

유·무선 통합망에서의 B-ISDN 구현방안

金道善, 柳承文
國防科學研究所

I. 서론

21세기의 문턱에서, 도래하는 정보화 사회에서 요구될 모든 서비스를 원활히 수행하기 위해서는 광대역 종합정보통신망(B-ISDN)의 구축이 필수적이며, 현재의 기술추세로 볼 때 B-ISDN이 ATM(asynchronous transfer mode) 전송 기술에 의해 구현될 것이라는 사실을 부인하는 사람은 없다.

불과 10여년 전만 하여도 꿈의 통신망으로 일컬어지며 각광을 받던 협대역 ISDN이 꽃을 피우기도 전에, B-ISDN의 관심이 고조되고 있는 것은 무엇보다 사용자의 다양한 서비스 수용면에서 융통성이 있으며, 기술 측면에서는 컴퓨터, 반도체 및 광통신기술 등에 힘입은 system 구현의 용이성 등으로 집약될 수 있다.^[1,2]

이와같은 배경에 따라 현재 CCITT를 중심으로 여러 나라에서 연구중인 B-ISDN 개념은 전송 링크로 BER이 10^{-9} 이하를 유지하는 광섬유의 사용을 전제로 하기 때문에 전송로의 특성이 악화되거나 무선망과 같이 전송품질이 열악한 구간에는 그대로 적용될 수 없다. 또한 통신관련 기술이 발전해감에 따라 종래의 획일적인 개념에서 용도, 환경조건에 따라 다양한 service가 요구되어지고 있으며^[3,4], 특히 service가 개별화됨에 따라 개인 통신망(PCN: personal communication network)^[5]의 출현이 확실해져 고정통신에 대한 이동통신의 비중은 오히려 더 높아질 전망이다. 이에따라 유선망과 무선망의 protocol이 다를 경우 매우 큰 불편이 예상되므로 만일 유선망과 무선망이 동일한 프로토콜을 사용할 수 있다면 모든 통신망의 구조가 단순해져 여러 분야에서 매우 큰 효과를 기대할 수 있다.

이런 관점에서 본 논문에서는 유·무선망의 차이점을

도출하여 해결방법을 알아보고, 이를 종합적으로 수용하여 유·무선망에 공히 같은 개념으로 적용될 수 있는 새로운 ATM 전송 및 교환기술로, GOSPEL(global octad switch packet with embedded label)^[6]의 개념을 제안하였다.

이 제안된 GOSPEL망은 end-to-end protocol을 사용하여 switch 구조를 단순화 하였고, 간접헤더 구조에 의해 헤더 정보의 크기를 작게하여 특별히 고안된 FEC(forward error correction) 기능을 추가함으로써 유·무선 구분없이 적용될 수 있는 ATM 교환망으로 요약된다.

물론 본 논문에서 제안된 ATM 패킷 구조가 헤더 크기와 패킷 크기만 기존의 ATM 패킷 구조와 동일하고 헤더의 내부 구성이 서로 달라 표준화에 순응하지 못하는 문제는 있으나 표준 ATM 통신망과 접속시 단순히 동일한 길이의 헤더만 교환함으로써 신호변환이 가능하므로 표준화가 굳이 요구되지 않는 유,무선 공용 PABX나 군용 통신망에는 훌륭하게 적용될 수 있다.

따라서 본 논문에서는 유·무선망에 적용될 수 있는 GOSPEL망의 개념, 원리, 구조 및 protocol등을 자세히 살펴본다.

II. 유·무선망의 차이점과 극복방안

일반적으로 유선망에 사용하는 protocol을 그대로 무선망에 사용하는 것은 유선망과 무선망이 갖는 근본적인 차이때문에 원칙적으로 불가능하다. 한편, 지금까지의 어떠한 통신체계든 그 구성은 유·무선망이 상존하는 구조로 되어왔기 때문에 상호 접속을 위한 protocol 및 H/W 구조가 복잡하다.

따라서 이장에서는 무선망이 유선망에 비해 다른점을 살펴보고, ATM에 기초한 통신체계 구축시 해결책을 알아본다.

