

예비 가상 경로를 이용한 ATM VC 교환기 고장 우회 방법

(Schemes to Overcome ATM VC Switch Failures using
Backup Virtual Paths)

유영환[†] 안상현^{**} 김종삼^{***}

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

요약 ATM 망에서의 장애는 가상 경로(virtual path; VP) 링크와 가상 경로 교환기(VP switches), 가상 채널 교환기(virtual channel switches)에서 발생할 수 있다. 이들 중에서 VP 링크와 VP 교환기의 장애를 복구하는 방법은 많이 연구되어 왔으나 보호 도메인(protection domain)의 종단점으로 사용되는 VC 교환기 장애에 대한 대응 방법은 아직 제안된 것이 없다. VC 교환기 장애는 단 한 번으로도 심각한 문제를 발생시킬 수 있기 때문에 이에 대한 새로운 고장 우회 방법이 절실히 요구된다.

지금까지 제안된 가상 경로 수준의 복구 방법은 크게 플러딩(flooding) 알고리즘을 사용하는 방법과 예비 가상 경로(backup virtual path; BVP)를 사용하는 방법으로 분류할 수 있다. 후자의 경우 예상치 못한 장애에 적절히 대응할 수 없다는 단점이 있기는 하지만, 복구 속도가 빠르고 필요 여유 용량(spare capacity)이 적다는 장점이 있다. 이 논문에서는 VC 교환기의 장애에 대응하기 위하여 새로운 형태의 BVP를 사용하는 복구 알고리즘을 제시한다. 제안된 알고리즘들은 공통적으로 VC 교환기의 고장으로 인해 단절된 가상 연결을 복구 할 수 있다는 장점을 가지며, 단순한 VP의 장애에 대해서도 기존의 연구와 비교하여 복구 능력의 저하가 없음을 실험을 통해 보인다.

Abstract Failures in ATM networks can occur at virtual path (VP) links, virtual path switches, and virtual channel (VC) switches. Restoration schemes have been proposed for VP link and VP switch failures, however, none for VC switch failures. In general, VC switches are used for edge nodes in protection domains. Since even only one VC switch failure can cause a critical problem, new restoration schemes for VC switch failures are highly required.

Restoration schemes at the VP level proposed so far can be categorized into those using the flooding algorithm and those using the backup virtual path (BVP) concept. Even though the latter cannot handle unpredictable failures, it has some advantages such as fast restoration and low spare capacity requirement. In this paper, we propose new restoration schemes using a new type of BVPs to handle VC switch failures. The simulation results show that the proposed schemes can restore virtual connection failures due to VC switch failures without degrading restorability for VP failures.

1. 서론

ATM 망에서의 장애는 가상 경로(virtual path; VP) 링크의 장애, VP 교환기의 장애, 그리고 가상 채널(virtual channel; VC) 교환기의 장애로 분류할 수 있다. 이들 중에서 VP 링크와 VP 교환기 장애를 함께 일컫는 VP 장애에 대해서는 여러 가지 복구 방법이 제안되었지만, VC 교환기 장애에 대한 대응 방법은 아직 제안된 것이 없다.

일반적으로 VC 교환기는 보호 도메인(protection

· 본 연구는 한국과학재단의 핵심전문연구지원사업의 지원에 의해 수행되었음 (과제 번호: 971-0905-033-1).

† 비 회 원 : 서울대학교 컴퓨터공학부
yhyoo@archi.snu.ac.kr

** 통신회원 : 서울시립대학교 전산통계학과 교수
(corresponding author.)

*** 통신회원 : 서울대학교 컴퓨터공학부 교수
ahn@venus.uos.ac.kr
cskim@archi.snu.ac.kr

논문접수 : 1998년 9월 28일

심사완료 : 2000년 4월 11일

domain)[1]의 중단점으로 사용된다. 그림 1은 보호 도메인의 개념을 나타낸 것이다. 보호 도메인이란 가상 연결(virtual connection)의 복구 단위로서 두 개의 VP와 두 개의 중단점으로 구성된다. 정상시에는 주 전송 경로(working transport entity)로 데이터를 전송하다가 주 전송 경로에 문제가 발생하면 예비로 마련해 둔 보호 전송 경로(protection transport entity)를 대신 사용하게 되는데, 이처럼 VP의 고장을 감지하고 복구 작업을 수행하는 것은 중단점으로 사용된 VC 교환기의 역할이다.

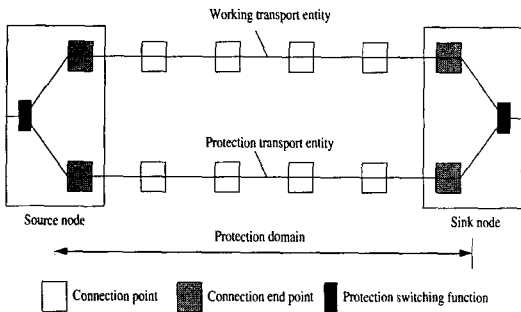


그림 1 보호 도메인(Protection domain)의 개념

VP에 문제가 발생한 경우는 VC 교환기의 주도로 복구를 수행할 수 있으나 VC 교환기 자체에 고장이 발생한 경우는 복구 방법이 없다. 더구나 어떤 VC 교환기가 여러 개의 보호 도메인에 맞물려 사용되고 있다면, 단 한번의 VC 교환기 장애가 발생시킬 수 있는 문제는 매우 심각해진다. 이런 이유로 VC 교환기의 고장 확률을 줄이기 위한 연구가 여러가지 방법으로 진행되어 왔다. 하나의 예로 교환기 내의 프로세서나 인터페이스 회로 등을 중복하여 설치하고, 이들 모두를 그물 형태로 연결하여 일부가 고장이 나더라도 교환기는 제 기능을 할 수 있도록 설계하였다. 이러한 구조의 교환기에서는 고장 확률이 극히 줄어들지만, 일반적인 교환기에 비해 네 배 내지 여섯 배의 비용을 필요로 하는 단점이 있다[2]. 그러나, 어떠한 교환기를 설치한다 하더라도 천재지변 등으로 인한 고장은 막을 수 없으며 일상적인 동작 중에도 고장이 발생할 확률은 존재한다. 그러므로, 교환기 자체의 기능에 의존하지 않는 효율적인 복구 방법이 절실히 요구된다. 본 논문에서는 VC 교환기 장애로 인한 심각한 손실을 막을 수 있는 새로운 복구 방법을 제안한다.

