

유무선 단말기를 이용한 분산 시뮬레이터 설계

조수현¹ · 이종식[†]

Design for Distributed Simulator using Wired and Wireless Devices

Soo-Hyun Jo · Jong-Sik Lee

ABSTRACT

Wireless communication technology developed, portable wireless devices based on wireless network including the smartphone rapidly developed, and products with functions similar to highly efficient computer are being launched. Therefore, active studies to enable performance of works, which were previously held in computers, on portable wireless terminal are being held. Until now, simulation was undertaken using computers which satisfy high performance computing resource. Such contains a problem that it cannot break away the limit of space. Portable wireless terminal has the merit that it can break away from the limit of space by using wireless network in comparison to wired environment. In this paper, it suggests a simulator based on wireless devices for distributed simulation. For such purpose, the performance time of simulation will be compared in regards to the case of performing nodes control simulation in a single highly efficient computer and the case of performing distributed simulation in many wired and wireless network environments. Through the comparison, the possibility for utility of portable wireless terminal as the object of distributed simulation will be verified, and it will be presented that simulation can be performed in a shorter time by performing distributed simulation under various wired and wireless network environments away from the limit of space.

Key words : Wireless Device, Distributed Simulator, DEVS

요약

무선 통신 기술이 발전함에 따라 스마트폰을 비롯한 무선 네트워크 기반의 휴대용 무선 단말기가 비약적으로 발전하여, 고성능 컴퓨터와 비슷한 성능의 제품이 출시되고 있다. 이에 따라 기존의 컴퓨터에서 수행하던 작업을 휴대용 무선 단말기에서 수행하려는 연구가 활발히 진행되고 있다. 지금까지 시뮬레이션은 고성능의 컴퓨팅 자원을 충족한 컴퓨터를 이용하여 수행해왔다. 이는 공간적인 제약에서 벗어날 수 없다는 문제점을 안고 있다. 휴대용 무선 단말기는 무선 네트워크를 이용하여 유선 환경에 비해 공간적인 제약에서 벗어날 수 있다는 장점을 지니고 있다. 본 논문에서는 분산 시뮬레이션을 위한 휴대용 무선 단말기 기반의 시뮬레이션 환경을 제안한다. 이를 위해 도식의 이동 노드의 시뮬레이션을 단일의 고성능 컴퓨터에서 시뮬레이션을 수행한 경우와 다수의 유선 및 무선 네트워크 환경의 단말기에서 분산 시뮬레이션을 수행한 경우에 대하여 시뮬레이션 수행 시간을 비교하고자 한다. 이를 통해 휴대용 무선 단말기의 분산 시뮬레이션 객체로서의 효율 가능성을 확인하고, 공간 제약에서 벗어나 다양한 유선 및 무선 네트워크 환경에서 분산 시뮬레이션을 수행함으로써 보다 짧은 시간 안에 시뮬레이션을 수행할 수 있음을 보여준다.

주요어 : 휴대용 무선 단말기, 분산 시뮬레이터, DEVS

접수일(2011년 4월 27일), 심사일(1차 : 2011년 9월 9일),
게재 확정일(2011년 9월 9일)

¹ 인하대학교 컴퓨터 정보공학과

주 저 자 : 조수현

교신저자 : 이종식

E-mail : jslee@inha.ac.kr

1. 서론

지금까지 시뮬레이션은 지역적으로 분산된 환경 하에서 진행하지 못하고, 정해진 공간에서 고성능 컴퓨터를 통해 수행되어 왔다. 하지만 근래에 와서 유비쿼터스 기술의 발전에 따라 작업 환경이 확장되고 있으며, 기존의 시뮬레

표 1. 세계 스마트폰 시장 전망^[2] (단위 : 백만대, %)

	휴대폰 판매대수	스마트폰	
		판매대수	비율
2007년	1,151	121	10.5
2008년	1,209	143	11.8
2009년	1,114	178	15.9
2010년	1,202	254	21.1
2011년	1,306	351	26.9
2012년	1,432	469	32.8
2013년	1,568	604	38.5

이션 환경은 유비쿼터스 기술의 이동성^[1]을 충족하지 못하고 있다.

최근 들어 무선 통신 기술이 발전함에 따라 스마트폰을 비롯한 무선 네트워크 기반의 휴대용 무선 단말기도 비약적으로 발전하여, 고성능 컴퓨터와 비슷한 성능의 제품이 출시되고 있으며, 이에 따라 기존의 컴퓨터에서 수행하던 작업을 휴대용 무선 단말기에서 수행하려는 연구가 활발히 진행되고 있다. 특히 스마트폰 시장은 표 1과 같이 높은 성장세를 지속하고 있다.

