

동적 라우팅기반의 점대다중점 라우팅 경로 선택

신현순*, 이상호**, 이경호*, 박권철*

A Point-to-Multipoint Routing Path Selection Algorithm for Dynamic Routing Based ATM Network

Hyun-Soon Shin*, Sang-Ho Lee*, Kyeong-Ho Lee*, Kwon-Chul Lee*

요 약

본 논문은 ATM 시스템에서 동적 소스 기반의 점대다중점 호출을 위한 라우팅 경로 선택 방법을 제안한다. 특히 최적의 경로 선택 보다는 리소스를 최대한 공유할 수 있고, 경로 계산 시간 단축 및 cycle 방지를 쉽게 보장할 수 있는 점대다중점 라우팅 경로 선택 방법을 제안하였다.

리소스를 최대한 공유할 수 있도록 하기 위해 착신지(destination) 노드로부터 가장 근거리의 branch point를 찾는 것이 이 알고리즘의 목표이다. 그러므로 목적(destination) 노드로부터 역 추적에 의해 인접한 노드중 같은 Call ID를 갖는 기존 경로상의 노드와 최초로 크로스(cross)되는 노드를 branch노드로 정하여 최적의 점대다중점 라우팅 경로를 구성하는 알고리즘이다. 크로스 노드를 찾을 때까지 역 추적에 의해 선택되는 최적의 노드는 다익스트라 알고리즘을 이용하여 선택된다. 즉 점대다중점 라우팅 경로 선택은 역추적에 의한 인접 노드중 기존 경로와 크로스되는 노드 선택 단계와 각 인접 노드중 최적의 노드를 선택하는 단계를 수행한다.

이 기법은 경로 설정을 위한 라우팅 정보 테이블의 검색 및 경로 계산 과정을 단축시키고, 경로 설정 시 cycle방지도 쉽게 해결한다.

ABSTRACT

This paper proposes the routing path selection mechanism for source routing-based PtMP (Point-to-Multipoint) call in ATM switching system.

Especially, it suggests PtMP routing path selection method that can share the maximum resource prior to the optimal path selection, guarantee the reduction of path calculation time and cycle prevention.

The searching for the nearest branch point from destination node to make the maximum share of resource is the purpose of this algorithm. Therefore among neighbor nodes from destination node by back-tracking, this algorithm fixes the node crossing first the node on existing path having the same Call ID as branch node, constructs the optimal PtMP routing path. The optimal node to be selected by back-tracking is selected by the use of Dijkstra algorithm. That is to say, PtMP routing path selection performs the step of cross node selection among neighboring nodes by back-tracking and the step of optimal node selection(optimal path calculation) among neighboring nodes by back-tracking. This technique reduces the process of search of routing information table for path selection and path calculation, also solves the cycle prevention easily during path establishment.

Keywords : PNNI, Point-to-Point, Point-to-Multipoint, Path Selection

* 한국전자통신연구원 네트워크연구소(hsshin@etri.re.kr), 한국전자통신연구원 네트워크연구소(kholee@etri.re.kr), 한국전자통신연구원 네트워크연구소(kcpark @etri.re.kr),

** 충북대학교 컴퓨터과학과 (shlee@cmlab.chungbuk.ac.kr)
논문번호: 030223-0526, 접수일자: 2003년 5월 26일

1. 개요

초고속 국가망은 ATM을 기반으로 구축되고 있다. 특히 초고속 국가망을 구성하는 ATM 시스템들은 동적 소스 라우팅 기반의 PNNI(Private Network to Network Interface)를 적용하고 있다. 소스 라우팅 방식은 소스(혹은 발신측) 노드에서 착신지 노드까지 연결되는 계층적 경로(hierarchical path)를 End-to-End로 소스 노드에서 결정하는 방식이다[7]. 그 경로상의 중간 경유노드에서는 소스 노드가 선택한 경로설정 명령을 따라야만 한다.

PNNI 망에서 점대다중점 호 서비스 역시 소스 라우팅 방식을 따른다. 그림 1에서 보이는 바와 같이 통상적으로 점대다중점 호 서비스에서 호가 발생한 노드를 Calling Party라 하고, 기존의 호가 이루어진 착신지(destination)를 Called Party라 하고, 새로이 추가되는 착신지를 Add Party라 한다.

점대다중점 호 서비스를 위한 Add Party 경로는 최대의 리소스 공유를 보장하는 최적의 경로 선택이 필요하다.

우리는 라우팅 관점에서 점대다중점 기능 구현을 고려해 보았다.

특히, 본 논문에서는 분산제어 ATM시스템에서 리소스를 최대한 공유하고 시스템의 부하도 최소화할 수 있는 점대다중점 라우팅 알고리즘을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 점대다중점 서비스를 위한 소프트웨어를 기술하고, 3장에서는 점대다중점 라우팅 구현을 위한 고려사항을 살펴보고 방안을 제시한다. 4장에서는 점대다중점 라우팅 경로 선택 알고리즘을 제시하고, 5장에서 결론을 맺고 향후 연구 계획을 기술한다.

