

실시간 ATM 망 관리 시스템 구현을 위한 성능 감시와 고장 노드 식별 방안

正會員 최 용 훈*, 이 길 흥*, 송 운 섭**, 이 준 호*,
이 재 용*, 이 상 배*

Performance Monitoring and Fault Node Identification Method for Real-Time ATM Network Management System

Yong-hoon Choi*, Kil-hung Lee*, Woon-seop Song**, Jun-ho Lee*,
Jai-yong Lee*, Sang-bae Lee* *Regular Members*

※본 연구는 한국과학재단의 특정기초 연구비 지원으로 수행됨

요 약

ATM은 다양한 서비스 품질을 가상 경로별로 보장해 주어야 하므로, 성능 관리가 매우 중요하다. 특히, 많은 양의 성능 정보들을 망 관리 시스템이 수집하여 신속한 판단을 내리는 것은 매우 어려운 일이다. 본 논문에서는 식별하고자 하는 고장 노드의 수에 따라 감시할 가상 경로를 선택하는 방안 (VP 선택 알고리즘)과, 선택된 가상 경로를 이용하여 고장 노드를 식별하는 법칙 (고장 식별 법칙)을 제안한다. 이 방안은 감시할 가상 경로의 수를 줄일 수 있으므로 신속한 고장 노드 식별을 가능하게 한다. 또한, Hard failure와 Soft failure가 발생한 노드를 식별할 수 있으므로, 망 내의 다양한 종류의 고장들을 인지해 내야 하는 망 관리 시스템의 성능 관리와 고장 관리 모듈에 유용하게 이용될 수 있다.

ABSTRACT

Performance management of ATM network is urgently required because a different quality of service should be guaranteed on each connection. A lot of performance measurement data increase the burden on agent systems and on management stations. In this paper, an effective OAM-based performance monitoring and faulty node

*연세대학교 전자공학과

**공군 본부 근무

論文番號:96375-1127

接受日字:1996年 11月 27日

identification technique is proposed. A proposed VP Selection Algorithm reduces management-related traffic and when the indication of hard or soft failure state is detected, failed node is identified by Fault Identification Rule.

I. 서 론

다양한 QoS를 보장하고, 광대역을 고속으로 서비스하는 ATM망이 안정적으로 운용되기 위해서는 신뢰성 있고 신속한 망의 성능 감시와 제어가 요구된다. ATM 망과 같은 대규모 망에 대한 적절한 성능 관리와 고장 관리를 수행하기 위해서는 관리 트래픽은 최소로 하면서 고장이나, 성능 저하의 원인과 위치를 신속하고 정확하게 파악할 수 있어야 한다. 이러한 목적을 달성하기 위해서 망 관리자는 망 내에 설정된 커넥션, 망 요소, 링크 등을 모니터링 하여야 하지만, 수천 개의 망 요소들로 구성되어 있는 ATM 망을 상세히 감시, 제어하는 것은 관리 트래픽의 증가로 인한 망의 성능 저하, 관리 정보의 폭주로 인한 신속한 제어력의 상실 등의 부작용을 유발할 수 있어 비효율적이다. 이러한 상반되는 관점에서 소기의 관리 목적을 달성하기 위해서 고속 망을 관리하는 망 관리 시스템은 최소한의 관리 트래픽으로 최대한의 관리 정보를 얻어낼 수 있어야 한다. 본 논문에서는 필요한 최소의 관리 트래픽으로 망 전체의 상황을 모니터링 할 수 있으면서도 모든 노드에서의 고장을 찾아낼 수 있는 방안을 제시하고자 한다.

망 내에서 발생하는 고장은 두 가지로 분류될 수 있다. 첫 째는 일시적인 혼잡이나, 망 요소의 오 동작으로 망이 동작을 계속하기는 하지만 그 성능이 기대치보다 떨어지는 현상인 Soft failure이고, 둘째는 망 구성 요소가 동작을 하지 않거나 상위 계층 프로토콜의 결함으로 연속성이 끊어진 경우인 Hard failure이다 [4]. Hard failure의 경우 고장이 발생한 망 요소 주변에서 다양한 고장 신호를 발생하므로 고장의 원인과 위치의 파악이 수월한 반면, Soft failure의 경우, 기대치를 약간 밑도는 수준에서 성능 저하가 발생하므로 이를 탐지해 내고 이에 대한 원인과 그 위치를 파악하는 것이 쉽지 않다.

망의 성능 관리와 고장 관리에 대한 연구들은 최근 들어 활발하게 진행되고 있다. 이중 Prem과 Venkataram [4]은 Realistic Abductive Reasoning Model을

이용하여 계층적 구조를 가지는 망에서의 Soft failure 문제를 다루었다. 관리 트래픽을 줄이는 방안에 대한 연구들도 다양하게 진행되어 왔는데 [5][6], 이중 Takano와 Fujita [5]는 관리 대상인 망에 레벨을 나누어서 망이 정상 동작 상태와 고장 상태인 때를 분리하여 감시하는 방안을 제시하였다. 고장 관리 분야에서는 추론 알고리즘과 knowledge base를 바탕으로 하는 전문가 시스템과 결부되어 많은 연구가 수행되어 왔으며 [14], 이러한 연구들의 대부분은 확률적인 접근을 기본으로 하고 있다. Hong과 Sen [7]은 이질적인 망들 사이에서 발생한 고장 문제를 확률적으로 다루었으며 Wang과 Schwartz [2]는 Dynamic-Routed Networks에서 고장 링크를 식별하는 방법을 확률적으로 다루었다. 그러나 Wang과 Schwartz의 연구는 Soft failure나 동시에 여러 개의 링크에서 발생한 고장은 식별할 수 없는 단점이 있다.

최근 들어서는 Bayesian network이 확률적인 분석에 기반을 둔 고장 진단 모델로 각광을 받고 있다 [9]. 이 방법은 고전적인 rule-based 모델에 비해 많은 장점을 가지고 있다. Deng [1]은 Linear Lightwave Networks (LLN)에서의 고장 진단 문제를 Bayesian network 모델을 이용하여 다루었다.

이러한 연구들은 대개 LLN과 같은 특정한 망 [1, 13]이나, 이질적인 망들의 연동 [7]에 있어서 발생하는 고장의 문제들을 주로 다루고 있으며, 고장도 대부분 Hard failure로 한정하고 있다. 따라서 커넥션마다 다른 QoS들을 보장해야 하는 ATM망에 이러한 연구들을 그대로 적용시키기에는 부적합하다. 본 논문에서는 ATM망에 있는 고유의 관리 기능인 Operation and Maintenance (OAM) 셀이 제공하는 관리 정보를 이용하여 ATM 망에서 반드시 고려해야 하는 Soft failure가 발생한 노드를 찾아내는 문제와 관리 트래픽을 줄이는 문제를 해결할 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 OAM 셀의 기본 원리와 동작을 설명하며 3장에서는 VP 선택 알고리즘과 고장 식별 법칙(Fault Identification Rule)을 제안하고, 간단한 예를 통해 이를 설명한다. 4장에

서는 3장에서 제안된 사항들을 종합하여, ATM망의 성능 감시와 고장 노드 식별을 행하는 방법을 제시한다. 5장에서는 모의 실험을 통해 제안된 사항들을 검증하며, 마지막으로 6장에서 결론을 맺는다.

