

■ 論 文 ■

# 중앙집중형 교통정보 시스템에 관한 연구

Research on a Centralized Traffic Information System

석 현 우

(고려대학교 산업공학과 석사과정)

한 민 홍

(고려대학교 산업공학과 교수)

## 목 차

### I. 서론

1. 연구배경 및 필요성
2. 연구의 목적 및 범위

### II. 본론

1. 중앙집중형 교통정보 시스템
2. 교통정보데이터베이스의 구성
3. 경로계획 알고리즘

4. 중앙집중형 교통정보 전달체계의 시험 모델 구현

### III. 결론

1. 연구결과
2. 앞으로의 연구방향

참고문헌

## 요 약

본 논문에서는 기존의 교통정보 시스템의 단점을 보완하는 중앙집중형 교통정보 시스템의 구축에 관하여 설명한다. 이 시스템은 중앙의 교통정보센터에서 전자지도 및 교통정보를 관리한다. 따라서, 사용자는 전자지도 및 교통정보 갱신의 필요가 없어지며, 교통정보 수신장치를 통해서 경로 정보를 원하는 시기에 제공받는다. 교통정보 수신장치는, 휴대용 노트북과 개인휴대 통신기기인 Cellular Phone이나 PCS Phone이 결합된 형태로 구현함으로써 대중적으로 널리 보급될 수 있는 교통정보시스템을 구축한다.

본 논문에서는 정확한 교통정보의 생성을 위하여 실시간 교통정보를 저장하는 새로운 교통데이터베이스의 구조를 제시하고 직진·우회전·좌회전·유턴 차량의 흐름을 모두 표현할 수 있는 새로운 교통 네트워크 모델을 제시한다. 많은 수의 노드와 아크로 이루어진 교통 네트워크 내에서, 경로계획 시간이 길어지는 문제 해결 방법으로 전체를 여러 개의 지역으로 나누는 방법을 제시하고, 각 지역 내의 경로계획에는 교통데이터베이스를 참조하는 Dijkstra 알고리즘을 사용한다.

## I. 서론

### 1. 연구배경 및 필요성

지능형 교통시스템(ITS: Intelligent Transport System) 구축 사업은 도로 및 차량의 첨단화, 교통관제 센터와 차량간의 효율적인 정보 교류 등을 통해 교통난을 해소하고, 신속하고 안전한 도로 교통 환경을 구축하는 것을 목표로 한다. 지능형 교통시스템이 성공적으로 구축되기 위해서는, 지속적으로 변화하는 교통정보 내에서 이루어지는 시스템인 만큼 이에 적합한 시스템 구조를 갖추어야 한다.

현재 국내에서 교통방송을 통해 주요 혼잡지역의 교통상황을 라디오로 방송하여 운전자로 하여금 막히는 길을 우회하도록 유도하고 있지만, 교통난을 해소하는 근본적인 해결책은 되지 못한다. 교통방송은 방송의 특성상 여러 지역의 교통 상황을 순차적으로밖에 알려줄 수 없으므로, 운전자가 자신이 필요한 때에 원하는 지역의 정보를 얻을 수 없으며, 교통상황이 바뀐 같은 지역의 정보를 다시 얻기 위해서는 장시간을 기다려야 한다.

현재의 교통정보시스템은 CNS(Car Navigation System: 차량항법장치)을 기반으로 하는 것이 가장 널리 알려진 형태인데, 기존의 CNS 장비에 교통정보 수신장치를 덧붙여야만 한다. 즉, 장비를 구입하거나 설치하는 초기 비용이 많이 들뿐만 아니라, 전자지도가 바뀌었을 때 CNS내에 설치된 전자지도를 갱신하는 비용과, 변경되는 교통정보를 수신하는 비용까지 사용자가 모두 부담해야 해야하기 때문에 일반 대중에게 널리 보급되기 힘든 시스템이다.

이런 문제점을 해결하기 위하여 본 논문에서 제시하는 중앙집중형 교통정보시스템은 중앙의 교통정보센터에서 모든 전자지도 및 교통정보를 관리한다. 이로써 개인의 지도 및 정보의 갱신비용을 줄일 수 있으며, 전체의 교통흐름 및 상황을 중앙의 교통정보센터에서 파악할 수 있다. 또한 사용자는 지속적으로 변화하는 교통정보를 신속하게 얻을 수 있다. 사용자의 교통정보 수신장치는 이미 널리 보급된 개인휴대통신기를 이용하게 되며, 화면 표시 장치 역시 전용 장비가 아닌 노트북 등과 같이 다목적으로 사용하는

장비를 활용한다.

본 논문은 이 중앙집중형 교통정보시스템을 통하여 향후 몇 년 내에 구축될 범국가적 교통정보데이터베이스 및 서비스 단말기의 형태를 제시한다. 본 연구를 통해 구현된 전달체계는 실제 교통 정보서비스의 방법을 결정짓는 참고자료로 충분히 활용될 것이다. 또한, 본 시스템은 기존의 시스템에 비하여 사용자에게 비교적 적은 비용부담을 주기 때문에, 교통정보의 대중화에 크게 공헌할 것이다. 또한 중앙에서 최적경로를 계산하여 각 사용자에게 보내주기 때문에, 자연적인 교통흐름의 분산을 이끌어낼 수 있다. 따라서 본 시스템의 장시간 운용을 통하여 교통체증으로 인한 비용문제 및 공해문제를 완화시킬 수 있을 것으로 기대한다.

### 2. 연구의 목적 및 범위

본 논문은 다음의 사항들에 대하여 기술한다.

