

## 二元 트래픽의 呼損率 計算

正會員 金 承 煥\* 正會員 成 檀 根\*\* 正會員 朴 鎮 秀\*\*\*

# Calculation of Blocking Probabilities in the Two-dimensional Multi-slot Connection Traffic

Seung Hwan KIM\*, Dan Keun SUNG\*\*, Jin Soo PARK\*\*\* *Regular Members*

**要 約** 二元 트래픽의 呼損率 計算을 위한 네가지의 알고리즘이 提示되고 乘除 計算의 複雜度, 所要 메모리, 計算可能한 채널數등이 서로 比較된다. 처음 세가지의 알고리즘은 오버플로우나 언더플로우 問題때문에 計算 채널數의 上限線이 制限되는 反面에 새로 提示한 알고리즘은 이러한 問題點을 解決할 수 있어 10000 채널 以上의 大容量 시스템의 呼損率 計算도 可能하다. 이러한 알고리즘을 利用하여 1 스톱呼와 6 스톱呼인 境遇에 대하여 각 呼損率을 計算하여 檢討해 보며, 또한 SLAM-II의 시뮬레이션을 통하여 計算된 結果와 比較하여 본다.

**ABSTRACT** Four computational algorithms are discussed and compared in terms of complexity, required memory, and the number of channels which calculate the blocking probabilities in the two-dimensional multi-slot connection traffic.

The overflow and underflow problems can be circumvented, and thus the blocking probabilities can be calculated even up to 10000 channels, by the proposed last algorithm. This algorithm is utilized in calculating the blocking probabilities for 1-slot and 6-slots calls. And the results of the simulation by the SLAM-II are compared with those of the analytic calculation.

### I. 序 論

현재 디지털網은 64kb/s의 基本速度로 網이 構成되어 音聲을 서비스하고 있는데, 綜合情報通信網(ISDN: Integrated Services Digital Networks)이 實現되어감에 따라 디지털網의 主 서비스인 電話音聲뿐만 아니라, 高速 데이터, 高速팩시밀리, 靜止畫 등 多種類의 서비스들을 하나의 通路 裝置로 交換하기 위해서는 交換의 基本單位인 타임스롯(time slot)을 使用하여 接續하는 多數 스톱(multi-slot) 接續이 必要하다.<sup>1)</sup> 즉, 64kb/s의 既存 디지털 交換機 構造에서 그 以上의 速度를 갖는 서비스 情報에 대해 하나의 呼

(call)에 m개의 64kb/s를 使用함으로써  $m \times 64$  kb/s의 回線 交換을 實現할 수 있다.

本 論文에서는 二元 트래픽 모델에서 呼種別 呼損率을 計算해 보고 計算하는 過程에서 發生하는 오버플로우(overflow)나 언더플로우(underflow) 問題를 解決할 수 있는 계산알고리즘을 提示하며 既存의 계산알고리즘과 比較 分析해 보고 SLAM-II에 의한 컴퓨터 시뮬레이션을 통해 確認해 본다.

### II. 二元트래픽 모델

일반적으로 두 種類의 呼가 通路網에 加해질 때 呼量이  $a_k(k=1,2)$ 이고 呼當 占有 채널數가  $m_k(k=1,2)$ 인 랜덤(random) 呼를 ( $X=a_k, Y=$

\*忠清專門大學 通信科  
Dept. of Communication Eng., Chung Cheong College  
\*\*韓國科學技術院 電氣·電子工學科  
Dept. of Electrical and Electronics Eng., KAIST  
\*\*\*清州大學校 電子工學科  
Dept. of Electronics Eng., Cheong Ju Univ.  
論文番號: 91-43(接受1991. 2. 4)

$m_k$ )로 나타낼 수 있는데 본 논문에서는呼量이  $a_1, a_2$ 이고呼當占有 채널數, 즉 64kb/s로換算된 타임스롯수가 1,  $m$ 인 두 種類의呼( $a_1, 1$ ), ( $a_2, m$ )가 全回線數가  $s$ 인回線에加해질때를 모델로 한다. 여기서 두 種類의呼를 64kb/s와 384kb/s로 例를 든다면  $m=6$ 이 된다.