첫째, 전송로의 오류율이 높다. 무선망은 신호가 공중으로 전파되기 때문에 주위의 환경(날씨, 지형, ECM 등)에 민감하고 주위환경은 항상 유동적이므로, 유선망에 비해 전송오류율이 불균일하게 매우 높다. 때문에 무선망에서는 FEC(forward error correction) 기능이 필수적으로 사용되어야 하는데, 현재 거론되는 상용 ATM망에는 진정한 의미의 error correction 기능이 없어^[6] 기존의 패킷구조를 그대로 무선망에 도입할 수 없다. 따라서 이에대한 해결책으로는 무선환경에 적용할 수 있는 FEC 알고리즘이 추가되어야 하는데, 현재와 같은 패킷 구조에 FEC 기능을 넣기 위해서는 헤더필드를 크게 하거나, 헤더정보를 줄이는 방법이 고려될 수 있다.

헤더필드를 크게 하는 것은 고정된 cell내에 사용자 정보필드를 줄이는 결과를 초래하여 전송효율을 감소시킨다. 반면에 헤더정보를 줄이는 개념은 패킷이 경유될 노드들에 라우팅(routing) 정보를 분산시킨후 패킷이 경유되면서 헤더가 교환되는 간접헤더 구조를 취함으로써 해결될 수 있다.

둘째, 전송속도가 제한된다. 무선망에서는 제한된 주파수 자원을 다수의 사용자가 공유하므로, 가능한 한 가용 채널수를 늘리기 위하여 채널당 허용 대역폭(전송속도)이 제한을 받는다. 이런 이유로 TDM에 기초한 회선교환망에서는 유·무선망 가입자간 통신시 protocol 및 통신체계가 복잡하다. 그러나, ATM 교환망은 가입자 대역폭이 많이 허용하는 범주내(수백 Mbps)에서 전송하는 단위시간당 cell의 수에 의해 결정되므로 유·무선망간의 통신시 근본적인 채널 대역폭의 차이를 극복하기 위해 단순히 cell의 갯수만 제어를 하면 된다.

셋째, 정보가 선택적으로 전송될 수 없다. 무선망에서는 한 노드에서 전파하는 신호를 망내의 모든 노드가 수신하기 때문에, 통신의 일반형태인 end user간의 point-to-point 통신이나, broadcasting과 같이 user를 선택하여 정보를 전달할 수 없다. 그러나 이러한 문제는 최근에 무선망을 위해 활발히 연구중인 부호 분할 다중채널(CDMA : code division multiple access) 방식^[7]을 이용, 서로 약속된 code를 사용함으로써 해결될 수 있다.

1. 개요

현재 여러나라에서 논의중인 상용 B-ISDN의 개념은 전송로 상태가 극히 양호한 광섬유의 사용을 전제로 하기 때문에 무선망과 같이 전송 품질이 열악한 환경에는 그대로 적용될 수 없다. 따라서 이와같은 B-ISDN 개념을 무선망과 같은 환경에 적용하기 위해서는 FEC(forward error correction) 기능이 필수적이다. 그러나 제한된 헤더 영역내에 이와같은 기능을 추가하면, 실제 전송되는 헤더 정보의 크기가 감소되며, 정보의 크기를 그대로 유지하려면 패킷의 헤더를 크게 하여야 하는데, 헤더가 증가하면 정해진 cell내의 사용자 정보 필드(field)가 감소되어 전송효율이 떨어진다. 따라서 본 논문에서 제안하는 GOSPEL망^[8]은 이와같은 상용 B-ISDN의 개념을 보완하여 유·무선망에 모두 적용할 수 있는 ATM 교환망의 구축에 목적이 있다.

때문에 GOSPEL망은 정보를 cell 단위로 나누어 ATM 방식으로 전송한다는 면에서는 상용 B-ISDN과 같지만, 패킷의 헤더가 그 패킷이 경유되는 노드에서 바뀌는 간접헤더구조(indirect header structure)를 갖는 특징이 있다. 이 간접헤더 구조 방식은 패킷이 목적지로 가기 위해 망에서 필요한 라우팅(routing) 정보들을 그 패킷이 경유될 노드들에 분산시킨후, 패킷이 경유되면서 단지 다음노드에서 필요한 정보만을 갖고 전송되기 때문에 헤더 정보의 크기를 줄일 수 있다. 또한 GOSPEL망은 이와같은 간접헤더구조 개념에 의해 헤더정보의 크기를 감소시킨 대신 헤더 정보의 오류를 제어하기 위한 FEC(forward error correction) 알고리즘으로써, 무선망과 같이 전송품질이 열악한 환경에 적합하도록 특별히 고안된 RECA(Ryu's error correction algorithm)를 사용한다.

