

텔레매틱스 서비스 인프라스트럭처를 위한 센서 네트워크 활용 및 데이터 관리

이현익 | 황인석 | 송준화
KAIST

요약

이동 통신 기술의 발달과 더불어, 점점 더 많은 사람들이 자동차 안에서 각종 정보 서비스를 제공받기 위해 텔레매틱스를 이용하고 있다. 텔레매틱스 환경에서의 다양한 서비스를 위해서는 데이터를 효과적으로 수집, 전달, 처리, 저장하는 것이 무엇보다 중요하다. 본 글에서는 텔레매틱스에서 요구되는 데이터의 수집과 처리에 센서 네트워크 기술의 활용 가능성과 실제 사례를 살펴보고, 더 나아가 센서 네트워크에서 발생하는 데이터를 효과적으로 관리하기 위해 제안된 방법들에 대해 알아보려고 한다.

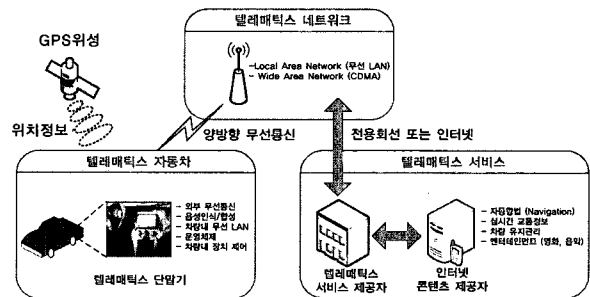
1. 서론

텔레매틱스(Telematics)란 텔레커뮤니케이션(Telecommunication)과 인포매틱스(Informatics)의 합성어로, 자동차 안의 단말기를 통해서 자동차와 운전자에게 다양한 종류의 정보서비스를 제공해주는 것을 의미한다. 운전자나 탑승자는 양방향 무선 네트워크와 인공위성 위치정보시스템을 이용하는 단말기를 통해 자동차 안에서 교통정보, 원격차량 진단, 모바일 전자상거래(M-Commerce)와 같은 각종 정보 서비스를 제공 받을 수 있다. 텔레매틱스는 이런 서비스들을 제공할 수 있는 유·무선네트워크, 단말기뿐만 아니라 전체 콘텐츠 및 서비스 등을 모두 포함한 전체 End-to-End 솔루션으로 정의될 수 있다. (그림 1)은 텔레매틱스를 개념

적으로 나타낸 그림이다.

(그림 1)에서 보는 것처럼 텔레매틱스 서비스 제공을 위한 기본적인 인프라는 크게 텔레매틱스 서비스 제공자 아키텍처, 무선네트워크 아키텍처와 단말기 아키텍처로 나뉘볼 수 있다. 텔레매틱스 서비스 제공자(Telematics Service Provider)는 줄여서 TSP라고도 부른다. TSP는 모든 서비스를 자체적으로 제공하는 대신 제 3의 인터넷 콘텐츠 제공자(3rd party)와 협력하여 서비스를 제공한다. 하지만, 기본적으로 텔레매틱스 사용자를 위한 서버 기능들, 예를 들면 PIM (Personal Information Management), 사용자의 단말기에 적합하도록 콘텐츠 변환/합성(Contents Adaptation and Integration), 혹은 단말기와의 네트워크 연결 관리 등은 TSP 서버에서 지원할 수 있다.

텔레매틱스는 무선 네트워크를 통하여 언제 어디서나 서비스가 가능하다는 특징이 있다. 이를 위한 무선 네트워크는 크게 무선 광대역 네트워크(Wide-Area-Network)과 무선



(그림 1) 텔레매틱스 개념도

근거리 네트워크(Local-Area-Network), DMB(Digital Multimedia Broadcasting) 세가지로 분류할 수 있다. 각각의 무선 네트워크는 단말기의 이동성이나 전송 속도 등의 차이가 있다. 운전 중에는 주로 무선 광대역 네트워크를, 정차한 경우는 고속의 무선 근거리 네트워크를 사용하게 된다. DMB의 경우는 운전 중이나 정차 시 모두 서비스를 받을 수 있다. 또한 최근에는 단말기의 이동성과 고속의 전송 속도를 모두 지원하는 네트워크가 등장하고 있다.

텔레매틱스 단말기는 다양한 종류의 하드웨어와 소프트웨어들이 탑재된다. 단말기 하드웨어구조는 크게 단말기와 차량내의 다른 하드웨어들과의 통신구조로 나뉘볼 수 있다. 단말기 자체는 무선네트워크가 가능한 일종의 컴퓨터라고 볼 수 있으며, 단말기 이외의 하드웨어로는 차량 제어장치, A/V장치, 사운드 및 기타 디스플레이 장치 등을 포함한다. 단말기에는 다양한 종류의 텔레매틱스 응용프로그램들이 설치되어 운영된다. 단말기 소프트웨어 구조(또는 소프트웨어 플랫폼)는 이러한 응용프로그램들의 설치, 실행 및 관리를 위한 기본적인 틀을 제공해 준다. 텔레매틱스 발전과 더불어 최근에는 텔레매틱스 단말기 및 단말 소프트웨어 플랫폼을 위한 많은 연구 및 표준이 제안되고 있다.

텔레매틱스로 인하여 자동차 안에서 가능해지는 서비스는 매우 다양하다. 우선, 무선네트워크를 통해 인터넷에 연결된 모든 인터넷 서비스를 이용할 수 있다. 인터넷에 기반한 텔레매틱스 서비스로는 형태에 따라서 실시간 교통 정보, 뉴스 및 날씨 정보 등의 일반 정보제공서비스와 이메일, 웹브라우저와 같은 모바일 인터넷서비스, 주식투자, 모바일 전자상거래(M-Commerce)와 같은 금융서비스와, VOD(Video-On-Demand), 음악스트리밍 서비스, 온라인 게임 등의 엔터테인먼트 서비스 등을 들 수 있다.