- 1) 중앙집중형 교통정보 시스템
- 2) 교통 상황을 반영하는 교통데이터베이스 구조
- 3) 경로 계획 알고리즘
- 4) 중앙집중형 교통정보 전달체계의 시험 모델 구현

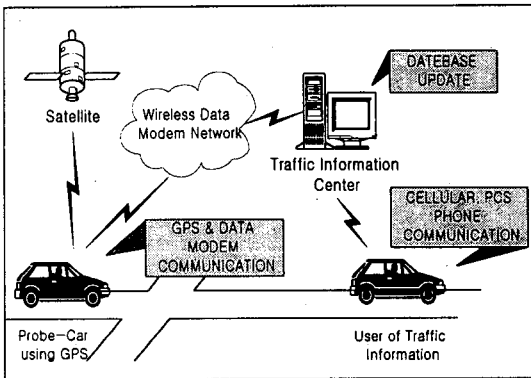
본 논문은 교통정보 수집체계에 의한 교통정보가 수렴되어 가는 현재 상황에서 실시간 교통정보를 전달하기 위한 교통정보데이터베이스의 구조를 제안하고 이를 바탕으로, 경로계획 알고리즘과 전달체계의 구성에 대한 방법을 제시한다. 특히, 기존의 분산형 교통정보시스템과 비교하여 보다 효율적인 중앙집중형 교통정보 시스템을 제안한다. 전달체계에서는 Cellular Phone과 PCS Phone을 이용한 교통정보 수신을 구현한다.

## II. 본론

### 1. 중앙집중형 교통정보시스템

기존의 교통정보시스템은 CNS(Car Navigation

System: 차량항법장치)를 기반으로 발달하였다. 초기에는 CD-ROM형태로 제공되는 전자지도 위에 최단거리 정도를 제공하는 수준이었으나 현재는 외부로부터 교통정보를 수신하여 전자지도 위에 교통정보를 함께 표시하는 기능을 가지고 있다. 본 논문에서 비교하는 분산형과 중앙집중형의 구분 기준은 경로계획의 주체가 누구인가이다. 분산형은 CNS를 기반으로 발전된 기존의 방식으로써, CNS장비와 교통정보 수신을 위한 통신장비로 구성되며, 수신된 교통정보를 이용하여 차량 내 단말기에서 경로계획을 하는 방식이다. 이에 반하여, 중앙집중형은 차량 단말기에는 전자지도 및 경로 정보를 가지지 않은 상태에서 필요시기에 중앙의 교통정보센터에 요청하여 해당 지역의 전자지도와 교통정보를 받아 화면에 표시하도록 설계된 시스템이다. 즉, 중앙의 교통정보센터는 사용자의 요구에 따른 최적경로를 계산하여, 경로를 포함하는 전자지도와 함께 전송하는 역할을 맡고 있다.



〈그림 1〉 중앙집중형 교통정보 시스템의 구조도

〈그림 1〉은 중앙집중형 교통정보 시스템에 대한 전반적인 구조를 나타낸다. 중앙집중형 교통정보 시스템은 크게 세 부분으로 이루어지는데, 교통정보를 수집하는 수집체계와 수집된 정보의 저장 및 검색을 맡은 교통정보센터, 그리고 사용자에게 전달하기 위한 전달체계로 구성된다.

중앙집중형 교통정보 시스템의 장점은 현재의 CNS와 같은 고가의 장비가 필요 없고, 중앙의 교통정보센터에서만 전자지도 및 교통정보를 교체하면 모든 사용자에게 변경된 정보를 제공할 수 있기 때문에, 사용자의 부담을 크게 덜어준다는 점이다. 그러나

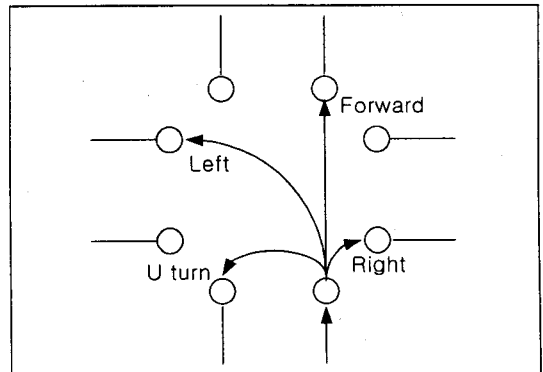
중앙의 교통정보센터의 역할이 그만큼 커지므로 다수 사용자로부터 받은 다양한 요청에 대응하기 위해서는 고성능 컴퓨터 시스템이 필요하다. 하지만 이러한 문제는 센터기능의 분산처리 및 경로계획 알고리즘의 개선 등을 통하여 해결될 수 있으며, 따라서 많은 비용을 들이지 않고 충분히 운영 가능한 시스템으로 구축할 수 있다.

## 2. 교통정보데이터베이스의 구성

교통정보를 실시간으로 수집하고 전달하기 위해서는 중앙 교통정보센터의 교통정보데이터베이스의 구조가 이를 잘 반영할 수 있는 형태를 지니고 있어야 한다. 다음에는 본 연구에서 이런 변화하는 교통정보를 받아들이기 위한 교통네트워크 모델과 교통정보데이터베이스를 정의한다.

### 1) 노드와 아크의 정의

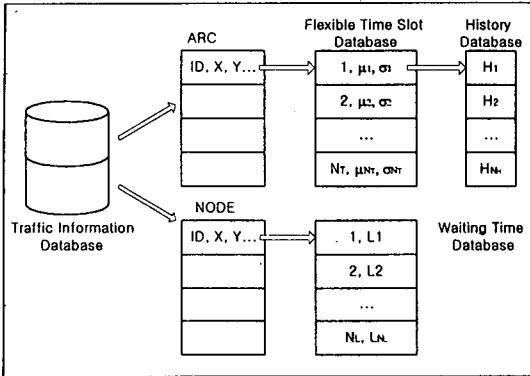
일반적으로 교통정보데이터베이스에 가장 기본적인 요소는 교차로와 도로의 정보이다. 기존의 시스템은 교차로를 하나의 노드로 도로를 하나의 아크로 나타내는 형태가 많다. 그러나, 이러한 교통 네트워크 구조는 복잡하고 현실적인 교통정보를 충분히 반영하기 힘든 구조이다. 본 논문에서 제시하는 교통 네트워크의 노드와 아크는 〈그림 2〉의 사거리의 예에서 잘 나타난다.



〈그림 2〉 노드와 아크의 연결

하나의 교차로는 여러 개의 노드로 분할되고 각 아크는 차량의 흐름을 나타낸다. 〈그림 2〉에서 예시한

사거리의 경우, 8개의 노드와 24개의 아크로 구성된다. 노드는 weight 값으로 신호주기와 관련이 있는 waiting time을 가지며, 아크는 travel time을 weight로 가진다. 이러한 교통 네트워크 구조는 기존의 구조와는 달리 차량의 직진, 우회전, 좌회전 및 유턴에 대한 정보를 각각 저장함으로써 보다 현실적인 교통정보를 나타낸다. 예를 들어, 좌회전이 막히는 교차로에서 직진이나 우회전 소통은 잘되는 경우, 제안된 교통 네트워크는 좌회전에 대한 흐름과 우회전이나 직진에 대한 흐름을 각각 다르게 저장할 수 있으므로 정확한 교통흐름을 나타낼 수 있다.