본 논문에서는 분산 시뮬레이션을 위한 휴대용 무선 단말기 기반의 시뮬레이션 환경을 제안한다. 이를 위해 도심의 이동 노드 흐름 제어 시뮬레이션을 Dijkstra algorithm과 Greedy algorithm을 이용하여 경로를 생성함으로써 단일의 고성능 컴퓨터에서 시뮬레이션 한 경우와, 다수의 유선 및 무선 네트워크 환경의 단말기에서 분산 시뮬레이션 한 경우에 대하여 시뮬레이션 수행 시간을 비교하고자 한다. 이를 통해 휴대용 무선 단말기의 분산 시뮬레이션 객체로서의 효용 가능성을 확인하고, 공간 제약에서 벗어나 다양한 유선 및 무선 네트워크 환경에서 분산 시뮬레이션을 수행할 수 있음을 보이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구에 관해 다루고, 3장에서는 본 논문에서 개발된 유무선 단말기를 이용한 분산 시뮬레이션과 Time manager에 관해 다루며, 4장에서는 Dijkstra algorithm과 Greedy algorithm을 도심 지역 데이터에 적용하여 단일 컴퓨터를 이용한 시뮬레이션과 유무선 단말기를 이용한 분산 시뮬레이션에서 이동 노드의 증가에 따른 시뮬레이션 수행 시간을 비교 실험하고 분석함으로써 유무선 단말기를 이용한 분산 시뮬레이션의 효율성을 입증한다. 마지막으로 5장에서는 향후 연구 방향과 연구의 필요성에 관해 정리함으로써 논문의 결론을 맺는다.

2. 관련 연구

2.1 DEVS 형식론

DEVS는 이산 사건 시스템을 위한 형식론^[3]이다. DEVS model은 Atomic model과 Coupled model로 구분되며, 이를 이용하여 계층적인 모델 구성이 가능하다. Atomic model은 시스템의 기본적인 구성요소를 나타내며, Coupled model은 연결을 기술한다. Atomic model 또는 Coupled model은 결합하여 새로운 Coupled model을 구성하여 더욱 큰 시스템을 나타낼 수 있다.

DEVS 형식론에서 각 모델은 미리 정해진 함수로 구성된 수동적인 객체이며, 실제 기능을 수행하는 것은 각 모델에 연결되어 있는 시뮬레이션 프로세스이다. Atomic model과 Coupled model에는 각각 시뮬레이터와 코디네이터라는 시뮬레이션 프로세스^[4]가 존재한다.

DEVS는 수학적 형식론에 기반 하여 모델의 재사용이 가능하며, 논리적인 분석 등의 기법을 활용할 수 있는 장점을 가지고 있다^[5].

2.2 분산 시뮬레이션

분산 시뮬레이션은 시뮬레이션 객체들을 다수의 컴퓨터들에 분산하여 시뮬레이션을 수행함으로써 실행 시간을 단축하는 데 그 목적이 있다. 분산 시뮬레이션은 비동기적인 분산된 프로세스들을 서로 연결하여 실행함으로써 구현할 수 있다. 동기화는 분산 시뮬레이션 알고리즘에서 중요한 요구 사항이다^[7].

분산 시뮬레이션은 하나의 프로세스가 다수의 논리 프로세스(LP : logical process)로 나누어 처리된다고 할 수 있다. 이 때, 각 LP는 할당된 사건에 대해 시뮬레이션을 진행하며 지역 가상 시간(LVT : logical virtual time)을 증가시킨다. 각 LP에 분산된 사건들은 인과성 제약(causality constraints)에 의해 단일 프로세스에서 시뮬레이션 되는 것과 같은 순서로 처리되어야 하며, 만약 순서대로 처리되지 않는다면, 인과성 오류(causality error)가 발생한다^[6].

분산 시뮬레이션의 동기화 방법은 크게 보수적 알고리즘과 낙관적 알고리즘으로 나눌 수 있다. 보수적 알고리즘은 시뮬레이션 실행 시간 동안 처리되는 사건의 순서를 엄격하게 유지하는 방법이며, 낙관적 알고리즘은 사건 처리 후, 인과성 오류가 발생 시 롤백을 통해 잘못 처리한 사건을 무효화한 후, 사건의 시간 순서대로 다시 처리하는 방법이다. 각 LP는 서로 영향을 받으며, 메시지의 교환을 통해 상호작용을 한다. 이 때, 메시지는 이전 메시지보다 큰 Time Stamp^[7]를 포함하여 인과성 오류를 방지한다.

