

웜홀 방식의 네트워크에서 경로전송을 기반으로 하는 효율적인 다중전송 알고리즘

(An Efficient Path-Based Multicast Algorithm in
Wormhole-Routed Networks)

김 시관*
(Si-Gwan Kim)

요약 본 논문에서는 웜홀 스위칭 방식을 이용하는 메쉬 구조를 갖는 네트워크에서 경로를 기반으로 하는 효율적인 다중전송 알고리즘을 제안한다. 제안한 다중전송 알고리즘은 웜홀 스위칭 방식의 경로 기반 다중전송이 갖는 특성을 이용하고, 종래의 하나의 해밀턴 경로만 이용한 알고리즘의 단점을 극복한다. 제안한 다중전송 알고리즘은 2개의 각 가상채널에 해밀턴 경로를 사용하여 다중전송 메시지가 트래픽의 양에 따라 임의의 해밀턴 경로를 선택하여 전송하기 때문에 기존의 단일 해밀턴 경로를 사용한 방법보다 링크를 보다 효율적으로 사용하여 전반적인 성능을 향상시킨다. 그리고, 본 논문에서 제안한 다중전송 알고리즘이 교착 상태가 없음을 증명하고 기존의 방법에 비해서 약 15~20%의 성능 향상을 보임을 시뮬레이션을 통하여 검증하였다.

Abstract In this paper, we present an efficient path-based multicast algorithm in wormhole-routed mesh networks. Our algorithm is based on a network partitioning strategy that uses two Hamiltonian paths. In the previous studies, only one Hamiltonian path was used. Thus messages traverse more horizontal channels than vertical ones, leading to earlier network congestion. By incorporating additional vertical Hamiltonian path as well as the horizontal Hamiltonian path, messages are distributed evenly as much as possible, thus making network performance better. We prove that this algorithm is deadlock-free. And by extensive simulations, we show that this algorithm is superior to the previous ones by 15~20%.

1. 서론

고성능 컴퓨터를 설계할 때 구성하는 프로세서의 수를 증가시키으로써 전체 시스템의 성능이 점차 향상시킬 수 있는 확장성이 있는 병렬 컴퓨터(Scalable Computer)[16]에 대한 연구가 이루어지고 있다. 이러한 병렬 컴퓨터 중에서, 다중컴퓨터(multicomputer)는 분산 메모리를 갖는 다중처리기로서 여러 개의 노드 컴퓨터로 이루어져 있으며, 이 노드 컴퓨터는 기억 장치가 물리적으로 분산되어 있기 때문에 상호연결망을 통해서 메시지를 주고 받음으로써 다른 노드 컴퓨터와 통신을 한다. 각 노드 컴퓨터는 프로세서와 지역 메모리 그리고 통신 장치인 라우터로 구성되어 있다.

다중컴퓨터의 성능은 노드 컴퓨터를 연결하는 상호연결망의 성능에 의해 큰 영향을 받으므로, 상호연결망의 성능을 향상시키는 효율적인 통신 방법이 필요하다[1, 3]. 다중컴퓨터의 통신 방법은 다음과 같이 세 가지로 분류할 수 있다. 첫째는 메시지를 보내기를 원하는 하나의 출발 노드가 하나의 목적 노드로 메시지를 보내는 단일전송(unicast, one-to-one) 기법이고, 둘째는 하나의 출발 노드가 하나 이상의 목적 노드로 동일한 메시지를 보내는 다중전송(multicast, one-to-many) 기법이고, 세 번째는 출발 노드가 자신을 제외한 모든 노드로 동일한 메시지를 보내는 방송(broadcast, one-to-all) 기법이다. 단일전송과 방송은 목적 노드가 하나이거나 모두인 경우의 다중전송의 특수한 경우이다.

*안동대학교 정보전자산업학부

다중전송 기법은 병렬 탐색 알고리즘이나 병렬 그래프 알고리즘과 같은 병렬 알고리즘 분야[4, 9]와 동일한 프로

그램이 다른 데이터를 가지고 다른 처리기에서 수행되는 SPMD(Single Program Multiple Data) 형태의 연산이 쓰이는 분야에 적용된다. 특히, 다중전송은 데이터 병렬 언어(data parallel language)에서 제공되는 경계선 동기화(barrier synchronization)[13]와 같은 연산에 필수적이다[5]. 또한 분산 공유 메모리 시스템에서 공유 자료들의 일관성(coherence) 유지 프로토콜을 위한 공유 자료의 수정과 무효화(invalidation)에도 효과적으로 사용될 수 있다[11].

다중전송 기법[6, 8, 12, 20, 22, 23]은 하드웨어적인 방법 또는 소프트웨어적인 방법으로 구현될 수 있다. 그러나, 대부분의 다중컴퓨터에서는 점대점 방식의 단일전송 기법만을 하드웨어적인 방법을 통해서 제공한다. 이러한 환경의 다중컴퓨터에서는, 다중전송 기법은 하나 또는 그 이상의 단일전송 메시지를 전달하는 방식을 통한 소프트웨어적인 방법으로 구현되어야 한다. 이러한 환경의 다중컴퓨터에서는, 다중전송 기법은 하나 또는 그 이상의 **단일전송을 기반으로 하는 다중전송(unicast-based multicast)**이라 부른다[14].

