

XML기반 Peer-to-Peer 엔진의 메시지 디스패처 설계

이재욱⁰ 이일수 권태숙 이승룡
경희대학교 전자계산공학과
(jaeki, primelee, suki, syllee)@oslab.kyunghee.ac.kr

The Design of Message Dispatcher for Peer-to-Peer Engine Based on XML

Jae-Wook Lee⁰ Il-Su Lee Tae-Suk Kwon Sungyoung Lee
Dept. of Computer Engineering, KyungHee University

요 약

본 논문에서는 웹, 모바일 환경에 적용 가능하고, 다양한 서비스의 지원이 가능한 글로벌 Peer-to-Peer(P2P) 엔진에서 서비스 모듈, 네트워크 모듈, 레퍼지토리 모듈간에 발생하는 메시지를 처리하는 메시지 디스패처의 설계 경험을 기술한다. P2P엔진이 이 기종 네트워크환경에 적용 가능하고, 다양한 서비스들을 동시에 지원하기 위해서는 피어 간 또는 엔진 내부에서 발생하는 많은 메시지들을 원활히 처리하는 모듈이 필요하다. 이를 위하여 메시지 처리를 위한 글로벌 P2P 엔진의 메시지 디스패처를 제안한다. 메시지 디스패처는 글로벌 P2P엔진에서 지원하는 서비스 또는 다른 모듈에서 발생하는 메시지들을 우선순위에 따라 시간을 이용하여 작업순서를 배열하는 Message Queue, Message Queue에 적체된 메시지들을 분석하여 처리하는 Message Processor, 메시지 디스패처와 서비스 또는 각 모듈간의 메시지의 전달을 위한 공유 버퍼인 Service Buffer로 구성되어 있다. P2P 엔진에서 메시지 디스패처를 이용하게 되면 다양한 서비스에 적용가능하고, 모듈간의 투명성을 제공하고, 신속한 메시지의 처리가 가능하게 된다.

1. 서 론

Peer-to-Peer(이하 P2P) 서비스는 인터넷상의 정보를 검색엔진을 거쳐 찾아야하는 기존방식과 달리 인터넷에 연결된 모든 개인 컴퓨터로부터 직접 정보를 제공받고 검색은 물론 내려 받기까지 할 수 있는 서비스를 말한다. 이 서비스로 인하여 웹사이트에 한정돼 있던 기존의 정보추출 경로를 개인이나 회사가 운영하는 데이터베이스로까지 확대할 수 있게 되었다. 즉 자신의 정보를 전국적 혹은 세계적으로 관리 운영하며, 회원상호간의 다양한 정보 공유뿐만 아니라 동일한 정보를 공유하고자 하는 회원간의 커뮤니티 형성이 가능하며, 그룹웨어로서 역할을 통해 원격회의, 원격 교육 등이 가능하다[1].

이러한 P2P 서비스는 인터넷 대역폭의 광대역, 고속화, 인터넷 사용자들의 PC 성능의 고도화와 마지막으로 서버 집중식 모델의 한계성과 비용의 문제와 같은 PC 환경이 변화하면서 등장하게 되었다. 그러나 기존의 냅스터(Napster)[2], 누텔라(Gnutella)[3]등과 같은 일반적인 P2P 서비스는 현재 단순한 파일 공유 서비스만을 지원하고, 모바일, 웹과 같은 다양한 네트워크 환경에 적합하지 않다는 단점을 가지고 있다. 이러한 단점을 보완하기 위하여 웹, 모바일 등의 다양한 네트워크 환경에 적용 가능하고, 다양한 서비스를 지원할 수 있는 XML 기반의 글로벌 P2P 엔진이 필요하게 된다. 또한 글로벌 P2P 엔진에서 다양한 서비스를 지원하게 되면 많은 메시지들이 발생하게 되는데 이러한 메시지들을 적절하게 처리하기 위해서 엔진내부에서 메시지를 처리해주는 디스패처가 필요하게 된다. 따라서 본 논문에서는 컴포넌트화된 다양한 서비스 모듈에게 XML포맷으로 구성되어진 메시지들을 신속하고, 효과적으로 처리하기 위한 메시지 디스

패처의 설계방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문이 제안하는 메시지 디스패처와 유사한 관련 연구를 소개하고, 3장에서는 제안하는 메시지 디스패처를 적용하고자 하는 글로벌 P2P엔진에 관하여 설명하고, 4장에서는 메시지 디스패처의 설계에 대하여 설명하고, 5장에서 결론을 내린다.