2.3 무선 단말기를 이용한 시뮬레이션 수행 방법

무선 단말기를 이용하여 시뮬레이션을 수행하는 방법에는 웹-서비스를 이용한 시뮬레이션, 소켓 통신을 이용한 시뮬레이션, 분산 객체를 이용한 시뮬레이션으로 총 세 가지^[1]가 있다.

첫 번째로, 웹-서비스를 이용한 시뮬레이션은 그림 1과 같이 서버의 역할을 담당하는 단일 컴퓨터에서 모든 시뮬레이션을 실행하고, 무선 단말기의 클라이언트는 지속적으로 시뮬레이션의 과정과 결과를 전달 받는 방법이다.

이 때, 웹 서비스의 사용은 무선 단말기에서 지원하는 웹 브라우저를 이용하여 접속하므로, 웹 브라우저만 지원한다면 일반적으로 사용하는 컴퓨터에서도 사용이 가능하며, 무선 단말기의 운영 체제에 관계없이 사용이 가능하다. 웹 브라우저의 페이지는 지속적으로 일정한 시간마다 화면을 갱신하여 출력한다^[1].

두 번째로, 소켓 통신을 이용한 시뮬레이션은 그림 2와 같다.

소켓 통신을 이용한 시뮬레이션은 웹-서비스를 이용한 시뮬레이션과 마찬가지로 서버를 담당하는 단일 컴퓨터에서 모든 시뮬레이션을 실행하고 무선 단말기의 클라이언트는 지속적으로 시뮬레이션의 과정과 결과를 전달 받는 방법이며, 웹 서비스가 아닌 소켓을 통해 시뮬레이션의 과정과 결과를 전달받는다는 점에서 웹-서비스를 이용한 시뮬레이션과 차이점이 있다^[1].

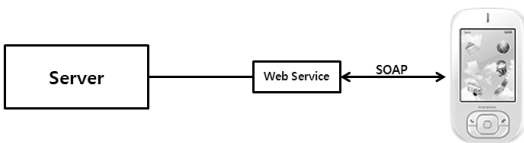


그림 1. 웹-서비스를 이용한 시뮬레이션

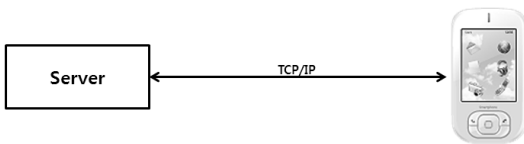


그림 2. 소켓 통신을 이용한 시뮬레이션



그림 3. 분산 객체를 이용한 시뮬레이션

마지막으로 그림 3은 분산 객체를 이용한 시뮬레이션이다.

분산 객체를 이용한 시뮬레이션은 각각의 무선 단말기가 분산 객체로서 시뮬레이션에 직접 참여하고 소켓을 통하여 타 분산 객체와 상호 작용하는 방법이다^[1].

본 논문에서는 분산 객체를 이용한 시뮬레이션 방법으로 실험을 진행하였다.

3. 유무선 단말기를 이용한 분산 시뮬레이션

3장에서는 본 논문에서 개발된 유무선 단말기를 혼용한 분산 시뮬레이션에 대해 소개한다.

본 논문에서 개발된 분산 시뮬레이터는 각 유무선 단말기마다 실행되어 객체로서 시뮬레이션을 수행할 수 있도록 개발되었다. 이 때, 모든 객체들을 관리하고 제어하는 시뮬레이션 서버가 존재하며, 시뮬레이션 서버는 각 객체의 IP 주소와 Port 번호를 매트릭스의 형태로 저장하여, 객체로부터 메시지 수신 시, 이를 목적지로 전달하고 큐에 저장하여 순차적으로 수행하여 시뮬레이션 시간 관리 및 시뮬레이션 제어를 담당한다. 이를 통해 객체 간의 상호 작용을 보장한다.

