하이퍼큐브 네트워크에서 벡터들을 이용한 우회경로 알고리즘

정회원 김 명 하*, 이 충 세**

Detour paths algorithm using the vectors in Hypercube Networks

MIng-He Jin*, Chung-Sei Rhee** *Regular Members*

요 약

네트워크 분야와 멀티미디어 기술의 발전에 따라 실시간 처리에 대한 연구가 각광을 받고 있다. 또한, 많은 용용 분야에서 신뢰도가 높은 실시간 통신에 대한 필요성이 대두되고 있다. 이 논문에서는 큐브 구조를 갖는 다중 컴퓨터에 대한 실시간 통신 알고리즘을 벡터를 이용하여 구현하려고 한다. 큐브 형태의 n 차원 다중 컴퓨터에서 n 개의 안전 벡터는 이웃한 노드들의 결함의 분포와 근사적인 결함의척도와 관련되어 있다. 이 연구에서는 실시간 채널에서 서로 설정하는 다른 우회 경로를 찾는 알고리즘을 개발한다. 개발한 알고리즘과 기존의 방법들을 비교하여 성능을 분석한다. 시뮬레이션 결과는 이미 제안된 알고리즘과 비교하여 보다 효율적임을 보이고 있다.

Key Words : Hypercube, Safety Vector, Fault Tolerant, Routing

ABSTRACT

The advances in networking technology and importance of multimedia communication require real time transaction. In many applications, high reliable real-time communications are required more frequently. In this paper, we propose a reliable communication in cube-based multi -computer using the safety vector. Each node in a cube-based n dimensional multi-computer is associated with a safety vector of n bits, which is an approximated measure of the number and distribution of faults in the neighborhood. We propose an algorithm that can establish detour paths using the safety vector. The established detour paths are disjoint with the primary real-time channel. Therefore, our algorithm is more efficient than earlier proposed algorithms.

Ⅰ. 서 론

실시간 통신에서는, 통신을 위한 정보의 생성과 전달 및 수신에 이르는 일련의 처리는 시간지연이 없이 이루어진다. 실시간 통신은 인터넷을 통한 멀티미디어 데이터통신의 필요성이 부각되고, 네트워킹 기술이 발전되면서 더욱 주목받고 있는 분야이다. 또한 실시간 통신의 필요성이 높아지고, 요구분야가 커질수록 시간제약을 보장함과 동시에 효율적인 비용으로 결함을 극복할 수 있는 신뢰도 높은 통신 서비스를 필요로 한다[Han98].

이 논문에서는 실시간 통신의 신뢰도를 높이기 위하여, 실시간 통신경로 상에 결함을 우회하는 함으로써 시간제약과 통신 신뢰도를 모두 만족시킬 수 있는 보다 효율적인 방법을 제안한다. 또한 네트워크의 토폴로지가 하이퍼큐브 형태를 가질 때, 실시간 채널에 결함이 발생하였을 경우에 대처하기 위한 효율적인 우회경로 설정 알고리즘을 제안한다.

* 충북대학교 전자계산학과, ** 충북대학교 전자정보대학(badistuta520@hanmail.net)
논문번호 : KICS2009-05-206, 접수일자 : 2009년 5월 18일, 최종논문접수일자 : 2009년 11월 19일

일반적인 실시간 채널의 특성과 실시간 채널 설정 방법을 검토한 다음, 결함을 허용할 수 있도록 실시간 채널을 설정 시 주 채널과 보조채널로 설정하는 기법을 소개한다. 또한 안전벡터(safety vector)의 개념과 성질을 이용하여 하이퍼큐브 네트워크에서 효율적인 우회경로 설정 알고리즘을 제안한다.

Ⅱ. 실시간 채널을 이용한 실시간 통신

2.1 실시간 채널 개념

실시간 채널[Ferr90]이란 사용자가 지정한 지연시간(delay bound) 내에 패킷이 송신노드(source node)로부터 수신노드(destination node)까지 전송되는 것을 보장하는 가상채널을 말한다.