인터넷을 통한 서비스와는 별도로, 또 하나의 중요한 서비스 형태로는 상황인식서비스(Context-Aware Service)가 있다. 기본적인 상황인식서비스는 GPS로부터 구해지는 위치 정보를 바탕으로 한 것이다. 예를 들면, 교통사고가 났을 경우 GPS위성을 이용해 자동적으로 사고차량의 위치를 추적, 가장 근접한 119구조대에 전달해 주는 응급구난 서비스를 제공할 수 있다. 또한, 자동 항법서비스(Navigation), 지리정보 및 교통상황정보를 기반으로 한 빠른 길 찾기 서비스 등이 그 예에 속한다. GPS 이외의 다른 센서정보를 이용한 상

황인식서비스도 있다. 예를 들면, 자동차의 제어장치 센서들과 연결된 단말기가 차량의 정보를 모아 무선 네트워크를 통해 차량 원격진단서비스 센터에 전송하면, 서비스 센터에서는 엔진 온도, 배기가스, 타이어, 오일등의 상태를 점검하여 이상 유무를 운전자에게 알려줄 수 있다. 더 나아가 차량의 속도계센서의 정보가 보험회사에 전송되고 보험회사는 이를 바탕으로 운전자의 자동차 보험 납부금을 정하는 서비스도 가능할 것이다.

이러한 센서기반의 텔레매틱스 서비스를 제공하기 위해서는 위에 언급한 '기본적인 텔레매틱스 인프라' 뿐만 아니라 '센서 데이터를 효과적으로 관리하는 인프라'가 필연적으로 요구된다. 그것이 바로 센서네트워크이다. 센서네트워크는 많은 영역에서 그 필요성이 제기되었고 그 동안 많은 연구가 있어왔다. 그리고 최근들어 텔레매틱스와 만나면서 서로의 영역이 확장되는 효과를 가져오고 있다. 다음 장에서는 센서네트워크에 대한 개략적인 설명과 함께 센서네트워크가 텔레매틱스를 어떻게 지원할 수 있는지에 대해 살펴보고자 한다.

II. 텔레매틱스 서비스 인프라스트럭처에 대한 센서 네트워크 활용

1. 텔레매틱스의 지속적인 정보 수요

텔레매틱스 서비스는 지속적인 정보 수집에 대한 다양한 수요를 포함한다. 예를 들면, 거시적인 측면에서는 대규모의 교통 정보 수집을 들 수 있다. 대부분의 자동차 운전자가 지속적으로 갖는 최대의 관심 중의 하나는 목적지까지 제 시간에 도착하는 것이다. 이러한 수요를 충족시키기 위해서는 현재 운전자의 위치와 목적지 사이의 경로 및 대체 경로 등을 포함하는 포괄적인 영역에 걸친 교통 상황 정보가 요구된다. 다른 한편으로 미시적인 측면으로는 차량의 기계적인 상태 정보 파악을 들 수 있다. 안전 운행에 필수적인 차량의 유지보수 및 고장 예방을 위해 엔진 등의 기계적인 상태에 대한 정보를 지속적으로 파악할 필요가 있다.

이러한 텔레매틱스 환경의 정보 수집 수요에는 센서 네트워크 기술이 적극적으로 활용될 수 있다. 센서 네트워크는

물리적인 대상이나 환경의 계측, 연산, 통신 기능을 갖춘 수 개~수십만 개 이상의 센서노드(sensor node)들 사이에 형성되는 네트워크 및 그 애플리케이션을 말한다. 교통 정보 수집을 위해서는 차량, 도로변, 톨게이트 등에서 차량 소통량 등을 감지하는 센서 네트워크를 구축할 수 있을 것이며, 차량 상태 파악을 위해서는 엔진룸 등에 진동이나 발열, 소음 등을 감지하는 센서 네트워크의 설치가 적합할 것이다.

적절한 기능의 센서들을 적재 적소에 설치하는 것은 센서 네트워크 활용의 시작에 지나지 않는다. 센서 네트워크는 저전력 통신, multi-hop 및 ad-hoc 네트워킹을 위한 라우팅 프로토콜, OS 및 DB 등의 경량(light-weight) 플랫폼, 센서 데이터 수집 및 처리를 위한 게이트웨이와 미들웨어, 네트워크 관리 및 유지 보수, 센서 네트워크를 활용하는 서비스의 편리한 개발을 목적으로 하는 개발 환경 및 인터페이스 등에 걸친 매우 포괄적인 주제이다. 센서 노드 하드웨어로는 미국 Berkeley 대학의 Mote 시리즈가 대표적이며, 저전력 근거리 통신 프로토콜로서는 이미 널리 보급된 Bluetooth와 Wi-Fi, 그리고 보다 저전력에 특화된 IEEE 802.15.4 규격에 기반한 ZigBee 등을 꼽을 수 있다. 한편, 이러한 센서 노드들 사이에 네트워크를 형성하는 데에는 기존의 통신 인프라를 활용하기 어려운 상황을 감안해 ad-hoc 방식의 multi-hop 네트워킹 기법이 널리 사용된다. 여기서 형성되는 네트워크의 topology에 따라 크게 flat routing과 hierarchical routing으로 나눌 수 있다. Flat routing은 네트워크를 구성하는 모든 센서 노드들이 동등한 역할을 수행하는 반면, hierarchical routing에서는 전체 네트워크를 다수의 계층적인 그룹으로 분할하여, 특정 센서 노드들이 해당 그룹의 대표 역할을 수행한다. Flat routing 방식으로 제안된 기법으로는 directed diffusion [19], SPIN [20], SAR [21] 등이 대표적이며, hierarchical routing 방식으로는 LEACH [22], TEEN [23], APTEEN [24] 등이 제안된 바 있다. 센서 플랫폼 및 데이터 관리 기술도 센서 네트워크의 중요한 주제이다. 컴퓨팅 환경이 열악한 센서 노드의 플랫폼으로서 TinyOS [29] 및 TinyDB [30] 등이 대표적으로 제안된 바 있다.