시뮬레이션 수행 시 그림 4와 같이 시뮬레이션 서버에

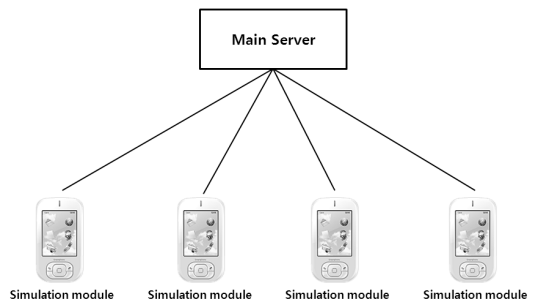


그림 4. 무선 단말기를 이용한 분산 시뮬레이션의 형태

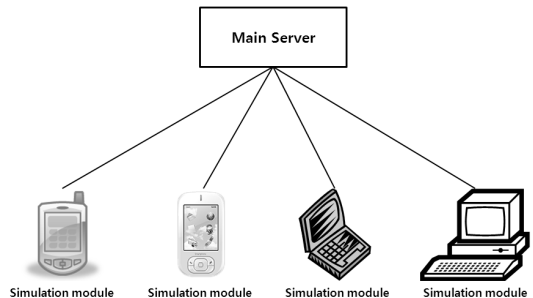


그림 5. 유무선 단말기를 혼용한 분산 시뮬레이션의 형태

무선 단말기들이 객체가 되어 접속하여 시뮬레이션을 수행할 수 있다.

그러나 본 논문에서 개발된 시뮬레이터는 어플리케이션의 형태로 구현되었으므로, 그림 5와 같이 무선 단말기는 물론 유선 네트워크 환경의 일반 컴퓨터에서도 동일한 사용이 가능하다.

3.1 분산 시뮬레이터 설계

개발된 분산 시뮬레이터는 그림 6과 같이 크게 시뮬레이션 서버와 시뮬레이션 클라이언트로 구성된다.

시뮬레이션 클라이언트는 유선 또는 무선 단말기들에서 구동되며, 각각 Atomic model이 되어, 시뮬레이션에 참여한다. 이 때, 시뮬레이션 서버는 전체 시뮬레이션을 제어하고 관리하며, 시뮬레이션 서버와 시뮬레이션 클라이언트는 .NET Framework 3.0에서 제공하는 TCP/IP 기반의 networking API를 통해 메시지를 전송한다.

시뮬레이션 클라이언트는 시뮬레이션 서버에 접속 후, 자신의 name과 source port와 destination port를 등록한다. 시뮬레이션 서버는 이를 매트릭스로 구성하여 관리한다.

시뮬레이션 서버와 시뮬레이션 클라이언트 간 사용되는 메시지 형식은 식 (1)과 같다.

$$\begin{aligned}
 & \text{order ID@source port name} \\
 & \text{@jobs index@simulation time} \\
 & \text{@value 1@value 2.. ..@value n/}
 \end{aligned} \tag{1}$$

order ID는 메시지에 포함된 명령에 관련된 식별자로 헤더에 해당된다. order ID는 다음과 같다.

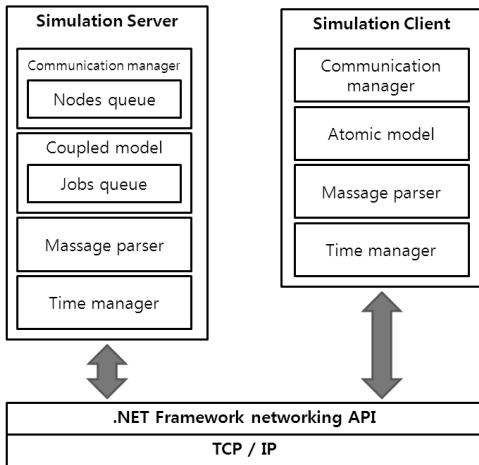


그림 6. 개발된 분산 시뮬레이터의 구성

- 1) reg : 서버에 Atomic model을 등록하고, name과 arrival time을 등록한다.
- 2) inp : Atomic model에 input port를 등록한다.
- 3) out : Atomic model에 out port를 등록한다.
- 4) con : 각 Atomic model의 input port 과 out port 사이에 coupling을 설정한다.
- 5) snd : output port를 통해 메시지를 서버에 전송한다.

source port name은 서버에 메시지를 전송한 out port이며, jobs index는 메시지의 번호이다. simulation time은 각 Atomic model에서 시뮬레이션을 수행한 후의 시뮬레이션 시간이며, value는 메시지에 포함하여 전송하는 값이다.

메시지를 전송 받은 시뮬레이션 서버는 시뮬레이션 메시지에 포함된 source port name을 통해 해당 out port에 연결된 input port를 탐색하여, 식 (2)와 같이 메시지에 destination port name을 추가한다.

$$\begin{aligned}
 & \text{order ID@source port name} \\
 & \text{@destination port name} \\
 & \text{@jobs index@simulation time} \\
 & \text{@value 1@value 2.. ..@value n/}
 \end{aligned} \tag{2}$$

destination port name 추가 후, 시간에 따라 정렬하여 큐에 저장하며, 정렬된 순서대로 목적지에 전달한다. 각 유무선 단말기는 전송 받은 메시지를 파싱한 후, 시뮬레이션을 진행하고, 결과를 다시 시뮬레이션 서버로 전송한다.