지금까지 제안된 VP 수준의 복구 방법은 크게 플러

딩(flooding) 알고리즘을 사용하는 방법과 예비 가상 경로(backup virtual path; BVP)를 사용하는 방법으로 분류할 수 있다. 후자의 경우 예상치 못한 장애에 적절히 대응할 수 없다는 단점이 있기는 하지만, 복구 속도가 빠르고 필요 여유 용량(spare capacity)이 적다는 장점이 있다. 이 논문에서는 VP 수준에서 VC 교환기의 장애에 대응하기 위하여 BVP를 사용하는 복구 알고리즘을 제시한다. 제안하는 복구 방법은 기존의 BVP와 조금 다른 형태의 BVP를 이용한다. 기존의 BVP는 주 가상 경로(primary VP; VP)와 중단점이 일치했으나 이 논문에서 제안하는 BVP는 한 점에서만 일치한다.

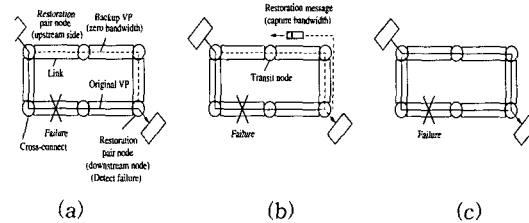


그림 2 0-대역폭 예비 가상 경로 방법

이 논문은 다음과 같이 구성된다. 2장에서는 관련 연구들을 언급하고, 3장에서는 '통합 예비 가상 경로(Integrated-BVP; IBVP)'라 부르는 복구 방법을 제안한다. 이 복구 방법을 실제로 구현하기 위해서는 ATM 신호 규약에 약간의 수정이 필요한데, 이를 4장에서 설명한다. 5장과 6장에서는 3장에서 제안한 방법의 문제점을 개선한 두 가지 복구 방법을 각각 기술한다. 7장은 실험 결과이다. 우리의 복구 방법이 별다른 성능의 저하 없이 VP의 장애는 물론 VC 교환기의 장애까지 복구할 수 있음을 실험을 통해 보일 것이다. 지금까지 VC 교환기 장애에 대한 VP 수준의 복구 방법이 제안된 적이 없으므로, 성능 비교는 주로 2장에서 설명하는 0-대역폭 예비 가상 경로 방법(zero-bandwidth BVP scheme)[3]을 기준으로 하였다. 마지막으로 8장에서 결론을 맺는다.

2. 관련 연구

VP 장애에 대한 VP 수준의 복구 방법으로 R. Kawamura는 0-대역폭 예비 가상 경로를 사용하는 방법을 제안하였다[3,4]. 물론 이전의 방법에서도 BVP를 사용하기는 했지만, 이들 방법에서는 BVP가 설정될 때 미리 대역폭을 할당했기 때문에 실제 고장이 발생하지 않는다면, '1+1' 또는 '1:1' 보호 도메인의 경우 전체 대

역폭의 절반을 낭비하게 되는 문제점이 있었다[5]. 반면, 0-대역폭 BVP는 경로만 미리 설정해 두고 실제 장애가 발생했을 때 비로소 대역폭을 할당하므로 대역폭을 사용하지 않고 낭비하는 일은 없다. 하지만, 복구를 수행하는 시점에 충분한 양의 대역폭이 남아있지 않으면 복구를 할 수 없는 문제가 있다. 하나의 장애로 인해 VP와 BVP가 동시에 단절되는 일을 방지하기 위하여 두 경로는 서로 교차하는 점이 없어야 한다.

그림 2는 0-대역폭 BVP 방법이 실제로 어떻게 동작하는지 나타낸다. 그림 2 (a)는 하나의 VP와 그 짝이 되는 0-대역폭 BVP의 그림이고, VP에 장애가 발생하면 (b)에서 보는 것과 같이 하향 노드(downstream node)가 BVP를 따라 복구 메시지(restoration message)를 전송한다. 이 복구 메시지는 VP를 복구하는 데 필요한 대역폭 정보를 담고 있어서 BVP상의 노드들은 이 값과 자신이 가지고 있는 여유 용량을 비교한다. 노드가 충분한 대역폭을 가지고 있다고 판단되면 계속해서 다음 노드에 복구 메시지를 전달하고, 그렇지 못한 경우에는 하향 노드 쪽으로 복구 대역폭 수용 불가 메시지(uncapturable message)를 보내 복구 작업을 중단시킨다. 복구 메시지가 상향 노드(upstream node)에 도착했다는 것은 모든 중간 노드들이 복구에 필요한 충분한 대역폭을 가지고 있다는 사실을 의미하므로, 이때부터 장애가 발생한 VP에 속한 모든 셀들은 BVP를 통해 전달된다(그림 2 (c)).

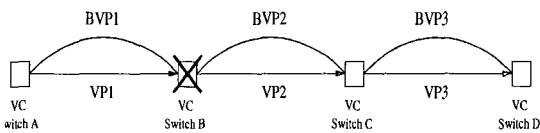


그림 3 기존의 예비 가상 경로

한편, E. Oki는 VC 수준의 복구 방법을 제안하였다 [2]. VP 수준의 복구 방법은 하나의 VP에 속한 모든 VC에게 같은 수준의 유용성(availability)을 제공하므로, 상대적으로 중요하지 않은 VC가 많은 경우 불필요하게 비용을 증가시키는 단점이 있다. E. Oki의 제안에서는 같은 VP에 속한다 할지라도 각 채널의 우선 순위에 따라 별도로 복구를 수행한다. 이 방법은 VC 수준 복구의 부수적인 효과로 VC 교환기의 장애에 대해서도 경제적이고 유연하게 대처할 수 있기는 하지만, VC 종단간(end-to-end)에 복구가 수행되기 때문에 시간이 많이 걸리고 VP 하나 당 최대 2^6 개나 되는 VC들을 별도

로 관리하는 데 따른 부하가 증가하게 된다.

또, S. Ahn은 VP 복구 시간과 BVP로 인한 비용을 줄이기 위하여 RBVP (Ring BVP) 방법을 제안하였다 [6]. 하지만, 역시 VC 교환기 장애 복구에 대한 언급은 없다.

3. 통합 예비 가상 경로 방법

이 장에서는 ‘통합 예비 가상 경로(Integrated-BVP; IBVP)’라는 새로운 형태의 BVP를 이용한 복구 방법에 대해서 설명한다. 이 방법은 VP 수준 복구 방법의 단순성과 빠른 속도는 물론, VC 수준 복구 방법에서 얻을 수 있는 유연성을 제공한다. 또한, 이 방법은 이 논문에서 목표로 하는 VC 교환기 장애를 복구할 수 있다.