실시간 통신에서의 결함허용은 다양한 분야에서 필요하지만, 실시간이라는 시간제약과 신뢰도라는 두 가지 조건을 모두 만족시키기 위해서는 특별한 방법이 필요하다. 실시간 채널을 이용한 실시간 통신에서는 수신노드와 송신노드가 같은 패킷이라면 모두 같은 경로를 통하여 전송되기 때문에, 그 경로상에 있는 하나의 컴포넌트라도 결함이 발생되면 전체적인 채널이 사용할 수 없게 된다[Zhen92].

안전벡터를 이용하여 네트워크의 정보를 각 노드가 갖게 함으로써, 결함이 있는 노드나 링크 쪽으로는 패킷이 전달되지 못하게 할 수 있다. n차원 하이퍼큐브 네트워크는 n개의 비트로 표현할 수 있는 2n개의 노드로 구성되어 있다. 이러한 성질을 이용하여 적응적 결함허용 라우팅을 효과적으로 수행할 수 있다[Bane89]. 이러한 방법으로 실시간통신의 결함극복에 이용한 연구는 아직 이루어지지 않았고, 이 논문에서는 이러한 방법을 이용한 알고리즘을 제안한다.

앞에서 살펴본 바와 같이 같은 내용의 패킷에 통신에 필요한 정보들을 추가하여 서로 다른 여러 개의 경로를 통해 동시에 전송한다. 수신노드는 각각에 전달된 패킷을 보팅(voting)하여 오류가 발생된 패킷을 분별할 수 있다. 이 방법은 보다 빠른 통신 서비스를 제공하기 위하여 제안된 방법이다. 그러나 이 방법은 수신노드에서 또 다시 패킷들을 재조합하여야 하므로 많은 오버헤드가 필요하다.

Ⅲ. 결함허용 라우팅

결함허용 라우팅(fault tolerant routing)은 통신할 경우에 통신경로상(링크나 노드)에서 발생할 수 있는 결함이 통신의 내용이나 질에 문제를 일으키지 않도록 처리해 주는 라우팅 기법을 말한다.

결함 허용 라우팅 방법은 여러 가지가 있는데 크게 분산(dispersity)과 중복(redundancy)방법으로 나누어 볼 수 있다. 결함허용 라우팅의 절차는 다음과 같은 과정을 아용하여 수행한다.

1) 발견(detection): 고장이 발생한 요소를 감지한다.
2) 경로설정: 결함허용의 가장 중요한 단계로써 결함에 영향을 받는 채널을 새로운 경로로 지정한다. 이때 중앙집중식 방법과 지역적인 방법이 있다.
3) 채널관리(channel management) : 새롭게 설정된 경로로 네트워크상의 자원들을 옮긴다.
4) 정상화 : 무결함의 상태로 정상적인 라우팅을 진행한다.

이 논문에서는 앞의 단계에서 결함이 발생된 후에 이를 발견한 후에 경로를 배정하여 결함을 극복할 수 있는 방법을 제시하는 두 번째 단계와 관련된 연구를 수행하였다.

Ⅳ. 하이퍼큐브 네트워크에서의 우회경로 설정방법

4.1 하이퍼큐브 네트워크

하이퍼큐브는 0번부터 2n-1까지의 수로 번호가 부여된 2n개의 노드가 자신의 노드번호와 서로 1비트만 다른 노드끼리 서로 연결되어있는 그래프이다. 노드u(i)는 노드 u의 i번째 차원에 속하는 이웃노드를 나타낸다. u(i)는 노드 u의 i번째 비트를 정의하거나 재정의함으로써 계산할 수 있다. 예를 들면, 1101(3) = 1001이다. 다시 말하면, 노드 1101의 3번째 차원에 속하는 이웃노드의 번호는 1001임을 알 수 있다[그림. 1 참조]. 이러한 성질은 큐브 내

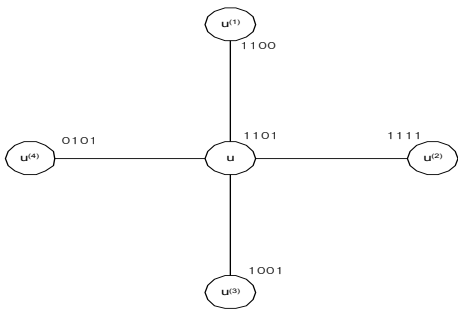

그림 1. 4-큐브에서 i차원 이웃노드 u(i)

에 결함(링크나 노드의 결함)이 있을 때 결함의 위치나 수에 따라 통신이 단절되거나 그렇지 않을 수 있다는 것을 의미한다. 송신노드와 수신노드 간의 경로의 길이가 해밍거리(hamming distance)와 같으면, 최소경로 또는 최적의 경로라 한다. 그러나 이러한 최소경로가 결함에 의해 모두 차단될 수 있다. 이러한 경우에 최단경로는 최소경로가 되지 못한다. 이 연구에서는 경로가 결함에 의해 차단되는 경우가 발생되지 않도록 하는 알고리즘을 제안한다.