2. 관련 연구

본 장에서는 P2P 환경에서의 메시지 처리 기술에 대한 관련 연구에 대하여 살펴보도록 하겠다.

2.1 누텔라(Gnutella), 프리넷(Freenet)

누텔라[3]는 Nullsoft사에서 개발하여 현재 대부분의 P2P 솔루션의 개발에 활용되고 있는 공개 소프트웨어이다. 누텔라는 서버가 존재하는 중앙 집중형의 하이브리드(Hybrid) 형태의 냅스터와 달리 이용자간에 디렉토리 정보를 릴레이식으로 중개해 주는 서버가 필요 없는 순수한 형태의 P2P 방식으로 플랫폼에 독립적인 P2P 솔루션이다. 누텔라는 Ping-Pong, 검색(Query), 다운로드의 세 가지 기능을 가진다. 누텔라의 메시지는 TCP/IP 프로토콜을 사용하며 다운로드에 관해서는 HTTP를 사용한다.

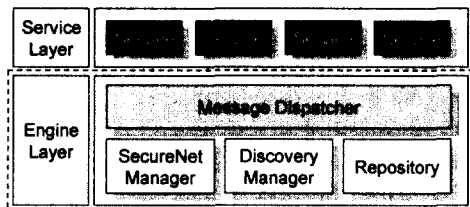
프리넷[4]은 Ian Clarke의 논문[5][6]을 바탕으로 익명성을 완전히 보장한 상태에서 분산된 정보의 저장과 검색을 하려는데 목적이 있다. 기본적으로 프리넷은 누텔라의 브로드캐스트(Broadcast)검색을 하지 않고 냅스터의 중앙집중식의 인덱스를 포함하지 않는다. 그리고 기본적으로 데이터가 자동적으로 정보의 인기도에 따라서 네트워크를 따라서 복사, 이동, 삭제된다.

누텔라와 프리넷은 P2P의 활용 예 중 서버가 없는 순수 P2P 모델을 이용함으로써 익명성을 제공하고 자원을 공유할 수 있었다. 그러나 P2P 응용은 개인경매(또는 EC), 지식공유, Collaboration과 같은 응용으로 발달하고 있는 현재 누텔라와 프리넷의 메시지 처리 기법은 순차적 방식만을 채택하고 있어 다양한 응용 서비스에 대처하는 유연한 구조를 가지고 못하고, 인터넷 환경에서의 메시지 표준으로 자리잡고 있는 XML을 지원하지 못한다.

3. 글로벌 P2P 엔진

본 장에서는 본 논문에서 제시하는 메시지 디스패처를 적용한 글로벌 P2P 엔진에 관하여 설명한다.

글로벌 P2P 엔진은 웹, 모바일 환경을 지원하는 자유로운 네트워크의 구성 및 유무선 네트워크 환경을 지원하고, 기존의 P2P 엔진들이 제공하는 파일, 지식공유와 같은 일반적인 서비스와 멀티미디어 스트리밍과 협력 작업과 같은 실시간 서비스를 지원하고, 콘텐츠의 투명성을 제공하는 데이터 교환 방법인 XML을 지원하고, 보안 기능이 강화된 P2P 엔진이다. 글로벌 P2P 엔진의 구조는 [그림 2]와 같고, 크게 상위계층의 서비스 레이어와 하위계층의 엔진 레이어로 나눌 수 있다.



[그림 1] 글로벌 P2P 엔진의 구조도

상위계층의 서비스 레이어는 쪽지, 채팅, 스트리밍서비스, 협력작업 서비스와 같은 서비스로 구성되어 있다.

하위 계층의 엔진 레이어는 각 모듈간에 전달되는 메시지들을 관리해주는 메시지 디스패처, 데이터를 관리해주는 레파지토리, 다른 피어를 검색하고, 생존 여부를 확인하는 디스커버리관리자, 다른 피어에게 메시지를 전달하기 위한 보안네트워크 관리자로 구성되어 있다. 보안 네트워크 관리자는 암호화 기능을 가지고 있는 모듈로써 사용자의 요구 또는 서비스의 종류에 따라서 다른 피어에게 전송되어진 메시지들을 암호화하거나 송신된 메시지를 복호화 하는 작업을 수행한다.