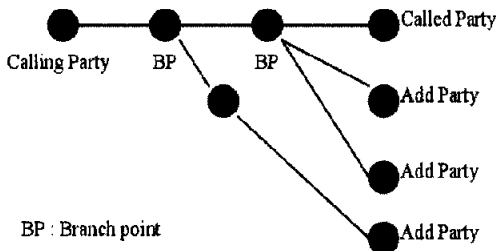


그림 1. 점대다중점 라우팅 경로

2. 시스템 및 소프트웨어 구성

본 논문이 적용 되는 시스템은 공중망과 사실망 인터페이스와 처리 기능을 포함하고 있는 ATM교환 시스템이다. ITU-T에서는 주로 공중망과 관련된 기능을 권고하며[1,2], ATM FORUM에서는 사실망 관련 기능을 권고한다[3,4,7]. 시스템의 대략적인 구조도는 그림 2와 같다. 각각의 접속시스템은 각종 인터페이스가 계위별로 접속 가능하며, 중앙 시스템에서는 접속시스템간의 연결 역할과 시스템의 공통 기능을 담당한다.

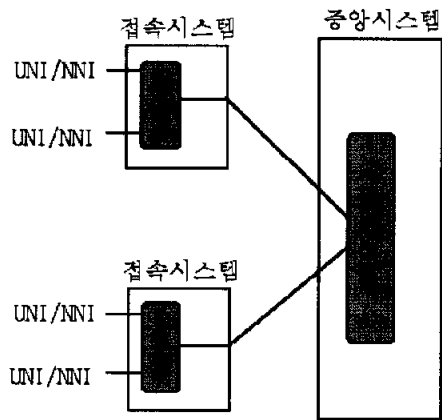


그림 2. ATM 시스템의 구조도

교환 시스템에서 호를 처리 하는 소프트웨어는 크게 프로토콜 제어 소프트웨어와 호 제어 소프트웨어로 나뉜다. 프로토콜 제어 소프트웨어는 규약된 시그널링 프로토콜을 약속 된 절차에 따라서 처리하고 송수신 하는 기능이다. 호 제어 소프트웨어는 교환 서비스를 위해서 규약 되지는 않았지만 시스템의 특성에 따르는 자원관리, 상태 관리, 번호 번역, 라우팅 등의 기능이다. 이러한 두 가지 제어 기능이 복합적으로 이루어졌을 때 정상적인 점대다중점 서비스가 가능하다.

2.1 시그널링 프로토콜 제어 기능

시그널링 프로토콜은 UNI4.0 Spec.에 따른 State Machine에 이벤트 처리를 기본으로 한다[4]. 점대다중점 서비스에서의 발신 단말은 망에서 점대점 서비스와 구분하기 위해서 Root 단말이라고 보통

말하며, 반대로 착신 단말을 Leaf 단말이라고 말한다. SETUP메시지의 BBC(Broadband Bearer Capability) 정보요소(Information Element)를 분석하여 점대점 서비스인지 점대다중점 서비스인지를 구분하며 만약 점대다중점 서비스인 경우에는 party의 identity를 위해서 ER(Endpoint Reference) 정보요소를 가지고 있어야 한다. 프로토콜 제어에는 Root 프로토콜 제어와 Leaf 프로토콜 제어가 있다. 그림 3에서는 점대다중점 시그널링 프로토콜의 진행 순서를 간략하게 나타낸다. 그림 3에서 대문자로 표기된 것은 일반적인 시스템의 시그널이 아닌 ATM시스템과 단말간에 규약된 메시지이다.

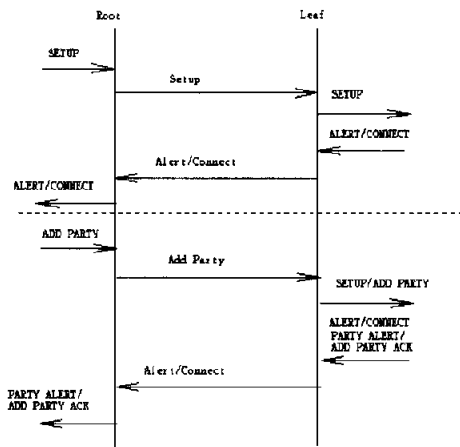


그림 3. 점대다중점 시그널링 절차

2.2 호 제어 기능

호 제어 기능은 앞 2.1에서 설명한 프로토콜 제어플 위해 교환 시스템 내에서 호의 모델을 설정하여 각종 서비스를 제공 하는 기능을 말한다. 대표적인 기능으로는 번호번역 기능, 라우팅 기능, 자원할당 기능, 스위치 링크 제어 기능 등이 있다.

우리는 호제어 기능 구현을 위하여 각 기능을 담당하는 기능 블록을 두었다. 호제어 기능을 담당하는 CCF(Call Control Function Block), 번호번역을 담당하는 NACF(Number and Address Control Function Block), 라우팅 경로 선택, DTL (Designated Transit List) 구성 및 DTL Processing을 담당하는 DRCF(Dynamic Routing Control Function Block), 스위치 및 링크 제어를 담당하는 GSCF(Global Switch Control Function

Block), 자원을 관리하는 CRCF(Call Resource Control Function Block), Signaling Protocol을 담당하는 SPCF(Signaling Protocol Function Block), 라우팅 프로토콜을 담당하는 DRPF(Dynamic Routing Protocol Function Block), 라우팅 정보 교환 및 encoding/decoding을 담당하는 DRMIF (Dynamic Routing Media Interface Function Block)을 구성하였다. 그림 4는 블록간 관계도를 보이고 있다.