〈그림 3〉 교통데이터베이스의 구조

〈그림 3〉은 본 논문에서 제시하는 교통정보데이터베이스의 구조이다. 교통정보데이터베이스는 크게 노드데이터베이스와 아크데이터베이스로 나누어지는데, 실시간 교통정보의 저장을 가능하도록 노드데이터베이스에는 교통신호주기와 연관된 Waiting Time Database를 참조하도록 구성하고 아크데이터베이스는 Flexible Time Slot Database와 History Database를 참조하도록 구성한다. ARC와 NODE 데이터베이스는 각 데이터마다 자신의 고유 ID와 절대 위치(X,Y)나 기타 속성을 나타내는 필드를 가지고 있다.

2) Waiting Time Database

각 교차로의 신호주기를 반영하는 데이터베이스로써 각 노드에서 참조한다.  $N_L$ 은 하루동안 바뀌는 신호주기의 수를 의미하며, 시간대에 따라서 변하는 신호주기 패턴을 반영한 테이블이다. 이 데이터는 신호주기가 변경될 때마다 테이블 자체를 갱신하는 방법

으로 유지한다.

3) Flexible Time Slot Database

실제 차량이 아크를 이동하는 시간에 대한 데이터베이스이다. 차량의 흐름은 시간에 따라 큰 차이를 보인다. 하루 24시간을 세분화하여 각 시간대를 대표하는 값을 저장하는 테이블을 구성한다. 각각의 세분화된 시간대를 time slot이라 한다.  $N_T$ 는 time slot의 수이며  $\mu_{N_T}$ ,  $\sigma_{N_T}$ 는 그 time slot의 차량 이동시간의 평균과 표준편차이다. 이 통계량은 각 아크와 연결되어 있는 History Database에서 산출된다.

time slot의 시간간격은 각 아크별로 교통량에 따라, 교통량이 집중하는 시간대에는 좁게, 교통량이 적은 시간대에는 넓게 책정하는 유연한 시간 간격을 적용하는 것이 보다 효율적이다. 예를 들어, 차량이 적게 통과하는 새벽이나 한 낮에는 2시간 내지 3시간을 하나의 time slot에 반영하고 차량 통행량이 많은 출·퇴근 시간에는 20분 내지 30분을 하나의 time slot에 반영한다. 〈표 1〉은 Time Slot Database의 구조를 나타낸다.

〈표 1〉 Time Slot Database의 구조

Time Slots	평균 시간	표준 편차
시간 1	$\mu_1$	$\sigma_1$
시간 2	$\mu_2$	$\sigma_2$
...	...	...
시간 N-1	$\mu_{N-1}$	$\sigma_{N-1}$
시간 N	$\mu_N$	$\sigma_N$

다음의 계산 예는 24시간을 30분 단위의 time slot으로 고정시킨 방법과, 차량이 집중되는 4시간만을 30분 단위로 하고 나머지 시간은 2시간 단위의 time slot을 적용하여 하나의 아크 저장공간을 비교한 것이다. 평균과 표준편차는 각각 4 bytes의 저장공간을 사용한다. Fixed Time Slot의 경우 총 48개의 time slot을 가지며 384 bytes의 저장공간이 필요하지만, Flexible Time Slot은 총 18개의 time slot을 가지며 68 bytes의 저장공간만이 필요하다. 따라서 Flexible Time Slot 방법은 Fixed Time Slot의 16.67% 정도의 저장공간만으로 데이터베이스를 구축할 수 있는 방법이다.

Fixed Time Slot

24시간 \* 60분/30분 = 48개

48 \* (4+4) bytes = 384bytes

Flexible Time Slot

$$(4시간 * 60분/30분) + (20시간/2시간) = 18개$$

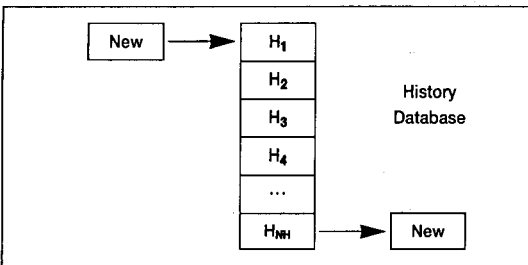
$$18 * (4+4)bytes = 68bytes$$

Efficiency

$$68/384 * 100 = 16.67%$$

적절한 time slot의 개수를 정하기 위하여 시뮬레이션 기법을 이용한 기초 작업과 실제 실험지역 선정을 통하여 얻어진 실측치 분석 작업이 함께 이루어져야 한다.

4) History Database



<그림 4> Queue구조의 History Database

History Database는 Time Slot Database를 보조하는 데이터베이스이다. 교통정보는 지속적으로 변화하지만, 도로별 하루 동안의 차량통행수 변화량을 기록하고 이를 여러 날 동안 관찰하면, 하루에 대한 통행수 변화 형태가 비슷한 분포를 나타낸다. History Database란, 이러한 차량통행수 변화량의 변화 경향을 파악하기 위한 각 도로별 통과 시간에 대한 Queue구조의 데이터베이스이다.  $H_{NH}$ 는 참조되는 아크의 실제 이동시간 정보이며,  $N_H$ 는 History Data의 개수이다. 따라서, History Database는  $N_H$ 개의 History Data를 가지며 각 아크의 통과시간의 변화를 저장하고 있다. 이 데이터베이스는 새로운 이동시간이 추가 될 때마다 가장 오래된 이동시간을 삭제하여  $N_H$ 개의 데이터를 유지한다. <그림 4>는 이러한 History Database의 Queue구조를 나타내고 있다.

