 2 완전 그물망 집단화 방법

이런 여러가지 문제에도 불구하고 여러 노드간에 주고받는 상태 정보의 양을 줄이기 위해 PNNI 계층 구조는 반드시 필요하며, 계층 구조에는 적절한 집단화 방법이 수반되어야 한다. 지금까지 제안된 대표적인 집단화 방법으로는 완전 그물망 방법(full-mesh approach)[3], 별 모양 방법(star approach)[3], 신장 트리 방법(spanning tree approach)[4] 등이 있다. 그러나, 완전 그물망 방법의 경우에는 정확한 상태 정보를 유지할 수는 있지만, 공간 복잡도가 $Q(N^2)$이어서 대규모의 망에 사용하기에 부적합하며, 별 모양 방법에서는 정보의 양을 줄이기는 했으나 그 정보가 매우 부정확해지는 단점이 있었다. 신장 트리 방법의 경우에는 이런 단점들을 보완하기는 했으나, 양방향으로 같은 전송 용량을 가지는 대칭 링크망에서만 사용할 수 있다는 한계가 있다.

따라서 본 논문에서는 셔플넷(shufflenet) 구조를 이용하는 새로운 집단화 방법을 제안한다. 이 방법은 실제 망이라 할 수 있는 비대칭 링크망에서도 사용이 가능하고, 유지해야 할 링크의 수가 pN(p는 정수, N은 경계 노드 수)으로 완전 그물망의 경우보다 훨씬 확장성(scalability)이 있으며, 상태 정보의 정확성도 완전 그물망과 큰 차이가 없다. 셔플넷 방법은 두 가지 측면에서 자원의 사용률을 높이는데, 먼저 집단화 정보를 정확하게 표현함으로써 자원이 사용되지 않고 낭비되는 것을 막을 수 있고, 둘째는 상태 정보의 양을 줄여서 PTSE 메시지 전송에 소모되는 자원의 양을 줄이기 때문에, 그만큼 다시 자원의 사용률을 향상시키는 효과를 얻을 수 있다.

이 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 살펴보고, 3장에서는 셔플넷(shufflenet)을 이용한 새로운 집단화 방법을 제안한다. 4장에서는 실험을 통해 셔플넷 집단화 방법의 효율성을 보이고, 마지막으로 5장에서 향후 과제와 함께 결론을 맺는다.

2. 관련 연구

ATM 포럼 표준에 제시된 기본적인 집단화 방법과 이를 개선한 방법들을 살펴본다.

2.1 PNNI 표준 집단화 방법

ATM 포럼의 PNNI 표준에는 집단화 방법에 대한 개념이 소개되어 있다 [3]. 기본적인 방법으로 대칭 노드 방법(symmetric-node approach)과 완전 그물망 방법(full-mesh approach)이 있고, 두 가지 방법이 절충된 형태의 별 모양 방법(star approach)이 있다. 경우에 따라서는 별 모양 방법을 대칭 별 모양(symmetric star)과 비대칭 별 모양(asymmetric star) 방법이라는 이름으로 구분하기도 하는데, 대칭 별 모양 방법은 대칭 노드 방법과 동일하므로 앞으로 별 모양 방법이라고 하면 모두 비대칭 별 모양 방법을 지칭한다. 이 절의 설명은 모든 링크가 양방향으로 동일한 QoS를 갖는 대칭 링크라고 가정하고 있음을 미리 밝힌다.

대칭 노드 방법은 링크 상태 정보를 줄이는 것에 중

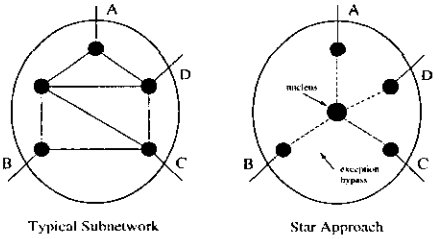


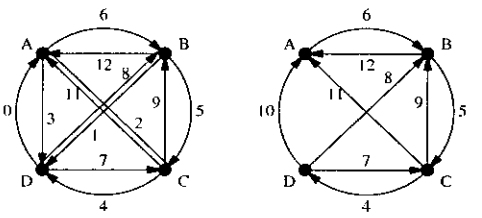
그림 3 별 모양 집단화 방법

점을 둔 방법으로 한 동료 그룹 내의 모든 노드가 하나의 가상 노드로 융합된다. 한 그룹 내에서 자신이 속한 동료 그룹을 벗어나 외부 동료 그룹으로의 링크가 있는 노드들을 경계 노드(border node)라고 하는데, 경계 노드 간 대역폭, 지연 시간을 비롯한 모든 상태 정보가 하나의 공통된 값[3]으로 대치된다. 상태 정보의 양이 다른 방법들보다 적다는 장점이 있지만, 정보의 부정확성이 증가하고 망 자원의 낭비가 심각해질 수 있다.

완전 그물망 방법은 모든 경계 노드들의 쌍에 대해 링크가 존재하는 완전 그물(full-mesh) 형태로 연결하고 각 링크마다의 정보를 따로 유지한다. 이 링크들은 논리적 링크로 실제 물리적 수준에서는 망 관리 정책에 따라 두 노드 사이의 최대 대역폭 경로이거나 최소 지연 시간 경로가 된다. 그림 2에서 왼쪽 그림은 우리가 집단화를 하려는 동료 그룹이고 오른쪽은 완전 그물망 방법으로 집단화한 형태이다. 대칭 노드 방법에 비해 상당히 정확한 링크 상태 정보를 유지할 수 있지만, 경계 노드의 수가 많을 때는 링크 상태 정보를 줄인다는 집단화의 기본 목적에 부합하지 못할 수도 있다. 경계 노드의 수를 N이라고 하면 상태 정보를 유지해야 하는 링크의 수가 N(N-1)/2로 급증하기 때문이다. 최악의 경우 어떤 동료 그룹이 링 형태의 망으로 구성되었다고 하면, 모든 노드가 경계 노드가 될 수도 있다.

마지막으로, 별 모양 집단화 방법에서는 가상의 핵(nucleus)을 동료 그룹 중앙에 두고 각각의 경계 노드로부터 중심의 핵으로 가상의 링크를 연결한다. 대칭 노드 방법에서는 모든 상태 정보가 하나의 공통된 값을 가지는 반면 별 모양 방법에서는 각 경계 노드 간 링크가 서로 다른 값을 가질 수 있다. 그림 3의 오른쪽 그림은 왼쪽의 실제 동료 그룹을 집단화한 형태이다. 이 경우 표현해야할 링크의 수는 O(N)으로 완전 그물망 방법의 1/N 수준이면서 대칭 노드 방법보다 정확한 링크 상태 정보를 유지할 수 있으므로 실제로 가장 많이 사용되는 방법이다. 별 모양 방법을 사용했을 경우는 손실되는 정보의 양을 줄이기 위하여 기본 링크 외에 예외 우회 링크(exception bypass link)를 추가할 수 있으나 모두 합해 총 3N개까지 둘 수 있다는 규정만 있을 뿐 명확한 설정 기준은 제시된 것이 없다.

대칭 노드 방법과 완전 그물망 방법은 각각 오로지 상태 정보의 양을 최소화하거나 정보의 정확성을 유지하는데 초점이 맞추어져 있다. 하지만, 이렇게 극단에 치우친 방법들은 링크 상태 정보가 지나치게 부정확해진다거나 상태 정보의 양이 너무 많아지는 부작용을 낳을수 있다. 별 모양 방법은 앞선 두 방법의 장점을 절충한 형태이지만, 동시에 두 방법의 단점도 여전히 지니고 있는 방법이라고 할 수 있다.