두 노드 s와 d사이의 해밍거리는 두 노드의 번호를 가지고 exclusive-or 연산을 취한 결과의 1의 개수가 된다. 또한, 이 결과 비트의 값이 1인 위치의 차원이 우선차원(preferred dimension)이 되고 비트의 값이 0인 차원은 보조차원(spare dimension)이 된다. 따라서 두 노드간의 경로를 설정할 때 우선차원에 있는 이웃노드만 이용하여 설정하는 경로가 최단경로이다. 그리고 각 노드의 번호는 n개의 비트를 이용하여 이진수로 표현하며, 노드의 상태를 표현하기 위하여 안전벡터를 이용한다.

4.2 안전벡터의 개념과 성질

n개의 비트로 표현하는 숫자 내에 노드나 링크의 결함에 대한 정보를 포함하고 있는 벡터를 안전벡터라고 정의한다. $(u_1,u_2,\ldots,u_n)$는 n큐브 내에 있는 각 노드 u와 관련된 결함에 대한 정보를 가지고 있다. 특히, u_k는 노드 u에서 k-해밍거리 노드까지 라우팅이 가능한지를 나타내는 것이다. 이진 하이퍼큐브에 기반한 네트워크에서는, 네트워크의 정보를 가지고 첫 번째 비트만 결정할 수 있다면, 이웃노드 안전벡터의 k-1번째 비트를 이용하여 현재 노드의 k번째 안전벡터를 계산할 수 있다.

정의 4.1 : $(u_1^{(i)},u_2^{(i)},\ldots,u_n^{(i)})$를 i차원에 속한 노드 u의 이웃노드의 안전벡터라고 하면, 다음과 같이 안전벡터를 정의한다. 결함노드의 안전벡터는 (0, 0, ...,0)이다. 만약 노드 u가 결함링크의 단 노드이면, 나머지 단 노드는 노드 u에 (0,0, ...,0)를 안전벡터로 정의한다. 첫 번째 비트를 이용하여 다음과 같이 정의한다.

$$u_k = \begin{cases} 0 & \text{if } \textit{node u is an end node of fault link} \\ 1 & \textit{other wise} \end{cases}$$

위의 정의를 이용하면 안전벡터 성질을 알 수 있다. 어떤 노드의 안전벡터의 k번째 비트가 “1”이면, k개의 이웃노드 가운데에 적어도 하나는 안전벡터의 k-1번째 비트가 “1”인 노드가 존재한다. 각 노드의 안전벡터를 계산하는 방법은 다음에 제시된 알고리즘4.1에 의한다. 알고리즘4.1에 의해 네트워크를 구성하는 모든 노드의 안전벡터를 구하려면 n-1번만 수행하면 된다. 먼저, 전제조건은 결함노드의 안전벡터는 (0, 0, ..., 0)으로, 결함이 없는 노드의 안전벡터는 초기 값으로는 (1, 1, ..., 1)으로, 그리고 알고리즘 4.1에 의하여 라우팅이 가능한 차원의 비트는 1로 그렇지 않은 비트는 0으로 한다.

다음 표는 그림 2의 4-큐브 네트워크를 구성하는 노드의 안전벡터를 알고리즘 4.1을 이용하여 계산하는 과정을 보인 표이다. 단, 노드 0011, 0100, 0110, 1001이 결함노드인 경우를 계산한 것이다. 4큐브 네트워크이므로 알고리즘4.1은 3번만 수행되면 위 표와 같이 각 노드의 안전벡터를 계산할 수 있다. 계산결과는 그림 3에 있다.