또한 센서 네트워크가 유용한 서비스로서 활용되기 위해서는 효과적인 데이터 관리가 필수적이다. 센서 네트워크에서 수집되는 데이터는 방대한 규모, 연속적인 전달과 같은 workload 특성이 있으며, 정보로서 유용한 가치를 가지려면

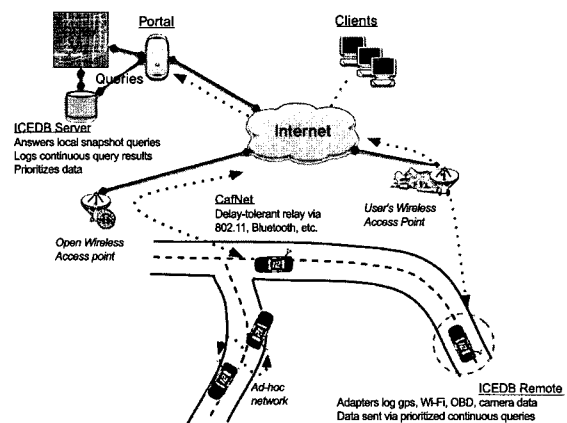
중복성 검사 및 신뢰성 보완과 같은 처리 등도 요구된다. 센서 네트워크 서비스에는 이러한 환경적 특징과 기능적 요구 사항을 충족하는 데이터 관리 기법의 지원이 전제되어야 하며, 이에 대해서는 III장에서 자세히 소개한다.

2. 센서 네트워크 활용 사례

여기서는 앞에서 간단히 예를 들었던 바와 같이, 텔레매틱스 서비스와 밀접하게 관련되었거나 활용이 가능한 실제 센서 네트워크 사례에 대해 살펴보고자 한다.

2.1 CarTel 프로젝트

MIT CSAIL에서 진행중인 CarTel [25] 프로젝트는 도로를 운행중인 자동차들에 각종 센서를 탑재하여, 이동하는 자동차들 사이의 센서 네트워크를 구현하려 한다. 개별 자동차에는 GPS 및 무선 통신 기능, 그리고 차량 엔진 진단 등을 비롯한 각종 센서가 포함된 장비가 장착된다. 자동차들은 일종의 센서 노드가 되어, 자신의 위치 정보 등을 비롯한 자기 상태 정보, 인근 차량 및 주변 환경에 관한 정보 등을 수집 또는 전달하는 주체가 된다. 이렇게 수집되는 정보는 인터넷을 통해 서버 측으로 전달되어 데이터베이스의 형태로 저장된다. 또한, 서버측에는 '포탈'이 있어서, CarTel 센서 네트워크에 대한 각종 질의(query)를 처리하고 서비스를 제공하는 역할을 수행한다. 그림 2에서 CarTel 시스템의 전반적인 구성을 소개하고 있다.



(그림 2) CarTel System Overview (출처: CarTel: A distributed mobile sensor computing system, SenSys 2006)

CarTel 시스템은 크게 다음의 세가지 요소로 구성된다.

CafNet: delay에 강인한 'carry and forward network' 스택. Multi-hop 네트워킹 방식으로 특정 자동차의 정보를 액세스하고 전달한다. Wi-Fi나 Bluetooth 등의 다양한 물리 계층 기술 위에서 동작 가능하다.

ICEDB: delay에 강인한 continuous query processor. 외부 사용자들은 포탈을 통해 CarTel에 대해 질의를 생성한다. 사용자의 질의는 일회성 질의(snapshot query) 또는 연속적 질의(continuous query)의 형태가 될 수 있으며, 이러한 질의에 대해 ICEDB 서버는 저장된 데이터를 대상으로 질의를 처리하거나 또는 CafNet을 통해 자동차들로부터 전달받는 스트림을 대상으로 질의를 처리한다.

AutoPortal: CarTel에서 얻어지는 정보를 웹 기반으로 시각화하고 질의를 입력받을 수 있는 사용자 인터페이스.

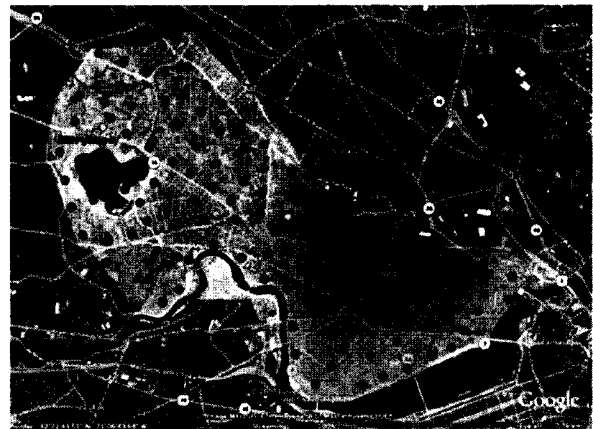
CarTel 시스템에서 얻어지는 자동차들의 GPS 위치 정보를 이용하면 교통 상황 모니터링 및 최적 경로 탐색과 같은 서비스를 제공할 수 있다. 위치 정보 외에 차량에 장착된 각종 진단 센서로부터 얻어지는 데이터를 활용하여 자동차 유지 및 보수 서비스도 가능하다. 또한 차량 및 교통 관련 정보에 국한되지 않고, 센서가 탑재된 차량이 넓은 지역에 걸쳐 이동성을 가진다는 점을 이용하여, 넓은 지역에 걸쳐 대기 오염이나 온도 등과 같은 각종 환경 정보, 그리고 도로 노면이나 교량의 상태나 파손 등에 대한 정보도 수집할 수 있을 것으로 전망하고 있다.

2.2 CitySense 프로젝트

텔레매틱스 애플리케이션에서 정보 생산 및 수집의 주체는 비단 자동차에 국한되지 않는다. 자동차들이 운행하는 도로 주변이나 신호등, 톨게이트 등에 센서를 위치시키면, 효과적으로 그 주변의 교통 관련 정보를 수집할 수 있을 것이다. 자동차에 설치되는 센서는 이동식인 반면, 이와 같이 도로상에 설치되는 센서는 고정식으로서, 고정된 위치에서 계속적으로 정보를 수집하고자 할 때 유용하다.