단일전송을 기반으로 하는 다중전송을 구현하는 방법 중 하나는 출발 노드가 전송할 메시지를 목적 노드의 갯수만큼 복사하여 직접 모든 목적 노드에 각각의 메시지를 전달하는 **개별 주소법(separate addressing)**이다. 그러나, 개별 주소법은 목적 노드의 수가 증가할수록 다중전송 메시지를 전달하는데 꽤나 많은 시간이 요구된다. 왜냐하면, 대부분의 시스템에서 네트워크를 구성하는 노드 프로세서는 동시에 단지 몇 개 이하의 메시지만을 보낼 수 밖에 없기 때문이다. 다른 방법으로는 출발 노드가 전송할 메시지들의 목적 노드들 중 일부분에만 메시지를 전달하는 **다중전송 트리(multicast tree)** 방법이다. 이 방법에서 메시지를 받는 노드들은 아직까지 메시지를 받지 않는 목적 노드들 중의 일부분에 메시지를 전달함으로써 다중전송을 구현한다[9].

점대점 방식의 다중컴퓨터 네트워크에서 소프트웨어적인 방법으로 다중전송을 구현하는 데의 주요 논점은 **McKinley**가 기술하였다[14]. 또한, **McKinley**는 다차원 메쉬(mesh) 구조를 갖는 점대점 방식의 네트워크에서 노드 프로세서가 동시에 보낼 수 있는 메시지가 하나로 제한되는 **단출구 통신(one-port communication)** 구조일 경우, 단일전송을 기반으로 하는 최적의 다중전송 알고리즘을 제안하였다.

최근에는 **전출구 통신(all-port communication)**을 지원하는 노드 프로세서를 가지는 점대점 방식의 네트워크에서

다중전송을 단일전송으로 효율적으로 구현한 알고리즘이 제안되었다[2, 8, 17, 18, 21]. 이러한 알고리즘 중에서 하이퍼큐브(hypercube) 네트워크에 대한 단일전송을 기반으로 하는 최적의 방송 알고리즘도 제안되었다[8, 17].

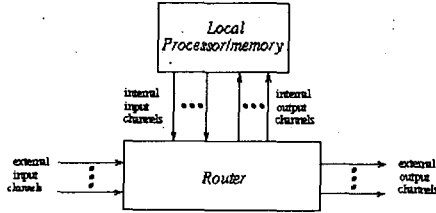
다중통신을 지원하기 위해서는 다음과 같은 조건이 필요하다. 첫째, 출발지에서 각각의 목적지로의 메시지 지연 시간은 되도록 짧은 것이 좋다. 가장 짧은 경로로 별개의 메시지를 각각의 목적지로 보내는 것도 하나의 방법이 될 수 있으나 트래픽이 많아지는 단점을 가지고 있다. 둘째로, 망의 트래픽 양이 최소화되어야 한다. 그러나, 지연시간과 트래픽의 양의 측면에서 볼 때, 최적의 다중 전송 경로를 찾는 것은 NP-hard 문제에 해당된다. 셋째로, 라우팅 알고리즘이 비교적 간단하여야 한다. 휴리스틱 알고리즘을 사용할 수 있지만 교착 상태가 없어야 한다는 점이다.

하이퍼큐브 구조에서는 트리 기반의 멀티캐스트 알고리즘은 XY 채널, 즉, 2개의 채널을 사용하여 해결한 알고리즘[10]이 제시되었지만 교착상태 발생의 문제점이 있다. **McKinley**[14]는 2개의 채널을 사용하여 메쉬에서 교착 상태 없는 알고리즘을 제시하였지만 시그널링 체계가 복잡한 단점을 가지고 있다. 최근에는 경로 기반의 다중전송 알고리즘이 제시되었는데 경로를 한 개만 사용하기 때문에 채널 이용률이 낮다는 단점을 가지고 있다.

본 논문에서는 메쉬 구조를 갖는 네트워크에서 경로에 기반을 둔 효율적인 다중전송 알고리즘을 제안한다. 제안하는 다중전송 알고리즘은 2개의 해밀톤 경로를 사용하여 트래픽의 양에 따라 경로를 선택할 수 있기 때문에 다중전송 때 뿐만 아니라 단일전송에서도 종래에 하나의 해밀톤 경로를 사용하는 것보다 더 높은 성능을 보여 준다. 그리고, 본 논문에서 제안한 다중전송 알고리즘이 논리 채널을 한 개만 사용하는 기존 알고리즘에 비해서 성능 향상을 시뮬레이션을 통하여 검증하고 교착 상태가 없음을 증명한다.

본 논문의 구성은 다음과 같다. 2절에서는 스위칭 기법과 네트워크 구조와 라우터의 기능 그리고 기존의 다중전송 알고리즘에 대해서 기술하고, 3절에서는 메쉬 구조를 갖는 네트워크에서의 경로 기반으로 하는 효율적인 다중전송 알고리즘을 제안한다. 4절에서 제안한 네트워크의 다중전송 알고리즘을 시뮬레이션을 통해서 성능을 비교하고, 5절에서 결론을 맺는다.