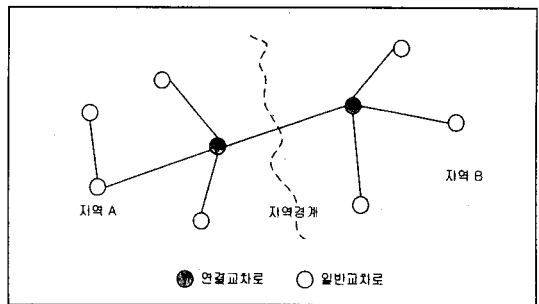
History Database는 아크의 이동시간에 대한 새로운 정보가 입수되었을 때, 이 정보가 해당 아크의 평균적 이동시간을 벗어나는지 테스트하는 근거가 된다. 새로운 이동시간이 현재의 데이터 집합에 포함되는지를 알 수 있는 방법으로 Regression 기법과 관리도 기법의 하나인  $\bar{X}-R$  chart등을 이용하는 방법이 있다.

History Database는 각 아크의 time slot별로 하나씩 참조하기 때문에, 상당한 저장 공간이 필요하다. 따라서 적절한 History Data의 개수를 정하는 것이 중요하다. 하나의 아크가 참조하는 History Database의 크기는  $N_H * N_T * 4$  bytes ( $N_T$ 는 Time Slot 수,  $N_H$ 는 History Data 수, 4 bytes는 Data의 크기)이다.  $N_T$ 를 18개로,  $N_H$ 를 10으로 가정할 때, 그 필요한 저장공간은 720bytes이고 <그림 2>와 같은 네거리 교차로를 표현하기 위해서는 17200 bytes의 저장공간이 필요하다.

3. 경로 계획 알고리즘

기존의 Dijkstra 알고리즘의 단점은 노드의 수가 증가함에 따라 계산량이 폭발적으로 증가한다는 것이다. 따라서 이를 해결하기 위하여 본 논문에서는 계층적 교통네트워크의 분할 방법을 제안한다.

1) Network의 분할

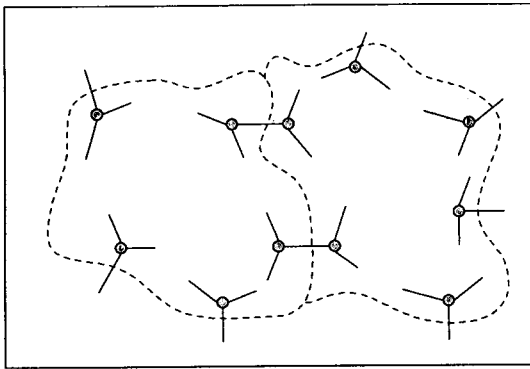


<그림 5> 일반교차로와 연결교차로

행정구역단위로 나뉘어있는 도로구조와 같이 교통네트워크 역시, 행정구역단위로 구분하여 관리하는 좋다. 그러나, 네트워크 계산상의 효율성을 고려한다면, 각각의 단위지역 내에는 교차로수가 비슷하게 포함되도록 구성하는 것이 이상적이다. 일반적으로 교차로는 하나의 지역내(intra-area)의 도로를 연결하는 교차로와 다른 지역간(inter-area)의 도로를 연결하는 교차로로 구분된다. 본 논문에서는 이들 교차로를 <그림 5>에서와 같이 일반교차로와 연결교차로로 칭한다.

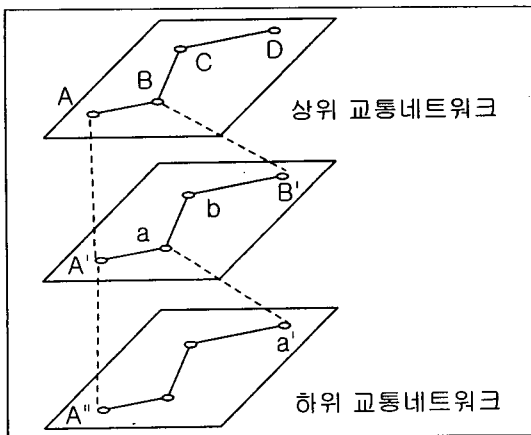
하나의 지역 내에서는 기존의 방법과 같이 Dijkstra 알고리즘으로 경로계획이 가능하다. 하지만 여러 지역간의 경로 계획을 위해서는 <그림 6>과 같은 연결

교차로만으로 구성되는 네트워크를 구성함으로써 계산에 필요한 노드와 아크의 수를 줄인다. 이렇게 연결교차로만으로 구성되는 네트워크는 지역내의 네트워크에서 경로를 검색하는 것과 동일하게 Dijkstra 알고리즘을 적용할 수 있다.



〈그림 6〉 연결교차로로 구성된 네트워크

더 큰 지역간에 경로의 검색 때에는, 단위지역을 마치 연결교차로와 일반교차로로 나누듯이 구분하여 네트워크를 구성하고 역시 Dijkstra 알고리즘을 적용한다. 즉, 〈그림 7〉과 같이 네트워크를 계층적으로 세분화함으로써 보다 네트워크의 구조를 간단하게 구성한다. 실제 사용자 측면에서의 경로의 검색은 상위 네트워크에서 이루어지지만, 상위 네트워크에서 탐색되는 경로는 그 보다 작은 하위 네트워크에서 이미 탐색된 경로를 바탕으로 계산되는 경로이므로 전체 네트워크 상의 모든 경로를 검색하는 것과 동일한 결



〈그림 7〉 계층 구조의 교통 네트워크

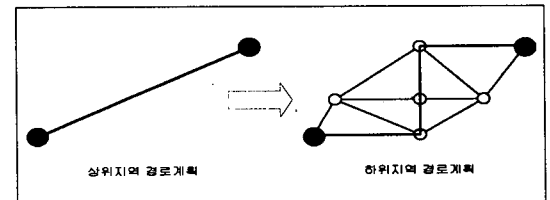
과를 얻어낸다.

〈그림 7〉을 통하여 본 논문에서 제시하는 경로 탐색의 방법을 자세히 설명하면 다음과 같다. 먼저 상위 교통네트워크에서 출발지를 A, 목적지를 D로 하는 경로를 탐색하도록 요청 받았다고 가정한다. 그리고, 네트워크 층은 그림에서처럼 3계층으로 나뉘어졌다고 가정한다.