Harvard 대학에서 진행중인 CitySense [26] 프로젝트는 도심 지역에 고정형 센서들을 실제로 설치하고, 이것이 각종 센서 네트워크 서비스 개발에 테스트베드가 될 수 있도록 개발 환경을 제공하고자 한다. CitySense 프로젝트는 특징적

으로 도시의 가로등에 센서 노드를 설치하려 하고 있다. 무선 통신 및 데이터 처리 기능을 갖춘 센서 노드를 가로등에 설치함으로써 손쉽게 설치 공간을 확보할 수 있고, 무엇보다 가로등의 전력을 빌려 쓸 수 있기 때문에 센서 네트워크에서 흔히 발생하는 전력 문제를 간단히 해결할 수 있다는 장점이 있다. CitySense에서는 센서 노드의 풍부한 전력을 바탕으로 복잡한 연산 및 데이터 처리 및 고속의 네트워킹을 가능케 하여, 센서 네트워크의 활용 가능성을 대폭 확대하려 하고 있다.



(그림 3) CitySense 개념도 (출처: <http://www.citysense.net>)

CitySense에서는 (그림 3)과 같이 도심 지역에 센서 및 게이트웨이를 설치하고, 이를 각종 센서 네트워크 서비스 개발에 테스트베드로서 제공한다는 계획을 세우고 있다. 분산 인터넷 서비스 개발을 위한 테스트베드로서 대표적인 PlanetLab [28]과 개념적으로 유사하다고 볼 수 있다. 이를 위해 CitySense에서는 센서 네트워크를 대상으로 하는 상위 수준의 프로그래밍 모델, 다수의 참여자들 사이의 자원 배분을 위한 자원 할당 방법 및 할당 정책 등의 세부 연구 주제가 있다. 또한 Harvard 대학에서 스트림 데이터 수집 및 처리를 위한 인프라 구조로서 제안하였던 Hourglass를 CitySense에 적용하려는 시도도 있으며, 대기 오염 모니터링을 일차적인 예제 애플리케이션으로서 기획 중이다. CitySense는 현재 초기 단계로서, 아직 텔레매틱스 애플리케이션에 직접적인 활용을 목표로 하고 있지는 않지만 그 가능성은 충분하다고 볼 수 있다.

2.3 Linear Road Benchmark

텔레매틱스 서비스를 위해서는 넓은 영역에 걸친 지역 및 그 지역을 다니는 수많은 자동차들로부터 데이터를 수집하게 될 것이다. 이렇게 수집되는 데이터는 매우 막대한 분량이 될 것이며 끊임없이 연속적으로 발생하는 특성을 지닐 것이다. Linear Road [27] 프로젝트에서는 시뮬레이션을 통해 고속도로 환경에서 각종 교통 상황을 모델링하고 여기서 각종 정보를 수집하였다. 그리고 이러한 시뮬레이션을 통해 얻어진 정보를 처리하는데 Aurora 스트림 프로세서 및 일반 Relational DB를 각각 적용하여 그 성능을 비교 평가하였다.

Linear Road에서는 고속도로를 운행하는 차량의 수, 개별 차량의 속도, 정차나 고장 또는 사고 등과 같은 개별적인 사건들, 톨게이트 통과 등과 같은 요소들을 모델링 한다. 이를 기반으로 개별 자동차의 위치 정보를 비롯한 각종 정보가 계속적으로 수집된다. 이러한 정보를 처리하여 예상 목적지 도착 시간 등과 같은 여러 가지의 질의를 처리하는 상황을 상정하였다. 성능 평가에는 3GHz Pentium CPU와 2GHz RAM을 장착한 Linux 서버가 사용되었으며, 총 10개의 고속도로 구간(각각의 고속도로는 100 마일 길이)에 대해서 성능 평가가 이루어졌다. 그 결과, Aurora 스트림 프로세서는 하나의 서버당 2.5개의 고속도로 구간에 대해 필요한 성능 조건을 유지할 수 있었던 반면, Relational DB의 경우 하나의 서버당 0.5개의 고속도로 구간 만을 담당할 수 있었다. 이는 텔레매틱스 환경에서 발생하는 workload 처리에 기존의 Relational DB를 적용하기에는 효율적이지 않으며, 보다 적합한 데이터 관리 기법을 적용할 경우 5배 이상의 성능 향상을 가져올 수 있다는 결과를 보여주고 있다. 다음 장에서는 이 같은 텔레매틱스 환경에서의 데이터 관리에 대해 구체적으로 소개한다.

III. 텔레매틱스를 위한 센서데이터 관리

텔레매틱스 서비스를 제공하기 위해서는 이와 같은 센서 네트워크에서 발생하는 센서데이터를 효과적으로 처리할 수 있어야 한다. 즉, 다양한 센서네트워크로부터 빠르고 지

속적으로 유입되는 대용량의 데이터를 효과적으로 저장, 관리, 전달하여야 하며 이들 데이터에 대한 질의들도 빠르고 효율적으로 처리할 수 있어야 한다. 기존의 데이터베이스 관리시스템 (DBMS, Database Management System)은 이렇게 빠른 속도로 계속 유입되는 데이터 처리에는 적합하지 않다. 기존의 DBMS는 이미 저장되어 있는 데이터를 효율적으로 관리하고, 이에 대한 질의 처리에 최적화되어 설계되어 있기 때문이다. 따라서, 텔레매틱스 환경에서의 데이터 관리를 위해서는 기존의 DBMS와는 다른 새로운 데이터 관리 시스템이 필요하다. 본 장에서는 텔레매틱스 및 센서네트워크 환경에서 데이터 및 질의의 구조에는 어떠한 특징이 있으며 이를 위해 고안된 데이터 관리 시스템에는 어떠한 것들이 있는지 살펴본다.

1. 센서데이터 및 질의의 특징

텔레매틱스가 구현될 컴퓨팅 환경은 종전의 환경과는 확연히 다른 차이점이 있다. 예를 들면 자동차의 위치나 속도, 엔진 및 타이어 등 자동차의 상태, 운전자의 상태 등이 센서에 의해 감지 되어 주기적으로 계속 보고된다. 이러한 종류의 데이터는 실시간으로 계속 유입되는 특성이 있어 스트림 데이터라고 한다. 스트림 데이터는 기존의 데이터베이스가 처리하던 데이터들과는 다음과 같은 점에서 다른 특성을 보인다.

