

서는 셀룰라 시스템과 Ad Hoc 네트워크를 이용하여 이런 문제점을 해결할 수 있는 방안을 제시한다.

3.1 시스템 모델

현재 차량 합법 시스템에서의 정보(교통정보 등)는 여러가지 방법(주간 검침, CCTV 등)수집되고 있다. 이렇게 수집된 정보는 정보센터로 보내져 가공되며, 이 가공된 정보들은 경로 계산이나 기타 컨텐츠로 이용되고 있다. 그러나 이런 현재의 시스템은 많은 부분에서 시간과 자원을 효율적으로 활용하지 못한다는 단점을 지니고 있다. 그중에서 수집된 정보를 가공하는 시간과 통신 상의 오버헤드를 예로 들 수 있다. 먼저 이동 호스트나 호스트들(Mobile hosts, Fixed hosts)이 정보센터에 접속하는 통신비용과 통신채널선점은 무선망의 오버헤드를 초래하며, 또한 이런 정보들을 가공하는 시간은 여러 수집된 정보를 정보센터라는 한 곳에 집중되기 때문에 이를 가공하기 위해서는 많은 시간이 필요하다. 이런 문제점을 해결하기 위해서 ARS(Ad Hoc Relaying Station)를 이용하여 기지국과 정보센터에 집중되는 문제점을 해결할 수 있으며, 또한 BTS의 call blocking 발생시 Ad Hoc 네트워크를 이용하여 BTS(Base Transceiver Station)에서의 호 집중을 분배하는 역할을 수행한다. 본 논문에서는 ARS를 호 집중의 분배 및 실시간 정보 수집에도 이용한다. ARS1과 ARS2는 BTS 1, BTS 2, BTS 3과 자유롭게 통신할 수 있으며, 이동이 가능하다. ARS1과 ARS2는 기존의 이동 호스트들과 같은 특성을 가지며, 단지 이동 호스트들과 다른 점은 Ad Hoc 네트워크가 가능하며, ARS 간 통신이 가능하다. 또한 ARS는 MSC(Mobile Switching Center)에 의해 통제되어진다. ARS는 단말기내에 네트워크가 가능한 무선 모듈이 탑재되어 있어야 하며, Ad Hoc 네트워크가 가능한 모듈이 탑재되어 있어야 한다. 그림 1에서와 같이 ARS는 이동하면서 자신의 현재 위치와 해당 구간을 지나간 시간을 근거로하여 속도를 계산하고 계산한 값들을 가공하여 정보센터로 보내는 작업을 수행한다. 또한 정보센터의 정보를 미리 가져와서 향후 다른 이동 호스트들(ARS가 아닌)에게 정보를 나누어주는 역할을 수행한다. 이는 프록시 서버(proxy server)나 게이트웨이(gateway)와 같은 역할을 수행한다. 이런 ARS는 이동성을 갖을 수도 있고 고정되어 있을 수도 있다. 그러므로 주유수 같은 고정적인 것을 이용할 수 있다. 이런 경우엔 Probe car와 통신이 가능한 인터페이스를 구성하면 된다. 물론 ARS는 cache 기능을 가지고 있어 정보센터에서 정보를 미리 caching 할 수 있다. ARS 시스템 구성을 위해서 ARS 간의 통신이 가능하도록 적당한 수의 ARS가 존재한다고 가정한다[1].

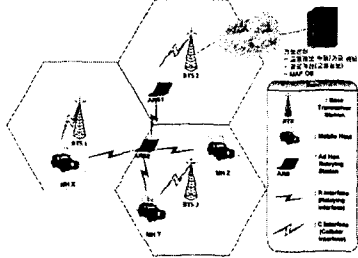


그림 1. ARS를 이용한 실시간 정보 제어 기법

그림 1에서 MH X, MH Y, MH Z는 경로 탐색을 위해 정보센터로부터 정보를 요구하는 이동 호스트를 나타낸다. 이들 이동 호스트들은 ARS에서 브로드캐스팅되는 정보를 받아서 자신이 필요한 정보들을 선별적으로 이용할 수 있다.