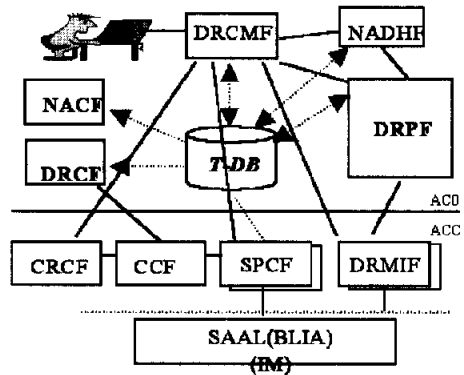


그림 4. 호제어 기능블럭 구성도

2.2.1 번호번역 기능

번호번역 기능은 호 제어 프로세스의 요구에 의해 Leaf의 주소를 번역하여 목적지 노드 및 포트(link)를 결정한다. 즉, Leaf 접속 시스템이 같은 스위칭 시스템 내에 있는지 아니면 다른 스위칭 시스템인지, 또한 같다면 같은 접속 시스템인지 다른 접속 시스템인지 구분한다. 이에 따라 스위치 링크를 접속하여 CCF는 해당 접속 시스템으로 Leaf 제어를 요구한다. 만약 서로 다른 스위칭 시스템이라면 이는 중계국을 통하여 다른 시스템으로 접속됨을 의미하므로 DRCF는 라우팅 기능을 수행하게 된다.

2.2.2 라우팅 기능

라우팅 기능은 번호번역 결과에 따라 처리방식을 달리한다. B-ISUP과 같은 hop by hop 라우팅의 경우에는 점대점 라우팅과 동일하게 처리하면 된다. PNNI와 같이 소스 라우팅을 지원해야 하는 경우에는 별도의 점대다중점 라우팅 알고리즘이 필요하다. 본 논문에서는 점대다중점 소스 라우팅 기능을 고려하고 있다. 자세한 내용은 3, 4장에서 다

루기로 한다.

2.2.3 자원 할당 기능

단말에서 요구하는 기능 중에는 실제로 데이터를 전송하는데 필요한 대역폭을 요구하고 있다. 이를 교환기에서 반영하기 위해서는 교환기 내의 자원을 관리 하여야 한다. 대표적인 자원관리로는 VPI, VCI 와 순방향, 역 방향의 대역폭을 할당하여야 한다.

2.2.4 스위치 링크 제어 기능

점대다중점의 호 제어 기능 중에서 가장 중요한 기능으로서 스위치 링크의 제어가 있다. 여러 개의 접속 시스템이 하나의 중앙 시스템에 접속되어 있는 시스템 구조에서 점대다중점 서비스를 위해서는 ATM셀의 복제 기능이 필요하다. 그림 5에서는 스위치 경로의 예를 나타낸다.

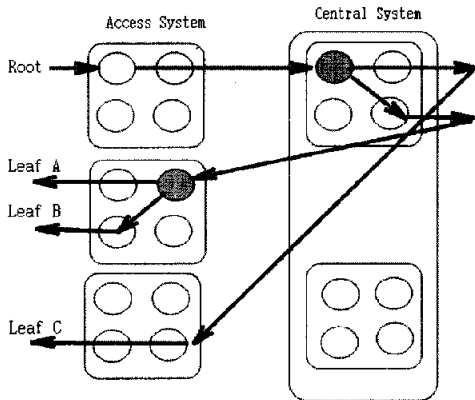


그림 5. 점대다중점 스위치 경로 설정의 예

그림에서 원은 스위치 모듈을 의미하며, 회색으로 칠해진 부분은 셀의 복제가 이루어지는 모듈이다. ATM 셀의 복제 기능은 동일한 정보가 동일한 Root에서 여러 개의 Leaf로 전달되기 위해서 교환 시스템에서 점대다중점의 Root에게 하나의 독립적인 스위치 채널을 할당 한다. 이 채널 번호를 key로 하여 추가 Leaf가 생성 된다(Leaf A). 이후 Add Party 요구가 있다고 가정 하면, 그림 3의 Leaf C와 같이 Root에서 서로 다른 접속 시스템의 Leaf인 경우에는 중앙 스위치에서 셀의 복제가 이루어진다. 서로 같은 접속 시스템인 경우에는 셀의 복제가 접

속 시스템의 스위치 모듈에서 이루어진다(Leaf B의 추가). Drop Party 인 경우에도 마찬가지로 하나의 접속 시스템에 더 이상의 Leaf 가 존재 하지 않는 경우에는 중앙 스위치에서 해당 스위치로 복제 중지를 요청하고 Leaf 프로세스는 소멸 된다.