입력받은 출발지와 목적지에 대한 첫 번째 네트워크에서의 Dijkstra 알고리즘 경로 탐색을 실시한다. 계산과정에서 각 노드와 노드간의 Weight 값이 필요하게 되는데, 예를 들어 A와 B구간에 대한 Weight 값이 필요해지면, 이 두 노드를 포함하는 두 번째 네트워크로 정보를 요청하게 된다. 그림에서 보듯이 두 번째 네트워크는 A'에서 B'까지의 정보를 다루는 네트워크이다. 이 네트워크에서 역시, A'에서 B'까지의 Dijkstra 알고리즘을 이용한 경로 탐색 과정이 실시된다. 마찬가지로 방법으로 두 번째 네트워크 층에서도 계산 중에 A'에서 a까지의 Weight값이 필요해지면, 다시 세 번째 네트워크로 정보를 요청하는 것이다. 즉, A''에서 a'까지를 포함하는 세 번째 네트워크로부터 경로 정보를 얻는다. 두 번째, 세 번째 네트워크는 분산처리방식으로 경로계산이 지속적으로 이루어지기 때문에 빠른 검색속도를 보일 수 있다.

상위 네트워크는 하위 네트워크에서 계산된 정보를 참조함으로써 경로 계획을 하게 된다. 이러한 방법으로 계속적인 계산 과정을 거쳐서 최종 A에서 D까지의 최적 경로 정보인 A, B, C, D를 얻을 수 있다.

2) Front Network Design



〈그림 8〉 계층적 교통네트워크에서의 경로 계획

계층 구조를 가지는 교통 네트워크에서 경로 계획이 실시간으로 처리되기 위해서는 분산처리 방식이 강력히 요구된다. 〈그림 8〉은 계층적 교통 네트워크

에서의 경로계획을 보여준다. Front Network의 기본 개념은 하위 네트워크내의 각 연결교차로간의 경로계획을 미리 계산하여 저장하고 상위 네트워크에서 관련 지역에 대한 경로계획이 이루어질 때, 이미 계산된 최적의 하위 경로 정보를 제시 해주는 것이다. 즉, 하위 지역 내 경로계획을 수행하는 프로세서는 지속적으로 교통 데이터베이스를 이용하여 해당 지역의 경로계획을 계산하고 얻어진 경로 정보를 Front Network Database에 저장한다. 따라서 실제적으로 경로계획을 하게 될 때에는 이 데이터베이스를 기준으로 데이터참조 형식의 경로계획이 이루어지기 때문에 많은 시간 단축 효과를 거둘 수 있다.

3) 기존의 경로 탐색 방법과의 비교

기존의 교통네트워크와 본 논문에서 제시하는 교통네트워크간의 구조상의 차이가 매우 크기 때문에, 동일한 실험 네트워크에서 두 방법의 경로탐색 성능을 비교하는 것은 부적합하다. 따라서, 본 논문에서는 각각의 경우의 단위 계산의 횟수를 분석함으로써, 그 우수성을 검증해보고자 한다. 이를 위하여 10000개의 노드 수를 가지는 100 by 100 Matrix 형태의 교통네트워크를 설정한다.

기존의 일반적인 Dijkstra 알고리즘의 Complexity는  $O(n^3)$ 로 계산 회수는  $3/2n^3$ 에 비례한다. 즉, 위의 예의 전체 네트워크를 Dijkstra 알고리즘을 사용하게 되면 최악의 경우  $3/2 \times 10000^3$  번의 계산을 수행해야 한다.

본 논문에서 제시하는 네트워크 분할 이론을 바탕으로 10000개의 노드를  $20 \times 20$ 의 작은 네트워크로 분할하면, 네트워크는 25개로 나뉘어진다. 하나의 하위 네트워크인  $20 \times 20$ 의 노드를  $4 \times 4$ 의 보다 작은 25개의 네트워크로 분할하면, 총 네트워크의 수는  $25 \times 25 = 625$ 개가 된다. 결론적으로, 네트워크의 개수는 많아지지만 전체 네트워크를 두 계층으로 분화함으로써, 하나의 네트워크에 포함되는 노드의 수를 10000개에서 16으로 줄였다. 따라서, 최악의 경우  $3/2 * 16^3$  회의 계산만으로 하위 네트워크의 경로 계획이 가능해진다. 만일, 모든 하위 네트워크에서 동일시간동안 경로가 계산되는 분산처리가 이루어진다면, 실시간 경로계획이 가능해진다.

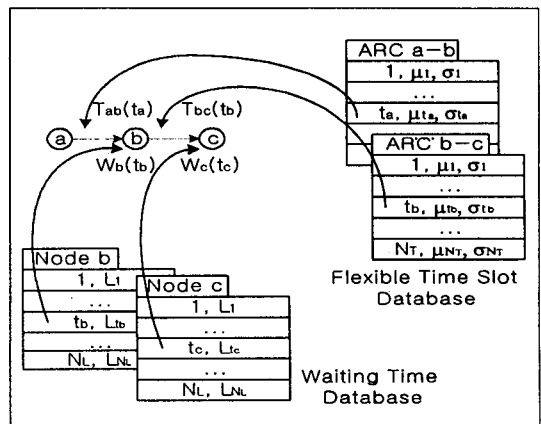
그런데, 본 논문에서 제시하는 교통네트워크는 분

명, 기존의 네트워크보다 많은 수의 노드와 아크를 포함하고 있다. 따라서, 위의 예에 적용하기 위해서는 8배 이상(그림 2)의 교통네트워크의 구조 참조) 노드수가 많은 90000개의 노드를 가지는 네트워크의 경우를 생각해 본다. 이때 네트워크를 두 계층으로 분할하여 645개의 하위 네트워크를 구성하면, 하나의 하위 네트워크에 포함되는 노드의 수는 144개가 된다. 만일, 네트워크 계층을 더 분할한다면, 포함되는 노드수를 더 줄일 수 있다.

4) 변화하는 교통정보를 반영하는 경로계획 방법

다음은 하나의 하위 네트워크의 경로계획 알고리즘에 대한 설명이다. 기본적인 절차는 기존의 Dijkstra 알고리즘과 비슷하지만, 변화하는 교통정보를 반영하기 위하여 아크나 노드내의 고정된 weight값을 사용하는 것이 아니라, 노드의 weight값은 Waiting Time Database를, 아크의 Weight 값은 Flexible Time Slot Database를 참조하여 경로를 찾아낸다.