이 RECA는 error가 균일하게 발생된다고 가정할 때, 10^{-2} 의 전송로 환경에서도 사용할 수 있는 성능이 우수한 FEC 알고리즘으로 개념 및 동작은 2-2절에서 다시 논의된다.

한편, GOSPEL망에서는 패킷전송에 관련된 모든 절차를 사용자간에 처리하는 end-to-end protocol을 사용하고, 망내의 switch는 유·무선 구분없이 단순히 패킷 헤더에 의한 H/W switching만을 하기 때문에 동일한 switch를 같은 개념으로 사용하는데 자세한 S/W 구조 및 패킷 교환원리는 각각 3, 4절에서 살펴본다.

2. 특징

간접 헤더 구조를 갖으므로써 헤더정보의 크기를 감소시키며, 대신 강력한 FEC 기능을 보강하여, 유 무선

III. GOSPEL Network의 개념

망에 동일개념으로 적용되는 GOSPEL망의 특징은 다음과 같다.

- 간접 헤더 구조 채택
- FEC 기능 적용(RECA)
- End-to-end protocol 사용

1) 간접 헤더 구조

간접 헤더 구조 방식이란 그림 1과 같이 패킷이 송신측에서 수신측으로 전송되면서, 경유되는 각 노드에서 패킷의 헤더가 다음노드에서 필요한 정보로 바뀌는 것을 의미한다. 이 간접헤더 구조는 GOSPEL망이 상용의 B-ISDN과 구별되는 가장 중요한 개념중의 하나로 그

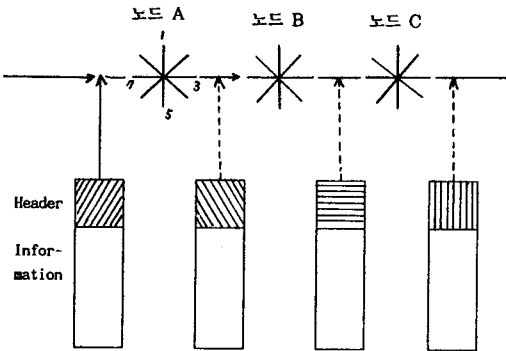


그림 1. 간접헤더 구조

원리는 call set-up 시 송신측에서 수신측으로 가는 경로(path)가 결정되면, 그 경로에 포함된 노드들이 패킷의 라우팅(routing)에 필요한 정보들을 분산하여 갖는데 기인한다. 즉, 각각의 노드는 이 라우팅 정보를 콜 정보 테이블(CIT: call information table)에 저장하고, 각각의 패킷은 자신이 도착될 노드에서 필요한 라우팅 정보(그 다음노드로 가기 위한 정보)가 저장된 CIT의 주소(address)만을 갖고 전송된다. 이와같이 GOSPEL망에서는 송신측에서 패킷이 전송될 때, 수신측까지 가는 모든 라우팅 정보를 갖고 떠나는 것이 아니고, 바로 도착될 노드에서 필요한 정보만 갖고 전송되며, 패킷을 수신한 각 노드에서는 CIT 정보를 참조로 패킷을 처리한 후 패킷헤더를 다음노드에서 필요한 정보로 교체하여 전송하기 때문에, 패킷의 헤더에는 항상 한 노드에서 필요한 정보만을 갖고 있기 때문에 헤더의 크기를 줄일 수 있다. 이와같이 패킷을 간접헤더 구조로 처리하기 위해 각 노드에서 필요한 라우팅 정보들을 저장하는 CIT의 구조는 그림 2와 같다.

CIT는 call set-up시 각 노드에서 가상채널(virtual

ADD	DIR	N.A	STA	TBD	
1					Call 1
2					Call 2
⋮					⋮
⋮					⋮
n					Call n

ADD : Address
 DIR : Direction STA : Status
 N.A : Next Address TBD : To be Determined

그림 2. CIT 구조

channel)을 설립하기 위해 1개의 CIT address를 할당하여 관련 정보들을 기록함으로써 작성되며, 이 CIT에는 라우팅 정보 외에 패킷의 처리에 필요한 각종 정보들이 함께 저장되는데 그 내용은 다음과 같다.