2. 시스템 모델(System Model)



<그림 1> 일반적인 노드의 구조

메시지 개시지연(startup latency)과 네트워크 지연(network latency), 그리고 블통 시간(blocking time)의 합으로 이루어지는 메시지 통신지연은 다중컴퓨터 시스템의 성능을 측정하는 척도이다[15]. 메시지 개시지연은 네트워크 시스템이 출발노드와 목적 노드 두 곳에서 메시지를 처리하는데 소요되는 시간이고, 네트워크 지연은 메시지의 시작 플릿(flit)이 출발 노드에서 네트워크에 들어간 시각부터 메시지의 끝 플릿이 목적 노드에 들어올 시각까지 소요된 시간이다. 블통 시간은 메시지의 전송기간 동안에 일어나는 가능한 모든 지연을 포함하고, 예를 들어, 망에서 통신 자원을 할당받기 위해서 경쟁하는데 소요되는 지연이 있다.

멀티캐스트의 성능을 측정하기 위한 멀티캐스트 지연(multicast latency)은 출발 노드가 메시지의 첫 복사본을 보내는 시각부터 마지막의 목적 노드가 메시지를 받는 시각까지 소요된 시간이다. 멀티캐스트 지연은 시스템 구조의 특성에 많은 영향을 받으며, 본 논문에서는 네 가지의 특성으로 구분되는 시스템 구조에서 효율적인 멀티캐스트 알고리즘을 구현하는 방법을 제안한다.

다중컴퓨터 시스템 구조는 메시지를 전달하는 스위칭 방식에 의해 결정된다. 대부분의 다중컴퓨터는 메시지를 여러 개의 고정된 크기의 플릿(flit)으로 분할하여 각각을 네트워크를 통해서 파이프라인 형태로 목적 노드까지 전달하는 웜홀(wormhole) 스위칭 방식이 널리 쓰이고 있다[4].

메시지의 길이가 길 경우에, 웜홀 라우팅의 파이프라인 효과로 인하여 메시지가 가야할 거리는 네트워크 지연에 큰 영향을 주지 않는다[15]. 메시지의 길이가 짧을 경우에, 메시지 개시지연은 단일전송 메시지에 대한 지연에 큰 영향을 미친다. 그러므로, 네트워크 자원에 대한 메시지들간의 경쟁이 없을 경우에는 웜홀 스위칭 방식에서 메시지 통신지연은 거의 거리에 무관하게 되며 메시지 개시지연 시간에 상당히 의존하게 된다. 웜홀 스위칭 방식의 가장 큰 문제점은 메시지들이 여러 채널을 점유하게 되므로 교착상태가 발생할 가능성이 크다는 점이다.

다중컴퓨터 시스템 구조를 결정하는 두 번째 요소는 네트워크의 구조이다. 본 논문에서 다루는 다중컴퓨터 시스템의 네트워크 구조인 메쉬(mesh)는 다음과 같이 정의된다. n 차원 메쉬일 경우, 각 차원을 i , $0 \leq i \leq n-1$ 이라 하고 각 차원 i 에서 $k_i (\geq 2)$ 개의 노드를 기반으로 이루어진다면, 다차원 메쉬 구조는 $k_0 \times k_1 \times \dots \times k_{n-2} \times k_{n-1}$ 개의 노드로 구성되어 있다. 임의의 노드 x 는, 각 차원 i , $0 \leq i \leq n-1$ 에 대해서 $0 \leq a_i(x) \leq k_i-1$ 라 하면, $a_0(x)a_1(x)\dots a_{n-1}(x)$ 의 n 개의 좌표로 정의된다. 두 노드 x 와 y 에 대해서, 하나의 차원 j 에 대해서만 $a_j(y) = a_j(x) \pm 1$ 이고 j 를 제외한 나머지 모든 차원 i 에 대해서 $a_i(y) = a_i(x)$ 이기만 하면, x 와 y 는 연결되어 있다. 다중컴퓨터 시스템의 대부분의 네트워크 구조는 2차원 메쉬, 3차원 메쉬 그리고 하이퍼큐브를 포함하는 다차원 메쉬 구조의 특별한 경우이다.

다중컴퓨터 구조를 결정하는 요소인 라우터(router)는 노드 프로세서에 개별적으로 구성되어 있으며, 노드 프로세서간의 통신을 처리한다. 그림 1의 노드 구조와 같이, 노드의 라우터가 여러 개의 외부 채널들을 통하여 이웃의 라우터와 연결되는데 이러한 외부 채널이 연결되는 방식은 다중컴퓨터 네트워크의 구조에 의해 결정된다[2]. 라우터로 들어오는 메시지들이 서로 다른 채널을 통해서 나가려고 한다면, 라우터는 여러 개의 메시지를 동시에 전달할 수 있다. 그와 더불어, 노드의 라우터와 이웃하는 라우터 사이에서 서로 반대의 방향으로 두개의 메시지를 동시에 전송하는 것도 가능하다. 라우터는 노드 프로세서와 여러 개의 내부 채널들로 연결되어 있다. 내부 채널은 노드 프로세서에서 들어오는 입력 채널과 노드 프로세서로 들어가는 출력 채널로 구분된다.