또한 글로벌 P2P 엔진에서 전달되는 모든 메시지들은 XML 포맷으로 구성되어 있다. 글로벌 P2P에서 전달되어지는 XML 문서 포맷은 <표 1>과 같다. 메시지의 포맷은 우선 순위, 생성일, 마감시간에 대한 정보를 가지고 있는 최상위의 <BODY>엘리먼트가 있으며 하위에는 다른 피어에게 전송할 메시지에 관한 정보를 가지는 <MSG>엘리먼트, 보안을 위한 보안정보를 가지고 있는 <SECURITY>엘리먼트, 레파지토리에 저장할 정보를 가지는 <REPOSITORY>엘리먼트로 구성되어 있다

[7].

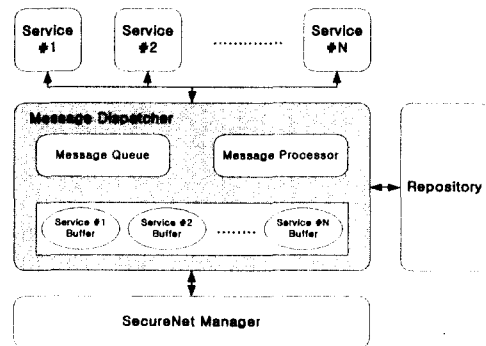
본 논문에서 제시하는 메시지 디스패처는 <BODY>엘리먼트의 우선 순위 값과 마감시간 값에 따라 Message Queue에 적재하고, Message Processor에서 서비스의 종류, 송신자, 수신자의 속성 값을 분석하여 알맞은 서비스 또는 다른 모듈에게 전달을 하는 역할을 한다.

<표 1> XML 문서 포맷

```
<?xml version="1.0" ?>
<!DOCTYPE Message SYSTEM "Message.dtd"
<BODY PRIORITY = "1" CREDATE="2001:03:01:18:05"
DUEDATE="2001:03:01:18:06">
<MSG>
<COMMAND> SEND </COMMAND>
<PROTOCOL> TCP </PROTOCOL>
<FROM> 163.180.116.185 </FROM>
<TO> 163.180.116.184 </TO>
<APP> CHAT </APP>
<CONTENT> ..... </CONTENT>
</MSG>
<SECURITY>
<LEVEL>1</LEVEL>
<ALGORITHM>DES</ALGORITHM>
<KEY LENGTH="56">Key</KEY>
</SECURITY>
<REPOSITORY>
.....
</REPOSITORY>
</BODY>
```

4. 메시지 디스패처 설계

본 장에서는 글로벌 P2P 엔진에서 각 모듈간에 발생하는 메시지들을 처리하는 메시지 디스패처의 설계 방법을 제시한다. 메시지 디스패처의 전체구조와 엔진의 타 모듈과의 연동되는 구조도는 [그림 2]와 같으며, 메시지 디스패처는 각 모듈에서 발생된 메시지를 Queue에 적재하는 Message Queue, Queue에 적재된 메시지들을 해당 모듈로 전달 해주는 역할을 하는 Message Processor, 메시지의 전달을 위해서 각 모듈과 메시지 디스패처가 사용하는 공유 버퍼인 Service Buffer로 구성되어져 있다.



[그림 2] 메시지 디스패처의 설계도

4.1 Message Queue

Message Queue는 P2P 엔진에서 발생하는 메시지를 Message Queue에 배치하는 역할을 한다. 이 Message Queue의 핵심은 P2P 엔진에서 발생하는 메시지들을 어떠한 정책에 의하여 Queue에 배치 할 것인가 하는 문제이다. 본 논문에서 언급하고 있는 글로벌 P2P 엔진은 다양한 서비스를 지원하며 각 서비스들은 실시간 특성을 지닌 서비스와 그렇지 않은 서비스로 구분할 수 있다.

따라서 본 논문에서는 메시지의 우선순위를 정하고 우선순위가 높은 것을 Queue의 앞쪽에 배치하며, 우선 순위가 같을 경우에는 메시지의 마감시간을 비교하여 마감시간이 짧은 것들을 앞쪽에 배치한다. 메시지들의 우선 순위는 VOD 서비스, 채팅, 협력작업 등과 같은 실시간 서비스들을 우선순위 1로, E-mail, Ftp와 같은 일반적인 서비스들 우선순위 2로 한다. 이러한 정책을 이용하여 Message Queue는 메시지의 우선순위와 마감시간을 검사하고, 각 메시지의 우선순위와 마감시간에 따라서 메시지들을 Queue에 배열하는 작업을 수행한다.

4.2 Message Processor

Message Processor은 Message Queue에 배치되어 있는 메시지들을 순차적으로 처리하는 모듈로써 XML 파서를 이용하여 메시지들을 분석하고 각 모듈로 메시지를 전달한다. Message Processor가 각 메시지들을 처리하는 방법은 <표 3>과 같다.