- DIR : 라우팅 정보가 있는 곳으로 도착된 패킷을 다음노드로 보내기 위한 출력(outgoing) 링크 번호가 저장된다.
- NA : 다음노드에서 필요한 정보들이 있는 CIT의 address가 저장되는 곳으로, 이 값이 정보패킷의 헤더가 된다.
- STA : 패킷의 우선순위, 음성/데이터 구분 및 요구 대역폭(band width)등 패킷의 품질을 보장하기 위해 패킷에 관련된 속성들이 저장된다.
- TBD : 추후 패킷의 품질을 높이거나, 새로운 서비스를 수용하기 위해 예약된(reserved) 필드이다.

이와같은 간접헤더 구조의 원리는 패킷의 처리에 필요한 모든 정보들을 CIT에 저장하고, 패킷의 헤더는 단지 각 노드에서 패킷을 처리하기 위해 필요한 정보들이 저장된 CIT의 add만 있는 unformatted 구조이기 때문에, 헤더의 크기가 항상 작으며 망에서 제공될 새로운 서비스 수용에 융통성이 있다.

2) FEC 기능

GOSPEL망이 상용 B-ISDN과 구별되는 또하나의 특징은 무선환경에 적응하기 위한 FEC(forward error correction) 알고리즘으로, 특별히 제안된 RECA(Ryu's error correction algorithm)를 사용하는데 있다. RECA의 개념은 송신하려는 정보(message)들을 4bit씩 나누어 표 1에서 미리 정해진 8bit의 RC(Ryu's code)로 치환(encoding)하여 전송하면, 역으로 수신측에서는 RC로부터 정보를 추출(decoding)하는 것으로

단순하다고 생각될 수 있으나, 핵심은 RC들이 갖는 상호간의 특수성에 있다.

표 1에서 볼 수 있듯이 어떤 Ryu's codeword든지 상호간에 hamming distance가 4 또는 8이므로, RC (Ryu's code) 8bit 중 1bit error가 생기면 정정(correction)이 가능하고, 2bit error가 생기면 검출(detection)이 가능하기 때문에, error가 균등하게 발생된다고 할 때 10^{-2} 의 링크에서도 cell 전송이 가능하다. 또한, 2bit error 검출기능을 이용하면 2.5×10^{-1} 의 링크에서도 ATM 망에서 서비스 품질로 중요하게 거론되는 CIR(cell insertion rate)을 감소시킬 수 있다.

표 1. Ryu's code table

No.	Message	Hexa Codeword	Ryu's Codeword	Antipodal	FEC표시
0	0000	55	0101 0101	15	0
1	0001	21	0010 0001	14	
2	0010	6F	0110 1111	13	
3	0011	48	0100 1000	12	
4	0100	1B	0001 1011	11	
5	0101	72	0111 0010	10	
6	0110	06	0000 0110	9	
7	0111	3C	0011 1100	8	
8	1000	C3	1100 0011	7	
9	1001	F9	1111 1001	6	
10	1010	8D	1000 1101	5	1
11	1011	E4	1110 0100	4	
12	1100	B7	1011 0100	3	
13	1101	90	1001 0000	2	
14	1110	DE	1101 1110	2	
15	1111	AA	1010 1010	1	

3. GOSPEL망의 노드 구조

GOSPEL망을 구성하는 노드는 단위 교환 switch의 조합으로 구성되며 하나의 단위교환 switch는 순수하게 패킷을 교환하는 8방향 switch와 이 switch 소자에 연결될 가입자 단말 또는 다른 switch의 정보들을 정합시키는 모듈(IM : interface module)들로 구성되며 전체 구조는 그림 3과 같다.

이와같은 구조의 특수성 때문에 IM은 단말의 종류에 따라, 그 특성에 맞는 다양한 모듈이 필요하고, switch와 switch간을 연결하기 위해서는 TIM(trunk interface module)이 필요하다.

그림 3과 같은 노드구조의 특징은 모듈별 명확한 기능분담에 따라 PIP 개념 적용이 용이하다는데 있다. 즉 8방향 switch 소자는 단순히 패킷을 교환하는 기능만 수행하고, 패킷전송에 필요한 모든 protocol 및 각종 단말 interface 기능들은 IM에서 수행되기 때문에, system 사용중 여러사항에 의해 up-grade 필요성이 생기거나 새로운 서비스가 도출될 때 단순히 해당 IM만 수정 또는 교체를 하면 된다.