各呼가 Poisson 過程에 따라  $\lambda_k$  ( $k=1,2$ )率로發生한다고 假定하면 平均 서미스時間(mean holding time)은  $1/\mu_k$  ( $k=1,2$ )로 指數分布(exponential distribution)하며 到着過程과 指數的으로 分布된 時間은 서로 獨立的이 된다. 또 各呼는  $dt$  時間 사이에 2個가 發生하지 않으며  $P(X=i, Y=j)=P(i, j)$ 는 統計的 平衡狀態의 同時接續確率(joint probabilities)이 된다<sup>2)</sup>. 여기서 호량  $a_k$  ( $k=1,2$ )는  $\lambda_k/\mu_k$ 가 되며 平衡狀態의 狀態遷移圖는 그림 1과 같다.

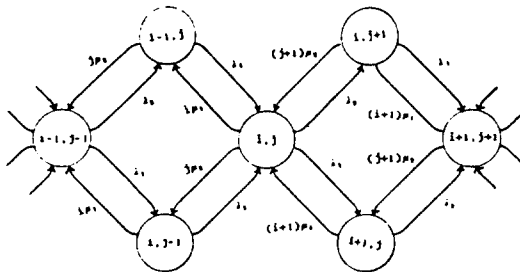


그림 1. 二元 트래픽 모델의 狀態 遷移圖  
Fig. 1. State Transition Diagram of the Two dimensional Multi slot Connection Traffic

이러한 狀態遷移圖로 부터 狀態平衡式을 구하여<sup>3)</sup>  $P(i, j)$ 를 구하면 다음과 같이  $P(0, 0)$ 의 함수로 주어진다.

$$P(i, j) = \frac{a_1^i}{i!} \frac{a_2^j}{j!} P(0, 0) \quad (1)$$

여기서  $s$ 는 全 채널수일때 두 종류의 호를 동시에 접속할 수 있는 모든 경우들을 집합  $A_s$ 로 표시하면  $A_s = \{(i, j) | i+mj \leq s\}$ 가 되며  $P(0, 0)$ 는 正規化 條件  $\sum_{(i, j) \in A_s} P(i, j) = 1$ 에서 求解

진다.

$(a_1, 1)$ 와  $(a_2, m)$ 인 두 種類呼의 各接續數를  $I$ 와  $J$ 로 할때 全接續 채널數  $S$ 는  $S=I+mJ$ 가 되며  $n$ 個의 채널이 占有될 確率을  $Q(n)$ 이라고 할때  $Q(n)=P\{S=n\}$ 는 다음과 같다<sup>4)</sup>.

$$Q(n) = P(0, 0) \sum_{m_j=0}^n \frac{a_1^{n-m_j}}{(n-m_j)!} \frac{a_2^j}{j!} \quad (2)$$

여기서 [ ]는 최대 정수를 취하는 Gauss 記號이며 正規化 條件을 利用하면  $P(0, 0)$ 는 다음과 같이 주어진다.

$$P(0, 0) = \sum_{i=0}^s \sum_{j=0}^{\lfloor s/i \rfloor} \frac{a_1^i}{i!} \frac{a_2^j}{j!} \quad (3)$$

式(2)의  $Q(n)$ 을 利用하여  $(a_1, 1), (a_2, m)$ 의 呼種別 呼損率  $B_1, B_m$ 을 計算하면 다음과 같다.

$$B_1 = Q(s) = P(0, 0) \sum_{m_j=0}^s \frac{a_1^{s-m_j}}{(s-m_j)!} \frac{a_2^j}{j!} \quad (4)$$

$$B_m = \sum_{n=0}^s Q(n) = P(0, 0) \sum_{m_j=0}^s \frac{a_1^{s-m_j}}{(s-m_j)!} \frac{a_2^j}{j!} \quad (5)$$