3. 점대다중점 라우팅 소프트웨어 설계

점대다중점 호에서 기존의 호가 이루어진 착신지를 Called Party라고 하고 새로이 추가 되는 목적지를 Add Party라고 할 때 Add Party는 전적으로 같은 Call ID를 갖는 기존 호에 의존한다.

라우팅을 위한 경로 선택 또한 기존 호 즉, 같은 Call ID를 가진 Called Party가 사용한 경로들을 기반으로 하여 구성하여야 한다.

자원(resource) 관리도 Called Party와 동일한 경로를 이용하는 노드까지는 변화가 없어야 하며 branch point 이후의 경로에 대해서만 자원을 관리하면 된다. 또한 점대다중점 호는 점대점 호와 구분하는 구분자가 반드시 있어야 하며, 같은 점대다중점 호는 Call ID와 Add Party ID로 구분한다.

점대다중점 라우팅 메커니즘 구현을 위해서는 몇가지 고려사항이 있다.

- 점대점 호와 점대다중점 호를 구분하는 방법
- Branch point를 결정하는 방법
- Branch point가 정해졌다면 Add Party의 경로를 구성하는 방법
- 리소스 관리 방법
- 각 노드에서 Local Port 의 선택 방법
- release처리 방법

각 사항들에 대한 해결 방안을 정리해 보았다.

3.1 점대점 호와 점대다중점 호의 구분

소스 라우팅에서는 점대점 호와 점대다중점 호는 반드시 구분을 해서 처리해야 한다. 이는 호 제어 기능 블록(CCF)에서 동적 라우팅 제어 기능 블록(DRCF)으로 라우팅(DTL)요구를 할 때 어떠한 형태(ReqType)의 call인지를 명시해 구분할 수 있다. 그리고 점대다중점 호인 경우 Call ID와 Add Party의 ID도 함께 명시해 주어야 한다. DRCF에서는 Call ID단위로 가 Add Party를 ID를 이용하여 관리할 수 있다. DRCF에서는 점대다중점 호에 대

해서는 어떠한 형태든 그 경로들(Called Party든 Add Party든)을 모두 관리하고 있어야 한다. 이를 위해 우리는 PtMP table을 두기로 한다.

3.2 Branch Point(Node) 결정

Branch node를 정하는 일은 점대다중점 라우팅 경로 선택에서 가장 중요한 일이다. 어느 노드를 branch node로 정하는가에 따라 얼마나 optimal한 경로를 구할 수 있는가 결정된다. 여기에도 다음과 같이 몇 가지 제약사항이 있다.

(1) ATM Forum의 PNNI spec.에는 branch가 일어난 이후에는 기존의 Called Party 경로와 cross가 일어나는 일이 없어야 한다고 제약하고 있다. 이것은 Add Party의 경로 계산 시에 발생할 지도 모르는 cycle을 미리 예방하기 위해서이다. 즉 아래의 그림6과 같이 branch가 일어난 이후에 다시 cross가 발생하는 경우는 비록 그 경로가 가장 optimal한 경우라 할지라도 사용해서는 안 된다. 이런 경우에는 둘 중에 하나만 branch point로 정하여야 하며 그 이후 경로는 다르게 설정해 주어야 한다.

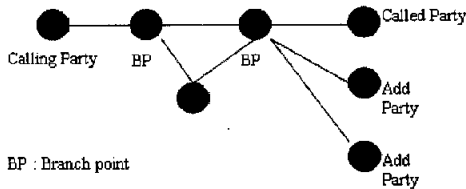


그림 6. Cycle이 발생한 라우팅 경로

(2) optimal한 경로를 찾기 위해 any node-to-any node의 경로 계산은 불가능하다. 기본적으로 소스 라우팅 기법은 소스 노드(calling party)에서 경로 계산 알고리즘에 따라 가장 optimal한 경로를 계산하는 방법이므로 Calling Party 입장에서는 다른 노드들을 착신지로 하여 가장 optimal한 경로를 구할 수 있다. 하지만 Add Party의 경우는 어느 시점을 branch 노드로 했을 때, Calling Party의 위치에서는 branch node에서 Add Party까지의 가장 optimal한 경로를 알기가 어렵다.

위와 같은 제약 사항 때문에 branch 노드는 기존의 Called Party의 경로와 Add Party의 경로를 비

교하여 마지막으로 공유하는 노드를 branch point로 잡는다. 따라서 Add Party 경로 설정시 반드시 Called Party에 대한 정보와 Called Party의 경로를 가지고 있어야만 한다. 그러므로 DRCF에서는 모든 점대다중점 호에 대해서 Call ID 단위로 PtMP table을 가지고 있어야 하며 Party ID별로 경로를 저장하고 있어야 한다. 그래서 똑같은 Call ID를 가진 Add Party 요구가 오면 기존에 설정된 똑같은 Call ID의 모든 호의 경로가 새로운 Add Party 경로 선택시 참조할 경로 정보가 되는 것이다.

우리는 위와 같은 조건을 만족하는 branch point를 쉽게 찾기 위한 방법으로 목적 노드부터 역 추적에 의해 인접노드를 분석하여 branch point를 찾는 방법을 선택하였다. 이 기법은 Add Party address (number)를 이용하여 착신 노드(destination node)를 찾고, 착신 노드로부터 역 추적에 의해 기존의 경로와 교차가 최초로 일어난 노드가 branch point가 되는 것이다. 자세한 알고리즘은 4장에서 기술한다.