II. OAM 셀의 동작과 MIB

ATM의 망 요소들은 OAM 셀을 이용하여 관리 정보를 교환할 수 있다. 이러한 통신 방법은 ATM망에 있는 고유한 방법이다[8]. OAM 셀을 이용하여 관리 정보를 교환할 경우 OAM 셀을 발생시킨 노드는 OAM 셀이 움직이는 가상 경로에 대한 정보를 얻을 수 있다. OAM 셀이 운반해 온 정보들은 성능, 고장에 관한 정보들이며 이러한 기능들에 대한 표준은 ITU-T 표준 I.610 [10]에 상세히 기술되어 있다. OAM 셀의 원리와 동작에 대한 자세한 설명은 참고문헌 [3, 10~12]를 참조할 수 있다. 본 논문에서는 ATM망의 성능 감시를 다루므로 이와 관련이 있는 성능 관리 OAM 셀에 초점을 맞춘다. ATM 계층의 성능 관리 OAM 셀은 종단간(end-to-end) 흐름과 세그먼트(segment) 흐름이 있으며 두개의 흐름 모두 forward monitoring과 backward reporting 기능을 가지고 있다. 우선 OAM 셀을 이용해서 관리 시스템이 망 관리를 행하기 위해서는 MIB를 관리 대행자 내에 구축해야 한다. MIB는 RFC 1155 SMI [15]를 사용하여 기술하였으며 참고문헌 [16~18]를 참조할 수 있다.

III. VP 선택 알고리즘과 고장 식별 법칙

3.1 가정과 기호

본 논문에서 사용되는 가정과 기호는 다음과 같다.

(가정)

- 관리 영역의 내부와 외부에 동시에 걸쳐서 존재하는 VP에 대해서는 고려하지 않는다.
- 임의의 VP에 대해서는 양방향으로의 경로(path)가 모두 존재한다고 가정한다.

(기호)

- $N = \{n_1, n_2, \dots, n_\alpha\}$: 망 내의 모든 노드들의 집합
- c_i : Virtual Path, Ordered set of ($n_j \in N, c_i = (1, 4, 6)$ 과 같은 방법으로 표현)

- $C = \{c_1, c_2, \dots, c_\beta\}$: 망 내의 모든 VP의 집합
- $D = \{c_k | c_k \in C, 1 \leq k \leq \beta\}$: 알고리즘을 통해 얻은 VP들의 집합.
- $N = \{n_e | n_e \in N, 1 \leq e \leq \alpha\}$: 알고리즘을 통해 얻은 노드들의 집합.
- $\phi(n_i)$: 노드 n_i 를 통과하거나 노드 n_i 에서 종료 또는 시작하는 VP의 개수.
- $\min[a_i | conditions]$: 조건(conditions)을 만족하는 a_i (대소 판단이 가능한 것이어야 한다.)들 중, 크기가 가장 작은 것을 취함.
- $c_i = 1$: VP i 가 고장 상태
- $c_i = 0$: VP i 가 정상 동작 상태
- $n_j = 1$: 노드 j 가 고장 상태
- $n_j = 0$: 노드 j 가 정상 동작 상태

3.2 예 제

그림 1과 같은 $N = \{n_1, n_2, n_3\}$, $C = \{c_1, c_2, c_3\}$ 인 망을 고려해 보자. 본 논문에서 고장은 hard failure와 soft failure 모두를 고려한다. n_1 에서 고장이 발생하면 $c_2 = 1, c_3 = 1$ 으로 모니터링될 것이며, n_1 을 지나지 않는 c_1 은 정상($c_1 = 0$)으로 모니터링될 것이다. 다시 말해, $c_2 = 1, c_3 = 1$ 로 관측되었다면, n_1 에서 고장이 발생했다고 할 수 있다. $c_1 = 1, c_3 = 1$ 으로 관측되었다면, n_2 가 고장이고, $c_1 = 1, c_2 = 1$ 으로 관측되었다면, n_3 가 고장이라고 할 수 있다. 좀더 정규화된 방식으로 표현하면, $n_1 = c_2 \wedge c_3, n_2 = c_1 \wedge c_3, n_3 = c_1 \wedge c_2$ 으로 표현된다. 만일, 망 내의 임의의 두 노드가 고장이라고 가정하면, $c_1 = 1, c_2 = 1, c_3 = 1$ 으로 관측될 것이며, 위의 표현 방법에 따르면 망 내의 3개의 노드를 모두 고장으로 잘못 판별하게 된다. 따라서 이 경우에 2개 이상의 고장에 대해서는 두 개 이상의 고장이 발생하였다는 사실만을 알 뿐, 고장 노드를 식별할 수는 없다.

다음으로, $N = \{n_1, n_2, n_3, n_4\}$, $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ 인 그림 2를 살펴보자. 이 경우에는 망내에서 발생한 2개의 고장에 대해서 그 고장 노드를 식별할 수 있다. $n_1 = c_2 \wedge c_3 \wedge c_5, n_2 = c_1 \wedge c_3 \wedge c_4, n_3 = c_1 \wedge c_2 \wedge c_6, n_4 = c_4 \wedge c_5 \wedge c_6$ 로 표현했을 때, n_1 과 n_2 가 고장이면, n_1 과 n_2 를 지나는 VP들인 c_1, c_2, c_3, c_4, c_5 에서 성능 저하가 관측될 것이며, n_1 과 n_2 를 지나지 않는 c_6 는 정상($c_6 = 0$)으로 관측될 것이다. c_6 로 인하여, n_3 와 n_4 는 정상 동작 상태임이 확인될 수 있다. 따라서, 망 내

에 임의의 2개의 노드에서 발생한 고장이라도 고장난 2개의 노드를 식별할 수 있다. 그러나 3개 이상의 고장에 대해서는 고장 노드들을 식별할 수 없다. 그림 3을 살펴보자.

그림 3은 그림 2를 c_6 가 n_1 까지 지나도록 변경시킨 것이다. 여기서 n_1 과 n_2 가 고장이면, n_1 과 n_2 를 지나는 $c_1, c_2, c_3, c_4, c_5, c_6$ 가 모두 성능 이상으로 관측될 것이므로 고장 노드들을 식별할 수 없게 된다. 그림 2와 그림 3의 차이를 살펴보자. 그림 2는 임의의 노드를 지나는 3개의 VP가 존재하며, 이 VP들은 그 노드 이외에는 다른 어느 노드에서도 서로 만나지 않는다. 그러나 그림 3은 n_4 를 지나는 VP중 c_5 와 c_6 가 n_1 에서 다시 만나고 n_3 를 지나는 VP중 c_2 와 c_6 가 n_1 에서 다시 만난다.