지금까지 BVP를 이용한 복구 방법들이 많이 제안되었다. 이들 방법에서 사용한 BVP는 주로 종단간 OAM(Operations And Maintenance) 셀의 전송 범위와 일치하였기 때문에 그림 3에서 보는 것처럼 양 종단점이 VP의 종단점과 일치하였다. 1장에서 언급했듯이 이 종단점 역할을 하는 것이 VC 교환기이다. VP 장애가 발생하면 VC 교환기가 Alarm Indication Signal(AIS)이나 Remote Defect Indication(RDI)[7]에 의해 고장을 감지하고, 고장으로 영향받은 채널들을 BVP로 옮긴다. 이런 복구 과정에서는 VC 교환기가 주된 역할을 하게 되는데, VP와 BVP가 종단점을 공유하기 때문에 VC 교환기가 고장났을 때는 복구가 불가능하다.

이 문제를 해결하기 위하여 그림 4와 같은 IBVP 방법을 제안한다. 그림 3과 같은 일반적인 망에 [3]에서 제안된 0-대역폭 BVP를 설정한 것으로서 그림에서의 BVP는 장애가 발생했을 때 사용되는 경로를 개념적으로 나타낸 것이지 실제 그러한 경로가 물리적으로 독립되어 존재하는 것은 아니다. 이 방법에서는 VP와 BVP의 상향 노드 종단점은 일치하지만, 하향 노드 종단점은 일치하지 않는다. 대신 연결한 각 VP 상의 VP 교환기 중에서 가장 가까운 교환기가 BVP의 다른 한 종단점으로 선택되고, 결과적으로 이전에는 보호 도메인의 시작과 끝 종단점 사이에 하나만 있으면 되었던 BVP가 연결한 VP의 개수만큼 필요하다. 즉, 그림 4의 VP1은 VP2, VP2'과 연결하고 있으므로, 연결한 각 VP 위의 VP 교환기 중에서 가장 가까운 교환기들과의 사이에 하나씩의 BVP를 준비하였다.

이상의 설명에서처럼 IBVP를 이용한 방법은 필요한 BVP의 개수가 연결 VP의 개수에 비례하여 증가하는 단점이 있지만, VC 교환기 장애 발생시 연결 VP와 BVP를 통합함으로써 복구를 수행할 수 있다. 여기서의

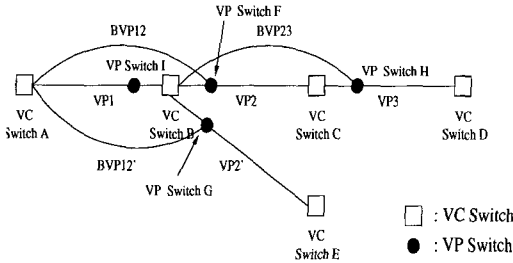


그림 4 통합 예비 가상 경로 (IBVP)

BVP는 0-대역폭 BVP이므로 BVP 수의 증가로 인한 대역폭의 낭비는 없다.

3.1 장애 복구

그림 4의 VP1에 장애가 발생했을 경우 장애를 감지한 VP 교환기는 VC 교환기 B에게 AIS를 보내 장애 발생 사실을 알린다. 그러면, VC 교환기 B는 VC 교환기 A에게 RDI를 보내게 되고, 이 과정에서 VP 상의 모든 교환기는 고장 사실을 알게 된다. 이후 VC 교환기 A는 뒤에서 서술할 복구 알고리즘을 수행한다.

한편, 그림 5의 VC 교환기 B에 장애가 발생하면 VP 교환기 I는 교환기 B로부터 일정 시간 이상 신호가 송출되지 않아서 물리층(SONET/SDH)에서 발생하는 PHY-AIS에 의해 VC 교환기 B의 고장을 알 수 있다. 물리층에서는 VC 교환기의 고장에 대응할 수가 없으므로, 복구 수준 상승(escalation)[1, 8]이 일어나 VC 교환기 A를 향해 VP-AIS를 전송한다. VC 교환기 A는 이 VP-AIS로 인해 고장을 감지하고 복구 알고리즘을 수행한다.

VC 교환기 A는 우선 VP 교환기 F와 G에게 복구 요청 메시지(Restoration Request Message)를 전송한다. 4장에서 설명할 개선된 ATM 신호 규약 덕분에 교환기 A는 교환기 B의 가상 채널 식별자(VC identifier; VCI) 테이블의 내용을 알고, BVP12와 BVP12'에 필요한 대역폭의 양을 복구 요청 메시지에 표시한다. BVP 상의 모든 노드가 충분한 여유 용량을 가지고 있다면 이 메시지를 계속 전달하고, 그렇지 않다면 역방향으로 중단 메시지(Cease Message)를 전송하여 복구 작업을 중단시킨다.

복구 요청 메시지가 교환기 F와 G에 도착하면, 이 교환기들은 복구가 완료되었음을 알리기 위하여 확인 메시지(Confirmation Message)로 응답한다. 이후부터 BVP12와 VP2가 하나의 VP가 되고, BVP12'과 VP2'이 또 하나의 VP로서 사용된다.

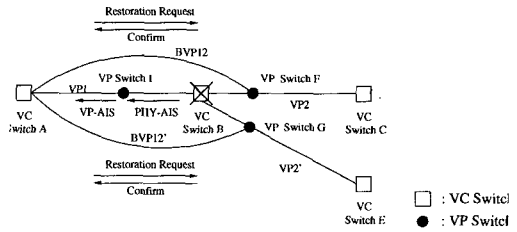


그림 5 IBVP에 의한 VC 교환기 장애 복구 과정

위의 두 가지 예에서 VP1의 장애이거나 VC 교환기 B의 장애이거나에 상관없이 모두 VC 교환기 A가 복구를 수행하게 되는데, 이는 IBVP 방법에서 BVP의 다른 중단점이 VP 교환기이므로 복구를 능동적으로 수행할 수 없기 때문이다.

IBVP 방법에서 원래 나뉘어 있었던 두 개의 VP가 하나의 VP로 통합되기 위해서는 두 가지 문제가 존재한다. 첫째, 통합된 이후 VC 교환기 A는 VP1에서 사용되던 VCI가 아닌 VP2와 VP2'에서 사용되는 VCI를 이용하여 셀을 전송해야 하므로, VC 교환기 A가 VP1 뿐 아니라 VP2, VP2'에서 사용하고 있는 VCI 정보를 알아야 한다. 이것은 4장에서 제안할 IBVP를 위한 신호 규약에 의해 가능하다.