3.2 정보 수집 및 전달

ARS의 기능은 위에서 언급했듯이 정보의 수집과 정보센터로의 전달, 그리고 주변의 이동 호스트들에게 브로드캐스팅하는 것이다. 이런 ARS는 실시간으로 정보를 수집하고 가공하는 기능을 가지며, 주변의 ARS와도 통신이 가능하여 정보를 교환할 수 있다. 본 절에서는 ARS의 실시간 기능과 정보 수집방법 그리고 정보 전달 기법에 대해서 설명한다. ARS는 이동 호스트와 비슷하게 동작한다. 이동성을 가지며, 기지국과 통신하는 것은 다른 이동 호스트들과 같은 기능을 갖는다. 그러나 이들은 Ad Hoc 네트워크가 가능한 무선 모듈을 탑재하고 있어 주변의 이동 호스트들에게 정보전달이 가능하며, ARS 간 통신이 가능하다. 또한 멀티 홉(Multi hop) 라우팅이 가능하여 주변의 노드들(ARS)에게 전달이 가능하여 특정 ARS에게 통신이 가능하다[2]. ARS는 GPS 위성으로부터 시그널을 수신할 수 있는 모듈을 탑재하고 있으며, 이를 이용하여 자신의 위치 좌표를 계산할 수 있다. 또한, 지속적으로 위치를 계산하여 이동한 거리대 시간을 계산하여 속도를 계산할 수 있는 연산 능력을 갖춘 시스템과 정보 저장장치를 위한 저장장치를 가지고 있다고 가정한다. ARS는 당시의 교통정보 수집 외에도 정보센터의 지도정보와 실제 도로의 지형을 수집할 수 있다. 기존 시스템에서의 정보의 가공은 정보센터에서 주로 담당하였다. 그러나 ARS는 충분한 자원과 파워를 바탕으로 정보를 가공하여 정보센터에 전달함으로써 정보센터에서의 부하를 최소화 시킬 수 있다.

3.3 실시간 정보 제어 기법

실제로 ARS는 이동성을 지니고 있으므로 이동 중 정보물 센터에 보내게 되는데 이런 정보를 이웃한 ARS에게로도 전송한다. 현재 링크 아이디, 종전 속도, 현재 속도, 종전속도대 현재 측정값에 대한 비율을

계산하여 비율이 클 경우, 모든 이웃한 호스트들에게 브로드캐스팅 한다. 이 때, 이동 호스트들과 셀 내의 BTS에 전달된(브로드캐스팅 정보) 정보는 R(ARS_id, link_id, present_time, pre_info, present_info, ratio)의 형태와 같이 전달된다. 전달된 R은 ARS의 정보를 뜻하며, ARS_id는 정보를 수집한 해당 ARS를 뜻하고, link_id는 정보를 수집한 해당 노드와 노드를 연결한 네트워크 도로의 링크를 뜻한다. Present_time은 현재의 시간을 의미하고 pre_info는 해당하는 링크의 센터에서의 당시 교통정보(속도, 또는 가중치)를 나타낸다. 또한, present_info는 현재 시간을 기준으로 ARS가 계산하여 가공된 교통정보(속도, 또는 가중치)를 나타낸다. 이렇게 ARS에 의해 수집되고 가공되어 이동 호스트들에게 전달된 실시간 정보는 이동 호스트들의 자신이 이동 중인 경로와 체크하게 된다. 또한 ratio는 센터가 가지고 있던 이전 정보와 현재 수집된 실시간 정보의 비율로써, 이 비율이 커질수록 정보의 변화가 심함을 뜻한다. 결과적으로 ratio가 커질수록 이웃한 이동 호스트들이 경로를 재탐색 할 확률이 높아질 것이다. ARS로부터 받은 정보가 유효(사고나 기타 극심한 정체로 상태가 바뀌었거나 그 반대의 경우)하다면 이를 이용할 수 있다. 이를 이용하려면 먼저 만약에 자신이 속한 경로상에 자신이 현재 보유하고 진행 중인 경로상에 해당 링크가 존재한다면, 현재 링크에 자신이 존재하고 있는지, 아니면 지나갔는지, 또는 아직 진입 전인지에 대한 판단을 수행한 후, 만약 진입 전이라면 경로를 재탐색하기 위해 센터에 다시 요청하여 빠른 새로운 경로를 다운로드할 수 있다.

```

알고리즘 I. ARS에서의 처리
링크별 수집된 정보를 가공한다.
loop
주기적/비주기적으로 해당하는 가공된 정보를 이동 호스트들(ARS 포함)에게 전달
    
```

```

알고리즘 II. 정보센터에서의 처리
주기적/비주기적으로 ARS로부터 들어오는 정보를 수신, 가공하여 저장
loop
이동 호스트로부터 탐색 요청 받음(이동 호스트의 현재위치 정보 포함).
해당하는 이동 호스트의 경로를 재탐색
if(이동 호스트가 요청한 경로가 현재 진행 중인 것까 다름) then
    새로운 경로가 타당하다면(최적) 이동 호스트에게 바뀐 경로를 다운로드
else
    경로 재탐색이 필요없음을 이동 호스트에게 알림
    
```

```

알고리즘 III. 이동호스트에서의 처리
이웃한 ARS로부터 정보를 받음
loop
if(ARS에서 보내는 정보 중 해당 link_id가 현재의 이동 호스트의
경로에 포함) then
    if(해당 link_id를 이미 지남 or link상에 있음) then
        경로 재탐색이 필요없음
    else
        재탐색을 정보센터에 문의
        현재 위치좌표 및 ARS에게서부터 받은 정보를 정보센터에 전달
    else 재탐색 필요 없음
    
```