(a) Original topology (b) Aggregated topology

그림 4 Lee의 비대칭망 집단화 방법 [6]

2.2 개선된 집단화 방법

위의 ATM 포럼 표준 외에도 기존 방법들의 단점을 극복하기 위하여 여러가지 새로운 방법들이 제안되었다.

Lee는 신장 트리(spanning tree)를 이용한 방법을 제안하였다 [4]. 이 방법은 우선 집단화할 대상 동료 그룹을 기반으로 완전 그물망을 구성한 후, 이 그물망에 기반하고 가중치(weight)가 각각 가용 대역폭과 지연 시간인 두 개의 신장 트리를 성한다. 이 두 개의 신장 트리는 다른 외부 동료 그룹들에게 전달되고, 외부 그룹의 노드들 중에서 대상 동료 그룹을 지나는 연결을 설정하고자 하는 노드는 신장 트리로부터 완전 그물망을 재구성한다.

각 동료 그룹의 정보가 두 개의 신장 트리로 표현되기 때문에 경계 노드의 수를 N이라고 했을 때 완전 그물망 방법의 N(N-1)/2보다 훨씬 적은 2(N-1) 개의 링크 정보를 가지게 되지만, 경로 대역폭 정보의 정확성은 완전 그물망 방법과 동일한 수준을 유지할 수 있다. 그러나, 지연 시간은 망의 구성 형태에 따라 실제보다 크게 부정확해질 수도 있다.

무엇보다도 가장 주목해야할 신장 트리 방법의 한계는 모든 링크가 양방향으로 동일한 QoS를 지원하는 대칭 링크를 가정하고 있다는 것이다. 실제 ATM 망은 양방향 링크가 물리적으로 서로 다른 용량을 갖는 것이 일반적이며, 물리적으로 대칭인 경우에도 가상 채널(virtual channel)의 양방향 자원 요구량이 다른 경우가 일반적이므로, 하나의 채널이 설정되고 해제될 때마다 양방향 자원의 양은 달라지게 된다 [5]. 이런 이유로 대칭 링크보다는 비대칭 링크로 이루어진 망이 현실적임

3) 정책에 따라 최악의 경우값이나 평균값이 사용될 수 있는데, 본 논문에서는 최악의 경우에 해당하는 값을 사용했다

에도 불구하고, 기존 대부분의 집단화 방법은 그 단순성 때문에 대칭 링크를 가정하고 제안되었다 [3, 4]. 한 예로 만일 비대칭망이었다면 완전 그물망에 필요한 링크의 개수는 위에서 말한 N(N-1)/2가 아니라 N(N-1)로 훨씬 복잡했을 것이다.

b_{ij}: the bandwidth of the link (i, j)
$b_{ij} = 0$ if the link (i, j) does not exist.
$P = \{1\}, B_1 = \infty, B_j = b_{1j}$ for $j \neq 1$
Step 1: Find $i \notin P$ such that
$$B_i = \max_{j \notin P} B_j$$
$P := P \cup \{i\}$.
If P contains all nodes, then stop.
Step 2: For all $j \notin P$
$$B_j := \max[B_j, \min[B_i, b_{ij}]]$$
Go to Step 1.

그림 6 노드 1에서 다른 모든 노드로의 최대 대역폭 경로를 찾는 알고리즘

비대칭 링크로 구성된 비대칭망에 대해서는 신장 트리 방법을 적용할 수 없기 때문에 별도의 정보 집단화 방법이 제안되었다. Lee는 [6]에서 비대칭망을 위한 집단화 방법을 간략히 제시하였다. 그림 4 (a)는 집단화하기 전의 원래 토폴로지이고 (b)는 집단화한 이후의 형태로 그 방법은 아주 단순하다. 두 노드를 잇는 각 링크에 대해 해당 링크를 사용하지 않고 여러 개의 다른 링크를 지나는 대역폭이 같거나 더 큰 경로가 있는지 살펴본다. 만일 그런 경로가 존재한다면 두 노드를 직접 연결하는 링크를 삭제(discard)하고 대신 그 경로를 사용하게 함으로써 정보의 양을 줄인다. 예를 들어, 그림 4 (a) 에서 노드 A와 D를 직접 연결하는 링크는 대역폭이 3으로 A-B-C-D를 거쳐 가는 경로의 대역폭 4보다 작다. 노드 A와 C를 연결하는 링크와 노드 B와 D를 연결하는 링크도 각각 A-B-C, B-C-D 경로보다 대역폭이 작으므로, 이들 세 개의 링크는 원래 토폴로지 정보에서 생략할 수 있다.

이 방법은 정확한 정보를 유지할 수 있다는 장점이 있지만, 공간 복잡도가 여전히 $O(N^2)$이고, 집단화 할 때와 이 집단화 정보로부터 다시 완전 그물망을 구성할 때 필요한 과도한 계산 시간 복잡도 때문에 실제 망에 사용하기에는 부적합하다. [6]에서는 구체적인 구현 방법을 제시하고 있지 않으므로, 본 논문의 실험에서 독자적으로 구현한 방식과 그 복잡도를 언급하면 다음과 같다. 이런 방식의 집단화 방법에서는 집단화하여 정보를 축약할 때와 축약된 정보로부터 원래 정보를 다시 복원할 때의 시간 복잡도가 따로 고려되어야 한다. 먼저, 집단화 할 때는 대역폭이 작은 링크부터 차례로 생략해 보고, 생략한 링크로 연결되어 있던

두 노드가 여전히 연결 가능한가를 검사해야 하는데, 이 작업에 필요한 시간 복잡도는 $O(N^2)$이다. 링크의 개수가 N^2이므로 총 시간 복잡도는 $O(N^4)$이고, 여기에 퀵소트(quicksort)를 이용하여 N^2개의 링크를 대역폭 크기 순으로 정렬하는 복잡도 $O(N^2 \log N)$이 추가되어야 한다. 반면, 집단화 정보로부터 완전 그물망을 복원할 때는 모든 노드 쌍에 대해 최대 대역폭 경로를 찾아야 하는데, 하나의 노드에서 모든 다른 노드로의 최대 대역폭 경로는 최단 거리 경로를 찾기 위한 Dijkstra 알고리즘[7]을 수정하여 구할 수 있다. 예를 들어,

노드 1에서 모든 다른 노드로의 경로를 찾는 알고리즘은 그림 5와 같다. 이 알고리즘의 시간 복잡도가 $O(N^2)$이고 노드의 개수가 N이므로, 전체 시간 복잡도는 $O(N^3)$이 된다.