둘째, VP 교환기 F와 G는 BVP12와 BVP12'으로 들어온 셀을 교환기 B에서 들어온 셀을 출력하는 포트와 동일한 포트에 내보내야 한다. 이는 BVP를 설정할 때 VP 교환기 F와 G가 VC 교환기 B로부터 입력되는 셀의 VPI와 BVP로부터 입력되는 셀의 VPI를 동일한 VPI로 교환하여 출력하도록 VPI 참조 테이블을 구성하면 된다.

3.2 성능 비교

이 장에서는 IBVP 방법과 R. Kawamura가 제안한 방법[3]을 VP 장애에 대한 복구 시간으로 비교한다. 메시지가 하나의 VP를 통과하는 시간을 1 시간 단위(time unit)로 잡고, 평균적으로 장애가 VP의 중간 지점에서 발생한다고 가정한다. R. Kawamura 방법의 경우(그림 2) VP-AIS를 하향 노드로 보내는 데 0.5 단위, 하향 노드가 복구 메시지로 응답하는 데 1 단위가 필요하여 결과적으로 총 1.5 단위의 시간이 필요하다.

반면, IBVP 방법에서는 전체 3.5 단위의 시간이 필요하다. 그림 4를 예로 설명하면, 만일 VP1에서 장애가 발생했다고 가정하자. 장애 발생 지점으로부터 하향 노드 B까지 VP-AIS가 전송되는 데 0.5 단위, 하향 노드 B가 VP-RDI를 상향 노드 A에 전송하는 데 1 단위,

또 상향 노드 A가 VP 교환기 F, G와 복구 요청 메시지와 확인 메시지를 주고받는 데 각각 1 단위씩 필요하다. R. Kawamura의 방법과 달리 VP-RDI의 전송이 필요한 이유는 OAM 셀이 복구 도메인인 VPI를 벗어나 VP2와 VP2'에 있는 VP 교환기 F, G에게 장애를 알려줄 수 없기 때문이다. 또한, VP 교환기는 능동적으로 복구를 수행할 수 없다고 가정하였기 때문에 상향 노드 A에게 장애 발생 사실을 알려야 한다.

복구 시간의 비교에 있어서는 R. Kawamura 방법이 IBVP 방법보다 유리하지만, IBVP 방법은 VC 교환기 장애를 복구할 수 있다는 장점이 있다. 5장과 6장에서는 IBVP 방법을 개선하여 복구 시간을 대폭 줄일 수 있음을 보인다.

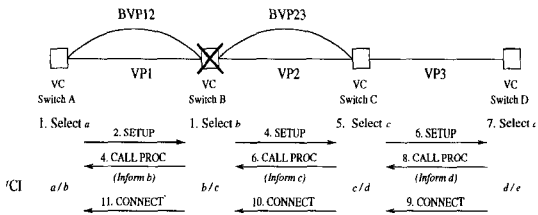


그림 6 ATM 포럼의 PNNI 신호 규약

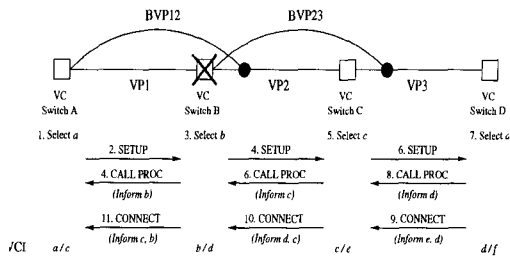


그림 7 제안하는 PNNI 신호 규약

4. IBVP를 위한 신호 규약

ATM 망에서 연결(connection)을 설정하고자 할 때 ATM 신호 규약과 신호 메시지들을 사용한다[9, 10, 11, 12]. PNNI 신호 메시지의 종류에는 설정(SETUP), 호 처리(CALL PROCEEDING), 연결(CONNECT), 해제

(RELEASE) 메시지 등이 있고, 각 메시지마다 9옥텟의 공통 필드 외에 다양한 길이의 정보 요소(Information Elements)가 추가된다. 공통 필드에는 신호 메시지의 종류와 길이, 현재 설정하고 있는 호를 구별하기 위한 식별자 등의 정보가 포함된다.

그림 6은 ATM 포럼에서 제안한 신호 규약이다[10]. 어떤 노드가 연결을 설정하고 싶으면 설정 메시지를 목적지 노드로 전송한다. 그림에서 이전 노드로부터 설정 메시지를 받은 VC 교환기 A는 호 수용 제어(Call Admission Control, CAC) 기법[7, 13]을 이용하여 연결을 수용할 수 있는지 검사한다. 이때 수용할 수 있다는 판단을 내리면, 앞 노드와 연결된 VP에서 사용할 VCI a를 선택하고, 호 처리 메시지와 설정 메시지를 각각 이전 노드와 다음 노드에 전송한다. 호 처리 메시지는 사용할 VCI를 알려주는 동시에 ACK의 역할을 한다. 설정 메시지가 목적지 노드에 도착하면, 성공적으로 연결이 설정되었다는 것을 알리기 위하여 목적지 노드는 원천지 노드에게 연결 메시지를 전송한다.

IBVP 복구 방법을 사용하기 위해서는 약간의 추가 기능이 필요하다. 이전에는 교환기가 자신의 VPI/VCI 테이블만 가지면 됐지만, IBVP 방법에서는 셀 진행 방향으로 다음 노드의 VPI/VCI 테이블을 갖고 있어야 한다²⁾. 예를 들어 그림 7의 VC 교환기 B가 고장이 났다고 가정하면, IBVP 방법에서는 BVP12와 VP2를 하나의 VP로 통합하여 복구를 완료하고, 이후 VC 교환기 A는 VPI에서의 VCI가 아니라 VP2에서의 VCI를 사용하여 셀을 전송한다. 즉, VC 교환기 C는 자신이 VP2에 할당한 VCI c를 교환기 B 뿐 아니라 A에게도 알려야 한다.

이런 기능은 별도의 심각한 부하를 유발하지 않고도 구현할 수 있다. ATM 포럼의 PNNI 연결 메시지는 연결 식별자(Connection Identifier) 필드를 갖지 않지만, 신호 메시지는 필요에 따라 다양한 정보 요소들을 추가할 수 있으므로³⁾ '연결 식별자 1'과 '연결 식별자 2'라는 두 개의 필드를 연결 메시지에 추가한다. 이 필드들은 각각 다음 노드와 그 다음 노드에 의해 해당 호에 할당된 VCI 값을 포함한다. 예를 들어 그림 7에서 9번의 연결 메시지는 교환기 D와 그 이후의 교환기가 할당한 VCI d와 e를 두 필드에 갖고 있다. 교환기 C는 이 중

1) 이 논문에서 가정하는 ATM 망이 NNI이므로, 신호 규약도 ATM 포럼의 사설 망-망 접면 (Private Network-Network Interface, PNNI) 신호 규약을 기준으로 한다.