- 데이터들이 네트워크를 통해 실시간으로 계속 들어온다.
- 데이터베이스 시스템이 데이터들이 들어 오는 순서를 조정할 수 없다.
- 데이터의 크기가 이론적으로 거의 무제한이다.
- 데이터의 형태가 매우 다양하다.

이러한 스트림 데이터에 대해서는 기존의 데이터와는 달리 연속 질의(Continuous Query)가 많이 발생한다. 연속 질의란 일정 시간 동안 데이터 관리 시스템에 있으면서 질의 조건이 맞으면 그 결과를 사용자에게 알려주는 질의를 말한다. 예를 들어 텔레매틱스 환경에서는 다음과 같은 연속질의가 빈번히 요구될 수 있다.

특정 영역 내에서 60Km 이상으로 달리는 차들을 알려달라

스트림 데이터와 연속 질의를 처리하기 위해 요구되는 사항은 다음과 같이 요약될 수 있다.

- 순서(order)와 시간(time) 기반의 연산을 위한 데이터 모델과 질의가 허용되어야 한다.
- 스트림 데이터를 전부 저장할 수 없기에 되도록 좋은 요약 정보를 만들 수 있어야 한다.
- 질의에 대한 결과를 얻기 위하여 입력되는 모든 스트림 데이터를 대상으로 하는 블로킹(blocking) 연산을 사용하지 않아야 한다.
- 성능 및 데이터 저장소에 대한 제약으로 인하여 스트림 데이터에 대한 백트래킹(backtracking)은 적절하지 않다.
- 실시간으로 스트림 데이터를 모니터링하는 응용프로그램은 비정상적인 데이터에 대하여 빠르게 대응할 수 있어야 한다.
- 확장성을 위하여 많은 연속 질의들의 실행이 공유되어야 한다.

이와 같은 요구사항을 만족하기 위하여 기존의 DBMS를 이용하여 스트림 데이터를 일반 테이블에 대한 삽입으로 처리하고, 연속 질의는 트리거(trigger) 또는 객체화된 뷰(materialized view)로 처리할 수 있으나, 다음과 같은 문제점을 갖는다.

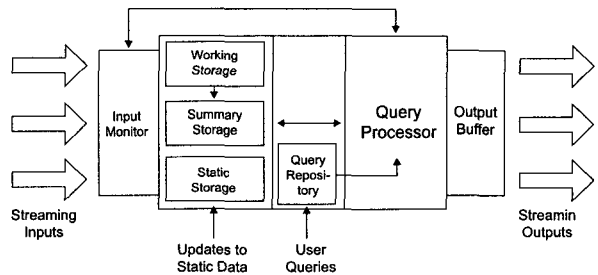
- 삽입 연산 부하가 크다.
- 트리거로 표현할 수 있는 조건이 제한적이며, 뷰에는 시퀀스를 표현할 수 없다.
- 근사치와 자원 할당을 표현할 수 없다.
- 지원하는 트리거의 수에 한계가 있다.
- 뷰는 스트림 결과를 제공하지 못한다.

이와 같은 문제점을 해결하고 위에서 언급한 데이터 스트림 및 연속질의를 처리하기 위해 요구되는 사항을 만족하기

위하여 데이터 스트림 매니지먼트 시스템(DSMS: Data Stream Management System)에 관한 연구가 진행되어 왔다.

(그림 4)는 DSMS의 일반적인 아키텍처를 보여준다. 입력 모니터(Input Monitor)는 입력되는 스트림 데이터의 양을 조절한다. 시스템이 스트림 데이터를 처리하는 속도가 입력되는 속도를 따라 잡지 못하면 입력되는 스트림 데이터를 버리는 것이 한 가지 조절 방법이 될 것이다. 데이터는 일반적으로 윈도우 질의(window query) 등을 처리하기 위해 임시적으로 저장되는 작업 저장소(Working Storage), 요약 정보를 지니기 위한 요약 저장소(Summary Storage), 데이터 소스의 위치와 같은 메타 정보를 저장하기 위한 정적 저장소(Static Storage) 중 하나에 저장된다. 연속 질의는 질의 저장소(Query Repository)에 등록되며, 공유 처리를 위해 그룹으로 묶여진다. 질의 처리기(Query Processor)는 입력 모니터에서 수집된 값을 기반으로 질의 처리 계획(query plan)에 대한 최적화와 실제 질의를 수행하여 결과를 출력 버퍼(Output Buffer)에 넣는 역할을 수행한다.

이 장의 나머지에서는 대표적인 데이터 스트림 매니지먼트 시스템에 대해서 간략히 설명하도록 하겠다.



(그림 4) 데이터 스트림 매니지먼트 시스템의 일반적인 아키텍처

현재까지 데이터 스트림 매니지먼트 시스템(DSMS: Data Stream Management System)에 관한 연구는 크게 두 단계로 진행되어 왔다. 첫 번째는 단일 노드에서 동작하는 DSMS이다. 초창기의 DSMS 연구는 주로 단일 노드에서, 위에서 언급한 데이터 스트림 및 질의의 새로운 요구 조건들을 어떻게 해결할 수 있을가에 대한 것이었다. 그리고 두 번째는 이

상의 단일 노드에서 동작하는 DSMS를 분산 시스템으로 확장하는 연구이다.

2.1 단일 노드 DSMS

대표적인 단일 노드 DSMS로는 위스콘신(Wisconsin)대학의 NiagaraCQ[14], 버클리(UC Berkeley) 대학의 TelegraphCQ[15], Brown 대학, Brandeis 대학, MIT 세 학교가 공동으로 개발한 Aurora[11] 등이 있다. 이상의 시스템들은 단일 노드에서 데이터 스트림을 관리한다는 점에서 공통점이 있지만, 그 각각이 주안점을 두어 개발한 부분, 즉 시스템의 핵심적인 특징은 서로 다르다.

NiagaraCQ 시스템은, 데이터를 XML로 모델링 했다는 점이 가장 큰 특징이다. NiagaraCQ 시스템은 인터넷 상에 있는 XML 문서들에 대해 XML-QL(Query Language)을 통해서 표현된 연속질의를 처리할 수 있도록 한다. NiagaraCQ에서는 많은 수의 연속 질의들이 그 형태가 유사하다는 관찰 하에 이러한 질의들을 그룹화해서 처리하도록 설계하였다.