이와 같은 사실들을 종합해 보면, 2개의 동시에 발생한 고장에 대해서 고장 노드를 식별하고자 하면 망 내의 임의의 노드에서 그 노드를 제외하고는 다른 어느 노드에서도 만나지 않는 VP가 3개 이상 존재해야 한다. 그림 3에서 n_3 와 n_4 는 위의 조건을 만족시키지 못한다. 따라서 2개의 고장에 대해서 고장 노드를 식별할 수 없다. 이 사실을 좀더 일반적인 환경으로 확장하면, k 개의 고장에 대해서 고장 노드를 식별하고자 하면, 망 내의 임의의 노드는 그 노드를 제외하고는 다른 어느 노드에서도 만나지 않는 VP가 $k+1$ 개 이상 존재해야 한다. 본 장에서는 망 관리자가 위에서 언급한 조건을 만족시키도록 망의 VP들을 선택하는 방법과 고장 노드를 식별하는 일반적인 방법에 대해서 다룬다. 3.3절에서는 망 내에 존재하는 VP중 고장 노드를 식별하는데 필요한 VP를 선택하는 방법에 대해서 언급할 것이며, 3.4절에서는 선택된 VP들을

이용해 고장 노드들을 식별하는 고장 식별 법칙 (Fault Identification Rule)에 대해서 다룬다.

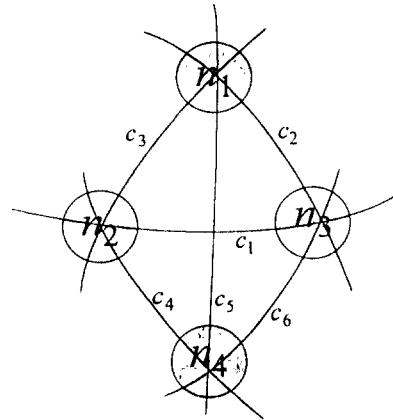


그림 2. four-node example

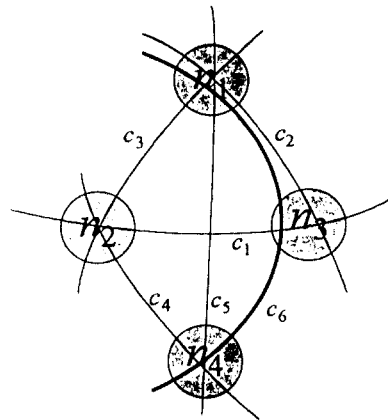


그림 3. four-node example

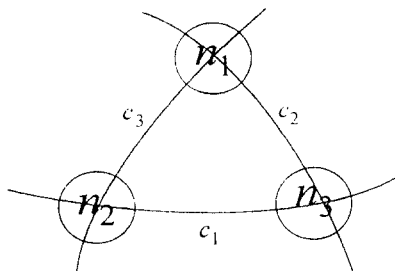


그림 1. A three-node example

3.3 VP 선택 알고리즘

망 내의 단일 고장에 대해서 고장 노드를 식별할 수 있기 위해서는 임의의 노드에 대해서 그 노드를 통과하는 VP는 그 노드를 제외하고는 다른 어느 노드에서도 만나지 않는 VP가 2개 이상 존재해야 한다. 아래의 알고리즘은 모든 노드가 위의 조건을 만족시키도록 하는 방법이다. 알고리즘은 다음과 같다.

<Algorithm 1: VP Selection for Single Fault Identi-

fication)

Step 1 $D = \emptyset, \bar{N} = \emptyset, M = \emptyset, T = \emptyset$ 으로 한다.

Step 2 $\phi(n_k) = \min[\phi(n_i) | \forall n_i \in N - M, 1 \leq i \leq \alpha]$ 으로 한다.

Step 3 if $\exists d_i, d_j \in D, (i \neq j)$ 에 대해서 $d_i \cap d_j = \{n_k\}$ 인 d_i 와 d_j 가 존재하면, Step 4로 간다.

else of $\exists d_i \in D, \exists c_i \in C - D$ 에 대해서 $d_i \cap c_i = \{n_k\}$ 을 만족하는 c_i 가 존재하면, 존재하는 모든 c_i 들을 T 의 원소로 한다.

$n(t_x) = \min[n(t_i) | \forall t_i \in T, 1 \leq i \leq \beta]$ 을 만족하는 $t_x \in T$ 를 D 의 원소로 한다.

$T = \emptyset$ 로 한다.

Step 4로 간다.

else if $\exists c_i, c_j \in C - D, (i \neq j)$ 에 대해서 $c_i \cap c_j = \{n_k\}$ 을 만족하는 c_i 와 c_j 가 존재하면, 존재하는 모든 c_i 와 c_j 에 대해서 $\{c_i, c_j\}$ 을 2^T 의 원소로 한다.

$n(t_x) + n(t_y) = \min[n(t_i) + n(t_j) | \forall \{t_i, t_j\} \in 2^T, 1 \leq i, j \leq \beta, i \neq j]$ 를 만족하는 $t_x, t_y \in T, (x \neq y)$ 를 D 의 원소로 한다.

$T = \emptyset$ 로 한다.

Step 4로 간다.

else n_k 를 \bar{N} 의 원소로 하고 Step 4로 간다.

Step 4 n_k 를 M 의 원소로 하고 $N = M$ 이 될 때까지 Step 2로 간다.

위의 알고리즘을 통해 얻어진 D 가 모니터링할 VP들의 집합이다 $\bar{N} = \emptyset$ 이면 망 내의 모든 노드들에서 발생한 단일 고장에 대해서 그 고장 노드를 식별할 수 있다. $n(\bar{N}) \geq 1$ 이면, $\forall n_i \in \bar{N}$ 인 노드들에 대해서는 고장 식별이 불가능하다. 알고리즘을 간략히 설명하면 다음과 같다. Step 1은 알고리즘에서 사용할 집합들에 대한 선언 및 초기화이다. Step 2에서는 VP가 가장 적게 지나는 노드를 찾는다. VP가 적게 지나는 노드 순서로 알고리즘을 수행하는 이유는 \bar{N} 에 속하는 노드를 최소화하기 위함이다. Step 3는 D 에 Step 2에서 선택된 노드의 고장을 식별할 수 있는 VP가 이미 존재하는 경우, D 와 $C - D$ 에 한 개씩 존재하는 경우, $C - D$ 에 두개 존재하는 경우, D 와 $C - D$ 를 모두 합쳐도 2개를 찾을 수 없는 경우로 나누어서 수행하는데, 가능한 VP중 길이가 가장 짧은 것을 찾아 D 의 원

소로 선택한다. Step 4에서는 종료할 조건을 명시했다. $\bar{N} = \emptyset$ 인 상태로 알고리즘이 끝나면, 망 내의 모든 노드들은 각 노드에서 그 노드를 제외하고는 다른 어느 노드에서도 만나지 않는 2개 이상의 VP가 존재한다. 이 알고리즘을 2개 이상의 고장 노드를 식별할 수 있도록 확장하려면, 3개 또는 그 이상의 VP를 추가했을 때, 추가된 VP와 기존에 D 에 존재하는 2개의 VP와 각각 교집합이 그 노드뿐인 VP를 Algorithm 1을 통해 얻어낸 D 에 추가하면 된다. K 개의 고장 노드를 식별하는 알고리즘은 다음과 같다.

