

* 본 연구는 고려대학교 특별연구비에 의하여 수행되었음.

WAP 상의 트랜잭션 서비스의 구현 및 성능평가

임경수,° 우시남, 안순신
고려대학교 전자공학과 네트워크 연구실
 {angus, niceguy, sunshin}@dsys.korea.ac.kr

The Performance Evaluation and Implementation of Transaction Service on Java-Based WAP

Kyungsoo Lim,° Sinam Woo, Sunshin An
Computer Network Lab., Dept. Of Electronic Eng., Korea University

요약

인터넷과 무선 이동통신 등이 보편화되면서 두 기술을 접목한 WAP (Wireless Application Protocol)[1]이 등장하였으며, 다수의 사용자 요구를 동시에 효율적으로 처리하면서 다양한 무선 단말기로부터 요청을 신속하게 처리할 수 있는 WAP 게이트웨이(gateway)에 대한 연구가 여러 연구 그룹에서 수행되고 있다.

본 논문에서는 Java의 기본적인 서비스인 멀티쓰레드를 이용하고, 컨테이너 유효 리소스의 효율적 관리를 위해 접속 풀링(connection pooling) 기능과 우선순위 큐를 적용하여 WTP (Wireless Transaction Protocol)[2]를 구현한다. 우선순위에 입각하여 구현된 트랜잭션 서비스를 시뮬레이션을 통하여 성능을 분석한다.

1. 서론

1990년대 이후로 인터넷은 급속히 확산되었으며 특히 WWW(World Wide Web)의 괄목할만한 성장을 가지고 왔다. 또한 무선이동통신 역시 빠른 속도로 그 시장을 넓히면서 발전해왔다. 다수의 사용자들은 무선 이동 통신 환경하에서 인터넷을 사용하고자 하는 욕구가 발생하였고 이에 서비스 제공업체측에서는 이러한 요구에 많은 관심이 집중되어 졌다.

무선 인터넷을 사용하기 위해서는 단말기와 무선 망에 위치한 무선 인터넷 게이트웨이 그리고 무선 인터넷 콘텐츠를 담고 있는 서버사이에 공통의 프로토콜을 사용하며, 이 공통의 프로토콜을 사용하여 단말기에서부터 무선망을 거쳐 무선 인터넷 콘텐츠 서버 사이에서 데이터를 주고 받는다. 그러나 단말기는 전력 소모량, 메모리 크기, 화면 표시 능력, 전송 속도, 안정성 등에 많은 한계를 가지고 있다. 이러한 제약 사항은 유선 인터넷 표준을 그대로 무선 환경에서 사용하기에는 불가능하므로, 기존 방식에 상당 부분 수용하면서 새로운 무선 환경에 적합한 프로토콜의 규정이 필요했다. 이러한 필요조건을 만족하게 해주는 것이 현재 표준으로 되어있는 WAP 프로토콜이다. WAP 프로토콜 스택[3]을 기반으로 한 WAP 게이트웨이에 대한 연구가 활발히 수행되고 있다. WAP 스택은 기본 데이터를 프로토콜에 맞추어 전송하고 수신하는 매체의 역할을 한다. 본 연구에서는 이러한 WAP의 스택중 트랜잭션 서비스를 담당하는 계층을 우선순위를 고려하여 설계하고 구현하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 WAP Gateway의 동작에 대해서 설명을 하고, 3 장에

서는 우선순위기능이 추가된 WTP를 구현하고 성능평가를 한다. 4 장에서 결론 및 향후연구를 기술하며 끝을 맺는다.

2. WAP Gateway

WapGateway 클래스는 여러 개의 클래스들의 조각들로 구성되어진다. Listener 클래스는 주어진 포트로 단말기의 요청이 들어오기를 기다리는 쓰레드이다. PoolManager 클래스는 현재 연결된 접속 리스트를 관리한다. Listener는 단말기로부터 접속요청이 들어오면, PoolManager에게 메시지를 보낸다. 이때 PoolManager는 접속 요청을 받아들일 수 있는가를 조사한 후 더 이상 접속 요청을 수용할 수 없으면 단말기의 요청을 거절한다. PoolManager가 단말기와의 접속이 가능하면 접속에 대한 전반적인 처리를 담당하는 새로운 ProtocolStack 객체를 생성한다