알고리즘 1. 안전벡터계산

```
begin
 forall u(안전벡터)
   if u is an end node of a
    faulty link
   then u1 = 0 else u1 = 1;
// 자신이 고장난 노드이면
// 첫 번째 bit를 0으로
// 그렇지 않으면 1로 set한다.
 for k = 2 step 1 to n
  forall u
 collects all the (k-1)th bits of u's
neigher's safety vector
// 그림 3참조
if  ∑ 1≤i≤n u (i) k-1 ≤ n − k then uk =
0 else uk=1
end
```

4.3 안전벡터를 이용한 결함허용 라우팅 알고리즘

각 노드의 안전벡터를 이용하여 알고리즘을 구현하였다. 라우팅 알고리즘의 패킷전달 방식은 유니캐스트(일대일통신) 방법을 기본으로 한다. 멀티캐스팅 엠본을 이용하지 않고 인터넷의 전형적인 통신형태인 일대일 통신(유니캐스트)방법을 바탕으로 한 많은 인터넷 실시간 멀티미디어 서비스가 소개되고 있다. 이 논문에서의 패킷전달방법으로 유니캐스팅

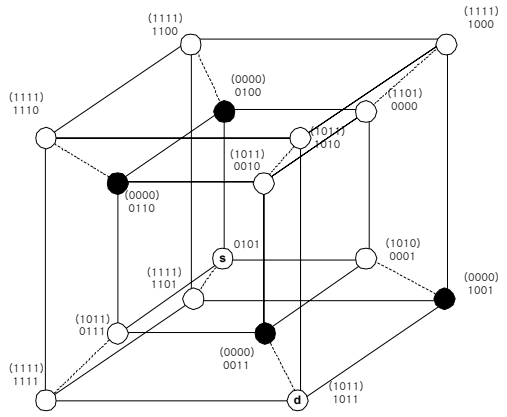

그림 3. 결함이 있는 4-큐브네트워크에서 안전벡터계산

을 사용한다.

많은 결함허용 유니캐스팅 방법들이 제안되었다([Chen90],[Chen90], [Gord88], [Lan94],[Lee92]). 이들 대부분의 방법들은 각 노드가 자신의 이웃노드에 대한 정보만을 알고 있거나(지역정보기반) 모든 노드의 상태를 알고 있는 것(전체정보기반)으로 가정하였다.

제안한 알고리즘에서는 송신노드에서 k-해밍거리만큼 떨어진 수신노드로 메시지를 전달하고자 한다면, 노드의 안전벡터의 k-번째 비트가 1이거나 또는 i차원에 속한 이웃노드의 k-1번째 비트가 1이면 최적임을 보일 수 있다. 즉, $s_{k-1}^{(i)} = 1,\ i \in \{1,2,..., n\}$을 만족하면 알고리즘이 최적이 된다. 따라서 처음 라우팅이 시작될 때는 안전벡터의 k-1번째 비트가 1인 우선이웃으로 메시지를 전달하면 안전벡터 r의 k-2번째 비트가 1인 우선이웃노드로 메시지를 전달한다. 이러한 방법을 이용하여 라우팅하는 것이 최적의 해를 얻는데 필요하다. 만약 k-1번째에서 1인 비트를 발견하지 못하면, 안전벡터의 k+1번째 비트 참조이웃노드 중 하나를 선택하여 라우팅한 다음 최적화 라우팅 알고리즘을 적용한다. 이 경우에 송신노드부터 수신노드까지의 거리는 해밍거리+2가 되며, 이러한 결과를 부분최적화이라고 정의한다.

송신노드에서 최적화인지 부분최적화인지를 결정하기 위해서는 다음의 정보를 이용한다.

1) 송신노드(s)부터 수신노드(d)까지 해밍거리 : H(s, d) = | s⊕d |
   s⊕d 값을 이용하여 우선이웃노드 집합과 참조이웃노드 집합을 구할 수 있다.
2) 송신노드 s의 안전벡터는 $(s_1, s_2, \cdots, s_n)$ 이다.
3) i 차원에 속한 이웃노드의 안전벡터는 $(s_0^{(i)}, s_1^{(i)}, s_2^{(i)}, \cdots, s_n^{(i)})$ 이다.