2) 교환기의 구조는 ATM 표준안에서 제외하여 각 개발자마다 독특한 기능을 가진 교환기 설계를 가능하게 했으므로[14], 이 정도의 기능 추가는 손쉽게 이루어질 수 있다.
3) 실제로 Fore System에서는 SPAN이라는 자신들의 독특한 신호 규약을 사용하고 있다[13].

VCI *e*를 취하고, 대신 자신이 할당된 VCI *c*를 필드에 채운다. 같은 방식으로 교환기 B는 VCI *d*를 취하고, 대신 VCI *b*를 채운다. 또, 각 필드마다 한 비트의 플래그를 두어 교환기가 어떤 VCI를 취해야 할지 알 수 있도록 한다. 이전의 신호 규약과 비교하여 추가된 메시지가 없으므로, IBVP를 사용함으로써 인한 별도의 부하가 발생하지 않는다.

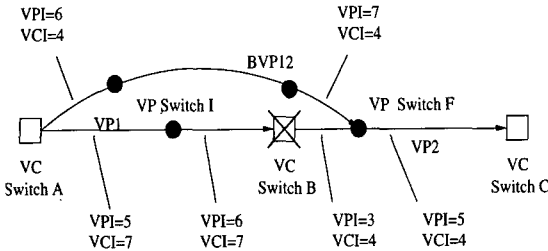


그림 8 장애 복구 후 VCI 교환 형태

장애가 발생했을 때 실제 이 VPI와 VCI 값이 어떻게 바뀌는지 그림 8을 통하여 살펴보자. VC 교환기 A에서 VPI=5, VCI=7로 출력된 셀이 VP 교환기 I에서 VPI=6으로 교환되고, VC 교환기 B를 통과할 때 VPI=3, VCI=4를 갖는다고 한다. 그리고, BVP12에서 VP 교환기 F로 들어오는 셀은 VPI=7을 가지며, 교환기 F는 BVP12를 설정할 때 VPI=3인 셀과 VPI=7인 셀을 모두 VPI=5로 교환하여 출력하도록 미리 VPI 참조 테이블에 정의하였다. 따라서, 교환기 B의 고장이 발생하면 VC 교환기 A는 IBVP 방법을 이용하여 BVP12와 VP2를 통합하고, VCI=7 대신 그림 7의 신호 규약을 이용하여 미리 알고 있던 VP2에서의 VCI인 4를 셀에 담아 BVP12를 통해 전송한다.

5. 개선된 IBVP 방법

VP 장애에 대하여 IBVP 방법은 이전의 방법들보다 더 많은 복구 시간을 필요로 하는 단점이 있다. 만일 VP 장애가 빈번하게 발생하고 VC 교환기 장애가 거의 발생하지 않는다면, IBVP 방법을 사용하는 것보다 기존의 방법들을 채택하는 것이 유리할 것이다. 따라서, 이 장에서는 개선된 IBVP(Advanced IBVP; AIBVP) 방법을 제안한다. 이 방법은 그림 9에서 보듯이 IBVP에 기존의 BVP를 추가한 것이다.

AIBVP 방법이 IBVP와 다른 점은 VP 장애와 VC

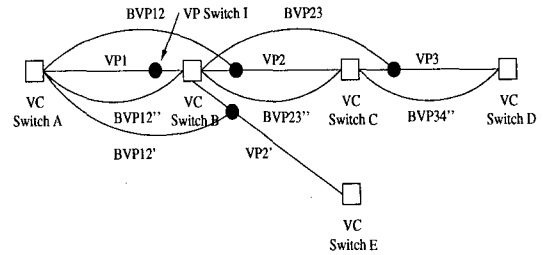


그림 9 개선된 통합 예비 가상 경로 방법 (AIBVP)

교환기 장애에 대해 서로 다른 복구 알고리즘을 수행한다는 것이다. VP 장애에 대해서는 R. Kawamura의 방법을 사용하고, VC 교환기 장애에 대해서는 IBVP와 같은 방법을 사용한다. 예를 들어, 그림 9의 VP1이나 VP2에 장애가 발생했을 경우에는 BVP12''이나 BVP23''을 사용하여 복구하고, VC 교환기 B에 문제가 발생했을 경우에는 BVP12와 BVP12'을 사용하여 복구한다. 단, VC 교환기 B와 BVP의 종단점으로 사용된 VP 교환기가 동시에 고장나는 일은 없다고 가정한다.

따라서, VP 장애 복구 시간은 R. Kawamura의 방법과 같으면서 VC 교환기 장애까지 복구할 수 있다. 하지만, 이 방법은 더 많은 BVP를 필요로 하고, VC 교환기 장애와 VP 장애를 구별할 수 있어야 한다. 두 종류의 장애를 구별하는 방법은 몇 가지가 있다. 쉽게 생각할 수 있는 방법은 AIS/RDI 셀의 장애 형태 필드(Defect Type Field)를 이용하는 것이지만, 아직 이 필드의 사용이 명확히 규정되지 않았다[11]. 다른 방법으로는 반드시 VP-AIS를 받은 VC 교환기가 복구 작업을 수행하도록 하는 것이다. 그림 4와 그림 5를 보면 VP 장애와 VC 교환기 장애에 대해 각각 하향 노드와 상향 노드가 VP-AIS를 받게 됨을 알 수 있다.

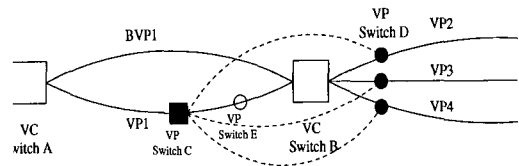


그림 10 분할 통합 예비 가상 경로 방법 (SIBVP)

6. 분할 IBVP 방법

이 장에서는 AIBVP의 변형인 분할 IBVP (Split-IBVP, SIBVP) 방법에 대해 기술한다. 이 방법에서는

그림 10의 교환기 C와 같이 실제로는 VC 교환기이지만, VP 교환기로 동작하고 있는 교환기가 BVP의 중단점으로 선택되어 복구 시간을 줄인다. VC 교환기를 평소에는 VP 교환기로 사용한다는 가정하므로, 비용을 증가시키더라도 안전성을 높이고자 할 때 적용될 수 있는 방법이다.