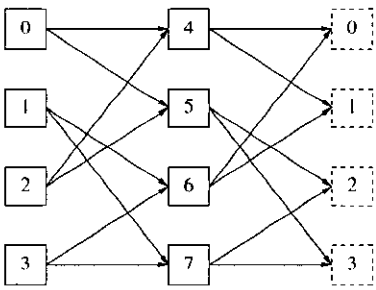


그림 7 (2, 2) 셔플넷의 형태

비대칭망에 사용할 수 있는 또 하나의 방법으로 Awerbuch와 Shavitt의 제안이 있다 [8]. 이 방법에서는 비대칭망을 먼저 대칭망으로 변경하는데, 대칭망 링크의 가중치는 비대칭망 양방향 링크 가중치 곱의 양의 제곱근으로 한다. 이렇게 구성된 동료 그룹의 경계 노드만으로 t-spanner 그래프를 구성하는데, t-spanner란 어떤 그래프의 서브그래프로서 모든 노드의 쌍에 대해 그 경로 가중치가 원래 그래프의 t배 이내로 제한되는 그래프를 말한다. t-spanner의 구성 방법은 Kruskal의 신장 트리 구성 알고리즘[7]과 마찬가지로 각 링크를 가중치의 오름차순으로 정리한 뒤 각 링크를 차례로 서브그래프에 포함시킬지 여부를 결정한다. 만일 현재까지 구성한 서브그래프에서 두 노드 u, v 간의 가중치[4]가

4) 두 노드를 잇는 경로가 존재하지 않는 경우 가중치는 ∞로 둔다.

원래 그래프 상의 가중치의 t배 이상이면 링크 (u, v)를 t-spanner 그래프에 포함시킨다. 따라서, t-spanner 그래프에서는 사이클(cycle)이 생길 수도 있다. 이 방법에서 가중치를 지연 시간으로 두면 지연 시간 정보의 부정확성을 t배 이내로 제한할 수 있다는 장점이 있으나, 비대칭망 양방향 링크의 가중치가 심하게 차이 나는 경우에는 대칭망으로 변경하는 과정에서 비대칭망의 특성을 크게 왜곡시키는 단점이 있다.

위의 방법들이 PNNI 표준에 제안된 기존 방법들의 공간 복잡도를 줄이는 것을 목표로 했던 반면에, 복잡도는 그대로 유지하되 QoS를 보다 정확하게 표현하기 위한 노력도 있었다. Iwata 등은 ATM 망에서 QoS를 고려한 경로 배정 방법을 제안하고, 이 경로 배정 방법이 효율적으로 동작하기 위해 기존의 별 모양 방법을 QoS 특성을 보다 잘 나타낼 수 있도록 개선한 방법을 제안하였다 [9]. 그리고, 기존의 여러 별 모양 방법들과 비교하여 호 봉쇄율(call blocking probability)이 현저히 개선될 수 있음을 실험을 통해 보였다.

3. 셔플넷(ShuffleNet)을 이용한 집단화

기존의 관련 연구들을 살펴보면 대부분 대칭 링크망을 가정하고 있으나 이는 알고리즘의 단순성을 위한 것이지 현실에는 맞지 않는 가정이다. 따라서, 이 장에서는 비대칭망을 위한 집단화 방법으로서 셔플넷(shufflenet) 구조를 이용하는 새로운 방법을 제안한다. 상태 정보 집단화 방법에 대한 연구에서 목표로 하는 것은 첫째, 완전 그물망 수준의 정확한 상태 정보 유지, 둘째, 노드간에 교환해야 하는 상태 정보의 최소화이다. 그러나, 이 두 가지 목표는 서로 상충되는 성질이 있으므로 적당한 절충점을 찾는 일이 필요하며, 본 논문에서 제안된 방법은 셔플넷 구조를 이용하여 이 문제를 해결하고자 한다.

먼저 3.1절에서 셔플넷의 성질을 알아보고, 3.2절에서 셔플넷을 이용한 집단화 방법을 예를 들어 설명한다. 3.3절에서는 셔플넷 방법이 가지는 몇 가지 문제와 해결책에 대해 기술한다.

3.1 셔플넷의 성질

(p, k) 셔플넷은 총 $N=kp^k$ 개의 노드를 k개의 열에 열 당 p^k 개씩 배열한 형태이다. 각 노드는 자신의 위치에 따라 고유 번호를 부여받는데, n (n = 1, 2, ..., k)번째 열, l (l = 1, 2, ..., p^k)번째 행 노드의 번호는 $(n-1)p^k+l-1$로 결정된다. 그리고, 각 노드는 다음 열 p^k 개의 노드 중에서 p 개의 노드로 향하는 출력 링크를 가지므로, (p, k) 셔플넷의 총 링크 개수는 pN이다. 각 링크의 상대 노드는 다음과 같이 결정된다. 만일, 어떤 노드의 고유 번호가 i라 하면 $j = (i \% p^k-1)p$가 되고, 노드 i는 다음 열의 노드들 중에서 위에서부터 j+1, j+2, ..., j+p번째 노드들에 대해 연결을 가진다. 그리고, 마지막 열에서는 첫번째 열로 연결을 가져 전체적으로 순환되는 형태를 가진다. 그림 6은 (2, 2) 셔플넷의 예이다.

셔플넷을 사용하면 모든 노드를 서로 직접 연결하지 않아도 각 노드에서 다른 모든 노드로의 가상 연결의 설립이 가능하고, 이때 한 노드에서 다른 노드로의 홉거리가 최대 (2k-1)로 제한된다는 장점이 있다. (p, k) 셔플넷의 임의의 한 노드에서 h 홉으로 갈 수 있는 노드의 수 n_h는 다음과 같이 주어진다.

$$n_h=\begin{cases} p^h & h=1, 2, \ldots, k-1 \\ p^k-p^{h-k} & h=k, k+1, \ldots, 2k-1 \end{cases} \quad [10] \qquad (1)$$

따라서, 한 노드에서 모든 다른 노드로 가는 데 필요한 홉 수를 모두 합한 후 이를 다른 노드의 개수 (kp^k - 1)로 나누면 평균 홉 수를 구할 수 있다.

$$h_{ave}=\sum_{h=1}^{2k-1} h n_h/(kp^k-1)=\frac{kp^k(p-1)(3k-1)-2k(p^k-1)}{2(p-1)(kp^k-1)} \qquad (2)$$

식 (2)에 따라 (2, 2) 셔플넷과 (2, 3) 셔플넷, (3, 2) 셔플넷 등은 각각 대략 2, 3, 2의 평균 홉 수를 가진다.

한편, 셔플넷은 정형 토폴로지이므로 노드의 개수가 반드시 kp^k(k, p는 정수) 개가 되어야 한다는 제약을 가진다. 이 제약 조건에 대한 해결책은 3.3절에서 설명한다.

3.2 셔플넷 집단화

이 장에서는 셔플넷 집단화(shufflenet aggregation) 방법을 제안한다. 이 방법은 대칭망에서도 사용할 수 있지만, 보다 일반적인 망에서 사용할 수 있도록 처음부터 비대칭망을 대상으로 고안되었다. 특별한 언급이 없는한 이제부터 가정하는 모든 망은 비대칭망이다.

집단화하려는 동료 그룹의 경계 노드가 여덟 개라고 가정하면, 먼저 이 동료 그룹을 대표하는 PGL(Peer Group Leader)은 이들 여덟 개의 노드를 가지고 그림 7과 같은 완전 그물망을 구성하는데, 비대칭망을 가정했으므로 논리 링크는 총 56개가 된다. 하나의 동료 그룹을 표현하는데 있어서 56개나 되는 논리 링크 정보는 상당히 큰 부담이 되므로 이를 적절한 수로 줄일 필요가 있다. 경계 노드의 수가 더 증가할수록 상태 정보의 축소 방법은 더욱 필수적이다. 그러나, 상태 정보를 축소하기 위해 별 모양 방법을 사용한다면, 논리 링크는 8개로 줄어들지만 상태 정보 역시 매우 부정확한 값을 갖게 되므로 다른 집단화 방법이 필요하다.