$s_0^{(i)}$ 는 라우팅 알고리즘을 단순히 하기 위한 여분의 비트이다. s와 s^i를 제외하고 $s_0^{(i)}$ =1이면, 결함이 있는 링크의 단 노드를 표현한다. 또한 진행벡터(navigation vector), N = s⊕d는 송신노드와 수신노드간의 관련벡터를 표현한다. 진행벡터는 송신노드에서 출발한 패킷이 어떤 이웃노드를 거쳐서 전달되는가를 결정하는 벡터이며, 패킷을 받았을 때 각 패킷의 진행노드를 바탕으로 중간노드들은 먼저 우선이웃노드와 참조이웃노드를 계산한다. 중간노드가 수신노드로부터 k+1 해밍거리만큼 떨어져 있으면, 안전벡터의 k번째 비트 값이 1인 참조이웃노드를 다음 경로로 선택한다. 그리고 패킷의 진행노드가 널(null)값이면 그 패킷의 전달이 완료된 것으로 수신노드에 도착한 것을 의미한다.

라우팅 과정은 송신노드에 적용하는 UNICASTING_AT_SOURCE_NODE와 중간노드에 적용하는 UNICASTING_AT_INTERMEDIATE_NODE로 구성된다. 첫 번째 라우팅 단계 후에 최적라우팅 알고리즘이든 부분 최적화 라우팅 알고리즘이든 패킷을 수신노드까지 전달하기 위하여 해밍거리인 경로를 선택하기 때문에 중간노드에서는 UNICASTING 과

알고리즘 4.2 송신노드에서의 라우팅 알고리즘

```
begin
// N은 송신노드와 수신 노드간의 관련 벡터
N = s⊕d ;
H = | s⊕d | ;
if S_H = 1 or ∃i( s_{H-1}^{(i)} = 1 and N^(i) =1)
   then 최적화_UNICASTING;
   send message m to s_{H-1}^{(i)} = 1 and N(i) = 1
        // 우선이웃노드
else if ∃i( s_{H-1}^{(i)} = 1 and N^(i) =0)
        then 부분최적화_UNICASTING;
      send message m to s^(i), where s_{H-1}^{(i)} = 1
        // 참조이웃노드
else failure
end
```

알고리즘 4.3 중간노드에서의 라우팅 알고리즘

```
begin
  if N=0 then stop;
  else send message m to U(i),  where
U(i) H-1=1 and N(i)=1
  // N은 s와 d의 관련벡터를 나타냄
end
```

정에서 부분최적화인지 최적화인지의 라우팅 유형을 분류할 필요는 없다. 다음의 정리3은 송신노드부터 수신노드간의 해밍거리, 이웃노드의 안전벡터 그리고 라우팅 경로의 길이간의 관계에 대한 것이다.

정리 4.1 송·수신노드간의 해밍거리를 k라고 가정할 때, 송신노드의 안전벡터의 k 번째 비트가 1이거나 송신노드의 우선이웃노드의 안전벡터의 k-1번째 비트가 1이면 주어진 라우팅 알고리즘에 의해 최적의 우회경로를 선택할 수 있다. 송신노드의 보조이웃노드의 안전벡터의 k+1번째 비트가 1이면 부분최적화를 보장한다.

4.4 우회경로 설정알고리즘

하이퍼큐브 구조로 네트워크를 설계하고, 최단경로 설정이 불가능할 경우엔 비 최단경로로(hamming distance 이상의 길이를 갖는) 라우팅하도록 하여 n차원 하이퍼큐브에서 n-1개의 노드 혹은 링크의 결함을 허용하는 알고리즘이 제안되었다[Han98]. 그러나 이 방법은 실시간을 고려하지 않았다. 하이퍼큐브 네트워크에서 실시간 통신을 할 때 n-1개의 결함을 허용한다는 것은 거의 불가능하다. 메쉬의 경우에서 우회할 수 있는 경로를 선택하는 경우의 수는 최대 3이다. 하지만 실시간 결함허용우회경로설정에서는 1개의 결함을 극복하는 것이 일반적인 경우이다. 하이퍼큐브 네트워크에서는 n-큐브일 경우 n-1이다. 그러나 패킷에 시간제약이 있으므로 허용가능한 결함의 수는 n-2개로 가정한다.