6.1 장애 복구

SIBVP 방법에서도 AIBVP 방법과 같이 VP 장애와 VC 교환기 장애에 대해 서로 다른 복구 과정을 거친다. 우선 VP 장애에 대해서는 R. Kawamura의 방법을 사용하므로, 1.5 단위 시간 내의 복구가 가능하다. 그림 11의 예를 이용하여 VC 교환기 고장으로 인해 단절된 가상 연결을 복구하는 과정을 살펴보자. VC 교환기 B에서 장애가 발생했다고 가정하면 물리층에서 발생한 PHY-AIS가 VP 교환기 E에 도착하고, 교환기 E는 이어서 VP-AIS를 발생시킨다. VP 교환기는 VP-AIS/RDI 셀을 감지할 수 있으므로, 교환기 C가 고장을 발견하고 복구 작업을 시작할 수 있다. 교환기 C는 그림 7과 같은 방법으로 VC 교환기 A가 연결을 설정할 때 신호 메시지를 읽어 교환기 A가 설정한 VC들의 상태를 알 수 있다. 따라서, 어떤 VC가 어떤 VP로 연결되어야 하는지 알고 있고, 이에 따라 복구 요청 메시지와 확인 메시지를 주고받는 과정은 IBVP, AIBVP와 동일하다. 단, 기존의 VP1이 분할되어 VC 교환기 A와 VP 교환기 C 사이가 하나의 완전한 VP가 되고, 그 이후와 VP2가 하나의 VP로 통합된다는 점이 SIBVP의 특징이다.

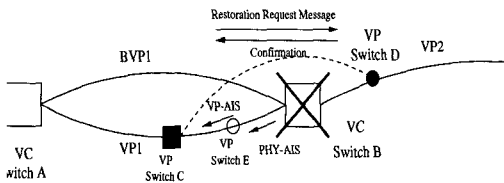


그림 11 SIBVP에 의한 VC 교환기 장애 복구 과정

표 1 복구 시간의 비교

(단위: 메시지가 통과하는 VP의 개수)

	Kawamura	IBVP	AIBVP	SIBVP
VP 장애	1.5	3.5	1.5	1.5
VC 교환기 장애	-	3	3	1.5

SIBVP 방법은 VC 교환기 장애에 대해서 AIBVP 방법을 향상시킨 것이다. 만일 그림 10의 VC 교환기 C가 VP1의 중간에 위치한다고 가정하면, VC 교환기 B의 고장으로 인한 VP-AIS가 교환기 C에 도착하는 데 0.5 단위, 복구 요청 메시지와 확인 메시지를 전송하는 데 각각 0.5 단위가 필요해서 전체적으로 1.5 단위 시간이 필요하다. 결국 R. Kawamura의 방법에서 VP 장애 복구에 필요한 시간과 같은 시간 내에 VC 교환기 고장으로 인한 가상 연결의 장애를 복구할 수 있다. 반면, IBVP 방법과 AIBVP 방법에서는 VP-AIS와 복구 요청 메시지, 그리고 확인 메시지 전송에 각각 1 단위가 필요하기 때문에 총 3 단위 시간이 필요하다. 각 복구 방법의 복구 시간 비교는 표 1에 나타내었다. 단위는 메시지가 통과하는 VP의 개수이다.

6.2 분할 IBVP 방법의 제한과 문제점

SIBVP 방법은 장애가 발생한 VP 상에 VC 교환 기능을 가진 교환기가 있을 때만 적용할 수 있다는 제한이 따른다. 또, 이 방법을 사용하기 위해서는 AIS/RDI 셀을 중간에서 가로채고 폐기해야 하는데, 지금까지의 AIS/RDI 셀은 언제나 중단간 OAM 셀이기 때문에 중단점에서만 읽거나 폐기할 수 있다[16]. 그러나, ITU-T에서는 최근에 부분 구간(segment) AIS/RDI 셀의 사용을 규정하고 이를 문서화하는 작업 중에 있다[17]. 만일, 그림 10에서 VC 교환기 B부터 VP 교환기 C까지를 하나의 부분 구간으로 정의하고, 부분 구간 AIS/RDI를 여기에 적용한다면 중단간 OAM 셀을 중간에서 폐기해야 하는 문제는 자연스럽게 해결된다.

7. 실험 결과

제안한 세 가지 방법과 R. Kawamura의 방법을 실험 비교하였다. 실험에 사용되는 망은 15개의 노드로 구성되어 있고, 각 노드는 모두 VC 교환 기능을 가지고 있어서 경우에 따라 VC 교환기 또는 VP 교환기로 사용된다. 각 노드의 입력 포트와 출력 포트는 각각 여덟 개씩으로 가정하고, 어떤 노드 간에 물리적 연결을 구성할 것인가는 확률 변수에 의해 임의로 결정하였다. 두 노드 간의 물리적 연결은 양방향성을 가지므로, 서로 데이터를 주고 받을 수 있다. 이와 같이 물리적 링크로 연결된 망에서 역시 임의로 60개의 VP를 구성하였는데, 어떤 VP는 하나의 링크로 이루어질 수도 있고, 또 어떤 VP는 여러 개의 링크로 이루어질 수도 있다. VP의 중단점으로 사용되는 모든 노드들은 VC 교환기의 역할을 하는 것이고, 여러 개의 링크로 이루어진 VP의 경우 중

단점을 제외한 중간의 노드들은 VC 교환 기능을 가지고는 있지만 VP 교환기로 사용되고 있다고 가정하였다.

이상과 같이 구성된 임의의 그물망(mesh network)에 대하여 각 복구 방법의 특징에 따라 BVP를 설립하였다. R. Kawamura의 방법에서는 모든 60개의 VP에 대하여 중단점이 서로 일치하는 하나씩의 BVP를 설립하였고, IBVP 방법에서는 두 개의 연결 VP 쌍에 대하여 하나씩의 통합 예비 가상 경로(IBVP)를 설립하였다. AIBVP에서는 모든 VP에 대해 중단점이 일치하는 BVP 하나씩과 연결 VP 쌍에 대해 역시 한 개씩의 IBVP를 설립하였고, SIBVP에서는 AIBVP와 같은 수의 BVP와 IBVP를 설립하되 IBVP의 경우 VP 중간에 위치한 VP 교환 기능만을 하는 VC 교환기에서부터 시작하는 IBVP를 설립하였다.