교통상황은 시간의 변화와 밀접하다. 따라서, 경로 계획에 있어서도 시간의 요소를 고려하여, 모든 weight값을 나타낼 때, 시간에 대한 파라미터를 지니는 식으로 표현한다. 즉, 각 노드 n에 차량이 위치할 때의 시점을 각각  $t_n$ , 시점  $t_n$ 일 때의 노드 n에서 노드 m을 연결하는 아크의 weight값을  $T_{nm}(t_n)$ . 시점  $t_n$ 일 때의 노드 n의 weight값을  $W_n(t_n)$ 이라고 정의한다. (그림 9)의 예는 노드 a에서 노드 c까지 이르는 경로계획을 나타낸다.



<그림 9> 노드 a에서 노드 c로의 경로계획

일반적인 경로계획의 절차는 다음과 같은 순서로 진행된다.

- step 1 : 출발지 노드를 선택한다
- step 2 : 목적지와 현재 선택된 노드와 비교한다
  - sub-step 1 : 목적지와 일치하면 경로를 완성하고 계산 루틴을 빠져나간다
  - sub-step 2 : 일치하지 않으면 step 3으로 간다
- step 3 : 선택된 노드들에서의 다음 연결된 노드까지의 weight sum을 계산한다
- step 4 : 계산된 weight sum을 모두 비교하여 가장 작은 값을 취한다.
- step 5 : 가장 작은 weight sum의 아크와 연결된 노드를 선택하고 step 2로 간다

본 논문의 경로계획에 있어서 가장 중요한 부분은 step 3 단계의 weight sum 계산이다. 즉, weight sum을 계산하는데, 본 논문에서 제시된 새로운 데이터베이스 구조를 활용함으로써, 시간에 따라 변화하는 교통정보가 반영된다. <그림 9>의 노드 a에서 노드 c까지의 최종 weight sum  $S_{ac}(t)$ 은 아크 a-b의 이동시간  $T_{ab}(t_a)$ 와 노드 b에서 대기시간  $W_b(t_b)$ , 아크 b-c의 이동시간  $T_{bc}(t_b)$ , 노드 c에서의 대기시간  $W_c(t_c)$ 의 합으로 표현된다. 또한 일반적으로 노드 n과 다음 노드 m사이에서,  $t_m$ 은  $t_n$ 과 노드 n에서 노드 m까지의 이동시간을 합한 시간이 된다. 즉,  $t_m = t_n + T_{nm}(t_n)$ 의 관계가 성립하므로, 예의 weight sum은 다음과 같다.

$$\begin{aligned}
 S_{ac}(t) &= T_{ab}(t_a) + W_b(t_b) + T_{bc}(t_b) + W_c(t_c) \\
 &= t_b - t_a + W_b(t_b) + t_c - t_b + W_c(t_c) \\
 &= t_c - t_a + W_b(t_b) + W_c(t_c) \\
 \therefore t_b &= t_a + T_{ab}(t_a), t_c = t_b + T_{bc}(t_b)
 \end{aligned}$$

또한 a, b, c, d 로 이어지는 경로에 대한  $S_{ad}(t)$ 는 다음과 같다.

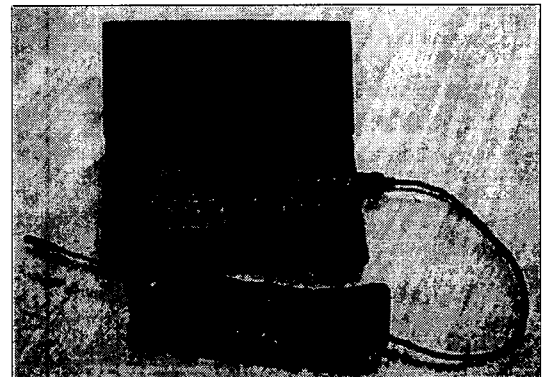
$$\begin{aligned}
 S_{ad}(t) &= T_{ab}(t_a) + W_b(t_b) + T_{bc}(t_b) + W_c(t_c) + \\
 &\quad T_{cd}(t_c) + W_d(t_d) \\
 &= t_b - t_a + W_b(t_b) + t_c - t_b + W_c(t_c) + \\
 &\quad t_d - t_c + W_d(t_d) \\
 &= t_d - t_a + W_b(t_b) + W_c(t_c) + W_d(t_d) \\
 \therefore t_b &= t_a + T_{ab}(t_a), t_c = t_b + T_{bc}(t_b), \\
 t_d &= t_c + T_{cd}(t_c)
 \end{aligned}$$

weight sum  $S_{nm}(t)$ 는 실제 노드 n에서 노드 m에 이르는 시간의 합이므로 도착예정시간을 나타낸다. 위 식의 전개와 같이 일반적으로 노드 n에서 노드 m까지의 weight sum  $S_{nm}(t)$ 은 다음과 같이 나타낸다.

$$\begin{aligned}
 S_{nm}(t) &= t_m - t_n + \sum_{n \rightarrow m} (W_k(t_k)) \\
 &\text{(단, 출발노드의 Waiting Time 즉, } W_n(t_n)=0\text{이다.)}
 \end{aligned}$$

본 논문에서 제시하는 경로계획 알고리즘의 우수성은, 알고리즘 진행 방법상은 기존의 Dijkstra와 동일한 형태이나 실제 참조되는 weight의 값을 새로운 교통데이터베이스에서 참조함으로써, 실제 교통 상황을 반영하여 통과시간을 정확히 예측할 수 있는 구조로 변화시킨 것이다. 따라서, 본 논문의 알고리즘으로 교통상황 예측과 더불어 보다 정확한 경로 계획이 가능하다.

#### 4. 중앙집중형 교통정보 전달체계의 시험 모델 구현



<그림 10> 단말장치의 구성

CNS를 기반으로 하는 교통정보 시스템은 교통 정보를 이용하기 위하여 필요한 장비의 설치비용이 매우 비싸기 때문에, 일반 대중에게 널리 보급되지 못하고 있다. 이러한 문제점을 해결하고 더 많은 사용자들을 확보하기 위해서, 저가의 단말기와 이미 널리 보급된 통신장비를 사용한다. 본 연구에서 제시하는 교통정보 단말장치는 <그림 10>과 같이 일반 노트북 컴퓨터와 Cellular Phone 이나 현재 사용자수가 급격히 증가하고 있는 PCS Phone을 연결한 형태이다. 사용자는 단말장치를 이용하여 중앙의 교통정보센터에



전화를 걸고 출발지와 목적지를 입력한 후, 서버로부터 계산된 경로정보를 받아보게 된다.