주 채널에 포함된 경로를 모두 배제한 보조채널을 설정한다. 왜냐하면, 주 채널과 공유하는 링크가 있을 경우에 그 공유링크에 결함이 발생된다면 보조 채널설정 의미가 없어지게 되며, 불필요한 자원 예약으로 네트워크의 효율만 떨어뜨리게 된다. 따라서 알고리즘 4.4를 이용하여 우회경로를 설정한다. 우회경로 알고리즘을 살펴보기에 앞서 다음과 같은 가정을 한다.

1) 보조 채널을 설정할 때 주 채널과 완전히 다른 경로를 사용하는 채널만 설정한다. 이때 비-최단경로도 허용한다.
2) 허용할 수 있는 결함의 수는 n-2개이다.

서브네트워크를 표현한 것으로 주 채널 Cp가 지나는 경로와 공통인 링크를 갖는 채널들은 설정하지 않고, 완전히 다른 경로를 갖는 채널만 설정한다. 송신노드가 0101, 수신노드가 1011이며, 0011노드가 결함이 있다고 가정하자. 송신노드와 수신노드간의 해밍거리가 3이므로 각 노드의 안전벡터를 계산한 것을 바탕으로 하여, 주 채널을 다음과 같이 설정한다. Cp = (0101, 0001, 1001, 1011) 그리고 보조채널은 다음과 같다. Cb1 = (0101, 1101, 1001, 1011), Cb2 = (0101, 1101, 1111, 1011), Cb3 = (0101, 0111, 1111, 1011)이 된다.

단, Cp는 주 채널의 경로, Cb는 보조 채널의 경로를 의미하며, 채널의 경로는 거치는 노드 번호를 이용하여 나타낸다. 즉, C=(Ns, N1, N2,...,Nd)로 표현한다. 여기서 Ns는 송신노드, Nd는 수신노드, Nn은 채널상의 노드번호를 의미한다. 알고리즘 4.4는 네트워크 상에 결함이 발생되었을 경우에 대비하여 주 채널의 결함을 우회하여 보조 채널을 설정하는 절차이다. 경로 상에 존재하는 노드들의 안전벡터를 참조함으로써 네트워크의 정보를 얻게 되고, 이 정보를 이용하여 보조 채널을 설정하게 된다. 기존의 연구에서는 사용되지 못하는 채널의 수만 늘려놓았지만, 이 논문에서 제안한 방법은 이러한 문제를 미리 방지할 수 한다. 또한 이렇게 설정되어 있는 보조 채널은 어떠한 경우라도 주 채널과 중복되는 링크가 포함되지 않으므로 선택된 채널에 대한 신뢰도가 향상된다. 알고리즘 4.4의 수행절차를 간단히 설명하면, 먼저 송신노드의 이웃노드(랜덤하게 선택한) 중에서 안전벡터 값을 이용하여 송신노드에서 패킷이 전달가능한 우선이웃노드를 결정한다. 이웃노드가 송신노드로부터 수신노드에 이르는 주채널에 포함된 노드인지 아닌지를 체크한 후 주 채널에 포함된 노드가 아니면 보조 채널에 포함시키고, 주 채널에 포함된 노드이면 다른 우선이웃노드를 선택하여 주 채널에 포함여부를 확인한다. 주 채널에 포함되지 않았으면 보조채널에 넣는다. 지금까지 제안된 알고리즘들은 대부분 네트워크의 정보를 수집하기 위하여 네트워크를 구성하는 모든 노드의 상태를 파악하는 프로세서가 필요하였으나 알고리즘 4.4는 안전벡터를 이용하여 네트워크의 정보

를 파악함으로써 오버헤드를 많이 줄일 수 있는 방법[Wu98]이다.

지금까지 살펴본 내용들은 모두 신뢰도 높은 실시간 통신을 위한 방법과 이론에 대한 것이다. 이 내용들을 요약하면 알고리즘 4.5와 같은 결과를 얻게 된다. 단계 4의 내용은 더 이상 사용되지 않을 채널을 제거하여 자원의 예약을 해제함으로써 자원의 가용성을 높인다. 자원을 효율성을 높이기 위한 방법에는 [Han98]에서 제안된 방법인 채널 멀티플렉싱 기법이 있다.

