

사용자 접속 히스토리를 이용한 P2PSIP에서의 호 설정 지연 감소 방안

이의택*, 송주석*

*연세대학교 컴퓨터과학과

{dtmaster, jssong}@emerald.yonsei.ac.kr

Reducing Call Setup Delay Using Connection history of the User in P2PSIP

UiTaek Lee*, JooSeok Song*

*Dept of Computer Science, Yonsei University

요 약

최근 Skype의 인기와 더불어 P2P기반의 VoIP에 대한 관심이 높아지고 있다. 이에 표준 콜 셋업 프로토콜인 SIP와 동등 계층 간 통신망 구성의 P2P 기술을 결합하여 분산된 VoIP 서비스를 제공하려는 연구가 활발하게 진행 중에 있다. P2P에서는 중앙 엔티티가 존재하지 않으므로 사용자의 위치정보를 효율적으로 분산시키고 신속한 검색 방법을 제공하는 것을 중요한 문제로 다루고 있다. P2PSIP에서는 파일공유 목적의 P2P 서비스와는 다르게 특정 사용자의 각종정보(위치, 프로파일 등)를 빠르고 정확하게 찾아내야한다. 분산된 사용자의 위치정보를 검색하기 위하여 주로 DHT를 사용하고 있다. 하지만 매번 새로운 접속마다 사용자의 위치를 다시 찾아야 하기 때문에 오버레이 네트워크의 트래픽을 증가시킨다. 또한, 오버레이 네트워크에 참여하는 노드의 수가 증가함에 따라 사용자의 검색시간이 길어지는 단점을 가지고 있다. 본 논문에서는 DHT로 찾아낸 사용자의 접속패턴 히스토리 정보를 캐쉬로 저장하여 호 설정 지연을 감소시키는 방안을 제안한다.

1. 서론

최근 패킷망기반의 음성통화 서비스의 인기와 더불어 P2P(Peer-to-Peer)기반 VoIP에 대한 관심이 높아지고 있다. 그 중 Skype[1]는 전 세계적으로 2억이 넘는 사용자를 보유하고 있을 정도로 가장 성공적인 VoIP(Voice over IP) 서비스로 인정받고 있다. 하지만 Skype는 비공개, 비표준의 프로토콜을 사용하고 있는 것으로 알려져 있다. 이에 따라서 표준 프로토콜을 이용한 VoIP서비스의 필요성이 대두되었고 시그널링 프로토콜인 SIP(Session Initiation Protocol)[2]를 동등계층 간 통신망 구성인 P2P와 결합하여 분산된 VoIP서비스를 제공하려는 연구가 활발히 진행 중이다. 또한 현재 IETF(Internet Engineering Task Force)의 P2PSIP 작업반(Working Group)을 중심으로 관련 표준화 작업이 진행 중에 있다.

일반적인 클라이언트-서버 방식의 VoIP 통신에서 유저는 로그인시 자신의 위치를 서버에 등록하고 서버에게 질의를 통해 상대의 현재 위치를 파악하거나 프록시(Proxy) 서버를 이용하여 통신을 하게 된다. 하지만 P2P에는 중앙 엔티티(Entity)가 존재하지 않기 때문에 사용자

의 정보를 어디에 저장하고 어떻게 찾는지가 중요한 문제로 떠오르게 된다.

P2P의 방식에는 크게 구조적(Structured)방식과 비구조적(Unstructured)방식이 존재 한다. 비구조적 방식에서는 질의를 플러딩(flooding)하여 자료를 찾는다. (Kazza[3]와 Gnutella[4]와 같은) 초기 P2P 프로그램이 이 방식에 해당한다. 일반적인 파일 공유 목적의 P2P와는 다르게 P2PSIP에서는 정확히 특정 사용자의 위치를 찾아야 하기 때문에 구조적 방식이 사용되고 있는 추세이다.

구조적 방식에서 가장 중요한 개념은 DHT(Distributed Hash Table)이다. 2000년대 초반부터 P2P네트워크에서 효율적인 자원보관과 검색서비스에 대한 요구로 인해 Chord[5], Pastry[6], Can[7] 등과 같은 DHT 알고리즘에 대한 연구가 진행되어 왔다. DHT는 특정 리소스를 누가 가지고 있는지에 대한 정보를 오버레이 네트워크에 참여하는 노드 각각에 분배하는 방식이다. 때문에 P2PSIP에서 DHT의 알고리즘은 호 설정 지연(Call Setup Delay)에 큰 영향을 미친다. 이 호 설정 지연은 SIP의 시그널링 퍼포먼스에서 중요한 역할을 하는 Quality of Service(QoS)의 지표중 하나이다.