<표 2> Message Processor의 메시지 처리방법

```

1: Analysis Message by use XML Parser
2: Store Message in Buffer
3: if send other Service then
    send Event in Service
4: else if send other Peer then
    send other Peer information
    in SecureNet Manager
    send Message in SecureNet Manager
5: else if send Repository then
    send Message type in Repository
    store Message
    
```

Message Processor의 메시지 처리방법은 먼저 전달되어진 메시지를 XML 파서를 이용하여 분석한다(Line 1). 분석이 끝난 메시지는 그 메시지에 해당하는 버퍼에 저장된다(Line 2). 다음으로 메시지의 종류에 따라 다른 처리과정을 거친다. 첫 번째로 다른 서비스에게 전달되어지는 메시지라면 해당하는 서비스에 이벤트를 발생시키고, 이벤트가 접수된 서비스는 버퍼로부터 메시지를 가져가게 된다(Line 3). 두 번째로 다른 피어에게 전달되는 메시지라면 다른 피어의 네트워크정보와 보안정보를 보안네트워크관리자에게 보내주고 버퍼에서 메시지를 가져가도록 한다(Line 4). 세 번째로 레파지토리에 전달되는 메시지라면 메시지 타입에 맞게 메시지를 저장하도록 메시지의 타입을 레파지토리에 알려주고 레파지토리에 저장한다(Line 5).

4.3 Service Buffer

Service Buffer는 각 서비스 또는 모듈별로 메시지 처리에 대한 결과물을 얻기 위한 메시지 디스패처와 공유된 버퍼로써, 서비스들이 엔진에 추가되면서 각 서비스별로 한 개씩의 버퍼를 생성한다. 또한 이 버퍼들은 다양한 서비스들의 특성을 고려하여, [8]의 Push와 Pull 기능을 동시에 지원하는 버퍼관리기법을 적용하여, Push와 Pull 방식을 모두 지원하는 버퍼이다.

본 논문에서 제시하는 메시지 디스패처는 앞에서 열거한 3가지 주요 모듈로 구성되어 있으며, 우선순위를 이용한 메시지 처리로 인하여 다양한 서비스의 동시 작업시 신속하고 빠른 메시지 처리를 할 수 있다는 장점을 가지고 있다. 또한 XML 포맷을 지원하기 때문에 플랫폼 독립적이고, 각 모듈이 컴퍼넌트화 되어 있기 때문에 P2P 엔진과 유사한 다른 시스템의 메시지 처리부분에 활용하기 용이하다.

5. 결론 및 향후 연구

본 논문에서는 글로벌 P2P 엔진의 주요 모듈인 메시지 디스패처의 설계 기법에 관하여 제시하였다.

메시지 디스패처는 XML을 이용하여 PC, 모바일, 웹과 같은 다양한 네트워크 환경에 적합하고, 다양한 서비스를 지원할 수 있는 글로벌 P2P 엔진의 지원을 위한 모듈로써 서비스와 다른 모듈에서 발생하는 메시지들을 분석하고, 알맞게 연결해줌으로써 효과적인 P2P 서비스를 할 수 있도록 하는 모델을 제시하였다.

하지만 본 논문에서 제시한 메시지 디스패처의 모델이 실제적으로 효율적이고, 신속한 메시지의 처리의 검증이 필요하다. 이 모델에 대한 검증을 위해서는 글로벌 P2P 엔진과 그 서비스들을 구현하여 메시지 디스패처의 성능을 검증하고, 다양한 테스트를 통하여 부족한 부분을 보완하는 것이 향후연구의 방향이다.

참고문헌

[1] Intel, PtP Working Group, <http://www.peer-to-peerwg.org>
 [2] <http://www.napster.com/>
 [3] <http://gnutella.wego.com/>
 [4] Freenet, <http://freenet.sourceforge.net>
 [5] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong, Freenet: A Distributed Anonymous Information Storage and Retrieval System in Designing Privacy Engancing Technologies: Anonymity and Unobservability, LNCS 2009, ed. by H. Federrath, Springer: New York, 2001.
 [6] I. Clarke, A Distributed of Informatics, University of Edinburgh, 1999.
 [7] "XML Stepbystep", Michael J. Young, 정보문화사, 2001년 10월
 [8] 정찬균, 이승룡, "멀티미디어 통신시스템을 위한 클라이언트에서의 Push/Pull 버퍼관리기법", 정보처리학회 멀티미디어 특집 논문집, pp. 721-732, 2000년 3월