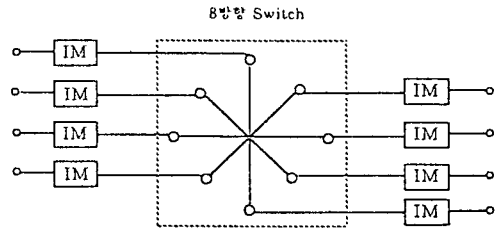


그림 3. GOSPEL 망의 노드 구조

그림 3에서 switch 소자의 각 가지를 노즐(nozzle)이라고 하는데, 이렇게 노즐의 수를 8개로 하면, 교환소자 제작시 byte 단위의 부품 및 8bit processor의 사용이 용이한 장점을 갖는다. GOS(global octad switch)라고 불리우는 이 8방향 switch는 임의의 한 노즐에서 입력된 패킷들을 어떠한 노즐로도 출력하는 교환기능을 갖는데 이것을 논리적으로 나타내면 그림 4와 같다. 그림 4에서 각각 번호가 붙여진 세모들을 형상에 기인하여 octopus라고 하는데, 하나의 octopus는 각각 1개의 nozzle과 8개의 leg를 갖는다.

4. 패킷 교환 원리

지금까지 설명된 내용을 중심으로 패킷이 교환되는 원리를 살펴보면 그림 5와 같다. 그림 5에서와 같이 7번 octopus에 CIT가 작성되었고, 그 nozzle로 헤더가 2

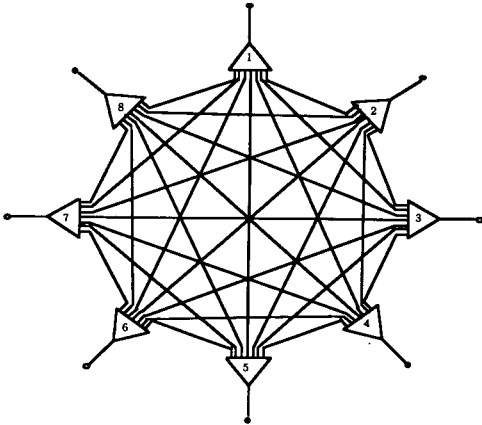


그림 4. Global octad switch

인 패킷이 입력된다면, 이 패킷을 수신하는 7번 octopus는 자신의 CIT에서 address가 2인 곳에 저장된 정보를 읽게 된다.

그림에서 DIR은 3, N.A(next address)는 7이므로, 7번 octopus는 수신된 패킷의 헤더를 7로 만든후 3번 링크로 송신하게 된다. 같은 방법으로 이 패킷을 수신하는 다음노드는 자신의 CIT 7번지에 있는 내용을 참조하여 이 패킷을 처리하게 된다.

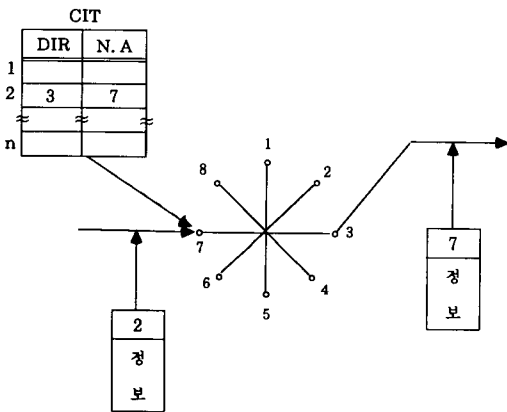


그림 5. 패킷교환 원리

IV. Protocol

GOSPEL망은 end-to-end protocol 개념에 입각하여,

패킷의 발생 및 라우팅을 IM에서 처리한다. IM에서 라우팅을 결정하는 방법은 자신이 자주 통신하는 가입자의 경로를 테이블(FCL : frequency call list)에 기록하였다가 알아내는 방법, 망내에서 경로를 가르쳐주는 센터에 문의하여 알아내는 방법 및 망내의 모든 노드에 문의하여(flooding) 알아내는 방법 등이 고려될 수 있다. 이 장에서는 라우팅 결정시 마지막 방법으로서 가장 많이 사용하게 되는 flooding 방식을 중심으로 이와 관련된 protocol을 살펴본다.