실험은 두 가지 형태로 수행하였다. 첫 번째는 망의 각 구성 요소마다 같은 확률로 고장이 일어나는 경우 각 방법의 복구 시간을 비교하였는데, 모든 VP에 한 번씩 장애가 발생하는 경우와 모든 VC 교환기에 한 번씩 장애가 발생하는 경우, 마지막으로 모든 VP와 VC 교환기에 한 번씩 고장이 일어나는 경우로 나누어 실험하였다. 두 번째는 VC 교환기 장애가 전체 장애에서 차지하는 비율이 변화함에 따라 전체 복구 시간이 어떻게 달라지는지 알아보고, 각 방법의 복구 시간을 비교하였다. 복구 시간은 장애 복구를 위한 메시지가 통과해야 하는 VP의 개수에 비례한다고 가정한다.

표 2 VC 교환기 또는 VP마다 고장이 한 번씩 일어나는 경우

(단위: 복구 메시지가 통과하는 VP 링크의 개수)

	Kawamura	IBVP	AIBVP	SIBVP
모든 VP에 장애가 한 번씩 발생하는 경우	3.12	5.79	3.12	3.12
모든 VC 교환기에 장애가 한 번씩 발생하는 경우	-	4.18	4.18	2.87
VP와 VC 교환기마다 장애가 한 번씩 발생하는 경우	-	5.47	3.33	3.07

표 2는 첫 번째 경우에 대한 결과로 망 내의 모든 VP와 VC 교환기에서 한 번씩의 장애를 발생시켜 복구 시간의 평균을 구했다. VC 교환기 장애를 복구할 수 없는 R. Kawamura의 방법에 대해서는 VP 장애만을 고려했다. 먼저 모든 VP에만 한 번씩의 장애가 발생하는 경우 AIBVP와 SIBVP 방법은 R. Kawamura의 방법과 같은 성능을 보였는데, 이는 두 방법이 VP 장애에

대해 R. Kawamura의 방법을 그대로 따르고 있으므로 당연한 결과이다. 한편, VC 교환기 장애만 발생하는 경우는 SIBVP 방법이 다른 방법들에 비해 우수한 성능을 나타내었다. IBVP와 AIBVP에서는 고장난 VC 교환기의 상대편 중단점이 복구를 수행하지만, SIBVP에서는 VP 연결점으로 사용되는 교환기가 복구를 수행할 수 있다고 가정했으므로 복구 시간이 짧아진다. 마지막으로 모든 VP와 모든 VC 교환기에 한 번씩의 장애가 발생하는 경우에는 IBVP 방법이 다른 두 방법에 비해 많은 복구 시간을 필요로 한다. 이것은 IBVP 방법이 VC 교환기 장애와 VP 장애에 대해 같은 방식의 복구를 수행함으로써 VP 장애를 복구하는 데 다른 방법에 비해 상대적으로 많은 시간을 필요로 하기 때문이다.

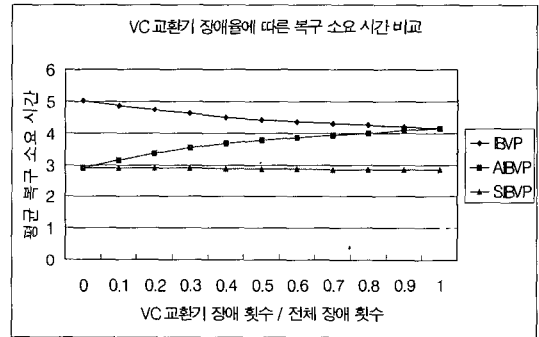


그림 12 복구 시간 비교

표 3 장애 형태에 따른 복구 가능성 비교

	Kawamura	IBVP	AIBVP	SIBVP
VC 교환기 장애	불가	가능	가능	가능
연결 VP의 동시 장애	가능	불가	가능	가능

그림 12는 실험의 두 번째 경우로 망에서 발생하는 전체 1000회의 장애 중에서 VC 교환기 장애가 차지하는 비율에 따라 변화하는 평균 복구 시간을 조사한 것이다. 그림의 가로축은 VC 교환기 장애가 전체 장애에서 차지하는 비율이고, 세로축은 평균 복구 시간이다. 예를 들어 VC 교환기 장애 비율이 0.2이면 1000회의 장애 중 VC 교환기 장애가 200회 발생한다는 뜻이다.

모든 장애가 VP 장애일 때, 다시 말해 VC 교환기 장애 비율이 0일 때는 IBVP 방법이 AIBVP와 SIBVP 방법의 거의 두 배에 달하는 복구 시간을 필요로 하였다. 이는 다른 두 방법이 VP 장애에 대해서는 별도로

R. Kawamura의 방법을 채택한 데 반해서 IBVP 방법은 알고리즘의 단순성을 유지하기 위하여 VP 장애에 대해서도 상대적으로 복구 시간이 긴 VC 교환기 복구 방법과 동일한 복구 과정을 거치기 때문이다. 또, IBVP와 SIBVP 방법에서는 전체 장애 중에 VC 교환기가 차지하는 비율이 증가할수록 평균 복구 시간이 짧아지는 반면에 AIBVP 방법에서는 VC 교환기 장애 비율이 증가할수록 필요한 복구 시간도 늘어난다. 이것은 표 1에서 볼 수 있듯이 다른 두 방법과 달리 AIBVP에서만 VC 교환기 장애 복구 시간이 VP 장애 복구 시간보다 크기 때문이다. 결론적으로 VC 교환기 장애 비율이 높을수록 SIBVP가 더 효율적이라고 할 수 있다. 그러나, VC 교환기 장애 비율이 낮을 경우는 SIBVP 방법의 구현상의 복잡성을 감안한다면 AIBVP 방법이 더 효과적일 수도 있다.

표 3은 각 복구 알고리즘을 복구 가능한 장애의 경우로 비교했다. 본 논문에서 제안한 방법들은 모두 VC 교환기 장애로 인해 단절된 가상 연결을 복구할 수 있지만, 그 중 IBVP 방법은 연결한 VP의 동시 고장을 복구할 수 없는 한계를 가진다.