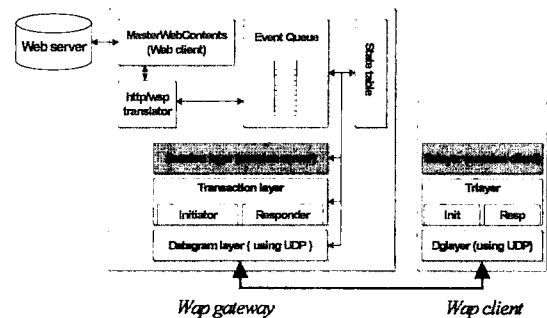


그림 1 WAP Gateway의 구조

[그림 1]은 이러한 WAP Gateway 의 전체 구조의 뼈대를 보여준다.

3. 우선순위가능이 추가된 Transaction Layer(WTP)의 구현

3.1 Transaction Layer

WTP 는 인터넷 접속 전용 컴퓨터에서 실행하기에 적당한 트랜잭션 프로토콜이다. WTP 는 보안 혹은 비 보안 무선 데이터그램 네트워크 위에서 모두 동작한다. 교환의 기본 단위는 Byte stream 이 아니라 Message 이다.

트랜잭션을 이용한 요청-응답을 기본으로 하는 연결-지향형 서비스를 통해 효율성을 증가시킨다. 또 Explicit 연결 설정 과정을 제거하여 무선 환경에서 통신상 발생할 수 있는 과부하를 제거하였다.

프로토콜을 구현하기 위해서는 트랜잭션 내부의 PDU 간에 기밀성을 제공하고, 전체적인 관리를 위해 생겨난 TID (Transaction Identifier)의 관리가 필요하다. 이때, TID 는 송신지 주소, 수신지 주소, 수신지 포트번호 등과 함께 작용하여 같은 트랜잭션 내의 모든 작업들이 유일함을 증명해준다. TIDFactory 클래스는 WtpResponder 객체가 주로 사용하며 TID 가 유효한지를 검사한다. TID 실증과정은 PDU 내부의 TIDnew 항목이 세트되었을 때 실행된다.

클라이언트와 서버에서 보내는 모든 PDU 에는 반드시 SendTID 를 포함해야 하고, RcvTID 는 처음 Invoke 메시지를 받으면서 생성되어 진다. 그리고 RcvTID 는 하나의 트랜잭션이 종료될 때까지 계속해서 사용된다. SendTID 와 RcvTID 는 XOR 의 관계를 갖는다.

그리고 WtpPDU 는 일반적인 특성의 Invoke, Result, Ack, Abort 과 선택 사항인 Segmented Invoke, Segmented Result, Negative Acknowledge 등으로 각각 구분하여 처리한다. WtpTimer 는 트랜잭션 수행 중에 서로 간의 상태를 관리하기 위한 타이머의 역할을 한다. 다음 [그림 2]는 위에서 설명한 WTP 를 간략하게 클래스 다이어그램으로 표현한 것이다.