다중컴퓨터 구조를 결정하는 네 번째 요소로 라우터가 동시에 복사할 수 있는 메시지의 갯수와 라우터로 노드 프로세서가 동시에 보낼 수 있는 메시지의 개수인 통신 출구의 개수이다. 단출구 통신은 메시지를 보내는 단계마다 최대 두 배만큼의 노드 프로세서가 메시지를 받을 수 있기 때문에 m 개의 멀티캐스트의 목적 노드로 메시지를 보내는 경우에 최소 $\lceil \log_2(m+1) \rceil$ 만큼의 단계가 메시지 전송 단계에 필요하다. 대부분의 멀티캐스트 알고리즘은 단출구 통신만을 지원하는 라우터를 사용하며 본 논문에서도 이를 가정한다.

노드 프로세서와 라우터가 동시에 여러 개의 메시지를 전송하기 위해서는 다음과 같은 기능이 필요하다. 노드 프로세서는 그림 1에서와 같이 라우터에서 노드 프로세서로 출력되는 메시지가 사용하는 출력 채널이 출력 버퍼로 향

하는 채널과 입력 채널로 향하는 채널 두 가지로 구분된다. 라우터는 출력 채널로 들어오는 메시지를 여러 개 복사할 수 있는 기능과 출력 버퍼로 향하는 채널로 하나를 보내고 나머지를 입력 버퍼로 향하는 채널로 보내는 기능이 필요하다. 이러한 기능을 갖는 라우터로써 동시에 여러 개의 메시지를 전송할 수 있는 다출구 통신을 이용한 멀티캐스트를 구현할 수 있다.

웜홀망에서 사용하는 채널을 구현하기 위해 각 채널을 물리적인 선들의 집합으로 구성하게 되면 비용이 올라가게 된다. 일반적으로 물리적인 채널은 약간의 하드웨어 비용 추가로 여러 개의 가상 채널로 구성하게 된다. 가상 채널을 사용하게 되면 물리적인 망은 여러 개의 독립적인 논리적 망으로 구성하게 되어 망의 접속율과 대역폭을 향상시키는 효과를 갖게 된다. 본 논문에서는 2개의 가상 채널을 사용한다고 가정한다.

3. 메시 네트워크에서 다중전송 알고리즘

경로를 기반으로 하는 다중전송 알고리즘을 이론적으로 기술하기 위해서, 다중컴퓨터 네트워크 구조를 노드 집합 $V(G)$ 와 연결선 집합 $E(G)$ 의 방향성 그래프인 (V, E) 로 표현한다.

노드 집합 $V(G)$ 의 한 노드 u 는 라우터를 가지는 노드 프로세서 u 를 나타내고, 연결선 집합 $E(G)$ 의 연결선 (u, v) 는 라우터 u 에서 라우터 v 로의 단방향 연결선을 나타낸다.

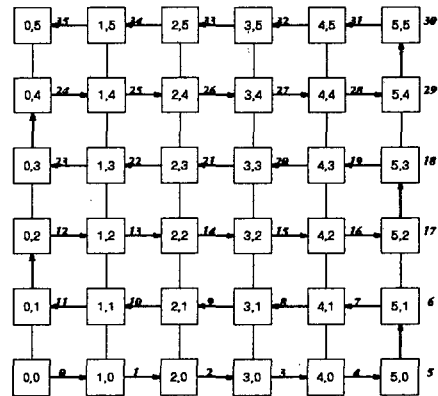
해밀턴 경로(Hamiltonian Paths)에 기반을 둔 망 분할 방법은 교착 상태가 없는 라우팅 알고리즘의 기본이 된다. 해밀턴 경로[7]는 그래프내에서 모든 노드를 단 한번만 방문하는 것이다. 2차원 메시에서는 여러 개의 해밀턴 경로가 존재한다. 여기서는 2개의 해밀턴 경로를 사용하는데 이는 다음과 같다.

정의 1 임의의 메시망에서 해밀턴 경로의 진행 방향이 메시의 최대 좌표와 최소 좌표에서의 방향 전환은 남쪽 혹은 북쪽 방향으로 이루어지고 그 이외는 동쪽 혹은 서쪽 방향으로만 전환되는 해밀턴 경로를 수평 해밀턴 경로(Horizontal Hamiltonian Path : *HHT*)라 한다.

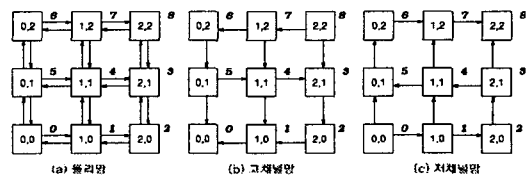
정의 2 임의의 메시망에서 해밀턴 경로의 진행 방향이 메시의 최대 좌표와 최소 좌표에서의 방향 전환은 동쪽 혹은 서쪽 방향으로 이루어지고 그 이외는 남쪽 혹은 북쪽 방향으로만 전환되는 해밀턴 경로를 수직 해밀턴 경로

(Vertical Hamiltonian Path : *VHT*)라 한다.