1. 패킷 구조 및 종류

GOSPEL망에서의 모든 패킷은 앞의 서론에서 잠시 언급되었듯이 상용 ATM망과의 연동을 고려하여, 그림 6과 같이 헤더 5bytes, 정보필드 48bytes로 구성하였다.

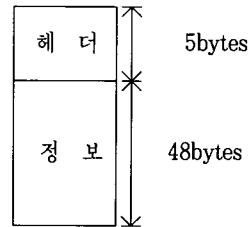


그림 6. 기본 패킷 구조

이것은 크기면에서 상용의 패킷과 같기 때문에, 추후 상용 ATM망과의 연동 필요성이 생기면, 헤더 부분만 교체하는 interface를 고려하면 된다.

한편, 현재까지 정의된 패킷은 사용자 정보를 전송하기 위해 가상채널을 설립하고, 정보를 전송한 후 가상회선을 해제하는데 기본적으로 필요한 5종류가 있으며 그 용도는 각각 다음과 같다.

- FP(flood packet) : 앞에 잠시 언급되었듯이 송신 IM이 수신측의 위치를 알기 위해 망으로 flooding하는 패킷으로, 이것을 수신하는 각 노드의 octopus는 패킷내의 정해진 위치에 순차적으로 자신의 번호를 기록하게 된다. 이렇게 망을 경유하여 이 패킷은 궁극적으로 망내의 모든 IM에 도착하게 되는데, 이때 이를 수신한 IM들은 패킷에 있는 수신자 번호와 자신에 연결된 사용자 단말 번호를 비교하여, 틀리면 cell를 폐기하고, 같으면 FAP(flood ack packet)를 이용하여 응답을 하게 된다.

- FAP : 앞에 언급되었듯이 이 패킷은 FP를 수신한 IM이 FP내의 수신자 번호와 자신의 가입자 주소가 일치하면, 자신의 상태(busy/idle) 및 위치를 송신 IM

에 알리기 위해 사용된다. 이때 이 FAP은 FP가 수신측으로 오면서 경유된 경로(octopus 번호)를 갖고 전송되기 때문에, FP가 경유되어온 경로를 따라 역으로 진행하게 되는데, FP때와 같이 이 FAP를 수신하는 각 octopus들도 패킷내의 정해진 위치에 순차적으로 자신의 번호를 기록하게 되는데 추후 이 정보들은 송신 IM에서 수신 IM으로 가는 경로 정보로 사용된다.

- CRT(call request packet) : 이 패킷은 FP를 송신한 IM이 FAP를 수신한 후, FAP의 상태(status) 필드가 idle일 때 즉 상대측이 call set-up 상태로 들어갈 수 있을때, 순 방향(forward) 가상채널을 형성하기 위해 전송한다. 따라서 이 패킷을 수신하는 각 octopus는 가상채널 형성을 위한 CIT를 작성한 후 이 패킷에 의해 지시된 다음노드로 전송하게 된다. 한편 이 패킷을 수신하는 IM에서는 역 방향(backward) 가상채널을 형성하기 위해 CRAP(call request ack packet)를 다시 전송하게 되는데, 이 패킷은 각 octopus에서 CRP와 동일하게 취급된다.

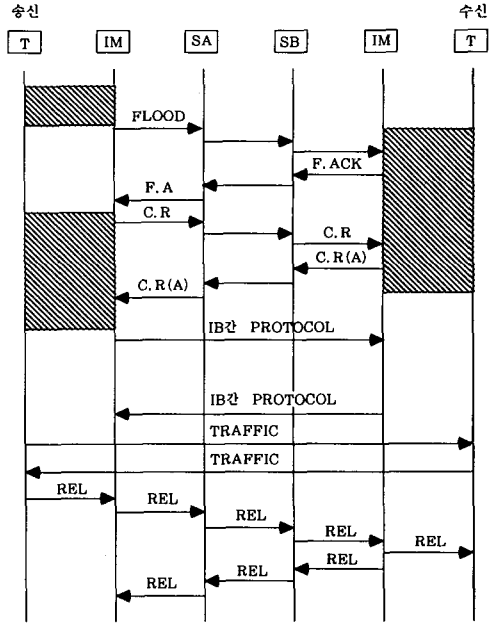
- TP(traffic packet) : 이 패킷은 가상채널이 형성된 후 사용자 정보를 전송하기 위해 사용된다. 지금까지 설명되었듯이, 이 패킷의 헤더에는 바로 도착될 노드에서 사용될 정보가 저장된 CIT 주소만 갖고 가며, 패킷이 도착되는 노드에서 CIT 정보에 의해 적절히 처리된 후 다음노드에서 사용될 헤더로 바뀌어 전송된다. 한편, 가상채널이 형성된 후 사용자 정보를 교환하기 전에 각 IM들간 제어신호를 주고받기 위해 IM간 protocol이 필요할 수 있는데, 이때도 각 IM들간의 약속하에 TP를 이용하게 된다.