(Algorithm 2: VP Selection for K Fault Identification)

Step 1 $\bar{N} = \emptyset, M = \emptyset, T = \emptyset, D = \{d_i | d_i \in D_{k-1}, 1 \leq i \leq \beta\}$ 으로 한다. (여기서, D_{k-1} 은 Algorithm 1을 통해 얻은 D 를 의미한다.)

Step 2 $\phi(n_k) = \min[\phi(n_i) | \forall n_i \in N - M, 1 \leq i \leq \alpha]$ 으로 한다.

Step 3 if $\forall d_i \in D$ 단, $\{n_k\} \subset d_i$ 를 만족하는 d_i 가 $K + 1$ 개 이상 존재하면 Step 4로 간다.

else if $\forall d_i \in D$ 단, $\{n_k\} \subset d_i$ 를 만족하는 d_i 가 모두 K 개 존재하고, $\exists c_i \in C - D$ 에 대해서 $d_i \cap c_i = \{n_k\}$ 을 만족하는 c_i 가 존재하면, 존재하는 모든 c_i 들을 T 의 원소로 한다.

$n(t_x) = \min[n(t_i) | \forall t_i \in T, 1 \leq i \leq \beta]$ 을 만족하는 $t_x \in T$ 를 D 의 원소로 한다.

$T = \emptyset$ 로 한다.

Step 4로 간다.

else n_k 를 \bar{N} 의 원소로 하고 Step 4로 간다.

Step 4 n_k 를 M 의 원소로 하고 Step 4로 간다.

Algorithm 1의 경우와 마찬가지로 D 가 K 개의 고장 노드를 식별하기 위해 모니터링해야 할 VP들의 집합이며, $\bar{N} = \emptyset$ 이면, 망 내의 모든 노드에서 K 개의 고장 노드를 식별할 수 있다. $n(\bar{N}) \geq 1$ 경우에는 $\forall n_i \in \bar{N}$ 인 노드들은 K 개의 고장에 대해 고장 발생 노드들을 식별할 수 없다. 본 논문에서는 VP선택 알고리즘을 통해 얻어낸 $\forall c_i \in D$ 와 $\forall n_i \in N - \bar{N}$ 로 구성된 망을 Reduced Network으로 명명한다.

3.4 고장 식별 법칙 (Fault Identification Rule)

Reduced Network에서 임의의 노드를 지나는 VP들

이 모두 성능 이상으로 관측되면, 그 노드는 고장이 다. 반면, 임의의 노드를 지나는 하나의 VP라도 정상으로 관측되면 그 노드는 정상 동작 상태이다. 이는 아래와 같이 표현할 수 있다.

<고장 식별 법칙 (Fault Identification Rule)>

$\exists n_k \in N - \bar{N}$ 에 대해 $\{n_k\} \subset d_i$ 인 d_i 들을 모두 찾아낸다. 찾아낸 모든 d_i 를 $d_1, d_2, d_3, \dots, d_r \in D$ 라 했을 때, n_k 에 대한 고장 식별 법칙은 아래와 같다.

$$n_k = (d_1 \wedge d_2 \wedge \dots \wedge d_r)$$

3.5 예제 : VP 선택 알고리즘, 고장 식별 법칙

3.5.1 1개의 고장 노드 식별

그림 4는 $N = \{n_1, n_2, \dots, n_7\}$, $C = \{c_1, c_2, \dots, c_{12}\}$ 로 구성된 망이다. 그림 4에 대해서 Algorithm 1을 수행하여 보자. $\phi(n_4) < \phi(n_1), \phi(n_5), \phi(n_6) < \phi(n_2), \phi(n_3), \phi(n_7)$ 이므로 Step 2에서 n_4 가 선택된다. Step 3에서는 n_4 에 대해 D의 원소로서 $c_7 = (1, 4, 7), c_8 = (2, 4, 6), c_9 = (3, 4, 5)$ 중 2개가 선택된다. 여기서는 c_7 과 c_8 을 선택하였다. 다음으로 Step 2에서 n_1, n_5, n_6 의 순서로 선택하여 알고리즘을 수행하였다. n_1 에 대하여 $c_1 = (1, 2, 5), c_2 = (1, 3, 6), c_3 = (3, 1, 2)$ 중 어느 하나가 D의 원소로 선택된다. 여기서는 c_1 을 D의 원소로 선택하였다. n_5 에 대하여는 $c_5 = (6, 7, 5)$ 나 $c_9 = (3, 4, 5)$ 중 어느 하나가 D의 원소가 될 자격이 있다. $c_6 = (2, 5, 7)$ 는 이미 D의 원소로 존재하는 c_1 에 대해서 2개의 원소를 교집합으로 가지므로 $(c_1 \cap c_6 = \{2, 5\})$ D의 원소가 될 수 없다. 여기서는 c_5 를 D의 원소로 선택하였다. n_6 에 대해서는 $c_5 \cap c_8 = \{6\}$ 을 만족하는 $c_5 = (6, 7, 5), c_8 = \{2, 4, 6\}$ 가 이미 D의 원소로서 존재하므로 D에 추가되는 VP는 없다.

그 다음으로는 Step 2에서 n_2, n_3, n_7 중 어느 하나가 선택될 수 있으며 여기서는 n_2, n_3, n_7 순서로 선택하여 알고리즘을 수행하였다. n_2 에 대하여 $c_1 \cap c_8 = \{2\}$ 를 만족하는 c_1 과 c_8 가 이미 D의 원소로서 존재하므로 D에 추가되는 VP는 없다. n_3 에 대해서는 $\{c_2, c_9\}, \{c_2, c_{10}\}, \{c_2, c_{12}\}, \{c_3, c_4\}, \{c_3, c_9\}, \{c_3, c_{12}\}, \{c_4, c_9\}, \{c_4, c_{10}\}, \{c_9, c_{10}\}, \{c_9, c_{12}\}$ 가 2^T 의 원소로 선택되며, 이 중 $n(c_{10}) + n(c_{12})$ 가 최소이므로 c_{10} 과 c_{12} 가 D의 원소로 추가된다. 마지막으로, $n(c_{10})$ 에 대해서는 $c_5 \cap c_7$

$= \{7\}$ 이 이미 D의 원소로 존재하므로 $(c_6 \cap c_7 = \{7\}, c_7 \cap c_{12} = \{2\})$ D에 추가되는 VP는 없다. 위의 과정을 통해 $D = \{c_1, c_5, c_7, c_8, c_{10}, c_{11}, c_{12}\}, \bar{N} = \emptyset$ 을 얻을 수 있다.

