

이동 컴퓨팅 환경에서 링크 상태 측정을 통한 원격 메소드 호출의 성능 개선

김현호 강대욱
전남대학교 전산학과
{freesite, dwkang}@cs.chonnam.ac.kr

Improving the Performance of Remote Method Invocation by Measurement of Link State in Mobile Computing Environments

Kim, Hyeon-ho Dae-wook Kang
Dept. of Computer Science, Chonnam natl. Univ.

요약

휴대용 컴퓨터와 무선통신 기술 발달로 인해 사용자들은 시간과 장소의 구애를 받지 않고 컴퓨팅 환경을 제공받을 수 있게 되었다. 그러나 일반적인 무선 환경에서는 핸드오프나 비트오류 등 단시간 단절이 매우 흔한 일이지만 TCP는 이를 유선상의 혼잡 신호로 인식하여 전송 윈도우의 크기를 줄이고 불필요한 대기시간을 설정하는 등의 혼잡제어 정책을 수행하게 된다. 따라서 이러한 문제를 해결하기 위해 여러 계층에서 시도하는 노력이 있었으나 링크 상태를 파악하여 응용계층에서 자료 전송 여부를 미리 결정할 수 있다면 불필요한 전송을 막을 수 있다. Remote Method Invocation(RMI)는 대화형 응용 프로그램과 서비스를 쉽게 만들 수 있도록 해주는 Java 기반의 하부 구조로써, 응용계층과 전송계층의 중간에 위치하여 사용자의 자료를 전달하는 역할을 하므로 이를 수정하는 것이 가장 바람직하다. 이 논문에서는 무선 상황에 적절하게 적용하는 RMI를 위해 동작과정을 확장하여 구현한다.

1. 서론

원래 그룹웨어의 형태로 시작된 분산 환경의 대화형 응용프로그램은 메신저 서비스나 쇼핑몰 등에서 사용되고 있다. 이러한 대화형 응용프로그램의 개발에서 객체 기술의 사용이 점차 증가하고 있는데 Java 언어는 서로 다른 서버와 클라이언트의 하드웨어 및 소프트웨어 플랫폼에서 발생하는 문제를 숨기므로 분산 서비스와 응용프로그램 개발에 용이하다. RMI는 자바 가상머신 위에서 작동중인 클라이언트 응용프로그램이 원격 자바 가상머신에서 구현된 서버 객체를 로컬 객체와 같은 방법으로 호출해서 사용할 수 있도록 하기 때문에 클라이언트/서버 개념의 분산 응용프로그램을 쉽게 개발하도록 지원한다.

그런데 현재 유선과 고정된 호스트들로 구성되어 있는 네트워크와는 달리 기술발전으로 조만간 이동 호스트와 무선으로 연결된 네트워크의 비율이 높아질 것이다. 하지만 TCP는 고정 네트워크에 적합하도록 설계되어 있기 때문에 무선상의 지연 및 손실된 패킷에 대해서도 혼잡 신호로 해석하게 되고, 이로 인해 일반적인 메시지 전달 프로그램과 비교할 때 부하가 많이 걸리는 RMI를 갖은 핸드오프와 비트 에러 비율이 높은 무선망에 그대로 적용하는 것은 적절치 못하다.

또한 혼잡 제어 정책을 수행하는 TCP가 현재의 유선 네트워크에서는 적절하게 적용하고 또한 널리 사용하고 있기 때문에 TCP 수정을 통해서 이 문제를 해결하려는 시도는 바람직하지 않다. 대신 응용계층에서 링크 상황을 판단하여 자료의 전송 유무를 결정함으로써 단절상황에서는 아예 전송요구가 발생

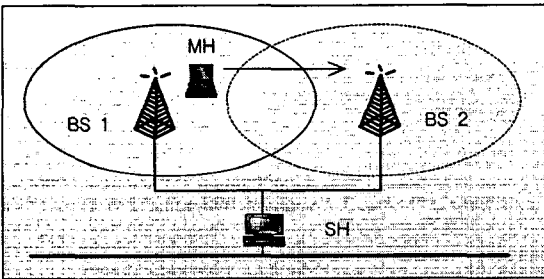
* 본 논문은 2000년도 정보통신 우수시범학교 지원사업(과제번호 2000-1) 연구비 지원에 의한 것이다

하지 않도록 하는 것이 중요하다. 이러한 문제는 RMI 구현 중에 현재의 링크 상태를 판단하는 linkStateAware 메소드를 추가하고 이 결과 값으로 자료 전송 여부를 판단한다.

미래의 무선망은 작은 반경의 피코 셀로 이루어져 잦은 핸드오프가 발생하고 무선망 자체가 가지는 전송 오류 때문에 단시간 링크단절에 적응하는 RMI의 성능향상은 필수적이다.