본 연구에서 구현한 전달체계는 교차로를 Node로, 도로를 Arc로 표현하는 기존의 교통네트워크 구조상에서 구현된 것이다. 지도는 서울 강북지역의 Vector 지도를 자체 제작하였고, 거리 데이터를 통해 산출된 가상 교통 데이터 값으로 경로를 계산한다.

전체 시스템은 중앙 교통정보센터의 역할을 하는 서버부분과 사용자 단말기 부분인 클라이언트 부분으로 나뉜다.

1) 서버의 구성 및 역할

서버는 통신을 위한 모뎀과 서버용 프로그램으로 구성되며 서버에서는 출발지와 목적지에 대한 최단시간 경로 탐색 알고리즘을 구동하며, 사용자에게 이를 전송하는 기능을 지니고 있다. 서버에서 클라이언트로 전송하는 데이터는 전체 Vector 지도에서 출발지와 목적지 정보를 포함하는 사각(Rectangle)영역을 잘라낸 전자 지도 정보와 Dijkstra 알고리즘에 의해서 얻어진 경로정보가 된다.

2) 클라이언트의 구성 및 역할

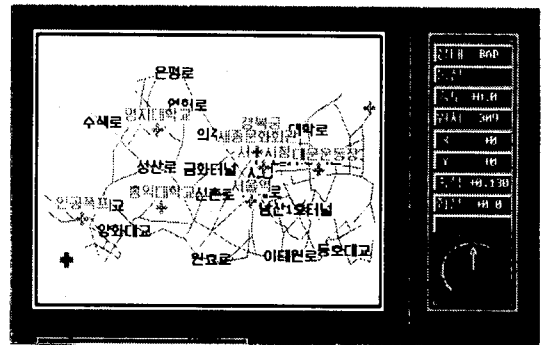
클라이언트는 서버에 접속할 수 있는 클라이언트용 프로그램과 프로그램 실행에 필요한 노트북 등의 소형 컴퓨터 그리고 통신을 위한 개인휴대이동통신기인 Cellular Phone 이나 PCS Phone으로 구성된다. 클라이언트에는 중앙집중형 교통정보시스템의 개념에 잘 나타난 것처럼 초기에는 어떠한 정보도 가지지 않는다. 즉, 사용자가 필요한 때에 원하는 전자지도와 경로정보를 받아 보게 된다. 또한 일단 교통정보센터로부터 데이터를 전송 받은 후에는 기존의 CNS 기능을 충분히 수행할 수 있다.

3) 개발 프로그램의 특징

(1) Layer 기능의 벡터 지도의 확대, 축소

벡터 지도의 장점은 확대나 축소 등의 지도 배율의 변화에 의하여 정보의 질이 떨어지지 않는다는 것이다. Layer기능이란 배율에 따라 보여지는 정보를 달리하는 기술로써, 본 개발 프로그램에서는 지도를 확

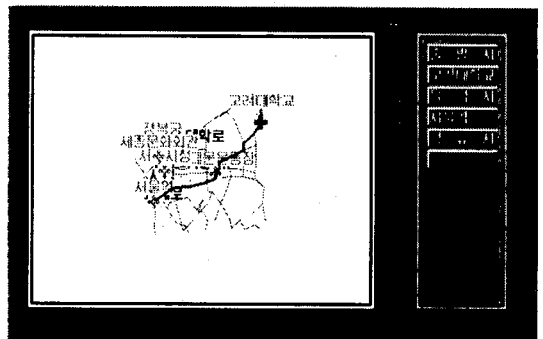
대했을 때, 도로 폭에 따라 표시하는 선의 굵기가 조정되고 더 많은 위치 정보를 나타낸다. <그림 11>는 서울 강북지역을 Vector 화한 전자 지도이다.



<그림 11> 벡터 방식의 전자 지도(서울 강북지역)

(2) Clipping된 지도정보 전송으로 전송 효율 향상

사용자에게 부과되는 통신비용을 줄이기 위한 통신시간 단축 기술이 필요하다. 본 개발 프로그램에서는 서버에서 전자지도 정보를 사용자가 원하는 출발지와 목적지까지의 경로를 포함하는 영역만 잘라(Clipping)내어 <그림 12>과 같은 간단한 데이터를 보내주게 된다. 이렇게 함으로써 전체 지도를 보내주는 경우에 비하여 평균 50%의 속도 향상을 보인다.



<그림 12> Clipping된 지도 및 교통정보

(3) 주행시 음성 안내

차량에서 사용하게 될 단말 프로그램은 운전자의 안전운행을 방해해서는 안 된다. 따라서 교통정보를 제공해줄 수 있는 수단이 연구되어야 하는데, 현 시점에서 가장 우수한 방법은 음성을 통한 정보의 전달이다. 음성에 의한 전달은 운전자에게 시야를 빼앗지

않을 뿐 아니라, 여러 가지 많은 정보도 충분히 소화해 낼 수 있는 전달 방법이다. 본 개발 프로그램에서는 단말 장치에서 필요한 단어들을 음성파일로 저장하여 이를 필요한 상황에 맞추어 조합하여 들려주는 방식을 채택했다.

**(4) 경로변경시 재전송 정책**

교통 정보는 지속적으로 변화하기 때문에, 운전자가 원하는 정보를 얻은 후 그 정보가 얼마 지나지 않아서 변경되는 경우가 종종 있다. 이렇게 이미 제공된 정보가 변경되었을 때, 대처하는 방법은 크게 두 가지로 나눌 수 있다. 사용자 자신이 직접 다시 교통정보센터에 접속하여 원하는 정보를 재수신하는 방법과, 교통정보센터에서 사용자에 대한 정보를 저장하고 있다가 제공했던 정보가 변경되었을 때, 다시 전화를 걸어서 알려주는 방법이다. 전자의 경우는 사용자에게는 불편함을 주는 방법이고, 후자의 경우는 중앙의 교통정보센터에서 재전송이 필요한 사용자의 정보를 기록해야하는 부담이 있다. 본 개발 프로그램에서는 두 가지의 방법을 모두 이용할 수 있게 하였다.