수평 해밀턴 경로와 수직 해밀턴 경로의 예가 그림 2와 그림 3에 있다. 본 알고리즘에서는 각 노드 u 마다 수평 해밀턴 경로에 해당되는 라벨 $l(u)$ 와 수직 해밀턴 경로에 해당되는 라벨 $m(u)$ 로 2개씩 주어진다. 라벨의 값은 N 개의 노드가 있다면 각 해밀턴 경로의 맨 처음 노드는 0인 값을 갖고 경로의 마지막 노드는 $N-1$ 인 값을 갖게 된다. 각 해밀턴 경로에 따라 라벨 값에 따라 그림 3과 같이 망을 2개로 분할할 수가 있다. 즉, 채널의 방향이 낮은 쪽에서 높은 곳으로 향하게 되는 고채널망과 높은 곳에서 낮은 곳으로 향하는 저채널망으로 분할할 수 있다. 다중전송뿐만 아니라 단일전송도 같은 라벨 기법에 의해 생성되는 경로를 따라서 라우팅한다. 즉, 목적 노드의 라벨의 크기가 출발 노드보다 크면 고채널망을 따라 라우팅되고 목적 노드의 라벨 크기가 출발 노드보다 작으면 저채널망을 따라 라우팅된다.



<그림 2> 2차원 메시에서 수평 해밀턴 경로의 예

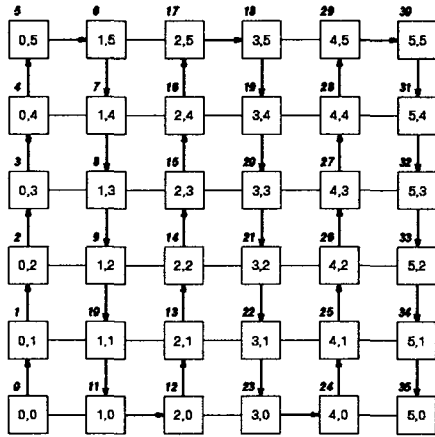


<그림 3> 수평 해밀턴 경로에서의 저채널망과 고채널망의 예

라벨 값은 노드의 위치에 해당되는 x 와 y 의 좌표에 따라 다음과 같은 함수 l 과 m 로 결정되며 주어진 망을 l 과 m 에 의해 노드에 라벨링한 예가 각각 그림 2와 그림 4에 나타나 있다.

$$l(x, y) = \begin{cases} y * n + x & \text{if } y \text{ is even} \\ y * n + n - x - 1 & \text{if } y \text{ is odd} \end{cases} \quad \text{--- (1)}$$

$$m(x, y) = \begin{cases} x * n & \text{if } x \text{ is even} \\ x * n + n - y - 1 & \text{if } x \text{ is odd} \end{cases} \quad \text{--- (2)}$$



<그림 4> 2차원 메쉬에서 수직 해밀톤 경로의 예

다중통신 알고리즘에서 임의의 메시지의 트래픽의 양에 따라 수평 해밀톤 경로 혹은 수직 해밀톤 경로를 택할 수 있다. 트래픽의 양은 동, 서, 남, 북방향의 각 입출력 채널을 통과하는 메시지의 양을 주기적으로 모니터링함으로써 측정할 수 있다. 그러므로, 메시지를 전송할 때는 각 채널의 트래픽 양을 검사한 뒤 트래픽의 양이 적은 채널로 전송을 하게 된다. 출발지에서 그 경로(즉, 채널)가 결정되면 메시지 전송 중에는 그 경로를 변경할 수 없다. 다중전송을 하기 위해서는 다음과 같은 준비 과정이 필요하다.

목적 노드 집합 D 를 두개의 부분집합 D_H 와 D_L 로 분할한다. D_H 는 출발지 노드 u 보다 라벨의 값이 큰 노드들로 구성되고 D_L 는 출발지 노드 u 보다 라벨의 값이 작은 노드들로 구성된다. 다중전송은 출발지 노드 u 에서 수평 해밀톤경로와 수직 해밀톤경로 중 한 경로를 선택하여 노드 집합 D_H 를 고체널망을 통해서 전달하고 노드 집합 D_L 을 저

체널망을 통해서 전달한다. 출발 노드에서 메시지를 라벨의 크기순으로 배열시켜 준비하는 과정은 [12]와 동일하다. 그림 5는 본 논문에서 제안하는 다중전송 알고리즘 MDP 를 보여준다.

다중전송 알고리즘은 라우팅의 결정이 매 중간 노드에서 이루어지는 분산 라우팅 알고리즘이다. 각각의 노드가 메시지를 수신하면 메시지 헤더의 첫 번째 목적지 주소와 일치하는지를 검사한다. 일치하면 그 첫 번째 주소는 메시지 헤더에서 삭제되고 메시지 내용을 프로세서부로 전달한다. 이 시점에 아직 메시지 헤더의 주소 부분이 비어있지 않으면 라우팅 함수를 사용하여 메시지 헤더의 첫 번째 주소에 해당되는 노드로 계속 전송이 된다.