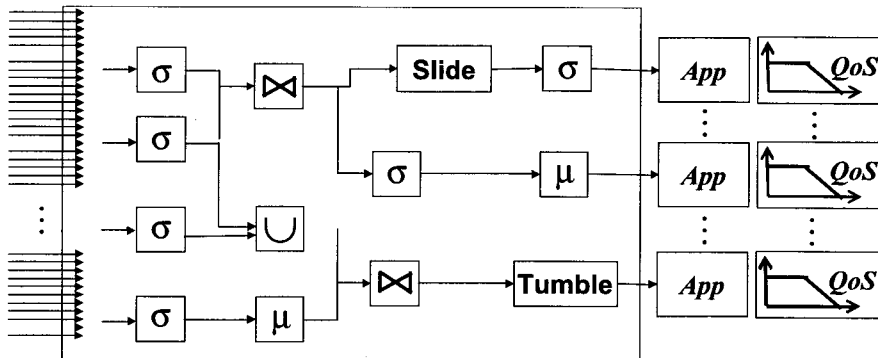
TelegraphCQ 시스템은 다른 DSMS와 같이 많은 양의 계속 변하는 스트림 데이터를 관리하고 그에 대한 연속 질의를 처리하는 시스템이다. TelegraphCQ는 연속 질의를 다루는 다른 시스템에 비하여 특히 공유(sharing)와 적응성(adaptivity)에 중점을 맞추어 디자인 되었다는 특징이 있다. 각 연속 질의는 여러 개의 질의처리모듈에 분리되어 저장되는데 이때 각 모듈에서는 저장되는 질의들을 공유 인덱스(index)로 관리함으로써 질의처리 효율을 높인다. 데이터의 처리는 이 데이터와 관련된 질의처리모듈을 거치면

(routing)서 이루어지는데, 이때 질의처리의 효율성, 우선 순위, 로드 밸런싱 등의 요건에 의해 그 라우팅(routing) 순서가 결정된다. 이를 적응적 라우팅(adaptive routing)이라고 한다.

Aurora 시스템도 대용량의 연속적인 스트림 데이터와 연속 질의를 처리하는 DSMS이다. Aurora의 가장 중요한 특징은 QoS(Quality of Service)에 기반하여 질의처리를 조정할 수 있다는 점에 있다. (그림 5)는 Aurora의 질의처리모델을 나타낸다. Aurora는 입력되는 데이터 스트림에 대하여 다수의 질의를 동시에 처리한 후 응용프로그램에게 질의 처리 결과를 전송해준다. 질의는 각 응용프로그램에 의해 정의된다. 그림에서 각 질의는 연산자 박스와 데이터 스트림을 나타내는 화살표로 구성된 워크 플로우 다이어그램으로 표현된다. 그리고 각 응용프로그램은 부분적인 질의 결과 또는 지연되는 질의 결과의 효율성을 설명하는 QoS를 정의한다. 스케줄러(Scheduler)는 QoS 요구사항을 참조하여 어느 한 시점에서 어떤 연산자 박스에서 처리할 지를 결정하여 그 박스를 스케줄링 해준다.

2.2 분산 DSMS

그 다음으로 분산 데이터 스트림 매니지먼트 시스템을 살펴보기로 하자. 현재 잘 알려진 분산 DSMS에는 UC Berkeley의 HiFi[16], Harvard의 Hourglass[17] 그리고 MIT, Brown 대학, Brandeis 대학의 Borealis[18]가 있다. 이 시스템들의 특징으로는 주로 단일 노드에서 동작하는 DSMS를 개발했던 대학 및 연구소에서 이를 분산화 하여 분산 DSMS를



(그림 5) Airpra 시스템 모델 (출처: Airpra: a new model and architecture for data stream management, In VLDB Journal 2003)

개발했다는 점이다. 시스템 별로 각각 특징을 가지지만, 분산 DSMS라면 반드시 풀어야 하는 몇몇 문제들이 공통적으로 존재한다. 우선, 분산 DSMS는 다양한 종류의 장치들, 즉 각종 센서, 컴퓨터, 다양한 단말기 등을 다루어야 한다. 둘째, 분산 DSMS에는 많은 양의 데이터 입력이 들어오기 때문에 이 데이터 스트림에 대한 질의를 확장성있게 효율적으로 처리할 수 있어야 한다. 이 장의 나머지 부분에서는 각 분산 DSMS의 특징과 함께 초점을 맞추고 있는 문제와 제시하고 있는 해결책을 살펴본다.

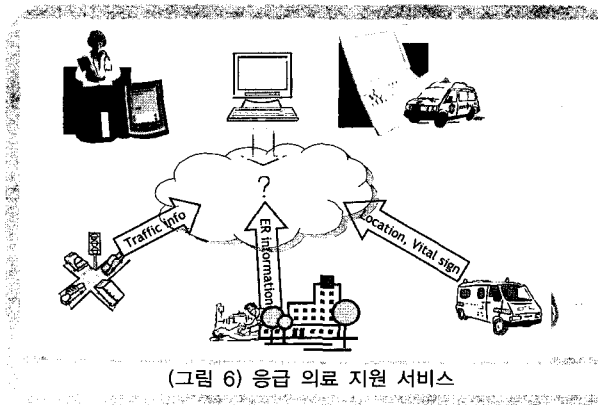
HiFi는 TelegraphCQ를 개발했던 UC Berkeley에서 개발한 분산 DSMS이다. HiFi란 이름은 'High Fan-in'의 약자로서 '다수의 입력 데이터를 효율적으로 처리해 주는 시스템'이라는 의미를 내포하고 있다. 즉 HiFi에서는 널리 퍼져있는 다수의 RFID 리더나 센서 등에서 데이터가 들어오고 각각의 입력 데이터의 양도 아주 많은 환경을 가정한다. 이 경우 그 대역폭(Bandwidth)과 처리 비용(Processing Cost)이 심각한 문제가 된다. HiFi에서는 전체 시스템을 완전한 계층구조(Hierarchical structure)로 구성함으로써 이러한 문제를 해결하고자 하였다. 즉, 입력 데이터가 일련의 호스트 컴퓨터들을 연속적으로 지나는 동안 그 양이 현격히 줄어들기 때문에 상위 단계에서는 그 처리 비용이 상당히 낮아진다. 이 때 중요한 점은 HiFi는 계층 구조의 각 계층에서 처리 되어야 하는 작업을 5단계로 정의했다는 점이다. 이를 CSAVA (Clean, Smooth Arbitrate, Validate, Analyze)처리라고 한다.