그림 5는 VP 선택 알고리즘을 수행하여 얻은 노드들과 VP들로 이루어진 Reduced network이다. Reduced network을 이용해 망 내에서 발생한 1개의 고장에 대해서 고장 노드를 식별할 수 있다. 표 1은 고장 식별 법칙을 Reduced network내의 모든 노드에 적용시켜 얻은 테이블이다. 본 논문에서는 이 테이블을 고장 식별 테이블이라고 명명한다.

3.5.2 2개의 고장 노드 식별

그림 4에 대해 Algorithm 2를 수행하여 보자. Step 1에서 $D = \{c_1, c_5, c_7, c_8, c_{10}, c_{12}\}, \bar{N} = \emptyset, M = \emptyset, T = \emptyset$ 으로 초기화되며, 앞 절에서와 같은 순서대로 노드를 택하여 알고리즘을 수행하면, Step 2에서 n_4 가 선택되고 Step 3에서는 D의 원소 중 c_7 과 c_8 이 c_9 과는 교집합으로 n_4 만을 가지므로 $c_9 = (3, 4, 5)$ 가 D의 원소로 추가된다. 다음으로, n_1 에 대하여는 D의 원소인 c_1 과 c_7 이 c_2 와는 교집합으로 n_1 만을 가진다. 따라서 $c_2 = (1, 3, 6)$ 이 D의 원소로서 존재한다. 따라서, 더 이상 D의 원소로 추가되는 VP는 없다. 알고리즘을 통해 $D = \{c_1, c_2, c_5, c_7, c_8, c_9, c_{10}, c_{12}\}, \bar{N} = \emptyset$ 을 얻게되며, 그림 6은 2개의 고장 노드를 식별하기 위한 Reduced network이며, 표 2는 고장 식별 테이블이다.

그림 4 모델은 망 내에서 발생한 3개 이상의 고장에 대해서는 고장 노드들을 식별할 수 없다. 3개 이상의 고장에 대해서 고장 노드들을 식별하기 위해서는 추가의 VP들이 요구된다.

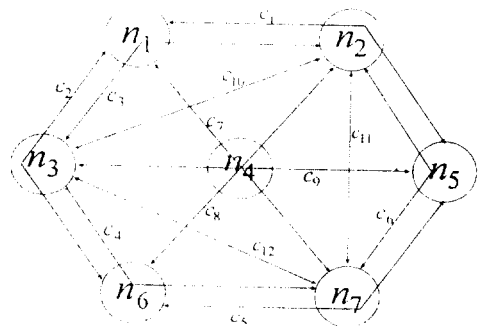


그림 4. A seven-node example

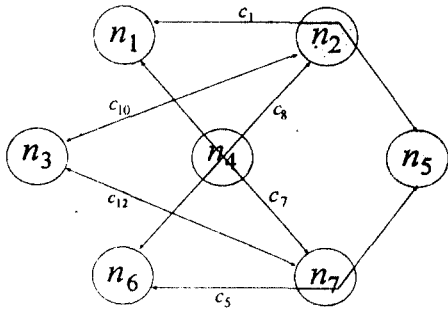


그림 5. 1개의 고장 노드 식별을 위한 Reduced Network

4	c_7, c_8, c_9	$n_3 = c_7 \wedge c_8 \wedge c_9$
5	c_1, c_5, c_9	$n_3 = c_1 \wedge c_5 \wedge c_9$
6	c_2, c_5, c_8	$n_3 = c_2 \wedge c_5 \wedge c_8$
7	c_5, c_7, c_{12}	$c_7 = c_5 \wedge c_7 \wedge c_{12}$

IV. 성능 감시를 통한 고장 노드 식별

전송 지연을 모니터링하기 위해서는 OAM MIB의 TSTP(f)와 TSTP(b)를 검색한다. 관리 객체들 중 TSTP(f)는 모니터링 OAM 셀이 전방향으로 삽입될 때의 time stamp 값이며, TSTP(b)는 OAM 셀이 후방향으로 삽입될 때의 time stamp 값이다. 따라서 이 두 값의 차가 전송 지연이 된다. 이 두 객체를 검색하기 위한 GetBulkRequest PDU의 variable-bindings 필드는 객체 TSTP(f)와 TSTP(b)를 포함한다. 전송 지연은 아래와 같이 계산된다.

$$\text{전송 지연: TSTP(b) - TSTP(f)}$$

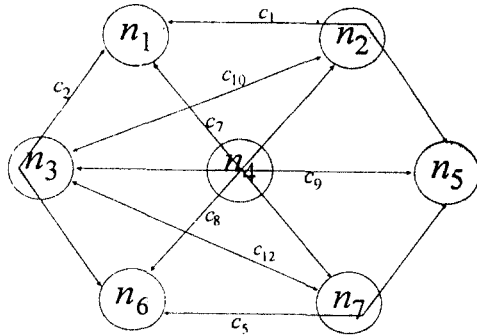


그림 6. 2개의 고장 노드 식별을 위한 Reduced Network

표 1. 고장 식별 테이블: 1개의 고장 노드 식별

노드	$\{n_j\} \subset c_k$ (단, $c_k \in D$)인 c_k	고장 식별 법칙
1	c_1, c_7	$n_1 = c_1 \wedge c_7$
2	c_1, c_8, c_{10}	$n_2 = c_1 \wedge c_8 \wedge c_{10}$
3	c_{10}, c_{12}	$n_3 = c_{10} \wedge c_{12}$
4	c_7, c_8	$n_4 = c_7 \wedge c_8$
5	c_1, c_5	$n_5 = c_1 \wedge c_5$
6	c_5, c_8	$n_6 = c_5 \wedge c_8$
7	c_5, c_7, c_{12}	$c_7 = c_5 \wedge c_7 \wedge c_{12}$

표 2. 고장 식별 테이블: 2개의 고장 노드 식별

노드	$\{n_j\} \subset c_k$ (단, $c_k \in D$)인 c_k	고장 식별 법칙
1	c_1, c_2, c_7	$n_1 = c_1 \wedge c_2 \wedge c_7$
2	c_1, c_8, c_{10}	$n_2 = c_1 \wedge c_8 \wedge c_{10}$
3	c_2, c_9, c_{10}, c_{12}	$n_3 = c_2 \wedge c_9 \wedge c_{10} \wedge c_{12}$

셀 손실과 셀 오삽입을 모니터링하기 위해서 OAM MIB의 backward reporting table의 관리 객체들을 검색한다. 관리 객체들 중 TUC(0), TUC(0+1)은 OAM 셀이 forward monitoring을 시작할 때의 Monitoring Block(MB)내의 사용자 셀의 개수이며, TRCC(0), TRCC(0+1)은 피드백 노드가 수신한 MB내의 사용자 셀의 개수이다. 따라서 TRC1(0)-TUC(0)는 CLP 비트가 0인 사용자 셀의 셀 손실/셀 오삽입의 개수이며 TRCC(0+1)-TUC(0+1)은 CLP 비트가 0과1인 사용자 셀의 셀 손실/셀 오삽입의 개수이다. 셀 손실 및 오삽입 개수는 아래와 같이 계산된다.

$$\begin{aligned} \text{셀 손실(CLP=0+1) 개수: TRCC(0+1) - TUC(0+1)} \\ \text{--- 음수이면 오삽입 셀 개수} \\ \text{셀 손실(CLP=0) 개수: TRCC(0) - TUC(0) --- 음수이면 오삽입 셀 개수} \end{aligned}$$

OAM MIB의 관리 객체들 중 BLER(06+1)은 MB내의 에러 블록의 개수이다. 에러 블록의 개수는 BLER(0+1)로부터 직접 얻어낼 수 있다. 그림 7은 성능 감

시와 고장 진단의 순서도이다.