3.3 Add Party의 경로 구성

Branch point가 정해지면 Add Party의 경로 구성은 쉽게 해결된다.

우선 branch 노드까지는 기존의 Called Party의 경로를 그대로 이용하고 branch 노드 이후는 새로 선택한 경로를 이용하여 Add Party의 경로를 작성하면 된다.

여기서 Called Party의 경로는 여러가지 이유로 인해서(대표적으로 AvCR(Available Cell Rate)이나 AW(Administrative Weight)의 값 변경으로 인한 경로 변화) 호 설정 시와는 달라졌을 가능성이 있다. 그러나 반드시 Called Party의 호가 설정될 당시 이용했던 그 경로라야만 한다.

우리는 이러한 사항을 고려하여 소스 노드에서 branch point까지는 PtMP table에 저장되어 있는 기존 경로를 이용하고 branch point를 찾을 때까지 역 추적에 의해 선택한 노드들을 역으로 나열하여 Add Party의 경로를 구성하도록 하였다.

또한 새로 작성된 Add Party의 경로는 Call ID 단위로 저장하는 PtMP table에 Add Party ID와 함께 release시점까지 저장하고 있어야 한다.

3.4 Resource 관리

원칙적으로 Add Party에 대한 리소스(resource) 관리는 branch 노드까지는 리소스의 할당 없이 기

존의 Called Party의 리소스 할당으로 사용한다.' 그리고 branch 노드 이후에는 Add Party가 필요로 하는 리소스를 allocation/re-allocation한다.

3.5 Local Port의 선택

그림7과 같이 노드간에는 여러 개의 link가 존재할 수가 있다. 각 노드에서는 DTL 처리를 할 때 소스 노드에서 정해진 경로에 따라 호가 진행되는 다음 노드와 local port를 선택해야 한다.

접대점 호일 경우에는 local port의 선택은 호가 요구하는 조건을 (주로 bandwidth)만족하는 port(link)중 어느 것을 선택해도 된다.

그러나, 점대다중점 호인 경우에는 그림7와 같이 Add Party1 호가 여러 개의 link 중 굵은 선 link를 사용했다고 가정할 때, Add Party2는 branch point까지는 반드시 Add Party1이 사용한 link들을 그대로 이용해야만 한다.

이러한 조건을 만족하기 위해 본 논문에서는 현재의 호가 사용하고 있는 port 정보를 알 수 있는 CCF에서 DRCF로 현재 점대다중점 Call ID가 사용하고 있는 port들을 list로 작성하여 DTL 처리 요구 시에 보내 주도록 하였다. DRCF에서는 DTL 처리시, 다음으로 이동할 노드를 찾고, port list 중 다음 노드로 갈 port가 존재한다면 그 port를 지정해주면 된다. 만일 다음 노드를 찾고 나서 port list 중에 다음 노드로 가는 port가 없다면 그 Add Party는 현재의 노드가 branch point임을 의미하므로 DRCF에서는 local port를 찾는 알고리즘(AW, AvCR, CDV(Cell Delay Variation), CTD(Cell Transfer Delay), 등을 고려하여)을 적용해서 찾아준다.

3.6 Release처리

점대다중점 release처리 시 고려할 기능은 라우팅 정보 관리와 리소스 관리이다. Release는 Add Party만 release시키는 Drop Party와 호 전체를 한꺼번에 제거하는 Release로 구분한다.

라우팅 정보 관리는 release가 일어나면 점대다중점 호인 경우 release type이라는 것을 두어 관리한다. Drop party인 경우에는 점대다중점 호를 관리하는 PtMP table에서 해당 party 호에 대한 정보만 삭제하면 되고, Call ID전체 호의 release인 경우에는 해당 Call ID의 PtMP table 전체를 삭제하면

된다.

자원 관리는 Drop Party인 경우에는 아직 다른 party가 사용 중인지 조사하여 사용 중인 경우에는 branch 노드 이후의 리소스만 되돌려 준다. 그리고 Called Party든 Add Party든 마지막으로 남은 party가 release될 때, 비로소 branch 노드 이후의 리소스뿐만 아니라 공통으로 사용하던 리소스(Calling Party에서부터 branch node까지)를 되돌려 준다.

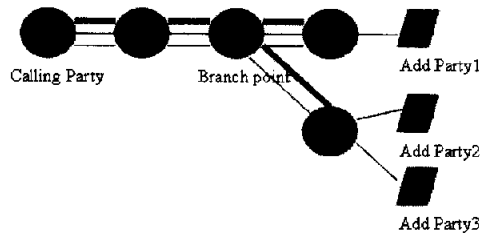


그림 7. 망간 링크(Port) 구성도

4. 점대다중점 라우팅 알고리즘

점대다중점 Add Party 호를 위한 점대다중점 경로 설정 알고리즘은 Call ID가 같은 기존 경로들(Called Party, Add Party1, Add Party2, ...)을 기준으로 하여 경로를 선택하는 것이 일반적이다.