$s$  回路에 加해지는 合成 呼量은  $a = a_1 + ma_2$ 이며 두 種類의呼에 대한 平均 呼損率  $B$ 는 다음과 같이 주어진다.

$$B = (a_1 B_1 + ma_2 B_m) / a \quad (6)$$

지금까지 二元 트래픽에 대한 呼損率 計算式을 誘導하였지만 이러한 式들로 呼損率을 計算하는 경우에 計算式에 包含되어 있는 "계승(factorial)" 計算에 의하여 오버플로우(overflow)나 언더플로우(underflow) 問題가 發生하여 낮은  $s$ 에 대해서도 計算上의 限界가 나타난다. 이러한 問題들을 解決하기 위하여  $Q(n)$ 을 回歸式(recursive equation)의 形態로 변환하면 "계승"의 計算시 發生하는 오버플로우 問題는 解決된

수 있으며 본래 2次元的인 問題를 1次元的인 問題로 줄일 수 있는데 n개의 채널이 占有된 確率 Q(n)을 回歸式의 形態로 表現하면 다음과 같다<sup>5)</sup>.

$$nQ(n) = a_1Q(n-1) + ma_2Q(n-m) \quad (7)$$

이러한 回歸式을 利用하여 B<sub>1</sub>과 B<sub>m</sub>을 구하면 다음과 같다.

$$B_1 = Q(s) \quad (8)$$

$$B_m = \sum_{n=m+1}^s Q(n) \quad (9)$$

式(7)을 利用하여 呼損率을 計算하는 경우 式(2), 式(3)을 利用하여 計算하는 것보다 簡單하고 效果적이지만 앞의 경우와 같이 初期值 즉, Q(0)를 計算해야 하므로 s가 큰 경우 Q(0)를 計算할때 오버플로우의 問題가 惹起되어 計算上의 限界를 나타낸다<sup>6)</sup>. 따라서 오버플로우나 언더플로우 問題를 解決할 수 있는 呼損率 計算 알고리즘이 必要하게 되는데 다음의 Ⅲ章에서 이러한 計算 알고리즘을 誘導하고 既存의 呼損率 計算 알고리즘과 比較해 본다.

### Ⅲ. 計算 알고리즘

Q(n-1)과 Q(n)의 比를 X(n), 즉 X(n) = Q(n-1)/Q(n)으로 表示하고 식(7)에서 簡略化하면 X(n)에 관한 回歸式이 誘導된다.

$$X(n) = \begin{cases} \frac{n}{a_1 + ma_2 \prod_{i=1}^{m-1} X(n-i)} & , n \geq m \\ \frac{n}{a_1} & , 0 < n < m \end{cases} \quad (10)$$

여기서 X(1)은 Q(0)와 Q(1)의 比이므로 初期條件 Q(0)값을 計算하지 않고 (10)식으로 간단

히 計算할 수가 있다.

또한 Q(n)을 X(n)으로 나타내면 다음과 같다.

$$Q(n) = \frac{Q(n-1)}{X(n)} = \prod_{i=1}^n \frac{1}{X(i)} Q(0) \quad (11)$$

Q(0)는 正規化 條件에서 다음과 같이 求해진다.

$$Q(0) = 1 / [1 + \prod_{i=1}^s \frac{1}{X(i)}] \approx 1 / \prod_{i=1}^s \frac{1}{X(i)} \quad (12)$$

X(n)을 利用하여 呼種別 呼損率 B<sub>1</sub>, B<sub>m</sub>을 구하면 다음과 같다.

$$B_1 = \frac{1}{1 + \sum_{i=1}^s \prod_{j=1}^i X(s-j)} \quad (13)$$

$$B_m = \frac{1 + \sum_{i=1}^{m-1} \prod_{j=1}^i X(s-j)}{1 + \sum_{i=1}^s \prod_{j=1}^i X(s-j)} \quad (14)$$