에러 블럭 개수: BLER(0 + 1)

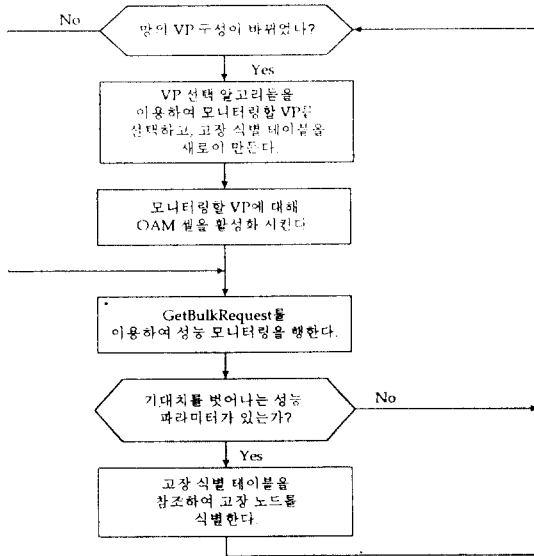


그림 7. 망의 성능 감시와 고장 진단 순서도

V. 모의 실험

5.1 모의 실험 모델과 조건

본 장에서는 그림 4의 망 모델에 대하여 모의 실험을 수행하여 고장 식별 법칙을 검증한다. 3장에서 언급했던 가정들은 모의 실험에도 역시 적용된다. 망 내의 모든 VP들은 양방향으로 경로가 존재하므로 VP의 양쪽 종단 점 중 한 곳에서 OAM 셀을 발생시키면 된다. 실험에서는 $c_i = (a, b, c)$ 로 표기된 VP에 대해 노드 a 에서 노드 c 로 모니터링 OAM 셀을 삽입하며, 노드 c 에서 노드 a 로 리포팅 OAM 셀을 삽입한다. c_i 의 성능 정보는 노드 a 의 data table에 기록되므로, 노드 a 의 MIB를 검색하면, c_i 의 성능 정보를 얻을 수 있다. 실험에서 모니터링하는 성능 파라미터는 전송 지연과 셀 손실이며, Soft failure가 발생하면, 고장 노드를 식별한다.

실험에서는 $c_1 \sim c_9$ 까지의 VP에는 평균 inter-arrival time이 0.5인 Poisson 분포로 셀을 발생시키고, $c_{10} \sim c_{12}$ 까지의 VP에는 평균 inter-arrival time이 2.5인

Poisson 분포로 셀을 발생시킨다. 모든 노드에서의 서비스 시간은 0.1 단위 시간으로 하였으며, 각 노드에서 queuing capacity는 500으로 설정해 500개의 사용자 셀을 수용할 수 있도록 한다. 또한 모든 VP에서 링크 지연은 0.1단위 시간으로 했으며 전송 지연은 링크 지연과 서비스 시간의 합으로 계산하였다. 표 3에 모의 실험 조건을 요약해 놓았다.

5.2 1개의 고장 노드 식별

3.5.1절에서는 망 내에서 발생한 1개의 고장에 대해 고장 노드를 식별하기 위해 VP 선택 알고리즘을 수행하여 Reduced network (그림 5)와 고장 식별 테이블 (표 1)을 얻었다. VP 선택 알고리즘을 통해 얻은 6개의 VP에 대해서는 MIB의 Control table에 새로운 row를 생성시킴으로써 OAM 셀을 활성화시킨다. OAM 셀이 활성화된 이후에는 MB 마다 Data table에 row가 하나씩 추가된다. 망 내의 노드들은 각각 자신이 발생시킨 OAM 셀이 피드백 되어 실어오는 성능 정보를 가지고 있으므로 관리자는 7개의 노드 중 OAM 셀을 발생시킨 노드들만 폴링한다. 이 모델에서는 n_1, n_2, n_3, n_6 가 OAM 셀을 발생시키는 노드이다. 실험에서는 $t = 1400$ 에서 노드 4의 서비스 시간을 0.1에서 0.18로 높여 주어 Soft failure가 발생되도록 하였으며, 셀 손실은 버퍼 오버플로우에 의한 손실을 측정하였다. $c_1 \sim c_9$ 는 사용자 셀이 평균 0.5 단위 시간마다 하나씩 생성되고, MB가 1000이며 $c_{10} \sim c_{12}$ 는

표 3. 모의 실험 조건

항목	값
노드의 수	7
VP의 수	12
MB	
$c_1 \sim c_9$	1000
$c_{10} \sim c_{12}$	200
노드의 서비스 시간	0.1
링크 지연	0.1
사용자 트래픽	Poisson
$c_1 \sim c_9$	평균 inter-arrival time: 0.5
$c_{10} \sim c_{12}$	평균 inter-arrival time: 2.5
Queue capacity	500

사용자 셀이 평균 2.5 단위 시간마다 하나씩 생성되고, MB가 200이므로 두 경우 모두 대략 500 단위 시간마다 OAM 정보가 Data table에 기록된다. 그림 8, 9는 forward monitoring table의 TSTP와 backward reporting table의 TSTP, TUC(0), TUC(0+1), TRCC(0), TRCC(0+1)의 값을 검색하여 전송 지연과 셀 손실을 매 주기마다 계산한 것이다.

그림 8에서 보듯이 고장이 발생한 1400 단위 시간부터 c_7 , c_8 의 전송 지연이 점차 증가하여, $t=3000$ 에서는 90단위 시간 정도로 수렴한다. 셀 손실의 경우(그림 9)도 전송 지연과 마찬가지로 고장 노드인 n_4 를 통과하는 c_7 , c_8 에서만 관측되었으며 이 노드를 통과하지 않는 c_1, c_2, c_{10}, c_{12} 는 셀 손실이 관측되지 않았다.