다중전송 알고리즘의 예가 그림 6에 나타나 있다. 이 예에서는 2개의 메시지 M_1 과 M_2 가 동시에 전송되는 예로서 기존의 알고리즘을 적용하면 메시지 M_1 에 의해서 수평 채널이 사용중이기 때문에 이 M_1 메시지의 개시가 지연이 되지만 제시한 알고리즘으로는 수직 해밀톤 경로를 사용할 수 있기 때문에 훨씬 더 짧은 지연시간이 있음을 알 수 있다. 이 그림에서 노드의 오른쪽 숫자는 수평 해밀톤 경로의 라벨 값이고 위쪽의 숫자는 수직 해밀톤 경로의 라벨 값을 나타낸다.

정리 다중전송 알고리즘 MDP 는 교착 상태가 없다.

증명 해밀톤 경로는 [7]에 의해 루프를 형성하지 않는다는 것을 알 수 있다. 그러므로, 교착 상태 발생 배제 조건, 즉, 루프를 형성하지 않게 자원을 배치하면 된다는 원칙에 의해 해밀톤 경로를 따라 메시지를 전송하면 교착 상태를 방지할 수 있다[15].

MDP 알고리즘에서는 다중 전송할 메시지의 목적지 노드들은 수평 해밀톤 경로를 사용하는 메시지들은 식 (1)에 의해서, 수직 해밀톤 경로를 사용하는 메시지들은 식(2)에 의해서 미리 정렬이 된다. 모든 다중 전송 메시지들은 전송 전에 수평/수직 해밀톤 경로의 선택을 미리 정하고 모든 메시지가 전송 중에는 경로를 바꾸지 않기 때문에 교착 상태가 발생할 수 없다. 그러므로, 알고리즘 MDP 은 교착 상태가 발생하지 않는다

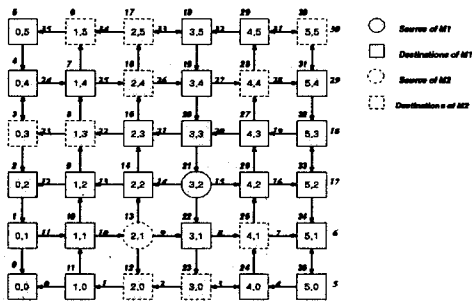
4. 알고리즘의 평가

다중전송 알고리즘의 성능은 시스템내의 임의의 노드에서 생성되는 메시지에 의한 상호간섭으로 인해 하나의 다중전송 메시지에 대한 지연이나 트래픽 증가만으로 평가하

알고리즘: D_{src} 는 f 개의 목적지로 메시지를 다중전송한다.
 입력: D_M : 목적지의 주소에 따라 정렬한 주소 리스트를 가진 메시지
 w : 현 노드의 주소
 출력: 메시지를 두 방향으로 전송한다.
 과정:

- 만약 $w = d_j$ 이면, $D'_M = D_M - d_j$ 하고, 메시지 복사본이 지역 노드로 전송이 된다.
 아니면, $D'_M = D_M$.
 (여기서, d_j 은 목적지 리스트 중 제일 첫 번째 주소를 의미.)
- 만약 $D'_M = 0$ 이면, 메시지 전송을 끝낸다.
- d 를 D'_M 의 첫 번째 주소로 한다.
 w' 는 다음에 전송될 새로운 노드의 번호로 세팅한다.
- 새로운 주소 리스트 D'_M 으로서 w' 노드로 메시지가 전송이 된다.

<그림 5> 메쉬 구조에서 다중전송 알고리즘 MDP



<그림 6> 6x6 메쉬에서의 다중전송 알고리즘의 예

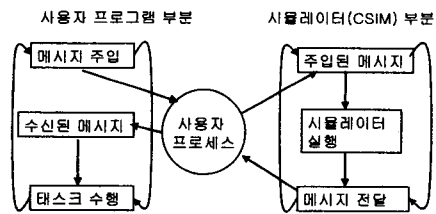
기 어렵다. 본 논문에서는 전송되는 메시지들 사이의 상호 간섭에 의한 전송지연을 측정하기 위해 16 x 16 메쉬 구조에서의 다중전송 알고리즘을 모의실험을 통하여 평가 및 분석한다. 평가 및 분석을 위한 모의실험은 C++ 언어와 사건구동(event-driven)을 이용한 모의실험 패키지(event-driven simulation package)인 CSIM[19]이다. CSIM 기반으로 작성된 프로그램의 구조는 그림 7과 같다.

본 논문에서 모의실험은 전송지연과 지연변위를 측정하여 두 다중전송 알고리즘의 성능을 비교 및 분석한다. 모의실험은 다음과 같은 가정을 따른다: 첫째, 네트워크의 임의의 노드에서 다중전송 메시지를 생성할 수 있으며, 메시지를 생성하는 확률은 매개변수인 부하(load)를 평균으로 하는 포아송분포를 따른다. 둘째, 메시지의 길이를 평균으로 하는 지수분포를 따른다.

용 분포를 살펴본다.