본 논문에서 제안되는 점대다중점 라우팅 알고리즘의 특징은 리소스를 최대한 공유할 수 있고 경로 계산 시간을 최소화 하는 것이다.

이를 위하여 Add Party 착신 노드에서 소스 노드까지 역 추적에 의해 기존 경로와 크로스되는 노드까지 Hop의 수를 가장 적게 구성하는 최적의 경로선택 기법을 사용하였다.

Call ID가 같은 기존 경로상에 있는 노드와 역 추적 경로 선택 시 처음으로 만나는 인접 노드(같은 노드)를 branch 노드로 정한다. 즉, 목적지 노드부터 역으로 인접 노드 중 기존 경로상의 노드와 같은 노드가 있는지를 찾는다. 만약 있을 경우에는 이 노드를 크로스 노드로 정한다. 만약 없을 경우에는 인접 노드중 최적의 경로 계산 과정을 통해 새로운 경로 구성을 위한 노드를 구한다. 새로운 노드를 기준으로 기존 경로상의 노드와 같은 노드이며 리소스를 만족하는 인접노드를 찾을 때까지 계속 역 추적 방법을 이용하여 크로스 노드를 찾는다.

Add Party정보의 구성은 크로스 노드까지는 기존의 경로를 이용하고, 그 다음의 경로는 역 추적에 의해 구해진 노드를 역으로 붙여서 최적의 경로를 만든다.

이와 같은 기법은 경로 설정을 위한 라우팅 정보 테이블의 검색 및 경로 계산 과정을 단축시키고, 경로 선택 시 cycle방지도 쉽게 해결한다. 또한 목적 노드에서 가장 가까운 노드를 branch 노드로 선택함으로써 소스 노드에서 branch 노드까지 리소스를 최대한 공유 할 수 있도록 한다. 이는 시스템의 성능도 높이고, 리소스도 효율적으로 사용할 수 있는 효과를 갖는다.

4.1 PtMP Table 구성

우리는 점대다중점 라우팅 경로 설정을 위해 그림8과 같이 Call ID단위로 기존 호에 대한 경로를 관리하는 테이블을 둔다. Call ID가 키가 되어 해당 table을 액세스 하며, PtMP Call ID 필드, Party Id 필드, 착신지(Destination) 노드 ID 필드, Routing Path 필드 등의 정보로 구성된다.

Call ID(1) Call ID(1)	Party ID(1) Party ID(2)	Destination ID	Node	Routing Path
Call ID(1)	Party ID(3)			
~	~	~	~	~
Call ID(2) Call ID(2)	Party ID(1) Party ID(2)	Destination ID	Node	Routing Path
~	~	~	~	~
Call ID(n) Call ID(n)	Party ID(1) Party ID(2)	Destination ID	Node	Routing Path
Call ID(n)	Party ID(3)			

그림 8. PtMP table

4.2 점대다중점 경로 선택

소스 라우팅 기반의 점대다중점 호를 위한 라우팅 경로 결정 알고리즘은 다음과 같다.

(1) 라우팅 요구를 받으면 착신측 주소(Called Party Number)를 분석하여 착신지 노드를 구한다.

(2) 라우팅 요구 종류(ReqType)를 분석하여 점

대점 라우팅 요구인지, 점대다중점 라우팅 요구인지를 조사한다.

(3) 조사 결과, 점대점 라우팅 요구의 경우, 소스 노드에서 목적 노드까지의 Dijkstra algorithm을 이용하여 최적의 PtP경로를 선택한다.

(4) 조사 결과, 점대다중점 라우팅 요구의 경우, 리소스를 최대한 공유할 수 있는 최적의 경로를 구성하기 위하여 라우팅 요구사 수신한 Call ID를 이용하여 해당 PtMP table을 찾아 기존 경로들을 구한다.

(5) 기존 경로가 없을 경우에는 소스 노드에서 목적 노드까지 Dijkstra algorithm을 이용하여 최적의 경로를 선택하고 PtMP table에 Call ID와 목적지 노드 그리고 구성된 최적의 경로를 저장하여 PtMP table을 생성시킨다.

(6) 기존 경로가 있는 경우에는 목적 노드부터 역으로 해당 노드(목적지 노드 또는 (7)과정에서 새로 선택된 노드)와 인접한 노드 중에 기존 경로상의 노드와 같은 노드(cross point)며 리소스를 만족하는 노드가 있는지를 조사한다.

(7) 조사 결과, 같은 노드가 없을 경우 Dijkstra algorithm을 이용하여 인접 노드 중 최적의 경로 노드를 선택하여 저장하고, 크로스 포인트 노드 또는 다음 경로 노드 선택을 위해 상기 (6)에서 (7)의 과정을 계속 수행한다.

(8) 조사 결과, 같은 노드가 있는 경우에는 이 노드를 크로스 노드로 정한다.

(9) Add Party정보의 구성은 Cross Point까지는 기존 경로를 이용하고, 이후는 상기 (7)에서 구해진 경로(path)를 역으로 붙여서 최적의 경로(best path)를 완성한다.