이와 같은 사실을 근거로 $c_1=0, c_5=0, c_7=1, c_8=1, c_{10}=c_{12}=0$ 로 하면 표 1에 의해서

$$\begin{aligned} n_1 &= (c_1=0) \wedge (c_7=1) \\ n_2 &= (c_1=0) \wedge (c_8=1) \wedge (c_{10}=0) \\ n_3 &= (c_{10}=0) \wedge (c_{12}=0) \\ n_4 &= (c_7=1) \wedge (c_8=1) \\ n_5 &= (c_1=0) \wedge (c_5=0) \\ n_6 &= (c_5=0) \wedge (c_8=1) \\ n_7 &= (c_5=0) \wedge (c_7=1) \wedge (c_{12}=0) \end{aligned}$$

을 얻는다. 따라서 $n_4=1$ 이고 나머지 노드들은 모두 0이므로 n_4 가 고장임을 판단할 수 있다.

5.3 2개의 고장 노드 식별

2개의 고장 노드를 식별하기 위해 3.5.2절에서는 Algorithm 2를 수행하였다. 이를 수행하여 얻어낸 Reduced network과 고장 식별 테이블은 각각 그림 6과 표 2이다. 실험에서는 Reduced network에 새로이 추가된 c_2 와 c_9 에 대해서 Control table에 새로운 row를 추가하여 OAM 셀을 활성화 시켰다. $t=1400$ 에서 노드 1과 노드 7의 서비스 시간을 0.1에서 0.13으로 높여 주어 두개의 노드에서 동시에 Soft failure가 생기도록 하였다. 그림 9, 10은 forward monitoring table의 객체인 TSTP와 backward reporting table의 객체인 TSTP, TRCC(0), TRCC(0+1), TUC(0), TUC(0+1)의 값을 이용하여 전송 지연과 셀 손실을 계산한 것

을 매 MB마다 보여준다.

그림 9에서 보듯이 고장이 발생한 1400 단위 시간부터 n_1 과 n_7 을 지나는 VP들인 $c_1, c_2, c_5, c_7, c_{12}$ 의 전송지연이 증가한다. 이중 c_7 의 전송 지연이 특히 높은 이유는 c_7 이 고장이 발생한 노드인 n_1 과 n_7 모두를 지나기 때문이다. 고장 노드를 지나지 않는 VP인 c_8, c_9, c_{10} 의 전송 지연은 증가하지 않는다. 셀 손실의 경우도 그림 10에서와 같이 $c_1, c_2, c_5, c_7, c_{12}$ 에서 관찰된다. 감시한 결과를 토대로, $c_1=1, c_2=1, c_5=1, c_7=1, c_8=0, c_9=0, c_{10}=0, c_{12}=1$ 로 하면 표 2에 의해,

$$\begin{aligned} n_1 &= (c_1=1) \wedge (c_2=1) \wedge (c_7=1) \\ n_2 &= (c_1=1) \wedge (c_8=0) \wedge (c_{10}=0) \\ n_3 &= (c_2=1) \wedge (c_9=0) \wedge (c_{10}=0) \wedge (c_{12}=1) \\ n_4 &= (c_7=1) \wedge (c_8=0) \wedge (c_9=0) \\ n_5 &= (c_1=1) \wedge (c_5=1) \wedge (c_9=0) \\ n_6 &= (c_1=1) \wedge (c_5=1) \wedge (c_8=0) \\ n_7 &= (c_5=1) \wedge (c_7=1) \wedge (c_{12}=1) \end{aligned}$$

를 얻을 수 있다. 고장 식별 법칙에 의해 n_1 과 n_7 이 고장임을 알 수 있다.

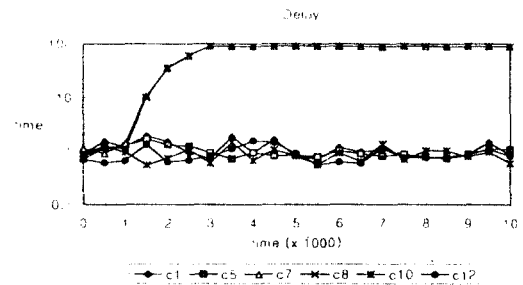


그림 8. 전송 지연-single fault

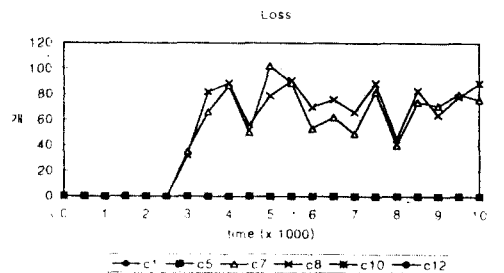


그림 9. 셀 손실-single fault

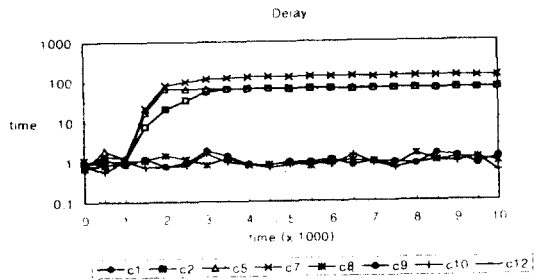


그림 10. 전송 지연-double fault

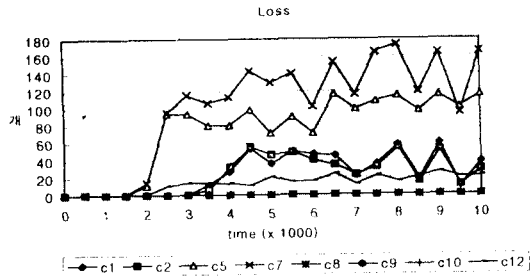


그림 11. 셀 손실-double fault

VI. 결 론

ATM망에서 성능 관리 OAM 셀이 활성화된 VPC/VCC에 대해서는 전송 지연, 셀 손실(율)/셀 오삽입(율), 에러 블럭의 수 등의 성능 정보를 알 수 있다. 링크에 고장(Hard failure)이 발생하면, 고장 발생 링크에 인접한 노드에서 중단 점으로 Alarm Indication Signal (AIS)셀을 발생시키며, AIS셀을 수신한 중단 노드는 반대편 중단 점으로 Remote Defect Indication (RDI)셀을 발생시켜 고장이 발생하였음을 중단 사용자에게 알린다. 그러나 노드에서 고장이 발생한 경우에는 그 증상이 다양하게 나타나는데, 이중, 성능 저하를 수반하는 Soft failure가 발생한 경우에는 이를 Alarm과 같은 event신호로 중단 사용자나 망 관리자에게 알려주는 방법이 없다. 따라서 event를 발생시키지 않는 Soft failure를 탐지하기 위해서 망 관리자는 망 내의 VP/VC를 모니터링할 수밖에 없다.