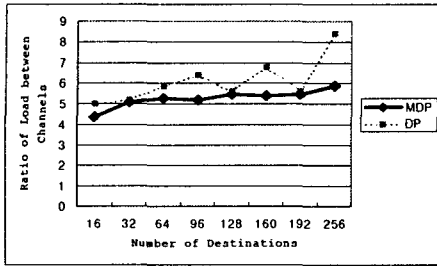
그림 8에서는 목적지의 갯수에 따른 평균 수평/수직 채널의 부하 비율을 나타낸 그래프이다.

X축은 한 메시지당 목적지의 수를 의미하고 Y축은 수평/수직 채널의 부하 비율을 의미하며 한 메시지의 길이는 128바이트로 가정하였다. DP는 기존의 알고리즘[14]을 적용한 경우이고 MDP는 본 논문에서 제안하는 알고리즘을 적용한 것이다. 목적지의 수가 적을 때는 두 알고리즘 모두 비교적 낮은 부하 비율을 나타내지만 목적지의 수가 많아짐에 따라 DP 알고리즘은 부하 비율이 높아짐을 볼 수 있다. 전반적으로 MDP 알고리즘이 15~20% 정도의 성능 향상을 보이고 있다. 수평/수직 채널의 부하 비율이 높다는 의미는 그 만큼 한 쪽 채널이 다른 채널보다 더 많이 사용된다는 것을 의미한다. MDP 알고리즘은 수평 채널뿐만 아니라 수직 채널을 골고루 사용하려는 시도를 하기 때문이다.



<그림 7> 시뮬레이터 프로그램의 구조도

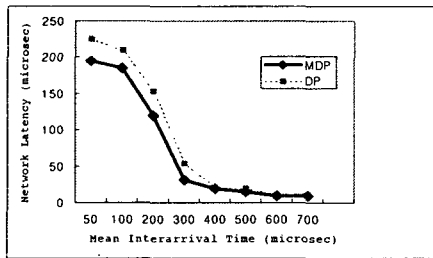
다음은 메시지당 평균 도착 시간에 따른 메시지의 평균 지연 시간 분포에 대한 모의실험의 결과는 그림 9와 그림



<그림 8> 메쉬망에서 여러 부하에서의 성능

10에 나타나 있다. 목적 노드의 개수는 각각 10개와 50개이며 메시지의 길이는 모두 128바이트, 메시지들은 포아송 분포를 따라 도착을 한다. 각 채널의 속도는 20 Mbytes/sec로 가정을 한다. 그림에서 보듯이 메시지의 평균 도착 시간이 늦은 경우, 즉, 부하가 낮은 경우에는 모든 경우에서 성능이 큰 차이는 나타나지 않는다. 그러나, 부하가 많아짐에 따라 DP 알고리즘은 수직 채널만 집중적으로 사용하기 때문에 성능이 본 논문에서 제시한 알고리즘보다 전반적으로 낮다는 것을 알 수가 있다. MDP 알고리즘이 최대 20% 정도의 성능을 향상시켰음을 알 수 있다.

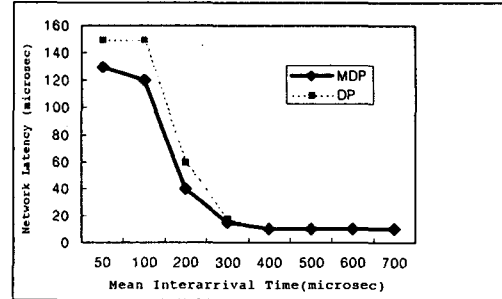
왜냐하면, 제안한 알고리즘은 수평 채널뿐만 아니라 수직 채널을 골고루 사용하기 때문이다. 또한, 목적 노드의 수가 증가하면 메시지가 망에 머무는 시간(즉, 메시지가 노드에서 처리되지 않고 채널을 점유하는 시간)이 증가하기 때문에 지연 시간이 증가하는 것도 알 수 있다.



<그림 10> 목적노드수(50개)에 따른 지연 성능

5. 결 론

본 논문에서는 워홀 스위칭 방식을 이용하는 메쉬 구조를 갖는 네트워크에서 경로를 기반으로 하는 효율적인 다중전송 알고리즘을 제안하였다. 이 알고리즘은 워홀 스위칭 방식의 단일전송 지연의 특성인 거리에 무관한 특성을 이용하였고 두 개의 가상채널에 별도의 해밀턴 경로를 사



<그림 9> 목적노드수(10개)에 따른 지연 성능

용함으로써 한 개의 해밀턴 경로만 사용하였던 종래의 방법보다 훨씬 채널 사용율을 높이고 이에 따른 성능도 향상시켰다. 이 알고리즘이 교착 상태가 없다는 것을 증명하였으며 시뮬레이션을 통하여 기존의 알고리즘보다 약 15~20% 정도 더 우수한 것을 검증하였다.