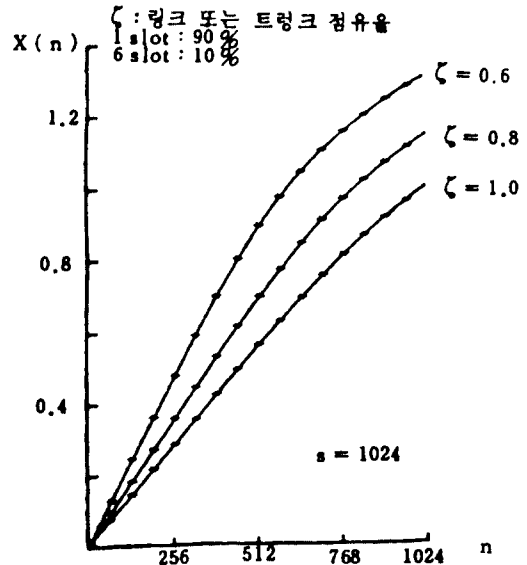


그림 2. X(n)의 計算  
 Fig. 2. Calculations of X(n)

呼損率  $B_1$ 과  $B_m$ 을 본 논문에서 제시한  $X(n)$  값으로 計算하면  $X(1)$ 은  $Q(0)$ 을 計算하지 않고 (10)式으로 쉽게 計算되어 오버플로우 문제는 발생하지 않는다. 또한 그림 2의  $X(n)$  計算例와 같이 個個의  $X(n)$  값이 크지 않으므로  $X(n)$  들의 곱에 의한 오버플로우 問題도 發生하지 않는다<sup>6)</sup>.

II章에서 呼損率을 計算하기 위해서 直接 計算 方法과 一次元的 回歸式으로 變換한 後에 計算하는 方法, 그리고 本 論文에서 提示한  $X(n)$ 을 利用한 呼損率 計算이 言及되었다. 여기에서는 위의 計算 方法들에 대해 計算 알고리즘의 分析을 통해 比較해 본다.

方法 1) 式(2), (4), (5)을 利用한 直接 計算法  
[스텝 1] 式(2)에서  $P'(0, 0) = \alpha$ 라고 놓으면  $Q'(n)$ 은 식 (15)와 같다.

$$Q'(n) = \sum_{m=1}^n \frac{a_1^{n-mj}}{(n-mj)!} \cdot \frac{a_2^j}{j!} \alpha \quad (15)$$

式(15)에 의하여  $n=1$ 에서  $n=s$ 까지  $Q'(n)$ 을 計算하여 記憶한다.

[스텝 2]  $\sum_{n=1}^s Q'(n) = \beta$ 에서  $\beta$ 를 計算하고, 여기서  $P(0, 0) = \alpha / \beta$ 를 구한다.

[스텝 3]  $B_1 = Q'(s) / \beta$ 를 計算한다.

[스텝 4]  $B_m = \sum_{n=1}^s Q(n) / \beta$ 를 計算한다.

方法 2) 式(1)에서 각  $P(i, j)$  값을 記憶하여  $B_1, B_m$ 을 計算

[스텝 1]  $P'(0, 0) = \alpha$ 라 놓으면  $P'(i, j)$ 는 식 (16)과 같다.

$$P'(i, j) = \frac{a_1^i}{i!} \cdot \frac{a_2^j}{j!} \cdot \alpha \quad (16)$$

[스텝 2]  $P'(i, j)$ 의 모든 境遇數의 合計  $\beta$ 를 計算하여  $P'(0, 0) = \alpha / \beta$ 를 구한다.

[스텝 3]  $\sum_{n=1}^m P'(s-mj-i, j) = \delta$ 를 計算하고  $B_m = \delta / \beta$ 를 구한다.

[스텝 4]  $\sum_{n=1}^m \sum_{j=0}^{\infty} P'(s-mj-i, j) = \delta$ 를 計算하

고  $B_m = \delta / \beta$ 를 구한다.