1.1 무선망에서의 단시간 단절

그림 1은 이동 호스트(MH)가 Base Station인 BS 1에서 BS 2로 이동하는 일반적인 핸드오프 상황을 보여주고 있다. 이 경우 MH에게 통신 단절이 나타난다. 이것은 BS 1과 통신을 하던 MH가 BS 2와 통신을 하기 위해 라우팅 정보를 수정하고 또한 BS 1와 BS 2의 자료 교환 및 고정 호스트인 SH의 정보의 수정 완료될 때까지 0.2초 이상 계속된다. 만일 두 셀의 중복지역이 거의 없거나 아예 서로 떨어져 있는 경우에는 1초 이상 지속되게 된다[3].



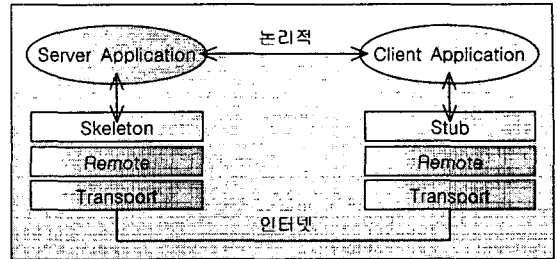
(그림 1) 일반적인 Hand Off 상황

무선망에서는 호스트의 움직임이 없어도 물리적인 전송 오류에 의해 패킷 손실이 자주 발생하고 이로 인해 단시간 단절이 발생한다. 다른 연구 결과에 의하면 연구실에서는 무선도 99% 이상의 놀라운 패킷 수집률을 보여준다[8]. 그러나 일반적인 환경에서는 패킷 손실 빈도수가 짧은 거리 이동 또는 안테나의 위치와 같은 요소들에 따라 다양하게 변화한다. 이것은 무선 장치가 주위의 노이즈와 여러 경로의 간섭에 영향을 받기 쉽기 때문에 무선망에서는 일반적이다.

1.2 RMI Framework 개요

그림 2는 RMI의 기본적인 계층모형을 보여주고 있다. RMI는 응용프로그램과 자바 가상머신 사이에 세 개의 계층으로 구현된다. 첫 번째는 클라이언트 쪽에 만들어지는 클래스인 스텝과 서버측에 만들어지는 스키텔레톤이다. 두 번째는 다양한 원격 참조자

나 호출 프로토콜을 지원할 수 있도록 해주는 Remote Reference Layer이다. 마지막으로 서로 다른 주소 공간 사이에서 마샬링 된 스트림을 흘려보내주는 하위 수준의 계층인 Transport Connection Layer가 있다.[1]



(그림 2) RMI 계층 모델

RMI는 자바 프로그래밍 언어와 개발환경을 사용하여 분산 네트워크 내에서 상호 작용하는 객체 지향 프로그램을 작성할 수 있는 방식이다. RMI는 요청과 함께 하나 이상의 객체들을 통과시키는 능력을 가지고 있다. 객체는 원격 컴퓨터 내에서 수행될 서비스를 변경하는 정보를 포함할 수 있다.

2. 관련연구

단시간 링크 단절에 효율적으로 적응하기 위한 노력은 많이 있었다. 다음은 각각 링크계층과 전송계층에서 해결책을 제시한 경우이다.

2.1 링크 계층에서의 단시간 링크 단절의 해결 노력

이것은 미디어 자체의 데이터 재전송을 통해 링크 계층에서 단시간 링크 단절 상황을 해결하는 시도이다. 이 방법의 장점은 상위계층 프로토콜에 의존하지 않고 신뢰성 있는 통신을 개선하는 접근방식이라는 점이다. 그러나 TCP가 이미 단대단 재전송 프로토콜을 구현하고 있기 때문에 중복된 일처리가 된다. 기존 연구에서는 이러한 독립적인 링크 레벨의 재전송 규칙이 오히려 성능을 감소시킬 수 있다는 것을 보여주고 있다[6].

2.2 TCP의 빠른 재전송 접근 방법

이것은 IP와 TCP를 수정하여 접근하는 방법이다. 즉 IP가 핸드오프 시작 및 종료의 신호를 인식하여 TCP에게 전달해 준다. TCP는 IP로부터 핸드오프 종료 신호를 받게 되면 즉시 전송을 시작하며 상대방에게 중복된 응답신호를 보낸다. 이러한 신호를 받은 상대방의 TCP 역시 핸드오프의 종료로 인식하고 바로 재전송을 시작하게 된다[3]. 하지만 이 방

법의 문제점은 IP 및 TCP를 수정해야 한다는 것이다. 앞서 기술했듯이 TCP가 광범위하게 사용되고 있는 상황에서 이러한 접근은 무리가 있다. 또한 이미 핸드오프 상황 중에 자료를 전송하는 문제가 있기 때문에 처음부터 자료를 전송하지 않는 방법을 생각할 수 있다.