본 논문에서는 DHT의 알고리즘과 관계없이 DHT를 사용하여 얻어진 사용자들의 접속패턴 히스토리를 이용하여 호 설정 지연을 줄이는 방법을 제안한다.

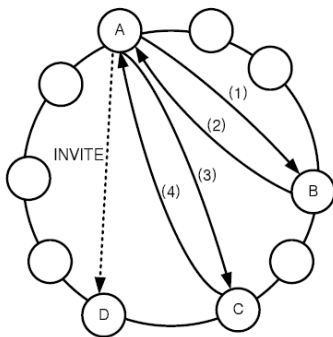
“이 논문은 2008년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (R01-2006-000-10614-0).”

2. 관련연구

SIP[2]는 시그널링 프로토콜로서 VoIP통신에서 호의 세션 수립에 사용된다. SIP의 주요 컴포넌트로는 User Agent(UA)와 Registrar서버, 그리고 Proxy서버 등이 있다. UA는 전화를 걸거나 받을 수 있는 단말로서 SIP전화기 이라고 생각할 수 있다. 각각의 UA는 유일한 주소인 AoR(Address of Record)이라는 SIP URI(Uniform Resource Identifier)로 구분된다. AoR은 “sip:foobar@office.com”과 같이 이메일과 유사한 형태로 표현된다. Registrar서버는 UA의 위치정보를 관리하며 Proxy서버는 UA들의 호를 릴레이 해주는 역할을 한다.

미국의 MIT와 UC Berkeley 대학의 학생들에 의해 고안된 Chord[5]는 DHT 알고리즘을 이용한 기법으로서, 링 구조로 이루어진 네트워크에 참여하는 노드들은 자신의 IP와 포트를 해쉬(hash)한 값을 통해 m-비트의 식별자, 노드ID를 생성한다. 또한 노드들이 가지고 있는 리소스 역시 동일한 방법으로 키를 계산한다. 이 노드ID와 키는 Chord-링이라는 가상의 공간의 원에 놓여지게 된다. 리소스의 키와 리소스의 URL(Uniform Resource Locator)은 현재 chord-링에 참여중인 노드들 중 자신의 노드ID보다 크거나 같은 최초의 노드가 저장하게 된다. 또한, 키 범위에 대한 효율적 관리를 위해 핑거 테이블이란 일종의 라우팅 테이블을 각 노드에서 관리하는데, 핑거테이블은 자신이 찾고자 하는 리소스를 찾기 위해 쿼리를 전송해야할 노드의 위치를 알려준다. 쿼리를 받은 노드는 자신이 알고 있는 리소스에 가장 근접한 노드를 알려주게 되고 이러한 쿼리를 반복하여 실제 리소스를 가지고 있는 노드의 위치를 최종적으로 찾아낼 수 있다.

일반적인 파일공유 목적의 P2P에서 리소스는 음악과 일과 같은 자원의 키워드에 해당되지만, P2PSIP의 경우 리소스는 UA의 AoR에 해당 된다.



(그림 1) Chord에서 사용자 검색과정

위의 (그림 1)을 보면 사용자 A가 D에게 전화를 걸고 있다. (1) A는 자신의 핑거 테이블을 확인하고 D의 AoR을 해쉬한 키에 가장 근접한 노드인 B에게 질의를 보내게 된다. (2) 동일한 방법으로 B는 자신의 핑거테이블을 검색

하여서 D의 키에 가장 근접한 C를 알려주게 된다. (3) 응답을 받은 A는 다시 C에게 질의를 보낸다. (4) C는 D의 키를 저장하는 책임을 가지고 있는 노드이다. 자신의 핑거 테이블에서 D의 위치정보를 발견하고 A에게 알려주게 된다. A는 D에게 INVITE 메시지를 보내 세션을 수립하게 된다. Chord는 검색 효율뿐 아니라 확장성(Scalability)과 로드밸런싱 등 많은 장점으로 인해 P2PSIP의 Underlying 프로토콜로 많이 사용되고 있다.

SOSIMPLE[8]에서는 SIP/SIMPLE 프로토콜을 이용한 분산된 VoIP, IM(Instance Messenger)를 제안하고 있다. Underlying 프로토콜로 Chord를 이용하고, REGISTER와 같은 SIP 메시지를 그대로 사용함으로써 기존에 존재하는 SIP/SIMPE 인프라스트럭처와의 호환성(Compatibility)을 보장하였다. 또한 완전히 분산된 구조를 이루기 위해 인증 및 보안상의 문제점 등을 제시하였다.