이 TP를 제외한 FP, FAP 및 CRP등의 제어패킷들은 노드에 CIT를 만들기 위한 패킷이었다면, TP는 CIT를 이용하여 사용자 정보를 전송하는 패킷으로 구분될 수 있다. 때문에 이 TP를 수신하는 노드는 단순히 헤더에 의해 CIT를 읽은 후, 그 정보에 따라 다음노드로 전송하기 때문에 TP는 각 노드에서 실시간으로 처리된다.

- RP(release packet) : 이 패킷은 송 수신간에 사용자 정보교환을 마친후 가상채널을 해제하기 위해 사용된다. 따라서 이 패킷을 수신하는 각 노드는 패킷의 헤더 정보에 따라 CIT의 해당 내용을 삭제한 후 다음노드로 전송한다.

2. Call Scenario

지금까지 설명된 기본 패킷들에 의해 망에서 수행되는 기본적인 call set-up 및 해제절차는 그림 7과 같다.



T : Terminal
IM : Interface Module
SA : Switch A
SB : Switch B

그림 7. Call set up 절차

그림에서 빗금친 부분은 각각의 단말에 의해 결정되는 terminal과 IM간의 고유 protocol이기 때문에, 망의 기본 절차에서는 제외되었다.

Call 설립절차는 먼저 송신단말에서 IM에 수신측 번호를 알려주면 IM에서는 정해진 위치에 수신측 번호등을 사용하여 FP를 만든 후 망으로 전송하면서 시작된다.

이 FP를 수신하는 각 노드는 수신된 octopus 번호를 정해진 필드에 기록한 후 자신에 연결된 모든 노드로 보내게(flooding) 된다. 따라서 하나의 call에 대해 망 내에는 많은 FP가 생성되어 폭주현상이 발생할 수 있으나, 이를 위한 flooding 제어방법은 임의의 노드당 동일한 FP를 한번만 flooding 하게 하는 방법과 FP에 망에서 경유될 수 있는 최대 hop 수를 제한하는 방법 등이 고려될 수 있는데 자세한 논의는 다음 기회로 미룬다.

한편 이 FP를 수신하는 각 IM은 수신자 번호를 자신의 단말 번호와 비교하여 같게되면 FAP으로 응답하게 된다.

이 FAP은 FP가 경유되어온 경로정보를 갖고 전송되기 때문에 FP가 경유된 하나의 경로를 통하여 FP를

송신했던 IM으로 가게된다. 이 FAP를 수신한 IM은 상대 IM의 상태(busy/idle)를 읽어 busy면 더 이상의 call set-up 절차를 중지하고 단말에 busy 신호를 전송한다. 그러나 상태정보가 idle이면 순방향 가상채널을 형성하기 위해 CRP를 송신하게 되고, 망에서 이를 수신하는 각 octopus는 CIT를 작성하여 다음노드로 보내게 된다. 한편 수신 IM에서 이 CRP를 수신하면 역방향 가상채널을 형성하기 위해 CRAP를 다시 망으로 전송하게 되고 망에서는 CRP와 같이 처리하여, 궁극적으로 이 CRAP이 처음 FP를 송신했던 IM으로 수신됨으로써 가상채널이 완성된다. 앞에서 잠시 언급되었듯이 이때부터 사용자 정보전송을 위한 TP를 전송할 수 있으나, 이에 앞서 IM간의 mode setting 등 필요한 protocol를 수행하게 된다.

이렇게 하여 사용자 정보의 전송이 끝난 후 각 IM에서 RP를 전송하고, 이것에 의해 망에서 CIT가 삭제됨으로써 가상채널이 해제된다.