**(5) GPS를 이용한 CNS기능 수행**

본 개발 프로그램은 교통 정보의 수신뿐만 아니라, 기존의 CNS의 기능을 충분히 수행한다. GPS 장비를 부착하면 곧 차량의 현재 위치를 지도 위에 표시해 줄 수가 있다. 이를 구현하기 위하여 통계적 맵매칭 기술을 이용한다. 맵매칭이란 GPS로부터 얻어진 현재 위치 정보를 전자 지도의 해당 위치에 나타내는 기술인데, 여기서 말하는 통계적 맵매칭 기술이란 이전의 차량 위치 정보를 저장하고 현재의 정보와 비교하여 위치를 표시함으로써 근접한 다른 도로에 있는 것으로 잘못 표시하는 것을 막기 위한 기술이다.

**III. 결론**

**1. 연구결과**

본 논문의 제안들은 실시간 교통정보시스템을 구성하는데 걸림돌이 되었던, 데이터베이스의 갱신과

경로탐색으로 대표되는 교통정보의 검색 방법에 대하여, 교통네트워크 분할 방법과 이에 따른 분산 처리 방법을 이용한 새로운 접근을 제시함으로써, 시스템 구성에 있어서 현실적 모델을 구성할 수 있는 방향을 보여주었다. 특히, 변화하는 교통 정보를 수용하기 위한 교통네트워크의 새로운 정의와 교통데이터베이스 구조 변경은 보다 정확한 교통 정보시스템을 이끌어 내는데, 주요한 역할을 할 것이며, 교통정보시스템 실용화를 앞당길 것이다.

〈표 2〉 PCS와 Cellular와의 통화 품질 비교 (성공률의 괄호 안은 Peak 시간대)

Time Slots	평균 시간	표준 편차
비교 대상	PCS (LG Ccm 1300F)	Cellular Phone (Motorola MicroTAC 5000)
평균 전송 시간	약 30 초	약 35 초
성공률(98 6월)	80% ( 50% )	90% ( 75% )
성공률(98 7월)	90% ( 75% )	85% ( 75% )
성공률(98 8월)	98% ( 80% )	75% ( 70% )
실패 원인	접속 실패	접속 실패, 통화중 끊김
특기사항	통화 장소 및 시간의 영향	통화 시간의 영향

개인통신장비를 이용한 교통정보 수신 프로그램 개발은 시스템의 보급 및 확산에 큰 영향을 끼칠 것으로 분석된다. 지속적인 전달체계 통신 실험 결과인 〈표 2〉는 현재 통신 시장의 방향을 보여준다. PCS의 사용자가 확대되고 통신망이 확충됨에 따라 몇 달 사이에 통신 성공률이 역전되는 현상을 볼 수 있었다. 또한, PCS의 경우 Cellular Phone에서 보이는 통신 중 끊김 현상이 거의 없기 때문에, 보다 안정적인 데이터 통신을 가능하게 하고 있다. 그러나, PCS의 경우, 전파의 특성상 직진성이 강하고 회절성이 약해서 장소의 특성을 타는 경향을 보이고 있다.

**2. 앞으로의 연구 방향**

중앙집중형 교통정보시스템은 앞으로 많은 개선을 필요로 한다. 실제 교통데이터베이스를 구축하기 위해서는 교통정보 수집체계와의 연계 방안이 모색되어야 하고 이를 위한 인프라 형성 방법에 대한 논의가 뒤따라야한다. 또한, 제안된 교통네트워크와 데이터베이스 모델에 대한 실제적인 적합성 및 실용성이 평가가 있어야 할 것이다. 전달체계에 대한 연구에서는 통신 효율을 높이기 위한 데이터 압축 기법과 보다 세밀한

통신 프로토콜 규약이 필요하며, 안정적인 데이터 통신을 위한 지속적인 통신망의 확충이 필요하며, 다양하고 대용량의 정보를 전송하기 위한 고속 통신망에 대한 연구가 함께 진행되어야 할 것으로 판단된다.

### 참고문헌

1. J.W.M BIESTERBOS & F. ZIJDERHAND "A Dynamic Car Navigation, Driver Information and Fleet Management System", Philips Journal of Research, Vol 48, pp.229~313, 1994.
2. Athanasios K. Ziliaskopoulos & Hani S. Mahmassani, "Time-Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway Systems Applications".
3. Vijay K. Garg & Joseph E. Wikes, "Wireless and Personal Communications Systems", Prentice Hall, 1996.
4. James R. Evans, "Optimization Algorithms for Networks and Graphs", MARCEL DEKKER, INC, 1992.
5. Lee, S., Choi, K., and Kim, G. (Korea Automotive Technology Institute), "KATECH's Efforts of the Standardization of Korean Digital Road Map for Navigation and Implementation of Shortest Path Algorithm Using the Vector Topology Data", IEEE Intelligent Vehicles Symposium, Seikei Univ. Tokyo, Japan, Sep 19-20, 1996.
6. Telecommunications Primer Signals, Building Blocks, and Networks E. Bryan Carne 1995.
7. "Montana 33.6 User's Guide", Motorola Inc., 1997
8. Davide C. March, "Database Design, Development, and Access Consideration for Automotive Navigation", IEEE Vehicle Navigation & Information Systems, 1989.
9. John Guzolek and Edward Koch, "Real-Time Route Planning in Road Networks", IEEE Vehicle Navigation & Information Systems, 1989.
10. Clay Collier and Jackson Wang, "Methods of Link Coding for Traffic Information Broadcast", IEEE-IEE Vehicle Navigation & Information Systems Conference, Ottawa-VNIS, 1993.
11. Bastiaansen, A.(Tele Atlas) "The Navigation Digital Street Map is the Critical Success Factor for Vehicle Navigation and Transport Information System in Europe", IEEE Intelligent Vehicles Symposium, Selkei Univ. Tokyo, Japan, Sep 19-20 1996.
12. 김대진, "차량항법 시스템에 관한 연구(A Research for CNS)", 고려대학교 대학원, 1994.
13. 윤형성, "동적 주행안내 체계 개발 연구(Development of Dynamic Route Guidance System)", 고려대학교 대학원, 1997.