따라서 위와 같은 내용을 고려해 봤을 때 전송계층과 애플리케이션 계층 중간에 위치하는 RMI를 수정하여 사용하는 것이 가장 좋은 방법이 될 수 있다.

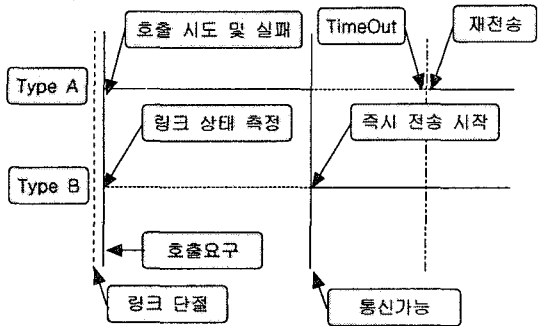
3. 단시간 단절에 적응하는 RMI의 효율적 제어

두 셀 사이를 이동하며 핸드오프를 수행하는 호스트나 전파가 약한 곳에 있는 호스트의 경우 단시간 단절은 매우 흔한 일이다. 이럴 때 전송하는 자료는 모두 손실 될 뿐만 아니라 TCP로 하여금 혼잡 제어 정책을 수행하게 하고 또한 불필요한 전송을 반복할 수 있다. 이러한 문제를 해결하기 위해 현재 폭넓게 사용되는 TCP를 수정하는 대신 현재의 링크 상태를 판단하는 linkStateAware 루틴을 RMI에 추가하여 이 결과 값에 따라 자료 전송 여부를 판단한다.

실험은 JDK 1.3.0 소스를 수정하고 Forte 6 C와 JDK1.3.0 binary를 이용하여 컴파일 된 RMI를 사용했다. 무선 링크상의 핸드오프 및 비트 오류 등으로 인한 서버와 클라이언트 사이의 단시간 단절은 리눅스용 ipchains 명령어와 셸 스크립트를 이용하여 소프트웨어적으로 구현을 했다. 실험 결과에 따르면 TCP 연결 과정에서 단절이 발생할 경우 연결 성립까지 평소보다 10배 이상의 시간이 소요되었으며 자료 전송 시에도 3배의 시간이 소요되었다. 이것은 TCP에 의한 재전송 정책과 slow-start 알고리즘에 기인한 것이다.

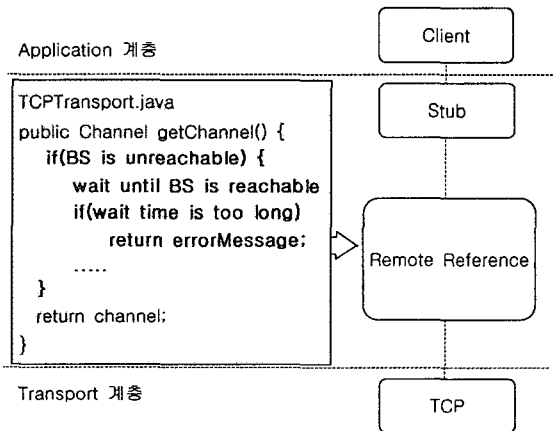
링크 단절시 자료를 전송하는 방법에는 그림 3과 같이 두 가지 방법이 있다. 첫째는 TCP에 의존하여 무조건 전송하는 기존의 방식(Type A)이고 두 번째는 단절 여부를 판단하여 링크 연결이 되었을 때 전송하는 방식(Type B)이다. Type B는 Type A에 비해서 링크 상태 측정에 소요되는 시간만큼 손해지만 링크 단절시 TCP의 재전송 시간에 의한 멈춤 현상과 slow-start 알고리즘이 작용하지 않기 때문에 효율적이다. TCP의 재전송 타이머는 지속적인 링크 단절에 대해서 지속적으로 증가하는 속성이 있기 때문에 단시간 링크 단절이 빈번한 곳에서는 간혹 자

료 전송이 마비되는 결과를 발생시키기도 한다. 하지만 Type B와 같은 방법으로 전송할 경우 이러한 일의 발생 확률은 낮아진다.