方法 3) 式(7), (8), (9)에 의한 呼損率 計算

[스텝 1]  $Q'(0) = \alpha$ 라 놓고  $Q'(n)$ 을 式(17)에서 구하여 記憶한다.

$$Q'(n) = \frac{1}{n} [a_1 Q'(n-1) + m a_2 Q'(n-m)] \quad (17)$$

[스텝 2]  $\sum Q'(n) = \beta$ 를 구하고  $Q(0) = \alpha / \beta$ 를 計算한다.

[스텝 3]  $B_1 = Q'(s) / \beta$ 를 計算한다.

[스텝 4]  $B_m = \sum_{n=1}^s Q'(n) / \beta$ 를 計算한다.

方法 4) 式(11), (13), (14)에 의한 呼損率 計算

[스텝 1]  $X(n)$ 을 順次的으로 計算하여 記憶한다.

[스텝 2] 式(13)에서  $B_1$ 을 計算한다.

[스텝 3] 式(14)에서  $B_m$ 을 計算한다( $B_1$ 과  $B_m$ 의 분모는 같다).

앞에서 提示한 4가지 方法에 대한 각 呼損率 計算을 乘除計算의 複雜度, 所要 메모리 및 計量 上の 問題點 등으로 比較해보면 表 1과 같다.

	乘除計算 複雜度	所要 메모리	計量上 問題點
방법 1	$O(s^3)$	$O(s)$	오버플로우 또는 언더플로우
방법 2	$O(s^3)$	$O(s^2)$	오버플로우 또는 언더플로우
방법 3	$O(s)$	$O(s)$	오버플로우
방법 4	$O(s^2)$	$O(s)$	—

表 1. 알고리즘의 比較分析  
Table 1. Comparison of 4 Algorithms

方法 1)은 計量上 오버플로우나 언더플로우 問題 때문에  $s$ 값이 작을때만 計算 可能하며 알고리즘의 乘除計算의 複雜度가  $O(s^3)$ 으로 計算 時間도 다른 3가지 方法보다 더 長이 걸린다. 方法 2)에 의한 呼損率 計算의 경우 方法 1)보다 是 計算의 複雜度가  $O(s^2)$ 으로 方法 1)보다 큰  $s$  값에 대하여 計算이 可能하나 이 方法도 오버플로우나 언더플로우 問題때문에 計算 可能한

s 값에 制限이 發生한다. 方法 3)에 의한 計算은 알고리즘의 複雜性和 所要 메모리 面에서 가장 좋은 알고리즘이나 오버플로우 問題가 惹起되어 最大 s에 制限이 있으며, 方法 4)에 의한 計算은 X(n)을 計算할때 s값이 아주 커져도 링크 占有度가 1일때 X(n)은 0과 1사이의 값을 갖게 되며 링크 占有度가 0.6으로 떨어져도 X(n)의 값이 크게 增加되지 않으며 X(n)을 利用한 計算에서는 初期值의 設定이 必要없이 呼損率 計算時 乘除에 의한 오버플로우 問題와 언더플로우 問題를 惹起하지 않고 B<sub>1</sub>과 B<sub>m</sub>을 求할 수 있으나, 方法 1), 2), 3)은 모두 初期值를 設定해야 하므로 初期值를 計算할때 오버플로우 問題가 發生한다.

#### IV. 計算結果

呼損率 計算을 위하여 개인용 컴퓨터가 使用되었는데 35!의 計算時 오버플로우 問題가 發生하므로 方法 1)에서는 s=34까지 計算이 可能하다. 方法 2)는 方法 1)에 比하여 最大 s 값에 약간의 改善이 있어 s=190 程度까지 計算이 可能하다. 또 方法 3)에 의해서는 計算 途中의 오버플로우 問題를 解決하기 위해 初期值로서의 값에 아주 작은 數를 繼續해서 곱하여 計算하면 s=1024 程度까지 計算 可能하게 된다. 그러나 方法 4)에서는 初期值의 設定이 必要없이 오버플로우 問題를 惹起하지 않고 s=10000 以上까지도 計算 可能하게 된다.