3.2 Time manager

Time manager는 전체 시뮬레이션 시간을 관리하며, 그림 7과 같이 객체로부터 메시지를 수신할 경우 메시지

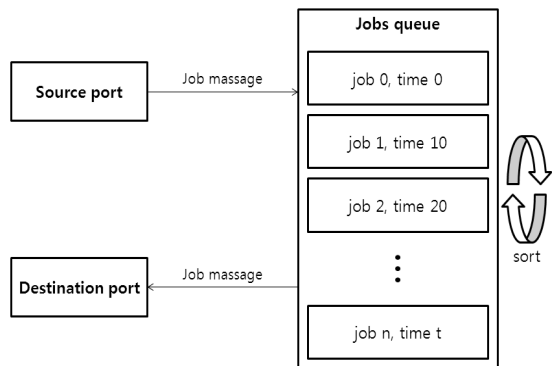


그림 7. Time manager의 동작

를 시뮬레이션 시간 T^{Total} 에 따라 정렬하여 큐에 저장한다. step 명령 시, 큐에 정렬된 순서대로 목적지 객체에 전달한다.

메시지를 수신 받은 Atomic model은 해당 메시지를 파싱하여 시뮬레이션을 수행한다. 시뮬레이션 시간 T^{Total} 은 식 (3)과 같이 T^{Server} 에 Arrival time T^{Atomic} 을 더함으로써 구할 수 있으며, Time manager는 T^{Total} 을 메시지에 포함 후, 다시 서버로 전송한다.

$$T^{Total} = T^{Server} + T^{Atomic} \quad (3)$$

다시 메시지를 수신 받은 서버는 Time manager에 전달하며, Time manager는 T^{Total} 에 따라 다시 정렬하여 큐에 저장한다. 다음 step 명령 시, 정렬된 순서에 따라 목적지 객체로 메시지를 전달한다.

3.3 DEVS Mobile

DEVS Mobile은 다양한 유무선 단말기에서 시뮬레이션을 수행하기 위한 도구로 서버 프로그램과 클라이언트 프로그램으로 구성되어 있다.

그림 8의 서버 프로그램은 유선 또는 무선 네트워크 환경에서 분산 객체로 시뮬레이션에 참여하는 유무선 단말기의 접속을 관리하는 역할을 한다.

서버 프로그램은 Time manager와 메시지 큐를 포함하여, 전체 시뮬레이션을 적절히 제어한다. Microsoft Windows XP 또는 그 이상의 운영체제에서 실행된다.

클라이언트 프로그램은 Microsoft Windows Mobile 6.5 기반의 휴대용 무선 단말기에서 동작하며, 그림 9와 같이 Porket PC 2003에서도 동작이 가능하다. 각 클라이언

트는 각각 독자적인 Atomic model로 동작하고, 서버 프로그램을 통해 메시지를 교환하며, 시뮬레이션 참여하여 분산 객체로서의 역할을 수행할 수 있다.

클라이언트 프로그램은 .NET Framework 3.0 기반으로 개발되었으므로 Microsoft Windows XP 또는 그 이상의 운영체제에서도 동일하게 실행된다.

서버 프로그램에서 step 신호를 전송하면, 클라이언트 프로그램은 경로를 생성한 후, 서버에 결과를 전송한다.

서버 프로그램은 그림 10과 같이 전체 시뮬레이션 시간을 측정하고, 전송 받은 결과를 시간에 따라 기록한다.