참 고 문 헌

- [1] W.C. Athas and C.L. Seitz, "Multicomputers: Message Passing Concurrent Computers", IEEE Computers, Vol. 21, No. 8, pp.9-24, Aug. 1988.
- [2] A. Bar-Noy, J. Bruck, C.-T. Ho, S. Kipnis, and B. Schieber, "Computing Global Combine Operations in the Multi-port Postal Model", Proc. of the fifth IEEE Symp. on Parallel and Distributed Processing, Dec. 1993, pp.336-343.
- [3] W. J. Dally and C. L. Seitz, "The Torus Routing Chips", Journal of Parallel and Distributed Computing, Vol. 1, No. 3, pp.167-196, Mar. 1986.
- [4] R.F. DeMara and D.I. Moldovan, "Performance Indices for Parallel Marker Propagation", Proc. of the 1991 Int'l Conf. on Parallel Processing, Aug. 1991, pp.658-659.
- [5] G. Fox, S. Hiranandani, K. Kennedy, C. Koelbel, U. Kremer, C.-W. Tseng, and M.-Y. Wu, "Fortran D Language Specification", Technical Report, TR90-141, Dept. of Com. Sci., Rice University, Dec. 1990.

- [6] A.J. Frank, L.D. Wittie, and A.J. Bernstein, "Multicast Communication on Network Computer", IEEE Software, Vol. 2, No. 3, pp.49-61, May 1985.
- [7] F. Harary, Graph Theory, Readings, Mass: Addison-Wesley, 1972.
- [8] C.-T. Ho and M.-Y. Kao, "Optimal Broadcast on Hypercubes with Wormhole and E-cube Routings," Proc. of the 1993 Int'l Conf. on Parallel and Distributed Systems, 1992, pp.694-697.
- [9] V. Kumar and V. Singh, "Scalability of Parallel Algorithms for the All-Pairs Shortest Path Problem," Technical Report, ACT-OODS-058-90, Rev. 1, MCC, Jan. 1991.
- [10] Y. Lan, A.H. Esfahanian, and L.M. Ni, "Multicast in Hypercube Multiprocessors", Journal on Parallel and Distributed Computing, Vol. 8, No. 1, pp.30-41, Aug. 1990.
- [11] K. Li and R. Schaefer, "A Hypercube Shared Virtual Memory", Proc. of the 1989 Int'l Conf. on Parallel Processing, Aug. 1989, Vol. I, pp.125-132.
- [12] X. Lin and L.M.Ni, "Multicast Communications in Multicomputer Networks", Proc. of Int'l Conf. on Parallel Processing, 1990, Vol. III, pp.114-118.
- [13] P. K. McKinley, H. Xu, and L. M. Ni, "Efficient Communication Services for Scalable Architectures", Technical Report, MSU-CPS-ACS-58, Dept. of Com. Sci., Michigan State University, East Lansing, MI, Apr. 1992.
- [14] P. K. McKinley, H. Xu, A. H. Esfahanian, and L. M. Ni, "Unicast-based Multicast Communication in Wormhole-Routed Direct Networks", Proc. of the 1992 Int'l Conf. on Parallel Processing, Aug. 1992, Vol. II, pp.10-19.
- [15] L. M. Ni and P .K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks", IEEE Computer, Vol. 26, No. 2, pp.62-76, Feb. 1993.
- [16] D. A. Reed and R. M. Fujimoto, Multicomputer Networks: Message Based Parallel Processing, MIT Press, Cambridge, MA, 1987.
- [17] D. F. Robinson, D. Judd, P. K. McKinley, and B. H. C. Cheng, "Efficient Collective Data Distribution in All-Port Wormhole-Routed Hypercubes", Proc. of Supercomputing '93, Nov. 1993, pp.792-801.
- [18] D. F. Robinson, P. K. McKinley, and B. H. C. Cheng, "Optimal Multicast Communication in Wormhole-Routed Torus Networks", Proc. of the 1994 Int'l Conf. on Parallel Processing, Aug. 1994, Vol. I, pp.134-141.
- [19] H. D. Schwetman, "CSIM:A C-based, process-oriented simulation language", Technical Report, PP-080-85, Microelectronics and Computer Corporation, 1985.
- [20] J.-P. Sheu and M.-Y. Su, "A Multicast Algorithm for Hypercube Multiprocessor", Proc. of Int'l Conf. on Parallel Processing, Aug. 1992, Vol. III, pp.18-22.
- [21] H. Xu, Y.-D. Gui, and L. M. Ni, "Optimal Software Multicast in Wormhole Routed Multistage Networks", Technical Report, MSU-CPS-ACS-91, Dept. of Com. Sci., Michigan State University, East Lansing, MI, Mar. 1994.
- [22] Yang-hua Chu, Sanjay G. Rao and Hui Zhang, "A Case For End System Multicast", Proceedings of ACM SIGMETRICS, Santa Clara, CA, June 2000, pp 1-12.
- [23] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture", Proceedings of ACM SIGCOMM, San Diego, CA, August 2001.



김 시 관 (Si-Gwan Kim)

1982년 2월 경북대학교 전자공학과
학사

1984년 2월 한국과학기술원 전산학과
석사

1984년~1994 년삼성전자, LG정보통신

2000년 2월 한국과학기술원 전자및전산학과 박사

현재 안동대학교 공과대학 재직

관심분야 : 멀티미디어 시스템, 컴퓨터 구조, 무선 통신