본 논문에서는 완전 그물망 방법의 복잡성과 별 모양 방법의 부정확성에 대한 대안으로 셔플넷 집단화 방법

을 제안한다. 셔플넷 구조의 도입에는 다음과 같은 세 가지 주요한 이유가 있다. 첫째, 셔플넷은 임의의 노드에서 모든 다른 노드로의 방향성 있는 연결을 제공할 수 있으므로, 비대칭망의 집단화에 적합한 구조이다. 둘째, 경계 노드 수가 $N = kp^k$ (p, k = 1, 2, 3, ...)인 동료 그룹을 (p, k) 셔플넷으로 표현할 경우 표현되는 논리 링크의 수가 pN이 되므로, N(N-1)인 완전 그물망의 경우보다 훨씬 확장성이 있다. 세째, 모든 노드 간의 연결이 (2k-1) 홉으로 제한되므로, 상태 정보 축소 이후에도 지연 시간이 크게 증가하지 않는다.

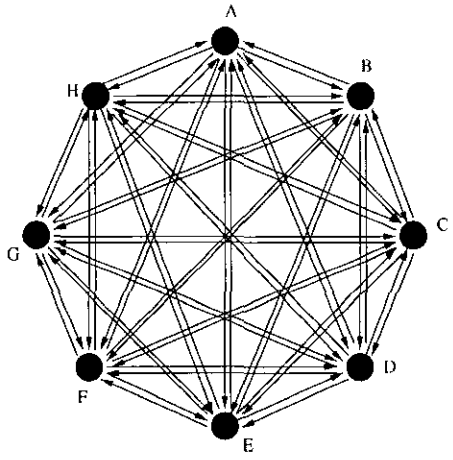


그림 8 경계 노드가 8개일 때의 완전 그물망

셔플넷 집단화 방법의 적용 순서는 먼저 동료 그룹을 완전 그물망 형태로 집단화 한 후 이를 셔플넷 형태로 축약하여 외부 동료 그룹에 전달하고, 외부 동료 그룹의 노드는 필요한 경우 전달받은 셔플넷 정보에서 완전 그물망을 복원하여 경로 배정 작업에 이용한다. 정보 축약 작업은 완전 그물망 N(N-1)개의 링크 중에서 정책에 따라 대역폭이나 지연 시간을 최대한 정확하게 유지할 수 있도록 pN개의 링크를 남기는 것이며, 정보 복원 작업은 셔플넷 구조에서 한 홉으로 연결된 경계 노드쌍에 대해서는 하나의 링크에서, 다른 노드쌍에 대해서는 여러 링크가 연결된 경로로부터 상태 정보를 계산하여 완전 그물망을 만들어내는 것이다. 대역폭 정보의 경우 대역폭이 가장 작은 링크가 전체 경로의 대역폭을 제한하는 성질이 있으므로, 셔플넷 정보로부터 재구성한 완전 그물망의 대역폭 정보가 원래 값에 근사하기 위해서는 완전 그물망의 링크 중에서 가능한한 대역폭이 큰 링크들만이 셔플넷에 표현되도록 해야 한다. 반면, 지연 시간은 경로를 구성하는 모든 링크 지연 시간의 합이 되므로, 지연 시간을 정확히 하고자 할때는 노드들을 가능한한 짧은 홉 수의 경로로 연결하거나 지연 시간이 짧은 링크들을 셔플넷에 남기면 된다.

그림 7의 완전 그물망을 셔플넷 형태로 표현하는 방법은 다음과 같다. 노드의 개수가 8개이므로 그림 6과 같은 (2, 2) 셔플넷을 사용한다. 우선 여덟 개의 경계 노드 A-H를 셔플넷 노드 0-7에 적절히 사상시켜야 하는데, 셔플넷 노드간의 연결은 3.1절의 링크 연결 규칙에 따라 미리 결정되어 있으므로, 경계 노드를 셔플넷 노드에 사상시키는 방법에 따라 완전 그물망을 셔플넷으로 축소시킬 때 어떤 링크 정보가 남느냐가 결정된다. 여덟 개의 노드를 순서대로 늘어놓는 경우의 수는 $_8P_8 = 8!$ 이나 되므로, 간단하고 효율적인 휴리스틱(heuristic) 알고리즘이 필요하다. 본 논문에서 사용한 방법의 목적은 대역폭 정보를 정확히 유지하기 위하여 셔플넷 구성 규칙을 만족하는 조건하에 가능한한 큰 대역폭 정보들이 셔플넷에 남게 하는 것이다.

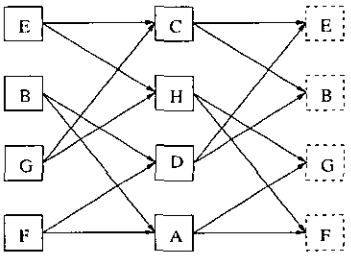


그림 9 경계 노드를 셔플넷 노드로 사상한 모습

경계 노드-셔플넷 노드 사상(mapping) 알고리즘

1. 현재까지 선택되지 않은 완전 그물망의 경로 중에서 최대 대역폭을 가진 경로를 선택한다. 만약 더이상 검사할 경로가 없으면 단계 6으로 간다.

2. 선택된 경로의 종단점인 두 경계 노드가 이미 셔플넷 노드에 할당되었으면, 단계 1로 간다.

3. 두 경계 노드 중 하나만 할당되어 있으면, 할당된 셔플넷 노드와 연결이 존재하는 셔플넷 노드 중 아직 사용되지 않은 노드가 있는지 조사한다. 비어 있는 셔플넷 노드가 있으면 나머지 경계 노드 하나를 빈 셔플넷 노드에 할당하고, 단계 5로 간다. 빈 셔플넷 노드가 없으면, 단계 1로 간다.

4. 두 경계 노드 모두 할당되어 있지 않으면 연결이

존재하는 사용가능한 두 셔플넷 노드가 있는지 조사한다. 빈 셔플넷 노드쌍이 존재하면, 두 경계 노드를 할당하고 단계 5로 간다. 빈 셔플넷 노드쌍이 존재하지 않으면, 단계 1로 간다.

5. 모든 경계 노드가 셔플넷 노드에 할당되었는지 검사하여 모두 할당되었으면 종료하고, 그렇지 않으면 단계 1로 간다.

6. 셔플넷 노드 중에서 비어있는 노드가 있으면 가장 적게 사상된 경계 노드 중에서 입출력 링크 대역폭의 합이 가장 큰 노드를 비어있는 셔플넷 노드에 할당한다. 모든 셔플넷 노드가 경계 노드를 할당받았으면 종료하고, 그렇지 않으면 단계 6을 반복한다.

위의 알고리즘에 따라 그림 7의 완전 그물망의 경계 노드가 그림 8처럼 셔플넷 노드에 할당되고 나면, 총 56개이던 논리 링크가 셔플넷 구성 규칙을 만족시키는 조건하에서 각 노드를 출발점으로 하는 링크 중 대역폭이 큰 것 두 개씩만 남아 총 16개로 줄어든다. 만일, 지연 시간을 정확히 하고자 한다면 먼저 최소 지연 시간 경로로 완전 그물망을 구성한 후 각 노드를 출발점으로 하는 링크 중 지연 시간이 작은 것을 두 개씩 남기면 된다.