그림 9. DEVS Mobile의 클라이언트 프로그램

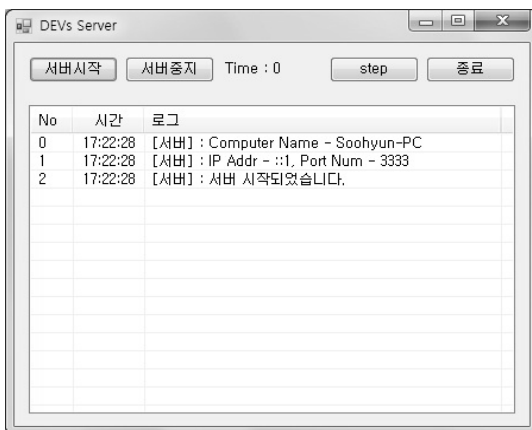


그림 8. DEVS Mobile의 서버 프로그램

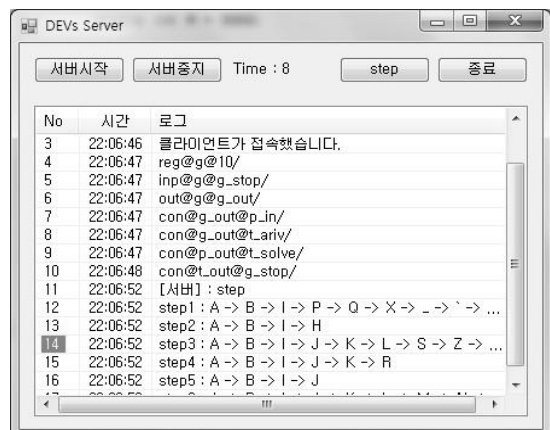


그림 10. 경로 생성 후 서버 프로그램의 화면

4. 실험 및 결과 분석

4.1 이동 경로 표현 방법

실험을 위해 그림 11과 같이 도심 지역을 일정한 크기에 따라 분할하여 맵을 구성한 후, Dijkstra algorithm과 Greedy algorithm을 적용하여 최적 경로를 생성하였다.

가로 7, 세로 7로 지역을 분할하여 그림 12와 같이 총 49개의 vertex를 가지는 맵을 구성하였다.

각 vertex 간의 edge는 랜덤하게 부여된 가중치를 가지며, 모든 이동 노드는 vertex A에서 출발하여, 랜덤하게 선택된 vertex로 이동하도록 하였다.

4.2 실험 구성

본 논문의 실험은 동일한 실험을 단일 컴퓨터 시뮬레이션과 다수의 유무선 단말기들을 이용한 유무선 단말기 분산 시뮬레이션에서 실행하여 비교, 분석함으로써 시스템의 효율성을 확인한다.

단일 컴퓨터 시뮬레이션은 그림 13과 같이 동작한다.

각 유무선 단말기를 통해 도심 지역의 실시간 정보를 획득하여, 서버의 도심 지역 Database에 저장하며, 이를 이용하여 서버에서 모든 이동 노드의 최적 경로를 생성한다.

본 논문에서 개발한 유무선 단말기 분산 시뮬레이터를 이용한 분산 시뮬레이션의 동작은 그림 14와 같다.

단일 컴퓨터 시뮬레이션과 달리 도심 지역 Database에 수집된 정보를, 각 유무선 단말기로 전송하여, 각각 최적 경로를 생성한다.

본 실험에서는 실험의 복잡성을 줄이기 위해 각 vertex 간의 가중치만 고려한다. 모든 이동 노드의 출발 지점은 동일하며, 목적 지점은 랜덤하게 생성한다.

본 실험에 서버로 사용된 컴퓨터와 클라이언트로 사용된 무선 단말기의 사양은 표 2와 같다.

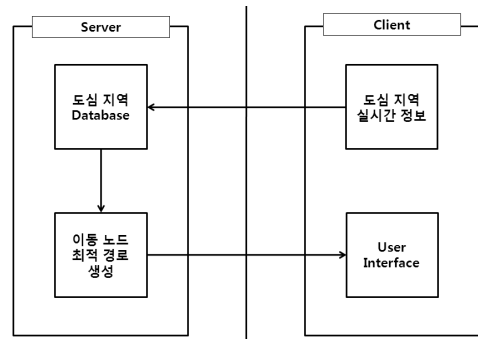


그림 13. 단일 컴퓨터 시뮬레이션

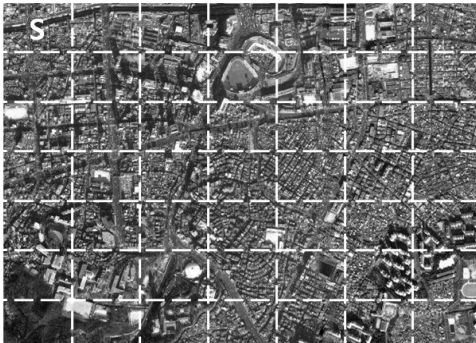


그림 11. 도심 지역을 일정한 크기로 분할^[8]