(10) 경로가 구성되면 PtMP table에 Call ID와 Party Id, Destination Node, 그리고 새로 구성된 최적의 Add Party 경로를 저장한다.

```

Search destination node
IF CInfo.ReqType = PTMP
    Search PtMPtable with Call ID;
    IFthere is the call of same Call ID;
        IF there is the path having same destination
            in the corresponding PtMP table;
            Use the existing path as is (bestpath =
                existing path);
        ELSE
            GOTO New path;
        FI;
    ELSE (in case of initial PtMP call of Call ID)
        Find the best path source to destination;
        Create & Initial PtMP table;
    FI;
    Store path information (Call ID, Party ID, Path)
    to PtMP table;
    FI;

New path :
    Search the neighbor node crossing the node
    on existing path;
    IF cross node and resource available;
        Up to cross point, use table path,and
        then finish one path attaching new path;
    ELSE
        Find the best path node by back-tracking;
        GOTO New path;
    FI;
    
```

그림 9. 역추적 점대다중점 경로선택 알고리즘

4.3 시험 및 검증

여러 노드들을 연결한 망을 구성하고, 여러 ATM시스템에 본 논문에서 제안된 알고리즘을 적용하여 멀티캐스팅 연결 시험을 수행 하였다. 시험 조건은 각 노드간의 링크 구조와 링크 정보 중

AW(Administrative Weight)를 바꾸어 가면서 점대다중점 경로 선택을 시도하였다. 그림 10는 제안된 알고리즘에 의해 구성된 Add Party 경로 선택의 예를 보이고 있다.

점선은 링크의 구성을 나타내고 굵은 선을 선택된 PtMP 라우팅 경로를 나타낸다.

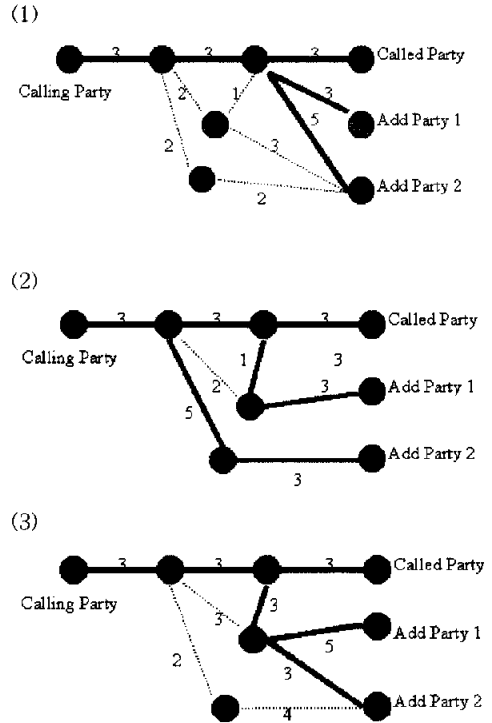


그림 10. Add Party 경로의 예

그림 10에 나타났듯이 기존의 최적 경로 선택에 의한 경로선택과 다른 경로가 선택됨을 확인할 수 있다. 즉, 착신 노드로부터 가장 Hop수가 적은 지점이 Branch Point로 결정됨을 알 수 있다. 이는 본 논문에서 제안된 알고리즘이 리소스의 최대 공유를 보장함을 의미한다.

기존에 제안된 알고리즘들도 리소스 공유와 최적 경로 선택에 대한 많은 기여를 하였다. 기존의 알고리즘 2가지와 본 논문에서 제안된 알고리즘의 상단점을 살펴보겠다.

기존 알고리즘 하나는 소스 노드에서 목적 노드 까지 선택 가능한 모든 정보를 선택한 후 기존 경

로(called Party Path)와 비교하여 가장 유사한 경로를 Add Party 경로로 선택한다[9]. 이 방법은 최대의 리소스 공유와 최적의 경로 선택을 만족시킬지는 모르지만 경로 계산 시간이 너무 많이 걸린다는 단점이 있다.

또 다른 기존 알고리즘은 임의의 노드를 그룹 노드로 정하여 점대다중점 호에 대하여 그룹 노드가 Branch Point 노드가 되도록 고정시켜 놓는다[10]. 즉 소스노드에서 그룹 노드까지 경로를 고정시켜 놓고 Add Party 호의 착신지로부터 역으로 그룹 노드까지 경로를 구하도록 하고 있다. 이 방법의 경우는 경로 선택의 단순화를 통한 경로 선택 시간의 단축의 효과는 있지만 Branch Point를 임의로 고정시킴으로써 리소스 공유에 있어 한계를 갖는다.

점대다중점 경로선택의 최우선 목표가 리소스의 최대 공유임을 고려할 때 본 논문에서 제안된 점대다중점 라우팅 경로 선택 알고리즘은 최대의 리소스 공유를 보장하고 경로 계산 시간을 단축시켰다. 즉, 위에서 살펴본 기존 알고리즘을 보다 개선시켰음을 확인할 수 있다.