s개의 채널중 n개의 채널을 1 스투트호와 m 스투트호가 90% : 10%의 比率로 占有한다고 假定할때 方法 3)과 方法 4)에 따른 計算例로서 링크 占有度(ξ)에 대한 B<sub>1</sub>과 B<sub>6</sub>의 呼損率이 그림 3)와 그림 4)에 各各 나타나 있는데 方法 3)은 s=1024 정도까지 計算할 수 있었다. 여기서 두 結果 사이의 差異는 方法 3)의 計算에서 初期值의 設定時 오버플로우 問題를 解決하기 위하여 初期值의 값을 줄이기 위하여 아주 작은 數를 繼續하여 곱하는 過程에서 誤差가 發生하기 때문

이다.

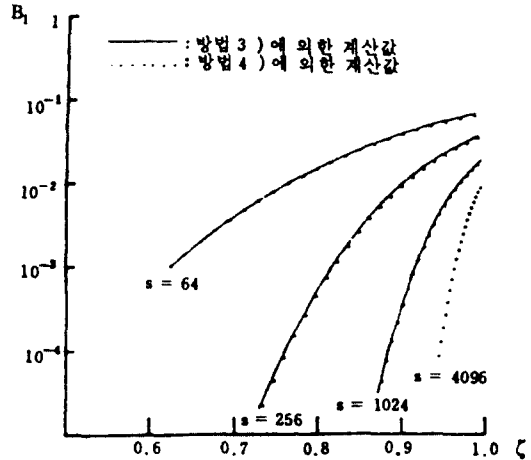


그림 3. B<sub>1</sub>의 計算  
Fig. 3. Calculation of B<sub>1</sub>

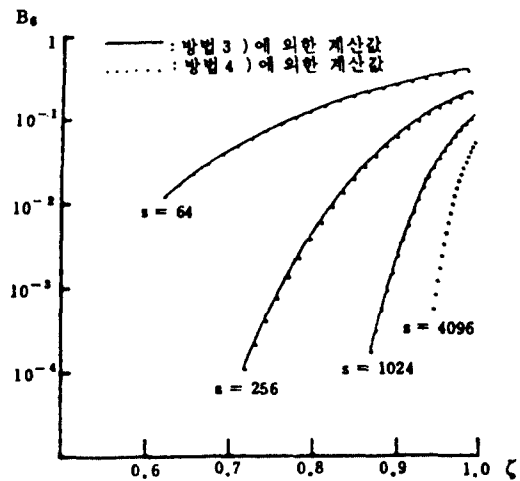


그림 4. B<sub>6</sub>의 計算  
Fig. 4. Calculation of B<sub>6</sub>

또한 그림 5)는 方法 4)의 數式에 의한 計算結果와 SLAM-II의 시뮬레이션에 의한 計算結果를 比較하여 나타낸 것인데 計算값들에 대한 다소의 差異는 있지만 같은 傾向을 나타내고 있음을 볼 수 있다.

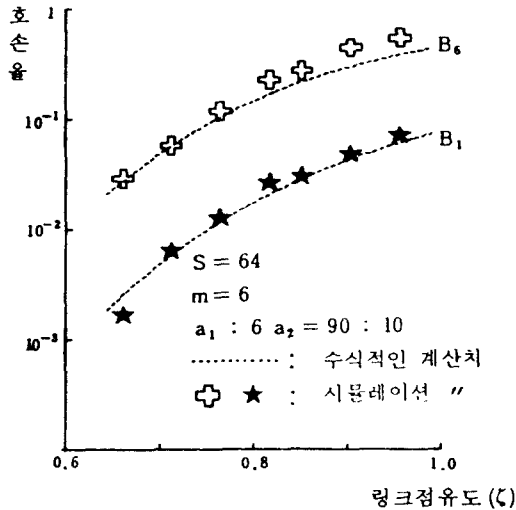


그림 5. 數式적인 計算結果와 시뮬레이션의 計算結果와의 比較  
 Fig. 5. Comparison between the analytic results and simulation results