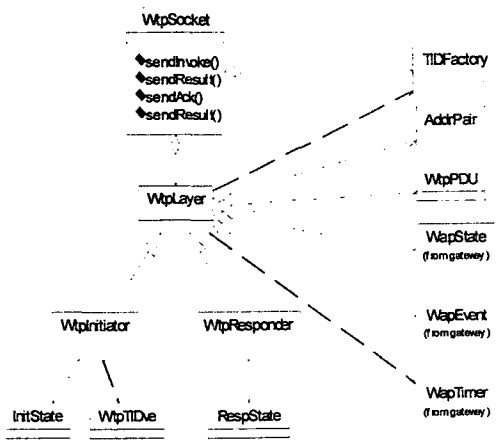


그림 2 WTP 클래스 다이어그램

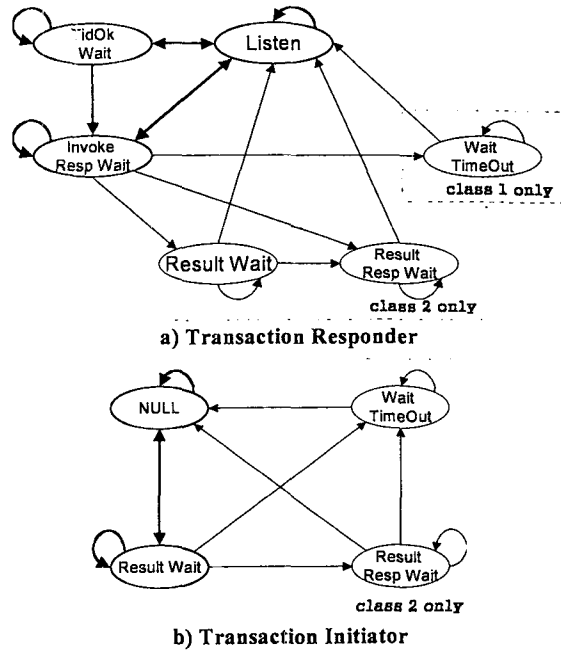


그림 3 트랜잭션 상태 다이어그램

[그림 3]은 Transaction Initiator 의 상태와 Transaction Responder 의 내부 상태의 흐름을 보여주는 상태 다이어그램이다. 여기서 각각의 상태 준이 이벤트는 매우 복잡한 관계로 그림 상에서 생략되었다. C0 서비스(Class 0 service or transaction)는 WtpInitiator 에 의해 수행되고, 주로 연결되지 않거나 확인을 받지 않는(상태를 관리하지 않는) 메시지를 보낼 때 사용한다. C1 서비스는 상태 관리를 위해 타이머를 사용하고, Initiator 와 Responder 사이에 메시지를 주고 받는다. 각 서비스에 대한 설명은 [그림 4]와 같다.

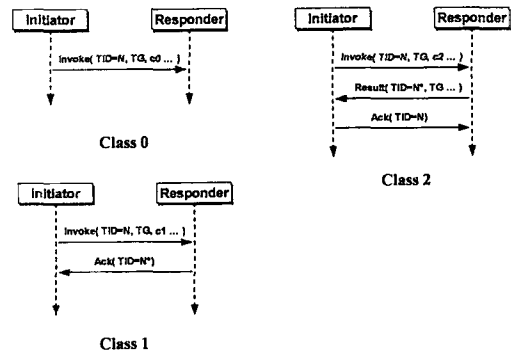


그림 4 기본 트랜잭션 분류

여기 핸들링은 대부분 상태를 가지고 있는 WtpInitiator 와 WtpResponder 에게 WtpEvent 를 넘겨주기 전에 이벤트와 상태정보를 비교, 분석하여 수행된다. 그렇지만 받은 PDU 자체 내에 잘못된 헤더파일을 포함하

고 있으면 WTP는 처리할 수 없다. 또한, 본 논문에서 주안점을 둔 각각의 PDU에 우선순위를 주기 위해 기본적인 PDU 헤더 바로 뒤 1 byte를 우선순위 등급을 표시할 수 있도록 예약하여 사용하였다.

3.2 일반적인 큐와 우선순위 큐

Queue 라는 인터페이스는 기본적인 함수들을 정리하였고, QueueImpl 클래스는 기본적인 큐의 구조를 수행하는 add()와 remove() 함수를 포함하고 있다. 큐의 요소로는 내부적으로 정보를 저장하는 Vector 의 속성을 가진 변수를 가지고 있다. 여기서 add()는 구성요소를 첨가하고, remove()는 삭제하며, size()는 몇 개의 구성요소를 포함하고 있는지 조사하고, isEmpty()는 큐가 비어있는지를 조사한다.

QueuePri 클래스는 QueueImpl 과 공용함수가 동일하며, QueuePri 는 add() 함수 내에 삽입할 장소를 결정하는 search() 함수가 첨가되어 있는 것이 다른 점이다. 이때, QueueImpl 과 QueuePri 클래스 모두는 Queue 인터페이스를 상속받는다.

4. 우선순위 큐의 성능 평가

실험은 LAN 과 WAN 상에서 메시지를 보내고 받는 기본적인 실험을 하였다. 실험환경은 한국통신 ADSL 상에 게이트웨이가 존재하고, 클라이언트는 고려대학교 네트워크 연구실에 위치하였다.