5. 결론

점대다중점 기능은 하나의 소스에서 여러 개의 착신지로 Broadcasting을 원하는 서비스이다. 이 서비스를 위해서 단말에서는 크게 요구 되는 기능이 없다. 그러나 망에서는 프로토콜 제어를 비롯하여, 자원 관리, 스위치 제어, 라우팅 제어 등에 많은 추가 기능이 수반된다.

본 논문은 ATM시스템에서의 소스 라우팅 기반 점대다중점 호를 위한 라우팅 경로 선택 방법을 제안하였다.

특히 최적의 경로 선택 보다는 리소스를 최대한 공유할 수 있고, 경로 계산시간 단축 및 cycle 방지를 쉽게 보장할 수 있는 점대다중점 라우팅 경로 선택 방법을 제안하였다.

기존 경로상에 있는 노드와 역 추적 경로 설정 시 처음으로 만나는 인접노드(같은 노드)를 branch 노드로 정함으로써 경로 설정을 위한 라우팅 정보 테이블의 검색 및 경로 계산 과정을 단축시키고, 점대다중점 경로 설정 시 cycle방지도 쉽게 해결하였다.

또한 목적 노드에서 가장 가까운(minimum hop counter) 노드를 branch 노드로 선택함으로써 소스

노드에서 branch 노드까지 리소스를 최대 공유 할 수 있도록 하였다. 이와 같은 기법은 시스템의 성능도 높이고, 리소스도 효율적으로 사용할 수 있는 효과를 갖는다.

점대다중점 기능을 이용한 서비스는 망에서 소요 되는 자원의 효율성이나, 서비스 성능을 고려하여 볼 때 공중망보다는 사실 망이나 단위 스위치에서 실제 서비스에 이용 되는 경우가 많을 것으로 예상되며, 기타 일반 방송 서비스에 응용이 가능하다. 점대다중점 메커니즘의 많은 활용을 기대해 본다.

참 고 문 헌

- [1] ITU-T Q.2931, "B-ISDN DSS2 UNI Layer 3 Spec for Basic Call/Connection Control", Feb. 1995.
- [2] ITU-T Q.2971, "B-ISDN DSS2 UNI Layer 3 Spec for Point-To-Multipoint Call/Connection Control", Oct. 1995.
- [3] ATM FORUM " ATM User-Network Interface Signalling Specification Version 3.1", Sept. 1994.
- [4] ATM FORUM " ATM User-Network Interface Signalling Specification Version 4.0", July. 1996.
- [5] Toshiharu Aoki, "Future Switching System Requirements", IEEE Communication Magazine, pp. 34-38, January 1993.
- [6] David E. MacDysan, Darren L. Spohn, "ATM Theory and Application", McGraw-Hill Series on Computer Communications, pp. 461-485, 1995.
- [7] ATM Forum, "Private Network-Network Interface Specification Version 1.0", Tech. Rep. AF-pnni-0055.000, ATM Forum, March 1996.
- [8] J M Scoot. and I G Hones : "The ATM Forum's private network/network interface", BT Technol J ,Vol 16, No 2, April. 1998.
- [9] H.S. Shin, S.S. Nam, K.C. Park, "Point-to-Multipoint Routing in ATM Switching", CEIC, 2002.
- [10] R.Venkateswaran, C.S.Raghavendra, X.Chen, V.P.Kumar, "Hierarchical Multicast Routing in ATM Networks", IEEE Trans. On Communication, Vol4, pp.1690-1696, 1996.

신 현 순(Hyun-Soon Shin)



1986년 2월 한남대학교
전자계산학과(이학사), 1990년
한남대학교 대학원
전자계산학과(이학석사),
2003년 2월 충북대학교 대학원

컴퓨터과학과 박사과정 수료,
1986년 3월~현재 한국전자통신연구원
네트워크연구소 선임연구원

<주관심분야> ATM 스위칭, 주소분석 및
관리, 망관리, 네트워크 프로토콜, 라우팅
알고리즘, QoS, 유무선 통합

박 권 철(Kwon-Chul Park)

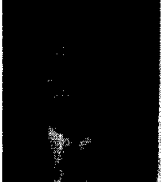


1977년: 고려대학교 전자공학
과(공학사)
1979년: 고려대학교 대학원 전
자공학과(공학석사)
1988년: 고려대학교 대학원 전

자공학과(공학박사)
1981년~현재 : 한국전자통신연구원 책임연구원,
부장

(주관심분야) 고속 스위칭, 네트워크진화, NGcN

이 상 호(Sang-Ho Lee)



1976년 숭실대학교 전자계산학
과(학사),
1981년 숭실대학교 대학원 전
자계산학과(석사),
1989년 숭실대학교 대학원 전

자계산학과(박사),
1976년 1월~1979년 5월:한국전력 전자계산소,
1981년 6월~현재:충북대학교 전기전자 및 컴퓨
터공학부 교수

<주관심분야> Protocol Engineering, Network
Security, Network Management, Network
Architecture

이 경 호(Kyeong-Ho Lee)



1980년: 광운대학교 전자통신
공학과(공학사)
1982년: 광운대학교 대학원 전
자통신공학과(공학석사)
1982년~현재 : 한국전자통신연
구원 책임연구원, 팀장

(주관심분야) 네트워크 시험망 구축 및 품질인
증, 검증