3. 제안 알고리즘

일반적인 사람들의 경우 대부분의 전화를 거는 대상은 친구에 국한된다. 또한 자주 이용하는 컴퓨터는 3대 정도에 불과하다. 예를 들면 오전에는 회사에 출근을 해서 A라는 IP의 개인전용 컴퓨터로 IM에 로그인을 하고 저녁에는 집에서 B라는 IP로 로그인을 하게 된다. 이처럼 한 사용자가 사용하는 IP와 IM에 접속하는 시간은 대부분 일정한 패턴을 나타내게 된다. 이러한 사용자의 접속시간과 IP의 패턴을 UA에 캐쉬(Cache)로 저장하고, 추후 로그인시에 버디의 위치를 파악할 때 사용하여 호 설정 지연을 감소시킬 수 있다. 또한 DHT를 이용한 검색 쿼리를 줄임으로써 오버레이 네트워크의 트래픽을 줄일 수 있다.

호 세션을 수립하기 위해서는 IP뿐 아니라 Port정보도 알아야 하지만 본 논문에서 접속포트는 항상 일정한 것으로 가정한다.

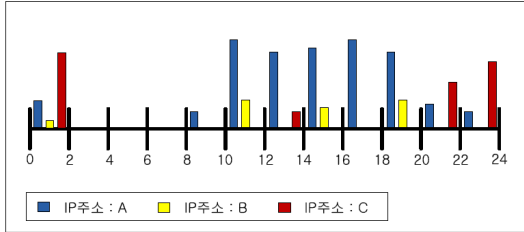
오버레이 네트워크에 접속하는 사용자는 로그인 서버를 통해 인증을 받은 후 자신의 UA에 저장되어 있는 버디의 접속패턴 테이블을 서버와 동기화하게 된다. 동기화를 통해 최신의 접속패턴 테이블을 UA에 저장한다.

SIMPLE[9] 프로토콜에서는 사용자의 프레즌스(Presence)를 확인하기 위해서 Subscribe/Notify 메시지를 이용한다. 버디에게 Subscribe 메시지를 보내서 현재 상태를 요청하고 버디는 Notify 메시지로 응답하여서 현재 상태를 알려주게 된다. 이후부터는 일정시간 간격으로 Notify 메시지를 보내서 자신의 상태를 통보해준다. 일반적인 IM과는 다르게 본 논문에서 Notify 메시지는 버디의 접속패턴 테이블을 만드는 목적으로만 이용되므로 30분에 한 번씩 보내는 것으로 설정한다.

접속패턴 테이블은 0시부터 23시까지의 시간으로 구성 되어있고, 각 시간에 해당하는 열에는 IP정보와 접속빈도가 저장된다. 버디의 로그인, 로그아웃 시간정보는 Subscribe/Notify 메시지가 최초로 도착하거나 더 이상 도

착하지 않는 시간을 이용한다. 접속빈도는 해당IP로 접속해 있는 동안 버디로부터 받은 Notify 횟수를 말하며, 동일한 시간에 여러 IP가 존재할 경우 우선순위를 정하는 척도로 사용된다.

아래의 (그림2)는 foobar라는 아이디를 가지고 있는 친구의 접속패턴 테이블을 도식화 한 것이다.



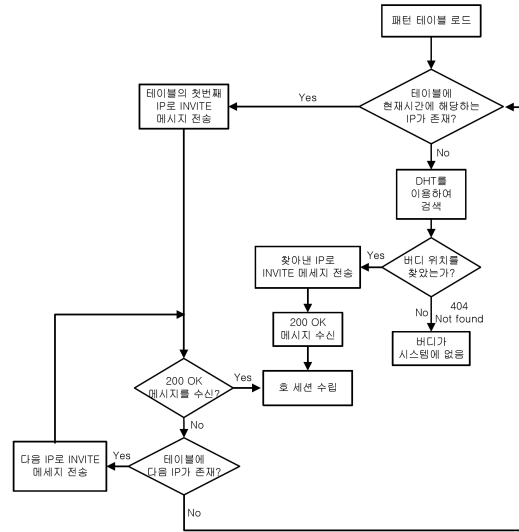
(그림 2) 사용자 foobar의 접속패턴 그래프

위의 그림에서 아래 숫자는 시간을 의미하며 막대 바는 접속빈도를 의미한다. 이 친구는 10시에 A라는 IP와 B라는 IP를 이용하여 접속한 적이 있었다. 접속빈도를 살펴보면 A가 B보다 훨씬 큰 것을 볼 수 있다. 때문에 10시에 이 친구에게 전화를 걸 때는 A에 먼저 INVITE 메시지를 보내게 된다.