V. 결 론

대부분의 통신체계는 유·무선망으로 구성되지만, 두 망이 갖는 근본적인 차이때문에 동일 노드 구조 및 protocol을 갖는 하나의 통신체계로 운용될 수 없다.

최근에 활발히 연구되고 있는 상용의 ATM 개념도 광섬유와 같은 전송링크에 기초한 개념이기 때문에, 이러한 구조를 그대로 무선망에 적용할 수 없다. 따라서 본 논문에서는 유·무선망의 차이점을 도출하여 해결방안을 살펴보고, 이를 종합적으로 수용할 수 있는 통신체계로 GOSPEL망을 제안하였다.


지면 관계상 본 논문에서는 발표내용을 GOSPEL망의 개념, 원리 및 기본 패킷교환을 위한 protocol에 중점을 두었기 때문에, 각종 부수적인 기능 및 망 관리 구현개념등에 관한 언급이 없어 완전하게 통신체계 전체를 소개할 수 없었던 것은 유감이지만, 이것은 다음 기회에 다루기로 한다.

현재 GOSPEL망의 개념에 기초한 test bed^[8]를 구축하여 기본적인 성능을 확인한 바 있으며, 이 test bed를 이용하여 망 관리 개념 및 성능분석을 비롯하여 각종

패킷구조와 protocol등의 표준화 연구가 진행중에 있다. 앞으로 모든 분야에서의 표준화가 완료되면 GOSPEL망은 유·무선망에 동일개념으로 적용될 수 있는 특수성 때문에, 무선망 위주의 군 통신체계나 군이 표준화가 요구되지 않는 상용의 유·무선 공용 PABX등에 경제적으로 적용될 수 있을 것으로 판단된다.

다만 상용 ATM 개념과 상이하여, 직접적인 연동은 곤란하지만 일단 패킷 및 헤더의 크기를 같게하여 연동시 발생될 기본적인 문제를 제거하였으므로 신호변환이 매우 쉬울 것으로 예상되며, 앞으로 표준화가 완료되면 이 분야에 대한 상세 연구가 필요할 것으로 판단된다.

參 考 文 獻

- [1] Henry Gilbert, Osama Aboul-Magd and Van Phung, "Developing a Cohesive Traffic Management Strategy for ATM Networks," IEEE Communications Magazine, pp.36~45, Oct. 1991.
- [2] Thomas R.Roubotham, "Local Loop Developments in the U.K.," IEEE Communications Magazine, pp.50~59, Mar. 1991.
- [3] Mike Frame, "Broadband Service Needs," IEEE Communications Magazine, pp.59~62, Apr. 1990.
- [4] Iwao Toda, "Migration to Broadband ISDN," IEEE Communications Magazine, pp.55~58, Apr. 1990.
- [5] 최두환, "PCN의 인식", 전자공학회지, 제18권 제 8호, 1991, 8.
- [6] 최대우, 송덕영, 박홍식, "ATM 교환기술 개발동향," 대한전자공학회, 텔레콤, 제 7권 제 2호, 1991. 11.
- [7] William C.Y.Lee, "Overview of cellular CDMA," IEEE Transactions on Vehicular Technology, vol. 40, no.Z, May. 1991.
- [8] 김도선, 이성민, 서학금, 한덕찬, 박영철, "GOSPEL 교환망 TEST BED 구현," 제 2회 통신정보 합동 학술대회 논문집, pp.208~211, 1992. 

筆者紹介



金道善

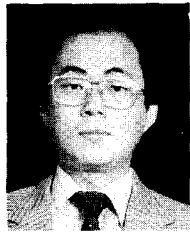
1958年 10月 3日生

1986年 2月 호서대학교 전자공학과(학사)

1988年 2月 인하대학교 대학원 전자공학과(석사)

1977年 ~ 1984年 금성전기(주) 연구소

1988年 2月 ~ 현재 국방과학연구소 선임연구원



柳承文

1953年 3月 19日生

1975年 2月 서울대학교 전기공학과(학사)

1980年 2月 한국과학기술원 전기 및 전자공학과(석사)

1985年 9月 한국과학기술원 전기 및 전자공학과(박사)

1975年 3月 ~ 현재 국방과학연구소 책임연구원