Harvard에서 개발한 Hourglass는 이름 그대로 마치 모래시계(Hourglass)의 가운데 잘룩한 부분이 연속된 모래의 흐름을 위에서 아래로 중계하는 것처럼, 다수의 센서 네트워크

로부터 발생된 데이터 스트림을 다수의 응용프로그램에 전달하는 인터넷 기반 시스템이다. Hourglass는 서로 연결된 다수의 서버들이 오버레이 네트워크를 이루는 구조를 지니며, 이 서버들은 서비스를 등록해주고 원하는 서비스를 찾아주며 데이터 스트림을 클라이언트에 전송하는 역할을 한다. 또한, Hourglass도 다른 분산 DSMS와 마찬가지로 데이터 스트림의 필터링, 통합, 압축, 버퍼링 등을 지원한다. Hourglass는 특정한 환경의 응용프로그램에 초점을 두고 있는 것이 아니라, 다수의 센서 네트워크와 다양한 응용프로그램을 지원하기 위한 기능과 안정적인 데이터 스트림의 전송에 초점을 두고 있다. 이렇게 다수의 센서 네트워크와 응용프로그램을 지원하기 위해 Hourglass에서 풀고자 하는 문제들은 다음과 같다. 첫째, 서비스 검색(Discovery)이다. 이는 응용프로그램이 원하는 서비스를 찾아줄 수 있어야 한다는 것이다. 이 경우 데이터 스트림의 제공도 이러한 서비스에 포함된다. 둘째, 빠른 데이터 전송(Fast Delivery)이다. 셋째, 장치의 다양성(Heterogeneity) 지원이다. 이는 장치마다 그 처리능력이나 저장능력, 사용하는 대역폭 등이 상당히 차가 있기 때문이다. 마지막으로 간헐적 연결(Intermittent connectivity) 지원이다. Hourglass에서는 다양한 종류의 장치들을 지원하기 때문에 무선으로 연결된 장치나 센서들의 경우에는 간헐적으로 그 연결이 끊길 수 있다.

이와 같은 환경의 예로써 다수의 병원과 의료 지원 시설, 앰블런스 회사, 119 서비스 등으로 구성된 응급 의료 서비스 시나리오를 들 수 있다. 이 시나리오는 다수의 센서 네트워크를 포함한다. 환자에 붙어 있는 무선 생체신호 센서(심전도, 심장 박동수 등), 교통 상황을 나타내는 교통 센서, GPS를 통한 앰블런스의 위치, 사용 가능한 병원과 병원 내의 침실 또는 응급실 정보, 보다 특별한 의료 서비스를 위한 의료 지원 시설 정보 등이 그 예이다. (그림 6참조) 다수의 센서 네트워크에서 생성된 정보들은 앰블런스나 병원, 의료지원 시설에 있는 휴대 컴퓨터나 PDA, 각 터미널들에 전달되어 실시간으로 환자의 상태나 앰블런스, 병원, 의료 시설 등의 위치를 획득하는 것이 가능하다.

Borealis도 역시 응용프로그램이 데이터 스트림에 대해 쉽게 질의할 수 있도록 도와준다. 하지만 Borealis는 다른 시스템과는 달리 실시간 질의 처리를 어떻게 하면 보다 효율적으로 할 수 있는 지에 대해 보다 초점을 맞추고 있다. 따라서



Borealis는 다음의 세가지 문제를 주로 다룬다. 첫째, 처리 결과의 능동적 수정이다. 센서 처리 결과는 일시적으로 불안정한 경우가 종종 발생하기 때문에 이런 경우가 발생했을 때 시스템이 이전 데이터 값을 정확히 수정해 주지 않는다면 응용프로그램은 계속해서 불안정한 결과를 유지할 수밖에 없게 된다. 둘째, 능동적인 질의의 수정이다. 일반적으로 많은 응용프로그램은 실시간으로 질의를 일부 바꾸기를 원하기 때문에 이를 잘 지원할 수 있어야 한다. 셋째, 확장성과 탄력성을 보장하는 질의 최적화이다. 분산 스트림 처리에 참여하는 장치들은 다양한 시스템 환경을 갖기 때문에 자원을 어떻게 할당해야 시스템 최적화를 이룰 수 있는가 는 쉽지 않은 문제이다.

이를 해결하기 위한 아이디어를 간단히 정리하면 다음과 같다. 우선 새로운 데이터 모델을 제안함으로써 처리 결과를 능동적으로 수정할 수 있도록 지원한다. 능동적인 질의 수정을 위해서는, 질의 전체를 바꾸는 방법 대신 파라미터를 바꾸는 방법으로 적은 오버헤드와 빠른 적용을 가능하게 한다. 마지막으로 확장성과 탄력성을 보장하는 질의 최적화를 위해서는, 데이터마다 VM(Vector of Metrics) QoS metric을 저장하게 하고 오퍼레이터를 거칠 때마다 VM이 어떻게 변하는 지에 대한 기록을 남긴 후, 이 VM의 정보를 이용해 QoS 함수를 적용함으로써 질의 최적화를 가능하게 한다.

IV. 결 론

텔레매틱스는 자동차 탑승자에게 다양한 종류의 정보서비스를 제공해준다. 특히, 다양한 센서 데이터를 활용함으로써 그 서비스의 종류와 질을 풍부하게 한다. 이를 위해서는 센서 데이터를 효과적으로 관리하는 것이 아주 중요하며 이를 위한 인프라가 바로 센서네트워크이다. 센서 데이터와 이에 관한 질의는 대용량과 연속성 등 그 특징이 기존의 것과는 상당히 다르다. 따라서 이를 효과적으로 처리할 수 있는 시스템이 최근 들어 많이 개발되고 있다. 본 글에서는 텔레매틱스 및 텔레매틱스 서비스를 위한 센서네트워크의 활용 방법, 실제 사례, 그리고 센서데이터를 효과적으로 관리하는 방법에 대해 알아보았다.