만약, 위치예측에 실패하게 되면 접속빈도를 하나씩 감소시킨다. 이것은 사용자의 이어나 이직 등으로 접속패턴이 변경되었을 경우에 빠르게 새로운 접속패턴이 적용 되도록 해준다. 최종 DHT 검색까지 실패했을 때에는 사용자를 찾지 못한 것이 아니라 사용자가 접속하지 않은 것이므로 접속빈도를 감소시켜서는 안 된다.

이렇게 생성된 사용자의 패턴정보는 UA에 캐쉬로 저장되어 있다가 로그아웃 시 서버와 동기화하여 사용자의 모델리터를 지원하게 된다.

아래 (그림3)은 만들어진 패턴 테이블을 이용하여 버디에게 전화를 거는 과정을 순서도로 나타낸 것이다. 버디에게 전화를 걸면 제일 먼저 테이블에 현재 시간에 해당하는 IP가 있는지 검색하게 된다. 만약 존재한다면 그 중에서 가장 접속빈도가 높은 IP부터 INVITE 메시지를 보낸다. 200 OK 응답을 받게 되면 호 세션이 수립된다. 만약, 응답을 받지 못했을 경우에는 테이블에서 다음으로 접속빈도가 높은 IP로 INVITE 메시지를 보낸다. 테이블에 저장된 모든 IP로부터 응답을 받지 못했다면 DHT를 이용하여 사용자를 검색한다. 검색결과 404 Not found 메시지를 받았다면 그 사용자는 현재시스템에 접속해 있지 않은 것이다.



(그림 3) 접속패턴을 이용한 사용자 검색 및 콜 세션 수립

3. 결론 및 향후 연구계획

본 논문에서는 사용자의 접속패턴 히스토리를 이용하여 호 설정 지연을 줄이는 방법을 제안하였다. 사용자의 접속패턴이 거의 일정할 경우 호 설정 지연이 현저히 감소할 뿐 아니라 오버레이 네트워크의 트래픽도 줄일 수 있을 것이다. Chord의 경우, 오버레이 네트워크에 참여한 노드의 총 개수가 N일 때 $O(\log N)$ 의 검색비용을 갖는다. 제안한 알고리즘에서는 최선의 경우에는 $O(1)$ 의 검색비용을 갖지만, 최악의 경우에는 오히려 검색효율이 떨어지게 된다. 사용자의 패턴을 오랜 기간 동안 축적할수록 알고리즘의 정확도는 더 높아질 것이라 예상된다. 하지만 사용자의 접속패턴이 급격히 변화할 경우에는 오랜 기간 축적된 자료가 더 정확도를 떨어뜨리는 요인으로 작용할 것이다. 이에 변화된 패턴을 빠르게 인식하고 적용할 수 있도록 알고리즘을 보완해야 할 것이다. 또한 최선의 경우와 최악의 경우를 확률적으로 분석하여 평균적인 효율을 계산할 것이다. 끝으로 시뮬레이션을 통한 성능의 평가를 보완할 예정이다.

참고문헌

- [1] "Skype: Free Internet telephony that just works," <http://www.skype.com>
- [2] J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.
- [3] "Kazaa: peer-to-peer file sharing software application," <http://www.kazaa.com>.
- [4] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network", in Proceedings of the 1st IEEE International Conference on Peer-to-Peer Computing,

Aug 2001.

[5] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in Proceedings of SIGCOMM 2001, August 2001.

[6] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems," Proc. Middleware, 2001.

[7] S. Ratnasamy et al., "A Scalable Content Addressable Network," Proc. ACM SIGCOMM, 2001, pp. 161 - 172.

[8] David A. Bryan and Bruce B. Lowekamp. "SOSIMPLE: A SIP/SIMPLE Based P2P VoIP and IM System," College of William and Mary, Computer Science Department, Technical Report, Nov 2005.

[9] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. RFC 3428 - session initiation protocol (SIP) extension for instant messaging. <http://www.ietf.org/rfc/rfc3428.txt>, Dec. 2002.