표 1 경계 노드 수가 N인 PG를 집단화 할 때의 시간 복잡도 비교

집단화 방법	정보 축약 시		정보 복원 시
	정렬	축약/변형	
Lee의 방법[6]	$O(N^2\log N)$	$O(N^2)$	$O(N^2)$
셔플넷 방법	$O(N^2\log N)$	$O(N^2)$	$O(N^2/(pk^2))$

이렇게 줄어든 정보를 다른 동료 그룹에 전송하고, 이웃 그룹의 노드는 연결을 설정하고자 할 때 이 셔플넷 정보로부터 완전 그물망을 재구성한다. 예를 들어 F-G 사이의 링크 대역폭 정보는 F-C와 C-G 대역폭 중 작은 값이 되고, 지연 시간은 두 링크 지연 시간의 합이다. F에서 G로의 경로로는 F-A-B-C-G, F-A-H-D-G 등도 가능하지만, 지연 시간이 너무 증가하므로 (2k-1) 홉 이내의 경로만 고려한다. 3.1절에서 언급했듯이 (2, 2) 셔플넷의 평균 홉 수가 2이므로, 지연 시간이 홉 수에 비례한다고 가정하면 평균 지연 시간은 원래 완전 그물망 정보의 2배 정도이다.

셔플넷 집단화 방법은 위에서도 언급했듯이 유지해야 할 정보의 양을 완전 그물망의 N^2에서 pN 수준으로 줄일뿐만 아니라 시간 복잡도 면에서도 비대칭망에 대한 Lee의 방법[6]보다 유리하다. 우선 정보 축소 과정을 살펴보면, Lee의 방법은 완전 그물망 N^2개의 링크를 대역폭 크기 순으로 정렬하는 복잡도 $O(N^2\log N)$과 생략 가능한 링크를 찾는 복잡도 $O(N^4)$이 필요하다. 반면, 셔플넷 방법은 휴리스틱 알고리즘을 사용할 경우 완전 그물망의 링크를 대역폭 크기 순으로 정렬하는 복잡도 $O(N^2\log N)$과 이 링크들이 연결하고 있는 N개의 노드들을 N개의 셔플넷 노드에 적절히 배치하는 복잡도 $O(N^2)$이 필요하다.

한편, 원래의 완전그물망으로 정보를 복원할 때의 시간 복잡도는 Lee의 방법의 경우 2.2절에서 설명했듯이 $O(N^3)$이고, 셔플넷의 경우에는 $O(N^3/(pk^2))$이다. 셔플넷 방법의 복잡도는 한 노드에서 다른 모든 노드로의 경로를 찾는 복잡도에 노드의 수 N을 곱함으로써 얻을 수 있다. 한 노드에서 다른 모든 노드로의 경로를 찾는 방법은 다음과 같다. 셔플넷은 일반적인 비정형 토폴로지와는 달리 한 노드와 직접 연결을 갖는 노드들이 미리 결정되어 있으므로, (p, k) 셔플넷의 한 노드에서 다른 모든 노드로의 경로를 결정하기 위해 방문해야 하는 노드의 수는 식 (3)으로 계산된다.

$$f(p,\ k) = \sum_{h=1}^{k-1} p^h \ + \ \sum_{h=k}^{2k-1} (p^k - p^{h\%k})p^{h\%k} \qquad (3)$$

홉 수 h가 열의 수 k보다 작을 때 정확히 h홉에 도달할 수 있는 노드의 수는 p^h 개이고, 홉 수가 k 이상이 되어 다시 첫 번째 열로 되돌아오면 다시 마지막 열에 도달할 때까지 매 홉마다 해당 열의 p^k개의 노드 중에 앞에서 이미 찾은 $p^h\%k$개를 제외한 나머지 노드로의 경로를 찾을 수 있다. 단, 이 때는 한 노드로의 가능한 경로가 $p^h\%k$개씩 생기므로, 이들을 서로 비교한 후 하나의 경로를 택하는데, 본 논문에서는 대역폭이 가장 큰 경로를 택했다.

식 (3)의 복잡도를 구하면 $O(N^2/(pk^2))$이 되므로, Lee의 비대칭망 방법에서 한 노드로부터 다른 모든 노드로의 경로를 찾는 알고리즘의 $O(N^2)$에 비해 훨씬 간단하다. 실제로 N=8 (p=2, k=2)인 경우 셔플넷 방법으로는 한 노드에서 다른 모든 노드로의 경로를 찾기 위해 필요한 노드 방문 횟수가 9회인 반면, Lee의 방법에서는 64회이다. N=18 (p=3, k=2)인 경우에는 각각 29회와 324회이고, N=24 (p=2, k=3)인 경우는 각각 41회와 576회로 크게 차이가 난다.

정보 축소 때보다 정보 복원 시의 시간 복잡도가 특히 더 중요한데, 그 이유는 축소는 하나의 동료 그룹에서 일어나지만, 복원은 다른 여러 동료 그룹에서 일어나

기 때문이다. 표 1은 Lee의 비대칭망 방법과 셔플넷 방법으로 N개의 경계 노드를 가진 동료 그룹을 집단화하고, 축약된 정보에서 다시 완전 그물망 정보를 복원할 때 필요한 시간 복잡도를 비교한 것이다. 셔플넷 방법의 완전 그물망 복원 시 복잡도에서 p와 k가 일반적으로 모두 2이상이므로, Lee의 방법보다 훨씬 간단한 것을 알 수 있다.

표 2 노드의 개수에 따른 셔플넷의 종류

$N = kp^k$	(p, k)
1	(1, 1)
2	(1, 2), (2, 1)
3	(1, 3), (3, 1)
...	...
7	(1, 7), (7, 1)
8	(1, 8), (8, 1), **(2, 2)**
18	(1, 18), (18, 1), **(3, 2)**
24	(1, 24), (24, 1), **(2, 3)**
...	...
kp^k	(1, N), (N, 1), ...

3.3 셔플넷 집단화의 문제와 해결 방안

표 2는 N개의 노드로 구성할 수 있는 셔플넷의 종류를 나타낸 것이다. (p, k) 셔플넷에 필요한 링크의 수는 pN이므로, (1, N) 셔플넷을 사용할 경우는 공간 복잡도가 별 모양 방법과 같은 O(N)이며, (N, 1) 셔플넷을 사용할 경우는 완전 그물망 방법과 같은 $O(N^2)$이다. 이때 정보의 정확성도 각각 별 모양 방법, 완전 그물망 방법과 동일하므로, 셔플넷 집단화의 특성을 이용하기 위해서는 p와 k가 1이 되지 않도록 하는 것이 중요하다.