### V. 結 論

二元 트래픽에 대한 呼損率 計算을 알아보고 4가지 方法으로 呼損率을 計算해 보았는데, 方法 4)의 計算 알고리즘은 初期값 設定이 必要없이 呼損率 計算時 알고리즘이 매우 簡單하고 計算 알고리즘상의 오비블로우 및 언더플로우 問題가 전혀 發生하지 않으므로 빠르고 正確한 計算을

行할 수 있다. 또한 方法 4)의 數式적인 計算값과 SLAM II의 시뮬레이션을 통한 計算값을 比較해 볼때 各各의 計算結果가 같은 傾向을 나타내고 있음을 볼 수 있는데 시뮬레이션을 통해 呼損率을 計算하려면 많은 時間이 필요하게 되나 方法 4)의 알고리즘으로 呼損率을 計算한다면 計算상의 오비블로우 問題를 發生하지 않고 쉽고 빠르게 呼損率을 計算할 수가 있다.

### 參 考 文 獻

1. Tatsuro TAKAHASHI, "Time Sequence Integrity for (Nx 64) kb/s Connection", Tran. IECE Japan, J69-B NO.10, P.1038 ~ 1045, Oct. 1986.
2. Cooper, R. B, *Introduction to Queuing Theory*, Second Edition, North Holland, 1977.
3. 秋山 稔, 近代通信交換工學, 電氣書院, 1973.
4. Haruhisa TAKAHASHI, Tetsuo TSUENIZUMI, Haruo AKIMARU, "Individual Call Losses for Multi-dimension Traffic", Paper of the Tech. Group, IECE Japan, SE83-138, P.17 ~ 20, 1983.
5. Joseph S. Kaufman, "Blocking in a Shared Resource Environment", IEEE Trans. Comm, Vol. Com 29, No. 10, Oct. P.1474 ~ 1481, 1981.
6. 成增根, 金承煥, 金大榮, "多元 트래픽의 呼損率 計算", 1987 年度 電氣·電子工學 學術大會 論文集(III), P. 958 ~ 961, 1987年 7月.



金承煥(Seung Hwan KIM) 正會員  
 1959年 8月19日生  
 1983年: 忠南大學校 工科大学 電子工學科(學士)  
 1988年: 忠南大學校 大學院 電子工學科(工學碩士)  
 1987年~1989年 6月: 忠南大學校 工科大学 電子工學科 助教  
 1989年 7月~1990年 2月: 韓國電子通信研究所 研究員  
 1990年 3月~現在: 忠清實業專門大學 通信科 專任講師



成權根(Dan Keun SUNG) 正會員  
 1952年 7月19日生  
 1975年 2月: 서울大學校 電子工學科 卒業  
 1977年 3月~1980年 7月: 韓國電子通信研究所 通話路研究室 專任研究員  
 1982年 8月: The University of Texas at Austin 電氣 및 컴퓨터工學科 卒業(M.S.)  
 1986年 5月: The University of Texas at Austin 電氣 및 컴퓨터工學科 卒業(Ph. D)  
 1986年 3月~現在: 韓國科學技術院 電氣·電子工學科 助教授



朴 鎮 秀 (Jin Soo PARK) 正會員  
1948年 8月30日生  
1975年 2月：漢陽大學校 電子工學科(學士)  
1977年 2月：漢陽大學校 大學院 電子通  
信工學科(碩士)  
1985年 2月：漢陽大學校 大學院 電子通  
信工學科(博士)  
1987年 2月～1988年 2月：Univ. of Colorado  
at Cololado Spring( Post  
Doc.)

1987年 2月～現在：清州大學校 電子工學科 教授

1988年 2月～現在：清州大學校 産業科學研究所 所長