성능 관리 OAM 관리 정보 베이스는 OAM 셀이 운반해온 성능 정보를 망 관리자가 검색하여 망의 성능 관리에 응용할 수 있도록 하는 역할을 한다. ATM

망은 대규모 망이기 때문에, 망 내에 존재하는 모든 VPC/VCC에 대해서 OAM 셀을 활성화시키고 이들을 전부 모니터링하면 관리 정보가 많아지기 때문에 신속한 판단 및 제어가 어려워질 뿐 아니라, 오히려 관리 트래픽이 망의 부하를 가중시키는 악영향을 미칠 수 있다. 따라서 망 내의 모든 VPC/VCC들을 모니터링하지 않고도 망 내에서 발생하는 Soft failure와 Hard failure들을 인지해 내고, 이들이 발생한 노드를 식별할 수 있는 방안이 필요하다. 본 논문에서 제안했던 VP 선택 알고리즘과 고장 식별 법칙은 이러한 문제를 해결하는 방안 중 하나가 될 것이다. VP 선택 알고리즘은 “몇 개의 고장 노

드를 찾아낼 것인가?”라는 관리자의 요구에 따라 다르게 수행될 수 있다. 모의 실험 모델에서는 2개의 고장 노드를 찾아낼 수 있었다. 만일 망의 VP구성이 관리자가 원하는 개수의 고장 노드를 찾을 수 없게 되어 있다면, 관리자는 VP 선택 알고리즘을 수행하여 얻어낸 N 에 속하는 노드들에 새로운 VP를 개설하면 원하는 개수의 고장 노드를 찾아낼 수 있을 것이다. 고장 식별 법칙은 OAM 셀이 운반해온 성능 정보를 이용하여 고장 노드를 찾는 방법이다. 이 방법은 간단한 논리 AND연산 수행으로 고장 노드를 식별할 수 있으나 임의의 노드를 지나는 모든 VP에서 일시적인 혼잡이 발생했을 때, 그 노드를 고장이라고 판단하는 오류를 빚을 수도 있다. 적어도 2이상의 MB 동안 성능 이상이 감지된 VP만을 0으로 설정하여 논리 AND연산을 수행하는 것은 이러한 오류를 피하는 방법 중 하나가 될 것이다.

참 고 문 헌

1. Robert H. Deng et al, "A Probabilistic Approach to Fault Diagnosis in Linear Lightwave Networks," IEEE J. Select. Areas Commun., vol. 11, No. 9, pp. 1438~1448, December 1993.
2. Clark Wang and Mischa Schwartz, "Identification of Faulty Links in Dynamic-Routed Networks," IEEE J. Select. Areas Commun., vol. 11, No. 9, pp. 1449~1460, December 1993.
3. Dominique Gaiti and Guy Pujolle, "Performance Management Issues in ATM Networks: Traffic and

Congestion Control," IEEE/ACM Trans. Networking, vol. 4, No. 2, pp. 249~257, pril 1996.

4. G. Prem Kumar and P. Venkataram, "Network Performance Management Using Realistic Abductive Reasoning Model," in Integrated Network Management IV, Chapman & Hall, pp. 187~198, 1995.

5. Makoto Takano and Katsumi Fujita, "Multilevel Network Management by means of System Identification," in Proc. IEEE INFOCOM95, pp. 538~544, 1995.

6. S. Ahn and J. Chung, "The Design of the Shuttle Protocol with Network Management Data Gathering," in Proc. IEEE ISNOM'96, pp. 180~185, April 1996.

7. P. Hong and P. Sen, "Incorporating nondeterministic reasoning in managing heterogeneous network faults," in Integrated Network Management II, Elsevier Science, 1991, pp. 481~492.

8. Henry J. Fowler, "TMN-based Broadband ATM Network Management," IEEE Comm. Mag., pp. 74~79, March 1995.

9. P. Cheeseman, "A method of computing generalized Bayesian probability values for expert systems," in Proc. 8th Int'l Joint Conf. Art. Intell., Los Angeles, CA, 1983.

10. ITU-T Draft Standards I.610, "B-ISDN Operation and Maintenance Principles and Functions," November 1994.

11. Stephen C. Farkouh, "Managing ATM-based Broadband Networks," IEEE Commun. Mag. vol. 31 No. 5, pp. 82~86, May 1993.

12. Thomas M. Chen et al, "Monitoring and Control of ATM Networks Using Special Cells," IEEE Network Mag. pp. 28~38, September/October 1996.

13. T. E. Stern, "Linear lightwave networks: How far can they go?," in Proc. IEEE Globecom, San Diego, CA, Dec. 2-5, 1990, pp. 1866~1872.

14. E. C. Ericson et al, Eds, Expert Systems Applications in Integrated Network Management. Boston, MA: Artech, 1989.

15. M. Rose and K. McCloghrie, "Structure and Identification of Management Information," RFC 1155, May 1990.

16. Yong-hoon Choi et al, "OAM-based ATM Network Management Information Architecture: SNMP Agent," in Proc. IEEE ISNOM'96 pp. 161~166, April 1996.

17. 최용훈, "실시간 ATM 망 관리 시스템 구현을 위한 성능 감시와 고장 노드 식별 방안," 연세대학교 석사학위논문, 1996. 12.

18. 이재용, "ATM 망 관리 시스템 개발," 중간 보고서 1996. 7



최 용 훈(Yong-hoon Choi) 정회원
1972년 2월 5일생
1991년~1995년:연세대학교 전자공학과 졸업
1995년~1997년:연세대학교 전자공학과 대학원(공학 석사)
1997년~현재:연세대학교 전자공학과 박사과정 재학중

※주관심 분야: Network management, Satellite commun.



이 길 흥(Kil-hung Lee) 정회원
1961년 11월 27일생
1985년~1989년:연세대학교 전자공학과 졸업
1989년~1991년:연세대학교 전자공학과 대학원(공학 석사)
1991년~1995년: LG 정보통신
1995년~현재:연세대학교 전자공학과 박사과정 재학중

※주관심분야: Network management, Satellite commun., Internet

송 운 섭(Woon-seop Song)

정회원

현재: 공군 본부 근무

한국통신학회논문지 제22권 3호 참조

이 준 호(Jun-ho Lee)

정회원

현재: 연세대학교 전자공학과 Post doc.

한국통신학회논문지 제20권 12호 참조



이재용(Jai-yong Lee) 정회원

1955년 3월 5일생

1977년:연세대학교 전자공학과 졸업

1977년~1982년:국방과학연구소 연구원

1984년:미국 Iowa State University, Computer Eng. (공학석사)

1987년:미국 Iowa State University, Computer Eng. (공학박사)

1987년~1994년:포항공과대학 전자계산학과 교수

1994년~현재:연세대학교 전자공학과 교수

※주관심 분야:Network management, Protocol engineering, Satellite commun., Wireless ATM

이상배(Sang-bae Lee)

정회원

1954년~1958년:공군사관학교 졸업

1951년~1961년:서울대학교 전자공학과 졸업

1962년~1964년:미국 Stanford University (공학석사)

1961년~1967년:공군사관학교 조교수

1967년~1969년:원자력 연구소 연구관

1969년~1979년:서울대학교 전자공학과 조교수

1972년~1975년:영국 Newcastle University(공학박사)

1979년~현재:연세대학교 전자공학과 교수

※주관심분야:Network management, Computer networks, ATM/B-ISDN