p와 k가 2이상일 때 경계 노드의 수가 셔플넷 노드의 수보다 적어서 셔플넷 노드를 모두 채울 수 없는 경우는 같은 경계 노드를 두 번 할당한다. 예를 들어 경계 노드의 개수가 7인 경우, 셔플넷의 정의에 따라 (1, 7) 또는 (7, 1) 셔플넷을 사용하기보다 일곱 개의 경계 노드 중 입출력 링크 대역폭의 합이 가장 큰 노드를 두 번 할당하여 (2, 2) 셔플넷을 구성한다. (1, 7) 셔플넷의 경우 별 모양 방법과 같으므로 표현되는 링크의 수는 일곱 개이지만 정보가 부정확하며, (7, 1) 셔플넷의 경우 완전 그물망 방법과 같으므로 정보는 정확하지만 표현되는 링크의 수가 49개가 된다. 반면, 이를 (2, 2) 셔플넷에 대응시키면 표현되는 논리 링크의 수는 2N (N=8), 즉 16개로서 (7, 1) 셔플넷의 49개보다 훨씬 적어지며, (1, 7) 셔플넷의 경우보다 정확한 정보를 유지할 수 있다. 그리고, 이렇게 하나의 노드를 복수 개의 셔플넷 노드에 할당하게 되면 한 노드와 연결된 더 많은 링크가 셔플넷에 나타나므로 완전 그물망 정보와의 오차를 줄이는 효과가 있다. 여덟 개의 경계 노드를 가진 동료 그룹을 (2, 2) 셔플넷을 이용하여 집단화한 경우 4장에서 보듯이 87.39%의 대역폭 정보 정확성을 나타낸 반면, 일곱 개의 경계 노드를 가진 그룹은 (2, 2) 셔플넷을 사용하면 하나의 노드가 셔플넷에 두번 나타나서 92.05%로 정확성이 향상되었다.

표 3 완전 그물망 대역폭 상태 정보 (경계 노드: 8개)

경계노드	A	B	C	D	E	F	G	H
A	-	917.16	1337.10	424.52	599.83	424.52	294.56	669.31
B	424.52		424.52	424.52	424.52	424.52	294.56	669.31
C	975.19	917.16		424.52	599.83	424.52	294.56	669.31
D	892.62	892.62	892.62	-	599.83	892.62	294.56	669.31
E	657.39	657.39	657.39	424.52	-	424.52	294.56	657.39
F	1739.54	917.16	2366.67	487.13	599.83	-	294.56	669.31
G	337.12	337.12	337.12	337.12	337.12	337.12	-	337.12
H	424.52	424.52	424.52	424.52	424.52	424.52	294.56	-

마지막으로, 만일 경계 노드의 개수 $N = k_1p_1^{k_1} = k_2p_2^{k_2}$이면, (p_1, k_1) 셔플넷과 (p_2, k_2) 셔플넷 중 어느 것을 사용할 것인지 선택해야 하는 문제가 생긴다. 여기서 선택의 기준이 되는 것은 유지해야할 정보의 양과 정보의 정확성의 상관 관계이다. (p, k) 셔플넷의 링크 수가 pN이므로 p 값이 작을수록 공간 복잡도는 낮아지지만, p 값이 작으면 k 값이 커진다. k 값이 커진다는 것은 논리 링크의 최대 홉 수 (2k-1)와 평균 홉 수가 커진다는 것을 의미하므로, 평균 지연 시간이 커지고 정보의 부정확성이 심화되는 결과를 가져온다. 따라서, N이 지나치게 큰 경우는 정보가 부정확해지더라도 p 값을 작게 할 필요가 있으며, N이 작은 경우는 k를 작게 하는 것이 좋다. 적절한 p와 k의 선택은 망 관리자의 몫이다.

예를 들어, 경계 노드의 수가 15개인 망이 있다고 하면, (1, 15) 셔플넷이나 (15, 1) 셔플넷을 사용하여 경계 노드 수와 셔플넷 노드 수를 정확히 일치시키는 것 보다는 필요없는 셔플넷 노드가 생기더라도 (3, 2) 셔플넷이나 (2, 3) 셔플넷을 사용하는 것이 공간 복잡도나 정보 정확성 측면에서 바람직하다. 이 때도 둘 중 어느 셔플넷을 사용할 것인가 선택해야 하는데, (3, 2) 셔플넷의 경우에는 링크의 개수가 $pN = p*kp^k$에 의해 54개이므로 48개의 링크가 필요한 (2, 3) 셔플넷보다 복잡하지만, 평균 홉 수에서는 식 (2)에 따라 각각 약 2와 3으로 (3,

2) 셔플넷이 유리하다. 앞에서도 언급했듯이 평균 홉 수가 크면 재구성한 완전 그물망의 논리 링크들이 더 많은 물리 링크를 거친다는 것을 의미하므로, 대역폭이나 지연 시간 면에서 더 많은 손해를 보게 된다. 따라서, 이런 경우는 필요한 링크의 개수가 조금 더 많더라도 (3, 2) 셔플넷을 사용하는 것이 낫다. 실제로 54개의 링크는 완전 그물망 방법에서 필요한 N(N-1) = 210개 보다는 훨씬 적은 수이면서도 완전 그물망 정보의 90%에 근접하는 정확성을 보였다.

4. 실험 및 결과 분석

셔플넷 집단화 방법의 효용성을 알아보기 위하여 모의 실험 프로그램을 작성하였다. 실험 방법은 임의로 구성한 동료 그룹의 자원을 완전 그물 방법, 별 모양 방법, Lee의 비대칭망 방법, 셔플넷 방법으로 각각 집단화한 후 각 방법이 얼마나 정확한 상태 정보를 유지하고 있는지 비교하고, 공간 복잡도와 시간 복잡도를 고려하여 각 방법의 효용성을 분석하였다. 단, 신장 트리 방법은 비대칭망에서 사용할 수 없는 방법이므로 비교 대상에서 제외하였다. 실험은 동료 그룹의 경계 노드가 8개일 때와 4개일 때 두 가지 경우로 나누어 진행하였다.

먼저 경계 노드가 8개일 때를 실험하였다. 실험에는 총 50개의 동료 그룹이 사용되는데, SMPL[11]의 랜덤 함수를 이용하여 각 노드 간 연결 상태를 모두 다르게 하였다. 단, 모든 그룹은 공통적으로 15개의 노드를 가지며, 각 노드의 입력 포트와 출력 포트를 4개씩으로 제한하였다. 모든 링크는 비대칭 링크로 각 링크의 전송 용량은 평균이 622Mbps인 지수 분포를 따르도록 하였다.

표 3은 실험에 사용된 50개의 동료 그룹 중 하나를 기반으로 구성한 완전 그물망 대역폭 상태 정보이다. 이 정보가 3.2절의 휴리스틱 알고리즘에 의해 셔플넷 형태로 축약되고, 완성된 셔플넷 정보는 외부 동료 그룹에 전달되어 외부 그룹의 모든 노드들이 이 셔플넷 형태의 정보를 갖게 된다. 그 후 외부 그룹의 노드가 이 그룹을 통과하는 연결을 설정하고자 할 때, 전달받은 셔플넷 정보로부터 완전 그물망을 재구성하여 연결 설정에 이용한다.