이동 호스트에서 ARS로부터 받은 정보에 대한 경로 재탐색 알고리즘 결정은 위와 같다. 위의 경로 재탐색 알고리즘은 주행 시간(travel time)을 기준하며, 이동 호스트는 전체 경로의 각 링크별 정보(속도, 거리, 시간)를 가지고 있다고 가정한다. 이동 호스트의 자원이 충분하지 않으므로 이동 호스트 내에서 경로의 재탐색 여부에 대한 판단은 불가능하다고 가정한다. 단지, 받은 정보(link_id, 현재시간, 등)를 토대로 자신이 현재 진행 중인 경로상에 해당하는 링크가 존재하는지에 대한 판단은 가능하다. 이것은 알고리즘 II의 정보센터에서 수행하는 기능의 일부를 단말기에서 수행할 수 있게하여 정보센터와의 통신을 최대한 줄일 수 있는 방안으로 이용할 수 있다. 또한, ARS가 정보센터로 정보를 보내는 간격은 주기적/비주기적일 수도 있다. 비주기적인 경우 ARS가 해당하는 링크를 지났을 때, 정보를 가공한다. 만약 가공한 정보가 센터에서 받은 정보와 비교하여 많은 차이를 보인다면 비주기적으로 전송을 진행할 수 있다(극심한 정체의 경우 등). 그러나 링크마다 ARS에 의해 이동 호스트에게 정보가 전달된다면, 이동 호스트가 정보센터에 최적의 경로를 요청하기 때문에 call blocking이 발생할 수 있다. 이런 경우엔 이웃한 ARS를 통해서 call을 분배할 수 있도록 한다. 또한, 이런 것은 이동 호스트의 재탐색 타당성을 요청하기 이전에 해당 링크가 전체 경로에 차지하는 비율을 계산하여 일정 비율이 아닐 경우 재탐색 타당성 요구를 정보센터에 보내지 않는다.

4. 성능 분석 및 평가

본 논문에서는 주행 시간(Travel time)을 기준으로 하여 경로를 계산한다고 가정한다. 주행시간에 대한 입의의 주행 경로물 요구하는 이동 호스트들에 대하여 각각 주행 경로에 공통의 링크가 존재한다고 가정하여 시간과 거리에 대하여 분석하였다.

4.1 ARS를 이용한 실시간 자동 경로 재탐색 기법

본 논문에서 제안하는 ARS를 이용한 기법에서의 시뮬레이션은 경로에 대한 시간과 거리로 표시하였으며, S사의 시뮬레이션 툴을 이용하여 분석하였다. 시뮬레이션 툴은 현실 세계의 정보를 반영하고 실제에 가까운 환경으로 구축되었다. 최적 경로란 목적지까지의 도달 주행 시간이 가장 짧은 경로를 뜻한다. 시뮬레이션에서 10개의 이동 호스트가 현재 진행중이고 보유중인 경로에는 공통되는 링크가 존재하며, 이를 주행한다고 가정하였다. 이런 이동 호스트의 전체 주행거리는 5~20km(이동 호스트가 받은 목적지까지 도달하는데까지의 전체 거리)로 가정하였다. 또한 ARS 간 통신이나, ARS와 이동 호스트 간 통신은 Ad Hoc 라우팅 프로토콜을 따른다[2].