참 고 문 헌

- [1] 송준화 외 공저, "텔레매틱스 개론 (개정판)", 홍릉과학출판사, 2007
- [2] Kyungmin Cho, Sungjae Jo, Hyukjae Jang, Su Myeon Kim, Junehwa Song, "DCF: An Efficient Data Stream Clustering Framework for Streaming Applications", Proc. of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), Krakow, Poland, Sep. 2006.
- [3] J. Lee, Y. Lee, S. Kang, S. Lee, H. Jin, B. Kim, and J. Song, "BMQ-Index: Shared and Incremental Processing of Border Monitoring Queries over Data Streams", Proc. of the 7th International Conference on Mobile Data Management (MDM'06), In cooperation with ACM SIGMOBILE and ACM SIGMOD, Nara, Japan, May 2006
- [4] Inseok Hwang, Qi Han, and Archan Misra, "MASTAQ: Middleware Architecture for Sensor Applications with Statistical Quality Constraints", Proc. of the First International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2005), in conjunction with PerCom'05, Kauai Island, Hawaii, USA, March 2005.
- [5] 김용관, "국내외 서비스 동향 및 시장동향", TTA저널 89호
- [6] 오현서, "텔레매틱스 무선 액세스 기술", TTA저널 89호
- [7] 이형석, "텔레매틱스 단말 플랫폼 기술", TTA저널 89호
- [8] 문형돈, "텔레매틱스 기술 및 시장 동향", 주가기술동향, 통권 1090호, ETRI, 2003.4
- [9] 윤두영, 김봉준, "텔레매틱스 서비스 현황 및 전망", 정보통신정책, 통권365호, KISDI, 2005.3
- [10] D. Carney, U. Cetintemel, M. Chemiack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik. Monitoring Streams: A New Class of Data Management Applications. In proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02), Hong Kong, China, August 2002

- [11] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik. Aurora: A New Model and Architecture for Data Stream Management. In VLDB Journal, August 2003
- [12] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. In Proc. ACM SIGMOD/SIGACT Conf. on Princ. of Database Syst., pages 1-16, Madison, Wisconsin, USA, June 2002.
- [13] P. Bonnet, J. Gehrke and P. Seshadri, "Querying the Physical World," IEEE Pers. Commun., vol. 7, Oct. 2000, pp. 10-15.
- [14] J. Chen, D. DeWitt, F. Tian, and Y. Wang. NiagaraCQ: A scalable continuous query system for internet databases. In Proc. of the ACM SIGMOD Conf. on Management of Data, 2000.
- [15] Sailesh Krishnamurthy, Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, Samuel R. Madden, Vijayshankar Raman, Fred Reiss, and Mehul A. Shah. TelegraphCQ: An Architectural Status Report. IEEE Data Engineering Bulletin, Vol 26(1), March 2003.
- [16] M. Franklin, S. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, and W. Hong, Design Considerations for High Fan-in Systems: The HiFi Approach, UCB, CIDR 2005.
- [17] Jeffrey Shneidman, Peter Pietzuch, Jonathan Ledlie, Mema Roussopoulos, Margo Seltzer, Matt Welsh, Hourglass: An Infrastructure for Connecting Sensor Networks and Applications, Harvard Technical Report TR-21-04, 2004.
- [18] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryzkina, N. Tatbul, Y. Xing, and S. Zdonik, The Design of the Borealis Stream Processing Engine, MIT, Brown, Brandeis, CIDR 2005.
- [19] C. Intanagonwiwat, et al., Directed Diffusion: A scalable and robust communication paradigm for sensor networks, In MobiCOM '00, August 2000.
- [20] W.R. Heizelman et al., "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," Proc. ACM Mobicom'99, 1999, pp.174-185
- [21] K. Sorabi et al., "Protocols for Self-Organization of a Wireless Sensor Network," IEEE Personal Communication, Vol.7, No.5, 2000, pp.16-27.
- [22] W. Heizelman, et al., "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Trans. on Wireless Communications, Vol.1, No.4, Oct. 2002, pp.660-670.
- [23] Arati Manjeshwar et al., "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," Proc. Second Int'l Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 2001.
- [24] Arati Manjeshwar et al., "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," IEEE Proc. Of the Int'l. Parallel and Distributed Processing Symposium(IPDPS'02), Apr. 2002, pp.195-202.
- [25] CarTel: <http://cartel.csail.mit.edu/>
- [26] CitySense: <http://citysense.net/>
- [27] A. Arasu, et al., Linear Road: A Stream Data Management Benchmark, VLDB 2004, August, 2004.
- [28] PlanetLab: <http://www.planet-lab.org/>
- [29] TinyOS: <http://www.tinyos.net/>
- [30] Sam Madden, Michael J. Franklin, Joseph M. Hellerstein and Wei Hong. TinyDB: An Acquisitional Query Processing System for Sensor Networks. ACM TODS, 2005.

약 력



2002년 KAIST 학사
2005년 KAIST 석사
2005년 ~ 현재 KAIST 박사과정
관심분야: 유비쿼터스 컴퓨팅, 분산 시스템

이 현 익



2001년 KAIST 학사
2002년 KAIST 석사
2002년 ~ 현재 KAIST 박사과정
관심분야: 유비쿼터스 시스템, 분산 시스템, 인터넷 플랫폼

황 인 석



1997년 University of Maryland 박사
2000년 ~ 현재 KAIST 전산학과 부교수
관심분야: 컴퓨터 네트워크, 유비쿼터스 플랫폼 및 시스템,
인터넷 기술, 무선 인터넷 및 이동 컴퓨팅 기술, 멀티미디어
기술, 분산시스템 기술, 컴퓨터 시스템, 문화 기술

송 준 화