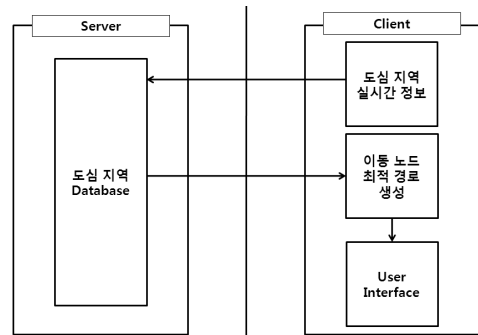


그림 14. 유무선 단말기 분산 시뮬레이션

A	2	B	5	C	5	D	2	E	4	F	2	G
4	1	1	2	4	1	2	2	1	2	1	1	1
H	1	I	2	J	2	K	1	L	2	M	1	N
5	1	2	4	3	1	5	4	3	4	2	5	U
6	4	4	4	4	3	4	1	4	5	3	3	U
V	4	4	4	1	4	4	2	1	4	5	3	W
2	1	1	1	1	3	3	1	5	2	3	3	W
1	1	^	5	.	1	_	2	a	2	b	1	c
8	5	1	1	2	1	2	4	4	2	1	4	j
0	5	e	1	f	1	g	1	4	2	i	4	j
0	6	1	5	5	1	5	1	1	7	1	1	j
k	2	l	2	m	4	n	6	1	7	1	1	q

그림 12. 실험에 사용된 맵(7x7)

표 2. 실험에 사용된 컴퓨터 및 무선 단말기의 사양

Server	
OS	Microsoft Windows 7
CPU	Intel Core 2 Duo E6750 2.66 GHz
Main Memory	2 GB
Client	
OS	Microsoft Windows Mobile 6.5
CPU	Samsung ARM11 800 Mhz
Main Memory	256 MB

4.3 실험 목적

도심 지역에서 이동 노드의 증가에 따라 단일 컴퓨터에서 모든 이동 노드의 최적 경로를 생성하거나 다수의 유무선 단말기에서 각각의 최적 경로를 생성하여 시뮬레이션 시간을 비교함으로써 보다 효율적인 방법을 선정하고자 한다.

4.4 실험 결과

먼저 서버 프로그램과 동일한 컴퓨터에서 실행된 클라이언트 프로그램에서 모든 이동 노드의 최적 경로를 생성하였으며, 이어서 서버 프로그램과 각기 다른 컴퓨터 또는 무선 단말기에서 실행된 클라이언트 프로그램에서 각각 이동 노드의 최적 경로를 생성하였다.

이 때, 이동 경로를 생성해야 하는 이동 노드의 수를 한 대에서부터 열 대까지 증가시키며 각각의 시뮬레이션 시간을 측정하였으며, 이동 노드의 증가에 따라 그림 15 및 그림 16과 같은 결과를 얻었다.

그림 15와 그림 16에서 모든 이동 노드의 최적 경로를

생성하기 위해 단일 컴퓨터 시뮬레이션을 이용한 경우, 이동 노드의 개수에 따라 지속적으로 증가함을 알 수 있다. 그러나 유무선 단말기 분산 시뮬레이션의 경우, 이동 노드의 최적 경로를 각각의 단말기에서 생성하므로, 이동 노드의 개수가 증가되더라도 큰 변화가 없음을 알 수 있었다.

4.5 결과 분석

단일 컴퓨터 시뮬레이션의 경우, 모든 이동 노드의 최적 경로를 구하기 위한 알고리즘을 순차적으로 반복하여 실행하므로 이동 노드의 개수의 증가에 따라 시뮬레이션 시간이 증가하지만, 유무선 단말기 분산 시뮬레이션의 경우, 각각의 유무선 단말기에서 분산되어 실행함으로써, 이동 노드의 개수가 증가하더라도 결과적으로 동시에 최적 경로를 생성하게 되어 시뮬레이션 시간의 증가가 근소한 것으로 판단된다.

이동 노드의 개수가 증가하더라도 시뮬레이션 시간의 증가가 근소하므로, 각 무선 단말기를 실제 환경에 설치하고 실시간으로 정보를 수집한 후, 이를 이용하여 복잡한 시뮬레이션을 수행할 시, 효과적인 시뮬레이션을 수행할 수 있을 것이라고 예상된다.

5. 향후 연구 과제 및 결론

도심 지역에서 이동 노드의 시뮬레이션을 단일 컴퓨터에서 모든 시뮬레이션을 수행하는 방법과 다수의 유선 및 무선 네트워크 환경의 단말기에서 분산 시뮬레이션을 수행하는 방법에 대해 시뮬레이션 시간을 비교하였다. 그 결과 유무선 단말기 분산 시뮬레이션은 분산 시뮬레이션을 수행하므로 이동 노드가 증가할수록 기존의 단일 컴퓨터 시뮬레이션에 비해 시뮬레이션 시간을 효과적으로 줄일 수 있음을 확인하였다.