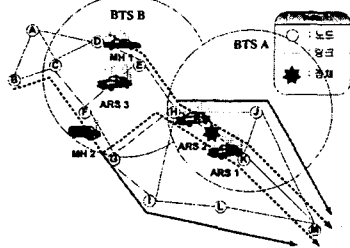


그림 2. ARS를 이용한 자동 경로 탐색 기법

그림 2는 ARS(Ad Hoc Relaying System)를 이용한 실시간 자동 경로 탐색 기법을 나타내며 ARS는 Ad Hoc Relaying Station을 의미하고 MH는 이동 호스트를 의미한다. 그림 2에서 MH1은 노드 D-E-H-K-M을 경유하는 경로를 정보센터로부터 받았다. 또한, MH2는 노드 B-C-F-H-K-M의 경로를 초기 경로로 하여 출발하였다. 이런 상황에서 ARS1은 노드 H에서 노드 K로 비슷한 시간에 이동하면서 교통정보를 수집한다. 수집한 정보는 정보센터에 보낸 뒤, 이를 정보센터에서 가공하여 사용자들에게 서비스할 수 있다. 또한, ARS2, ARS3 역시 ARS1과 같은 기능을 수행하고 있다고 가정한다. 만약 이 상황에서 노드 H와 K 간 링크에서 교통사고나 기타 정체 현상이 발생하여 H-K 링크를 지나는데 많은 시간이 소요됨을 ARS1에 의해 수집되었다고 가정하면, ARS1은 주변의 이동 호스트나 ARS에게 브로드캐스팅하게 된다. 만약 현재 위치가 H-K 링크 상에 존재하는 이동 호스트가 이 정보를 받았다면, 이동 호스트는 자신의 주행 경로상에 ARS1이 보내준 링크가 존재하는지 확인한다. 만약 자신의 주행 경로에 H-K 링크를 포함한다면, 이미 지나친 길인지, 현재 그 링크에 있는지에 대한 것을 자신이 가진 경로와 위치에 대해서 비교한다. 현재 H-K 상에 존재하므로 정보를 폐기한다. 그림 2에서 MH1과 MH2는 ARS3으로부터 교통정보를 전송받는데, 이 정보는 ARS1에서 ARS2를 통해 ARS3로 전달된다. 정보를 받은 MH1과 2는 위에서 언급한 과정을 통하여 자신이 수행할 경로상에 정체 현상이 발생함을 알 수 있다. 정체 현상을 감지한 MH1과 2는 정보센터에 접속하여 재탐색의 필요성을 문의한다. 이 경우 정보센터에서는 이미 H-K 링크의 교통정보를 ARS1으로부터 수신하였기 때문이다. 재탐색 요청을 문의받은 정보센터는 기존의 경로와 비교하여 최적의 경로를 계산한다. 계산한 경로가 초기의 경로보다 타당하다면 새로운 경로를 MH1과 MH2에게 전달한다. 전체의 경로가 길다면 짧은 경로에 비해 다른 대안 경로(재탐색 경로)가 필요 없을 수도 있다. 현실에서는 많은 다른 요소가 복합적으로 발생하므로 대안 경로가 초기 경로보다 더 빠른 경로가 될 수도 있다. 그 결과 MH1은 D-E-H-J-M을, MH2는 B-C-F-G-I-L-M을 경유하는 경로로 다시 설정하였다. ARS로부터 정보를 받아서 정보센터에 재탐색을 요청하는 방법 이외에도 이동 호스트가 자신이 주기적으로 최적경로를 재탐색하는 방안이 있을 수 있다. 그러나, 이런 방법은 통신에 대한 오버헤드가 너무 크므로 ARS를 이용하는 기법에 비해 효율적이지 못하다. 시뮬레이션 환경은 실제로 현실세계에서 사용되는 교통정보를 토대로 계산되었으며, 경로 계산은 dijkstra의 algorithm을 이용하였다. 시뮬레이션 파라미터로는

- 이동 호스트 수를 10 개로,
- 주행 거리를 5 ~ 20 Km 로,
- 시뮬레이션 지역은 서울 전 지역으로 설정하였다.

그림 3과 4에서 X축의 오른쪽으로 진행할수록 전체 경로의 거리가 길어짐을 뜻한다. 그림 3은 주행 시간에 대한 시뮬레이션 결과를 나타낸다. Y축은 시간(sec)을 나타내며, X축은 이동 호스트를 나타낸다. 그림 4에서 이동 호스트(MH1 - MH10)는 임의의 주행 거리를 가지고 각자 다른 목적지를 주행하고 있으며 10개의 이동 호스트 모두 경로상에 공통된 링크를 통과한다고 가정하였다. 실제로 초기경로시간은 최초 정보센터에서 탐색된 경로로 진행했을 때의 목적지까지의 주행 시간을 나타낸다. 그러나 실제로 해당하는 경로상에서 교통정보는 실시간으로 변화하기 때문에 실제의 주행 시간은 최초의 시간과 다를 수 있다. 주행시간에 대한 시뮬레이션은 실제 교통정보를 토대로 임의의 목적지와 출발지를 설정하여 측정하였다.