테스트는 한번에 동시에 작동할 수 있는 Thread 를 40 개로 제한하고, 우선순위는 4 개로 분리하였다. 이때 우선순위는 r=0 가 가장 낮고, r=3 이 가장 높다. 트랜잭션에 사용되는 메시지의 크기는 258bytes 로 고정시켰고, 4300 개에서부터 수렴을 하여 그보다 많은 수인 6000 개 트랜잭션을 수행하였다.

[그림 5]와 [그림 6]은 각각 LAN 과 WAN 에서 처리하는데 걸리는 시간의 평균을 계산하여 통계를 낸 것이다

5. 결론 및 향후 연구

본 논문에서는 WAP 상에서 트랜잭션의 일괄적인 처리 방식을, 중요한 트랜잭션을 우선시하여 처리하는 방식으로 변환하는 방법을 설계 및 구현하였고, 우선순위가 높은 클래스를 갖는 트랜잭션은 FCFS 로 설계한 것에 비해 빠르게 처리된다는 것을 보였다.

본 논문에서는 우선순위가 높은 클래스에 대해서는 서비스를 받기까지 기다리는 시간이 작았지만, 상대적으로 낮은 클래스는 많은 시간을 기다려야 하는 문제점이 존재한다. 향후 좀더 효율적인 우선순위 정책을 제공하는 전체 시스템 성능향상 방안이 연구되어야 한다.

추가적으로 Mobile IP 와 같은 다른 종류의 네트워크 환경에서의 테스트를 통하여 다른 프로토콜 스택과 독립적으로 연동한다는 것을 증명하는 과제가 남아있다. 또, WSP 및 WTLS 를 추가 제작하여 무선 통신의 안정성을 더해야 하는 과제가 남아있다.

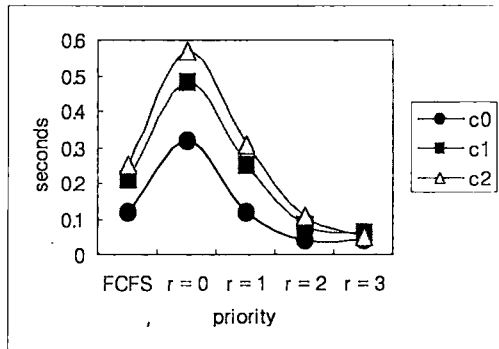


그림 5 Lan 상에서의 대기시간

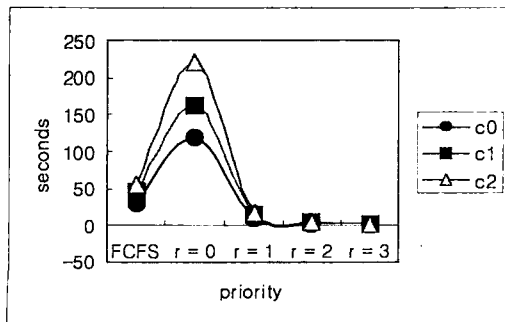


그림 6 WAN 상에서의 대기시간

참고 문헌

- [1] WAP Forum, "WAP Architecture Specification", 30-April-1998.
- [2] WAP Forum, "Wireless Datagram Protocol Specification", 19-February-2000.
- [3] 이준규, 김동호, 김상경, 안순신, "자바 기반의 WAP 스택 설계" 정보과학회 학술발표 논문집, 2001년 4월
- [4] WAP Forum, "Wireless Transaction Protocol Specification", 19-February-2000.
- [5] WAP Forum, "Wireless Application Environment Specification", 29-March-2000.
- [6] WAP Forum, "Wireless Markup Language Specification", 19-February-2000.
- [7] WAP Forum, "Wireless Session Protocol Specification", 4-May-2000.
- [8] Bolch, Greiner, de Meer, and Trivedi, "Professional Java Server Programming", WROX, 1998.
- [9] Ted Neward, "Server-Based Java Programming", MANNING, 2000.
- [10] Merlin Hughes, Michael Shoffner, Derek Hamner, and Umesh Bellur, "Java Network Programming", MANNING, 1999.
- [11] Richard Monson-Haefel, "Enterprise Java Beans", O'REILLY, 1999.
- [12] Bill Venners, "Inside the JAVA 2 Virtual Machine", McGrawHill, 2000.
- [13] Ken Arnold, and James Gosling, "The Java Programming Language", Addison Wesley, 1998.