표 4 표 3을 기반으로 한 셔플넷 정보에서 재구성한 완전 그물망 대역폭 상태 정보 (경계 노드: 8개)

경계노드	A	B	C	D	E	F	G	H
A	-	917.16	424.52 (31.75)	424.52	424.52 (70.77)	424.52	294.56	669.31
B	424.52	-	424.52	424.52	424.52	424.52	294.56	424.52 (63.43)
C	424.52 (43.53)	424.52 (46.29)	-	294.56 (69.39)	294.56 (49.11)	424.52	294.56	424.52 (63.43)
D	892.62	892.62	892.62	-	294.56 (49.11)	892.62	294.56	669.31
E	424.52 (64.58)	657.39	424.52 (64.58)	424.52	-	424.52	294.56	657.39
F	1739.54	917.16	2366.67	424.52 (87.15)	424.52 (70.77)		294.56	669.31
G	337.12	337.12	337.12	337.12	337.12	337.12	-	337.12
H	424.52	424.52	424.52	424.52	424.52	424.52	294.56	-

표 5 각 집단화 방법의 성능 비교 (경계 노드: 8개)

(완전 그물망 기준)

	대역폭 백분율 (%)			값이 달라진 링크의 개수		
	최대값	최소값	평균	최대값	최소값	평균
별 모양	83.04	8.64	45.73	48	37	44.98
셔플넷	96.03	72.05	87.39	22	5	15.28
Lee의 방법	100	100	100	0	0	0

표 4는 표 3을 기반으로 작성한 셔플넷 정보로부터 재구성한 완전 그물망 상태 정보이다. 굵게 표시된 16개의 값은 원래 완전 그물망의 56개 링크 중에서 생략되지 않고 셔플넷 집단화 정보에 그대로 남아 있었던 링크들의 값이다. 표 3의 A-C 간이나 C-A 간과 같은 상대적으로 큰 값들도 그보다 더 큰 용량을 가진 링크가 먼저 선택되어 각 경계 노드가 셔플넷 노드로 사상되면 셔플넷에 표시되지 못할 수도 있다. 한편, 표 4 중에서 괄호 내의 값들은 해당 링크의 대역폭을 셔플넷 형태로 축소되기 이전의 값과 비교한 백분율이다. 예를 들어 A-E의 칸에 '424.52 (70.77)'이라고 되어 있는 것은 재구성한 완전 그물망에서 A-E 링크의 대역폭이 424.52이고, 이는 원래 완전 그물망 정보에 나타난 대역폭의 70.77%에 해당한다는 의미이다. 괄호와 함께 쓰이지 않은 값들은 원래 완전 그물망 정보와 동일한 경우인데, 이 표의 예에서는 56개의 링크 대역폭 정보 중에서 13개가 원래의 값과 달라졌고, 전체 대역폭을 비교했을 경우에는 재구성한 망이 원래 완전 그물망 대역폭의 87.77%를 나타낸다. 이는 별 모양 방법으로 실험했을 때 나타난 50.52%보다 상당히 정확한 값이고, (2, 2) 셔플넷이므로 집단화 정보에 나타나는 링크의 수도 경계 노드의 수가 N일 때 2N에 지나지 않는다. 단, 별 모양 방법의 예외 우회 링크(exception bypass link)를 설정하는 기준이 명확치 않아 이 논문의 실험에서는 기본 링크(default link)만을 사용하였음을 미리 밝히며, 셔플넷 방법에서도 표 4의 A-C 구간처럼 집단화 이전의 정보와 이후 셔플넷에서 얻어낸 정보가 크게 차이가 나는

링크의 경우 우회 링크를 이용하여 보정할 수 있으나, 별 모양 방법과의 공정한 비교를 위하여 사용하지 않았다.

별 모양 방법과 셔플넷 방법, Lee의 방법을 50개의 동료 그룹에 대해 적용한 결과는 표 5와 같다. 원래의 완전 그물망을 기준으로 했을 때 별 모양 방법으로 표현할 수 있는 대역폭의 범위는 최소 8.64%에서 최대 83.04%였고, 평균은 45.73%였다. 기본적으로 모든 망의 링크 대역폭 평균이 비슷함에도 불구하고 최대값과 최소값이 이렇게 크게 차이가 나는 이유는 별 모양 방법의 기본 링크가 하나의 경계 노드에서 다른 모든 경계 노드로의 최대 대역폭 경로 중 최소값으로 결정되기 때문이다. 하나의 동료 그룹에 속한 여러 링크들의 대역폭이 크게 다르면, 집단화로 인한 링크 상태 정보의 손실은 더 클 수밖에 없다. 오른쪽의 링크 개수는 별 모양 방법으로 바꾸었을 때 완전 그물망의 링크 중에 대역폭 값이 달라진 링크가 몇 개인지를 나타낸다.

한편, 셔플넷 방법으로는 완전 그물망 상태 정보에 나타나는 대역폭의 최소 72.05%에서 최대 96.03%까지 표현할 수 있고, 평균은 87.39%에 달했다. 값이 달라진 링크의 개수는 총 56개의 완전 그물망 링크 중에서 평균 15.28개였다. 셔플넷 방법에 필요한 링크의 수가 pN(p: 정수, N:경계 노드 수)으로 별 모양 방법의 N보다 복잡하긴 하지만, 정보의 정확도가 모든 동료 그룹에서 별 모양 방법보다 나았고, 평균적으로는 두 배에 가까웠다. 공간 복잡도가 $O(N^2)$인 완전 그물망과 비교해서도 평균 13% 정도의 오차밖에 없었고, 이는 셔플넷 방법이 완전 그물망 방법의 경우보다 훨씬 확장성이 있어서 대규모의 망에 적용할 수 있음을 고려한다면 큰 단점은 아니다.

마지막으로 Lee의 방법은 대역폭 정보가 완전 그물망의 정보와 달라질 경우에는 정보를 축약하지 않으므로 정확성은 언제나 완전 그물망과 같다. 그러나, Lee의 방법은 공간 복잡도가 $O(N^2)$으로 경우에 따라서 정보의 양을 거의 줄이지 못할 수 있으며[5] 3.2절에서 이미 보았듯이 계산 시간 복잡도가 셔플넷 방법에 비해 훨씬 높다. 특히 축소된 정보로부터 완전 그물망을 재구성하는 일은 주위 여러 동료 그룹 내의 노드들에서 하게 되므로, 이 복잡도가 낮다는 것은 Lee의 방법에 비해 셔플넷 방법이 크게 유리한 점이다.

표 6 평균 홉 수의 비교 (경계 노드: 8개)

(단위: 홉 수)

망 번호	완전 그물	별 모양	셔플넷	Lee의 방법
1	3.36	4.75	5.39	12.35
2	3.41	4.38	5.46	9.27
3	4.46	4.38	8.43	12.90
4	3.27	4.12	4.66	4.88
5	2.59	3.38	3.59	6.79

지연 시간은 대역폭과 함께 링크 상태 정보에 있어서 가장 중요한 인자이다. 이 논문에서는 망을 구성하는 모든 물리적 링크의 지연 시간이 같다고 가정하고, 집단화한 이후의 모든 경로의 지연 시간은 물리 링크의 홉 수에 비례한다고 가정하였다. 표 6은 각 집단화 방법에 나타난 경로의 평균 홉 수를 나타낸다. 대체로 완전 그물망 경로의 홉 수가 가장 작고, Lee의 방법으로부터 재구성한 망의 평균 홉 수가 가장 크다. (2, 2) 셔플넷의 평균 홉 수는 3.1절의 식 (2)에 의해 2라는 것을 알았는데, 이는 셔플넷 상의 하나의 링크가 완전 그물망 링크 하나와 대응되므로 셔플넷 상의 두 노드간 경로는 평균적으로 완전 그물망 링크 두 개의 연결과 같다는 것을 의미한다. 따라서 셔플넷으로부터 재구성한 완전 그물망 링크의 평균 홉 수는 집단화 이전의 완전 그물망 링크 홉 수의 두 배 정도가 되어야 하는데, 실제 실험에서도 일반적으로 두 배에 약간 못 미치는 값으로 나타났다. 앞에서 모든 물리 링크의 지연 시간이 동일하다고 가정했으므로, 셔플넷에서 재구성한 완전 그물망은 원래의 완전 그물망에 비해 지연 시간이 두 배 정도로 표현됨을 알 수 있다. 여기서 주의해야 할 것은 별 모양 방법이 지연 시간면에서 셔플넷 방법보다 우수한 것처럼 보이지만, 대신 대역폭면에서 크게 손실이 있음을 간과해