본 논문에서 개발된 시뮬레이터는 공간 제약에서 벗어나 다양한 유선 및 무선 네트워크 환경에서 분산 시뮬레이션을 수행할 수 있기 때문에, 실시간으로 변화하는 실제 도로 교통 정보를 이용하여 도심 지역에서 이동 노드의 시뮬레이션을 수행한다면 효율적이라 생각된다.

현재 구현된 유무선 단말기 시뮬레이터는 .NET Framework 3.0기반으로 개발되어 현재 널리 보급된 안드로이드 기반 스마트폰에 적용할 수 없다는 단점이 있다. 향후 다양한 플랫폼을 고려하여, 보다 다양한 환경에서 활용할 수 있도록 할 것이다.

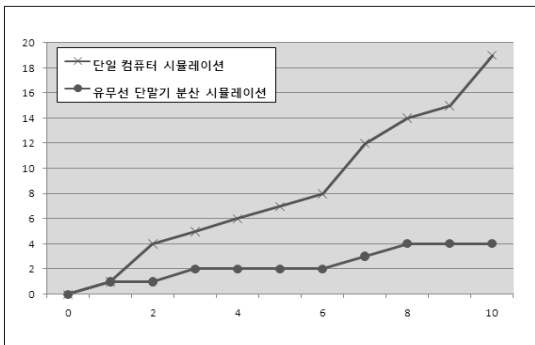


그림 15. 이동 노드의 증가에 따른 시뮬레이션 시간 (Dijkstra algorithm)

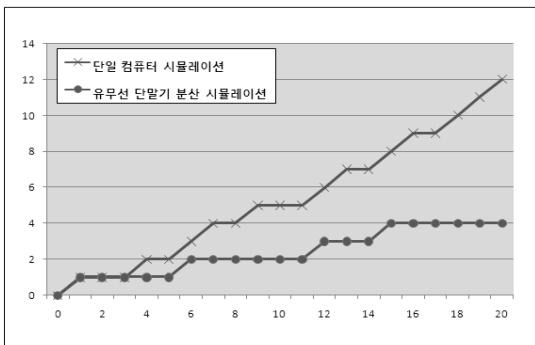


그림 16. 이동 노드의 증가에 따른 시뮬레이션 시간 (Greedy algorithm)

참 고 문 헌

1. 김현휘, 김태섭, 이강선, “스마트폰을 이용한 국방 시뮬레이션,” 한국시뮬레이션학회 '10 춘계학술대회 논문집, pp. 77-85, 2010년 5월.
2. 권기덕, 임태운, 최우석, 박성배, 오동현, “스마트폰이 열어 가는 미래,” CEO Information, 741, pp. 1-22, 2010년 2월.
3. Bernard P.Zeigler, Herbert Praehofer, Tag Gon Kim, Theory of Modeling and Simulation, 2nd Ed., AcademicPress, Jan. 2000.
4. 서경민, 성장호, 김탁근, “MATLAB/Simulink를 이용한 DEVS 모델의 시뮬레이션 환경 구축,” 한국시뮬레이션학회 논문지, 17(4), pp. 219-227, 2008년 12월.
5. 강원석, 사공봉, 홍준성, 김기형, “GUI를 이용한 병렬/분산 시뮬레이션 환경의 개발,” 한국시뮬레이션학회 '99 춘계학술대회 논문집, pp. 174-178, 1999년 4월.
6. 김방현, 김종현, “대규모 센서 네트워크를 위한 최적-동기식 병렬 시뮬레이션,” 정보과학회 논문지 : 시스템 및 이론, 35(5), pp. 199-212, 2008년 6월.
7. 정진립, 우영제, 정창성, “웹을 이용한 분산 시뮬레이션 환경,” 한국시뮬레이션학회 '98 춘계학술대회 논문집, pp. 96-100, 1998년 5월.
8. 이성용, “유전자 알고리즘을 이용한 최적경로 시스템 설계,” 석사학위논문, 인하대학교, 2011년 2월.



조 수 현 (swordlevross@nate.com)

2009 한남대학교 전자공학과 학사
2011 인하대학교 컴퓨터·정보공학과 석사

관심분야 : 모델링&시뮬레이션, 유비쿼터스 컴퓨팅, 클라우드 컴퓨팅



이 종 식 (jslee@inha.ac.kr)

1993 인하대학교 전자공학과 학사
1995 인하대학교 전자공학과 석사
2001 미국 애리조나대 전기·컴퓨터공학과 박사
2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
2003~2006 인하대학교 컴퓨터공학부 조교수
2006~현재 인하대학교 컴퓨터정보공학부 부교수

관심분야 : 시스템 모델링&시뮬레이션, 그리드 컴퓨팅, 소프트웨어공학