그림 3에서 기존 기법으로 이동 호스트가 정체 구간을 통과하는데 소요되는 시간은 ARS를 이용한 시간에 비해 상대적으로 많은 시간이 소요된다. 물론 상대적으로 차이가 발생하는 시간의 차는 정체가 풀리는데 소요되는 시간과 동일하다.

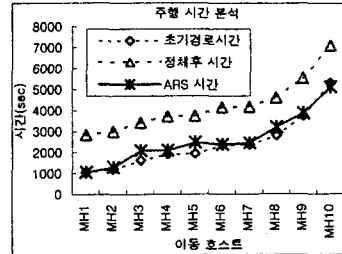


그림 3. 주행 시간 분석

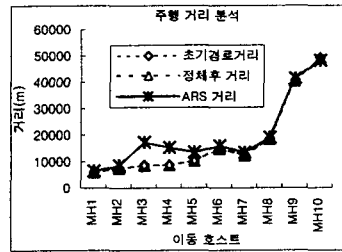


그림 4. 주행 거리 분석

그림 4는 주행 거리에 대한 시뮬레이션 결과를 나타낸다. 세로 축은 시간(sec)을 뜻하며, 가로축은 정보센터로부터 받은 전체 경로에서 공통으로 해당 링크를 가진 이동 호스트를 나타내며, 오른쪽으로 갈수록 전체 경로가 길어짐을 뜻한다. 그림에서 볼 수 있듯이 실제로 주행한 전체 주행 거리가 길어질수록 시간은 증가하며, ARS로부터 경로에 대한 정보를 받고 다시 정보센터에 정보를 요청한 경우 실제로 초기 경로에 비해 전체 경로의 거리는 길어진다. 그러나 그림 3에서와 같이 시간은 줄어들게 되어 목적지까지 도착하는데 까지 소요되는 시간이 짧아지게 된다.

5. 결론 및 향후 연구

본 논문에서는 차량 합법 시스템에서의 효율적인 실시간 정보의 제공 기법에 대해서 다루었다. 특히, MANET을 이용한 ARS(Ad Hoc Relaying System) 기법을 토대로 실시간으로 교통정보를 수집하고 이를 효과적으로 이용하는 방법에 대해서 설명하였다. 이런 ARS를 이용한 시스템은 기존의 초기 경로대로 주행하는 시스템에 비해 효과적인 최적의 경로를 제공할 수 있으며, 교통정보 수집에도 효율적으로 대처할 수 있다. 시뮬레이션 분석 결과 실제로 기존의 경로를 초기 탐색 후 목적지까지 계속 주행하는 방법이나 주기적으로 자동으로 정보센터에서 경로를 설정하여 이동 호스트에게 전달하는 기법보다 ARS를 이용하여 실시간으로 경로 재설정 시 필요한 구간에서만 이동 호스트에서 경로를 재설정하는 제안 기법이 더욱 효율적임을 알 수 있다.

참고문헌

- [1] H. Wu, C. Quao, S. De, and O. Tonguz, "An Integrated Cellular and Ad-hoc Relaying System: iCAR," IEEE Journal on Selected Areas in Communication(JSAC) (special issue on Mobility and Resource Management in Next Generation Wireless System), Oct., 2001.
- [2] Charles E. Perkins, "Ad Hoc Networking," Addison Wesley, 2000.
- [3] <http://www.entrac.co.kr/>
- [4] Y. D. Lin and Y. C. Hsu, "Multihop cellular: A new architecture for wireless communication," in IEEE INFOCOM, pp. 1273-1282, 2000.
- [5] <http://www.fleetnet.de/>
- [6] IEEE Standards Board, Part 11: Wireless LAN medium access control(MAC) and physical layer(PHY) specifications, The Institute of Electrical and Electronics, Inc., IEEE Std 802.11-1997.
- [7] C. Qiao, H. Wu, "iCAR: an Integrated Cellular and Ad-hoc Relaying System," IEEE International Conference on Computer Communication and Network, pp. 154-161, Oct. 2000.
- [8] G. Dommety and Raj Jain, "Potential Networking Applications of Global Positioning Systems (GPS)," OSU TR-24, April 1996.
- [9] J. Caffery, Jr. and G. Stuber, "Vehicle location and tracking for IVHS in CDMA microcells," in IEEE Personal Indoor Mobile Radio Conference, pp. 1227-1231, 1994.
- [10] S. Hameed and N. H. Vaidya, "Efficient Algorithm for Scheduling Data Broadcast," ACM/Baltzer Wireless Networks(WINET), 1999.
- [11] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," Tech. Rep. TR-24, CS Dept., The Ohio State University, April 1996.