(그림 3) 링크 단절시 두 가지 전송방식

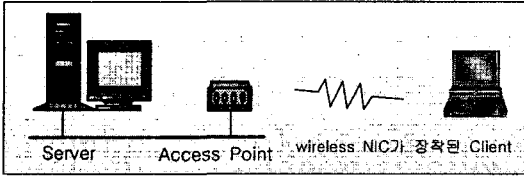
RMI에서는 원격 객체 검색 전에 TCP 연결을 시도하고 메소드 호출시 채널을 얻는 과정을 거친다. 즉 두 과정 모두 TCPTransport.getChannel()을 수행하게 되는데 이렇게 채널을 확보하기 전에 링크 상태를 측정하는 루틴을 삽입하고 이의 결과에 따라 전송 여부를 결정할 수 있다. ICMP 메시지를 이용한 링크 상태 측정에 필요한 시간은 3ms 이하이고 좀 더 효율적인 구현을 통해 단축할 수 있다.



(그림 4) 링크 상태 확인

4. 테스트 환경

실험은 서버와 클라이언트 사이를 무선장비를 이용하여 연결하고 일반적인 연구실 환경에서 실시하였다. 서버는 Ultra Sparc 60이며 운영체제는 솔라리스 2.7이다. 클라이언트는 리눅스를 이용하였다. 무선장비는 Orinoco Wireless network Interface Card (11Mbps)를 사용하였다. 시스템 구성은 (그림 4)와 같다.



(그림 5) 테스트 시스템 구성

5. 결론 및 향후 연구 방향

모바일 환경의 통신에서 피할 수 없는 문제는 단시간 링크 단절이다. 이것은 셀 네트워크에서의 잦은 핸드오프와 무선 자체의 물리적인 오류로 인한 패킷 손실에 의해 일어난다. 기존의 유선망에 적합하게 되어 있는 TCP와 수정되지 않은 RMI를 사용할 경우 이러한 단절 상황에 적응하지 못하고, 혼잡 제어 정책으로 인해 불필요한 전송을 초래하게 된다. 이 문제를 해결하기 위해 RMI를 수정해야 하고 이것은 다음과 같은 장점을 갖는다.

- (1) 기존의 RMI 코드를 이용하여 일부만을 확장하기 때문에 쉽게 적용할 수 있고, 상대방의 RMI 및 애플리케이션, TCP, IP등의 소스를 수정하지 않으므로 기존의 네트워크에 투명하게 적용할 수 있다.
- (2) 단시간 링크 단절 상황에 적절하게 대응하므로 무선 호스트 및 고정 호스트에서 불필요한 전송을 줄일 수 있고, TCP의 재전송 정책에 의존하지 않으므로 링크 단절 종료 시 바로 전송을 재개할 수 있다.

문제점으로는 핸드오프나 비트 오류가 없는 곳에서도 불필요하게 링크상태를 확인함으로써 인해서 생길 수 있는 성능저하의 우려가 있다. 따라서 앞으로의 연구에서는 지속적인 링크상태 확인이 아니라 상황에 따라 선택적으로 확인 여부를 결정하는 방법을 모색해야 한다. 핸드오프나 비트 오류가 잦은 지역 또는 이동 호스트가 빠른 속도로 움직이는 등의 상황에서는 매 전송 시 링크 상태 확인을 하고 비교적 고정된 위치에 있어 단시간 단절이 적은 경우에는 링크상태 측정을 회수를 줄이고 기존의 TCP에 의존하도록 해야한다.

참고문헌

[1] V. Krishnaswamy, D. Walther, S. Bhola, E. Bommaiah, G. Riley, B. Topol and M. Ahamad. Efficient Implementations of Java RMI, In 4th USENIX Conference on Object-Oriented Technologies and Systems (COOTS'98), Santa Fe,

NM, 1998.

[2] G. Welling, M. Ott, Structuring Remote Object Systems for Mobile Hosts with Intermittent Connectivity, In Proc. of the 18th International Conference on Distributed Computing Systems, Amsterdam, The Netherlands, May 1998

[3] A. Caceres, The Effect of Mobility on Reliable Transport Protocols, In Proc. of the 14th Intl. Conf. on Distributed Computing Systems, p.p 12-20, June 1994.

[4] H. Balakrishnan, S. Seshan, and R.H. Katz., Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks, ACM Wireless Networks, 1(4), December 1995.

[5] A. DeSimone, M. C. Chuah, and O.-C. Yue. Throughput performance of transport-layer protocols over wireless LANs, In Proceedings of IEEE GLOBECOM 1993, p.p. 542-549, December 1993.

[6] S. Paul, E. Ayanoglu, T. F. LaPorta, K. H. Chen, K. K. Sabnani, and R. D. Gitlin, An asymmetric-link-layer protocol for digital cellular communications, in Proc. InfoComm '95 (1995).

[7] Yi-Bing Lin, Imrich Chlamtac, Wireless and Mobile Network Architectures, WILEY, 2001.

[8] D. Duchamp and N. F. Reynolds, Measured performance of a wireless LAN, In Proceedings of the 17th Conference on Local Computer Networks, p.p 494-499. IEEE, Sep 1992.