알고리즘 4. 우회경로설정(s, d)

```
{
   t = s; // t는 source node
   sv = 안전벡터 of s; // sv는 source node의 안전벡터
   while( t=d){
    for( i=0; i<n; i++){
        j = random(); // 다음 노드를 랜덤 하게 선택한다.
      // 단, j번째 비트 값이 1인 것을 선택한다.
            if( sv_j≠0){
             t_j = complement(t_j);
             t = (t_0, t_1, ..., t_j, ..., t_n)
            if (t ∈ Cp){ t_j = complement(t_j);
             t = (t_0, t_1, ⋯, t_j, ⋯, t_n) }
      // 선택된 노드가 주채널에 포함되어있으면 다시.
      // 원위치로 돌아감
            else  t add to Cb;
      // 그렇지 않으면 보조채널에 포함시킴
                  } }
        } return Cb;
}
```

알고리즘 5. 결함허용실시간통신을 위한 채널설정 알고리즘

step 1. 알고리즘 4.2를 이용하여 n-큐브 네트워크를 구성하는 각 노드의 안전벡터를 계산한다.
step 2.채널 설정 요구가 있으면, 알고리즘2.1을 이용하여 실시간 채널로 주채널과 보조채널로 설정한다. 보조채널로 설정하는 것은 알고리즘 4.4를 따른다.
step 3. 결함이 발견되면, 안전벡터를 수정하고 알고리즘 4.4를 이용하여 다시 보조채널을 설정한다.
step 4. 주기적으로 사용되지 않을 채널을 제거한다.

V. 결 론

네트워크 기술이 고도화됨에 따라 실시간 통신의 응용분야도 다양해짐은 물론 통신에 대한 높은 신뢰도를 필요로 한다. 실시간이라는 시간제약을 만족함과 동시에 신뢰도 높은 통신을 수행해야 한다. 실시간 통신과 결함허용을 결합하기 위한 효과적인 방법들이 많이 제안되었다. 이 연구에서는 제안된 여러 가지 방법들 중에서 실시간 채널을 이용한 방법을 개선하여 주 채널과 보조 채널로 통신경로 설정을 한 후 안전벡터를 이용하여 우회경로를 설정하는 방법을 연구하였다. 제안한 알고리즘은 기존의 문제점을 해결한 여러 가지 알고리즘을 서로 연결하여 실시간 통신의 시간적 제약을 만족하면서, 좀 더 효율적으로 통신의 신뢰도 향상시킬 수 있다.

제안된 알고리즘은 하이퍼큐브 네트워크의 전체적인 정보를 제한적으로 이용하여 안전벡터를 구하고, 안전벡터 계산하는 방법과 또한 이웃노드와 메시지교환을 단 n-1번 수행하기 때문에 시간적인 면에서 효율적이다. 송신노드는 각 노드에 계산되어 있는 안전벡터 값과 송신노드와 수신노드간의 해밍거리(최단거리)를 참조하여 최적의 라우팅 알고리즘을 이용할 것인지, 부분최적화 라우팅 알고리즘을 이용할 것인지를 결정할 수 있다[Wu98].

이 방법은 휴리스틱과 GREEDY알고리즘을 모두 이용하였으며 네트워크를 구성하는 노드에 결함이 많이 발생한 경우에도 매우 효율적이며 유용한 방법이다. 그러나 각 노드마다 안전벡터를 가지고 있어야 하기 때문에 전체적인 프로세서가 별도로 존재하는 알고리즘에 비하여 메모리의 효율면에서는 떨어진다. 하지만 전체적인 프로세서가 각 노드의 정보를 파악하여 네크워크의 정보를 습득하는 방법보다 소요되는 시간복잡도면에서는 효율적이다.

참 고 문 헌

〔1〕 A. Banerjea, "Fault Management for Realtime Networks", 1989.

〔2〕 L. N. Bhuyan and D.P. Agrawal, "Gerneralized Hypercube and Hyberbus Structures for a Computer Network," IEEE Trans. Computers, Vol.32, No.4, pp.323-333, Apr. 1984.