표 4 60개의 VP를 가진 망에서의 BVP 개수

	Kawamura	IBVP	AIBVP	SIBVP
예비 가상 경로의 개수	60	228	288	288

표 5 필요한 VPI/VCI 테이블 엔트리 수 비교

	Kawamura	IBVP	AIBVP	SIBVP
VPI/VCI 엔트리 수	559.80	1430.40	1737.84	1737.84

마지막으로 제안한 방법들은 기존의 방법보다 많은 수의 BVP를 요구하는 것이 단점이다. 표 4는 60개의 VP를 가진 망에서 각각의 복구 방법에 따라 요구되는 BVP의 수를 나타낸 것인데, BVP 개수에 비례하여 VPI/VCI 테이블 엔트리의 수도 증가해야 한다. 하나의 VP가 N홉의 링크로 구성되었다고 하면 시작 노드까지 포함하여 총 (N+1)개의 노드가 이 VP를 위한 엔트리를 포함해야 하므로, 하나의 망이 가져야 하는 총 VPI/VCI 엔트리의 수는 다음과 같다.

$$(\text{총 VP의 수}) \times (\text{VP의 평균 홉 수} + 1) + (\text{총 BVP의 수}) \\ \times (\text{BVP의 평균 홉 수} + 1)$$

위의 식에 따르면 Kawamura의 방법의 경우 VP와 BVP의 수가 각각 60개이고, 그들의 평균 홉 수가 각각 3.15와 4.18이어서 필요한 전체 VPI/VCI 엔트리 수는 559.8개였다. 반면, 본 논문의 제안 방법에서는 표 5에 나타난 바와 같이 Kawamura 방법보다 2.5배 내지 3배의 엔트리를 저장해야 했다.

8. 결론

VP 수준에서 제안된 지금까지의 복구 방법들은 VC 교환기 장애에 대응할 수가 없었으나, 본 논문의 제안 방법들은 VP 장애는 물론 VC 교환기 장애까지 복구할 수 있다.

IBVP 방법은 알고리즘의 단순성을 유지하면서 VC 교환기 고장에 대응하려 했기 때문에 VP 장애에 대해서는 낮은 효율을 보인다. 하지만, 이 문제는 AIBVP 방법에서 R. Kawamura의 VP 장애 복구 방법을 도입함으로써 해결하였다. SIBVP 방법으로는 특별한 경우에 한하기는 했지만, VC 교환기 고장으로 인한 가상 연결 장애 복구 시간도 줄일 수 있었다.

실험을 통해 AIBVP와 SIBVP 방법은 R. Kawamura의 방법과 비교하여 같거나 더 나은 효율로 VC 교환기 복구까지 가능함을 알 수 있었다. 무엇보다도 본 논문에서 제안한 방법들을 이용하면, 현재 VC 교환기들처럼 고가의 회로를 중복해서 설치할 필요가 없기 때문에 VC 교환기의 비용을 크게 줄일 수 있다. 단, 제안한 방법들에서는 한 교환기가 쉘 진행 방향으로 이웃한 교환기의 VPI/VCI 테이블까지 알고 있어야 하므로, 더 많은 VPI/VCI 테이블 엔트리를 필요로 하는 단점이 있다.

또, 제안 방법은 하나의 가상 경로에 속하는 VC들이 서로 다른 BVP를 이용하여 복구가 될 경우도 있으므로 교환기에 유지해야 할 정보의 양이 많아진다. 이에 대한 해결책으로 인터넷 주소의 서브넷(subnet) 개념을 이용하여 각 VC마다의 정보를 별도로 테이블에 유지하지 않고, 같은 BVP를 사용하여 복구해야 하는 VC들의 VCI들이 공통된 시작값(prefix)을 갖도록 하는 방법을 생각할 수 있는데, 이것은 향후 과제로 남긴다.

참고 문헌

[1] ITU-T Draft Recommendation I.630. *ATM Protection Switching*. Jun. 1998.
 [2] E. Oki, N. Yamanaka, and F. Pitcho. Multiple-Availability-Level ATM Network Architecture. *IEEE Communications Magazine*, 33(9):80-88, Sep.

1995.

- [3] R. Kawamura and I. Tokizawa. Self-healing Virtual Path Architecture in ATM Networks. *IEEE Communications Magazine*, 33(9):72-79, Sep. 1995.
- [4] R. Kawamura, K. Sato, and I. Tokizawa. Self-healing ATM Networks Based on Virtual Path Concept. *IEEE Journal on Selected Areas in Communications*, 12(1):120-127, Jan. 1994.
- [5] T. H. Wu. *Fiber Network Service Survivability*. Artech House, 1992.
- [6] S. Ahn. A Fast VP Restoration Scheme using Ring-Shaped Sharable Backup VPs. In *Proceedings of IEEE GLOBECOM*, pages 1383-1387, 1997.
- [7] D. E. McDysan and D. L. Spohn. *ATM Theory and Application*. McGraw-Hill, 1994.
- [8] L. Nederlof, K. Struyve, C. O'Shea, H. Misser, Y. Du, and B. Tamayo. End-to-End Survivable Broadband Networks. *IEEE Communications Magazine*, 33(9):63-70, Sep. 1995.
- [9] The ATM Forum. *ATM User-Network Interface Signalling Specification Version 4.0*. 1996.
- [10] The ATM Forum. *Private Network-Network Interface Specification Version 1.0 (PNNI 1.0)*. 1996.
- [11] ITU-T Recommendation I.610. *B-ISDN Operation and Maintenance Principles and Functions*. Nov. 1996.
- [12] R. O. Onvural. *Asynchronous Transfer Mode Networks Performance Issues*. Artech House, 1995.
- [13] W. J. Goralski. *Introduction to ATM Networking*. McGraw-Hill, 1996.
- [14] U. Black. *ATM Foundation for Broadband Networks*. Prentice-Hall, 1995.
- [15] E. Biagioni, E. Cooper, and R. Sansom. Designing a Practical ATM LAN. *IEEE Network*, 7(2):32-39, Mar. 1993.
- [16] O. Kyas. *ATM Networks*. International Thomson Computer Press, 1995.
- [17] ITU-T Draft Revised Recommendation I.610. *B-ISDN Operation and Maintenance Principles and Functions*. Jun. 1998.



안 상 현

1986년 서울대학교 컴퓨터공학과 졸업.
1988년 서울대학교 컴퓨터공학과 석사.
1993년 University of Minnesota 전산학 박사. 1988년 ~ 1989년 데이콤 사원.
1994년 3월 ~ 1998년 2월 세종대학교 전산과학과 교수. 1998년 3월 ~ 현재 서울시립대학교 전산통계학과 교수



김 종 상

1960년 서울대학교 공과대학 전자공학학사. 1965년 서울대학교 공과대학 전자공학과 석사. 1975년 서울대학교 공과대학 전자공학학사. 1979년 ~ 현재 서울대학교 공과대학 컴퓨터공학부 교수. 1986년 ~ 1988년 한국정보과학회 회장. 1992년 ~ 현재 서울대학교 컴퓨터신기술 연구소 소장.



유 영 환

1996년 서울대학교 컴퓨터공학과 졸업.
1998년 서울대학교 컴퓨터공학과 석사.
1998년 ~ 현재 서울대학교 컴퓨터공학부 박사 과정. 관심분야는 ATM 네트워크, QoS 라우팅, 무선 통신