표 7 평균 홉 수의 비교 (경계 노드: 4개)

(단위: 홉 수)

망 번호	완전 그물	별 모양	셔플넷	Lee의 방법
1	3.83	5.25	5.33	6.12
2	4.00	4.75	5.00	5.19
3	3.67	5.00	4.00	7.38
4	3.75	3.50	4.58	4.45
5	4.42	5.50	5.58	8.01

5) 우리의 실험에서는 경계 노드가 8개인 경우 Lee의 방법에 나타나는 링크의 수가 14개에서 32개로 다양하게 나타났다. 반면 셔플넷 방법으로는 (2, 2) 셔플넷을 사용할 수 있으므로 언제나 16개이다. 또한, 경계 노드가 12개일 때 셔플넷 방법에서는 (3, 2) 셔플넷을 사용하느냐 (2, 3) 셔플넷을 사용하느냐에 따라 54개 또는 48개의 링크 정보가 필요하나 Lee의 방법에서는 최소 44개에서 최대 104개로 변화가 심했다.

서는 안된다. 또, Lee의 방법의 경우에는 대역폭 정보는 정확하지만 지연 시간을 크게 왜곡시키는 경향이 있음도 알 수 있다.

표 7과 표 8은 경계 노드가 4개일 때의 결과이다. 망의 구성 방법은 이전 실험과 동일하고 이번에도 역시 50개의 동료 그룹에 대해 실험했다.

5. 결론 및 향후 과제

근원지 경로 배정을 하는 PNNI의 각 노드는 효율적인 경로 배정을 하기 위해서 시시각각으로 변화하는 망 전체의 자원 분포 형태를 항상 알고 있어야만 한다. 대표적인 망 자원으로는 가용 대역폭과 지연 시간을 들 수 있고, 이들 자원에 대한 정보는 PTSE 메시지에 담겨 주기적으로 또는 노드와 링크 상태 정보에 중요한 변화가 있을 때, PNNI 계층 구조를 따라 올라가며 각 노드가 속한 동료 그룹 내에 플러딩 된다. 이때, 만일 망의 규모가 매우 커지게 되면 경로 배정 테이블 관리를 위한 PTSE 메시지 자체가 심각한 전송 부담을 줄 수 있으므로, 망 자원 정보를 효과적으로 간략화 할 수 있는 방법에 대한 연구가 필요하다. 대표적인 기존 연구로는 완전 그물망 방법과 별 모양 방법이 있으나, 전자는 정보의 양을 축소하는 정도가 미미하여 규모가 큰 망에 대한 확장성이 떨어진다는 단점이 있고, 후자는 정보가 심각하게 부정확해진다는 단점이 있다.

표 8 평균 홉 수의 비교 (경계 노드: 4개)

(단위: 홉 수)

망 번호	완전 그물	별 모양	셔플넷	Lee의 방법
1	3.83	5.25	5.33	6.12
2	4.00	4.75	5.00	5.19
3	3.67	5.00	4.00	7.38
4	3.75	3.50	4.58	4.45
5	4.42	5.50	5.58	8.01

본 논문에서는 완전 그물망 집단화 방법의 복잡성과 별 모양 방법의 정보 부정확성으로 인한 망 효율 저하 문제를 해결하기 위하여 셔플넷을 이용한 방법을 제안하였다. 셔플넷 방법은 집단화하려는 동료 그룹의 경계 노드를 셔플넷 노드에 사상시킨 후 셔플넷 링크 설립 규칙에 따라 링크가 생기는 두 경계 노드 사이의 링크 정보만 남기고 나머지는 생략하는 방법이다. (p, k) 셔플넷의 경우에 총 pN (N은 경계 노드 수)개의 링크가 생기므로, $O(N^2)$ 개의 링크를 표현해야 하는 완전 그물망 방법에 비해 망 크기의 증가에 따른 확장성 면에서 훨씬 유리하고, 정보의 정확성에서도 가용 대역폭 정보의 경우에는 완전 그물망에 근접하는 수준을 얻을 수 있음을 실험을 통해 알 수 있었다. 단, 지연 시간은 정보의 양을 축소하는 정도에 비례하여 증가하는 현상을 보였다.

셔플넷 집단화가 가지는 또 하나의 장점은 양방향으로 서로 다른 전송 용량을 갖는 비대칭망에서 사용할 수 있는 방법이라는 것이다. 신장 트리 방법을 비롯한 이전의 많은 방법들이 대칭망을 가정하고 제안되었기 때문에 단순하긴 하지만, 실제 망에 적용되기에는 어려운 점이 있었던 것을 해결한 방법이라고 할 수 있다. 또한, 비대칭망 집단화를 위해 Lee가 제안한 방법[6]과 비교해서는 셔플넷 집단화에 필요한 시간 복잡도가 현저히 낮아서 과도한 계산 작업을 요구했던 문제를 개선했다. Lee의 방법은 공간 복잡도가 역전히 $O(N^2)$이고, 경우에 따라서는 지연 시간 정보를 심각하게 왜곡시킬 수 있는 단점이 있었다.

앞으로 남아있는 문제는 집단화에 수반하는 지연 시간 정보의 악화를 최대한 억제할 수 있는 방법을 찾는 것이다.

참 고 문 헌

[1] P. V. Mieghem. Estimation of an Optimal PNNI Topology. In Proceedings of IEEE ATM, pages 570-577, 1997.

[2] R. Guérin. QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms. IBM Research Report, Jul. 1996.

[3] The ATM Forum. Private Network-Network Interface Specification Version 1.0 (PNNI 1.0). 1996.

[4] W. C. Lee. Spanning Tree for Link State Aggregation in Large Communication Networks. In Proceedings of IEEE INFOCOM, pages 297-302, 1995.

[5] D. E. McDysan and D. L. Spohn. ATM Theory and Application (2nd ed.). McGraw-Hill, 1999.

[6] W. C. Lee. Topology Aggregation for Hierarchical Routing in ATM Networks. ACM SIGCOMM Compuer Communication Review, 25(2):82-92, Apr. 1995.

[7] D. Bertsekas and R. Gallager. Data Networks. Prentice-Hall, 1992.

[8] B. Awerbuch and Y. Shavitt. Topology Aggregation for Directed Graph. In Proceedings of the 3rd IEEE Symposium on Computers and Communications, pages 47-52, 1998.

[9] A. Iwata, H. Suzuki, R. Izmailov, and B. Sengupta. QOS Aggregation Algorithms in Hierarchical ATM Networks. In IEEE International Conference on Communications, pages 243-248, 1998.
[10] B. Mukherjee. Optical Communication Networks. McGraw-Hill, 1997.
[11] M. H. MacDougall. Simulating Computer Systems Techniques and Tools. MIT Press, 1987.

유 영 환
정보과학회논문지 : 정보통신
제 27 권 제 2 호 참조

안 상 현
정보과학회논문지 : 정보통신
제 27 권 제 2 호 참조

김 종 상
정보과학회논문지 : 정보통신
제 27 권 제 2 호 참조