〔3〕 M. S. Chen and K.G.Shin, "Adaptive Fault-tolerant Routing in Hypercube Multicomputers," IEEE. Trans. Computers,

Vol.39, No.12, pp.1406-1416, Dec. 1990.
〔4〕 M. S. Chen and K. G. Shin, “Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers,” IEEE Trans. Parrallel and Distributed Systems, Vol.1, No.2, pp.152-159, Apr. 1990.
〔5〕 B. Chen, S. Kamat, and W. Zaho, “Fault-Tolerant Real-Time Communication in FDDI-Based Networks,” Proc. Real-Time Systems Symp., 1995
〔6〕 R. L. Cruz,“A Calculus for Network Delay, Part I: Netwrok Elements in isolation,” IEEE Trans. on Information Theory, Vol.37, No.1, pp.132-141, Jan. 1991.
〔7〕 D. Ferrari and D. C. Verma, “A Schem for Real-Time Channel Establishment in Wide-Area Networks,” IEEE J. Selected Areas Comm., pp.368-379, 1990.
〔8〕 S. J. Golestan, “A Framing Strategy for Congestion Management,” IEEE Jornal on Selected Areas In Communication
〔9〕 J. M. Gordon and Q. F. Stout, “Hypercube Message Routing in the Presence of Faults,” Proc. Third Conf. Hypercube Concurrent Computers and Applications, pp.251-263, Jan. 1988.
〔10〕 Seungjea Han, K. G. Shin, “A Primary-Backup Channel Approach to Dependable Real-Time Communication in Multi-hop Networks,” IEEE Trans. on Computers Vol.47, No.1, pp.46-61, 1998.
〔11〕 Y. Lan. “A Fault-Tolerant Routing Algortithm in Hypercubes,” Proc. 1994 Int’l Conf. Parallel Processing, pp.Ⅱ 163-Ⅱ 166, Aug. 1994.
〔12〕 T. C. Lee and J. P. Hayes, “A Fault-Tolerant Communication Scheme for Hypercube Computers,” IEEE Trans. Computers, Vol.41, No.10, pp.1242-1256, Oct. 1992.
[13] J. Wu, “Adaptive Fault-Tolerant routing in Cube-Based Multicomputers Using Safety vectors”, IEEE Trans. on Parallel and Distributed Systems, Vol. 9, No.4, pp.321-334, Apr. 1998.
〔14〕 Q. Zheng and K. G. Shin, “On the Ability of Establishing Real-Time Channels in Point-To-Point Connected Packet Switching Networks,” IEEE Trans. Comm., Vol.42, nos.2/3/4, pp.1096-1105, Feb./Apr. 1992.
〔15〕 Q. Zheng and K. G. Shin, “Fault-Tolerant Real-Time Communication in Distributed Computing Systems,” IEEE Trans. Parrallel and Distributed sys. Vol.9. No.5. pp.470-480, May 1998.
〔16〕 심재철, 김동승, “하이퍼큐브 컴 퓨터에서의 결 함허용 경 로배정기법,” 한국정보과학회 논 문지, 제23권, 5호, pp.509-518, 1996년 5월.
〔17〕 J. Chen, I. A. Kanj, G Wang, “Hypercube Network Fault Tolerance: A Probabilistic Approach,” Parallel Processing, 2002. Proceedings. International Conference on, pp.65-72 Aug. 2002.
〔18〕 Ruei-Yu Wu, J. G. Chang, Gen-Huey Chen, “Node-disjoint paths in hierarchical hypercube networks,” Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International. pp.5, 2006.
〔19〕 D. Xiang, “Fault-tolerant routing hypercube multicomputers using local safety information,” IEEE Trans. Parallel Distrib. Syst., Vol.12, No.9, pp.942 – -951, Dec. 2001.
〔20〕 J. Wu, F. Gao, Z. Li, Y. Min, “Optimal, and reliable communication in hypercubes using extended safety vectors” Reliability, IEEE Transactions on Vol.54, Issue 3, pp.402- 411, 2005.

김 명 하 (MIng-He Jim) 정회원

2005년 7월 중국 ChungChun Normal University 학사
2008년 8월 한국충북대학교 전자계산학과 석사
2008년 9월~현재 한국충북대교 전지계산학과 박사과정
<관심분야> 알고리즘, 정보보안, 네트워크

이 충 세 (Chung Sei Rhee) 정회원

1989년 Univ. of South Carolina 컴퓨터과학과 (박사)

University of North Dakota 전산학과 조교수

1991년~현재 충북대학교 전자정보대학교 교수

<관심분야> 결함허용, 알고리즘 및 전문가 시스템, 정보보안