

호 설정 지연을 줄이기 위한 계층적 P2PSIP에서의 효과적인 캐싱방법

이의택, 송주석
연세대학교 컴퓨터과학과
e-mail:{dtmaster, jssong}@emerald.yonsei.ac.kr

An Efficient Caching Mechanism for Reducing Call Setup Delay in Hierarchical P2PSIP

UiTaek Lee, JooSeok Song
Dept of Computer Science, Yonsei University

요 약

VoIP 통신에서 단말간의 세션을 수립하고 종료하는 역할을 하는 SIP는 기본적으로 서버-클라이언트 방식으로 이루어져있다. 하지만 이는 확장성과 설치비용, 단일실패 지점 측면에서 큰 단점을 갖는다. 최근에 DHT를 이용한 구조적인 P2P를 SIP와 결합하여 위의 단점들을 극복하기 위한 연구들이 활발히 진행되고 있다.

하지만 DHT를 이용한 방식은 기존 서버-클라이언트 방식에 비해 검색지연이 늘어난다는 큰 단점을 가지고 있다. 검색지연은 SIP에서 QoS의 중요한 지표중 하나인 호 설정 지연에 60~80% 정도의 큰 영향을 미치게 되므로 해결해야할 중요한 문제이다.

P2PSIP에서는 종단 노드가 검색한 사용자의 주소를 캐싱하여 같은 사용자를 재 검색시에 사용하고 있다. 이는 네트워크 전체 오버헤드 감소에 큰 영향을 주지 못할뿐만 아니라 발생된 검색비용을 자신만이 사용할 수 있으므로 비효율적이다.

이에 본 논문에서는 도메인으로 구분된 계층적 P2PSIP네트워크에서 프록시 서버의 역할을 하는 슈퍼노드를 이용한 효과적인 캐싱 방법을 이용하여 호 설정 지연을 단축할 수 있는 방안을 제안한다.

1. 서론

VoIP(Voice over IP) 통신에서 단말간의 세션을 수립하고 종료하는 역할을 하는 SIP(Session Initiation Protocol)[1]는 기본적으로 서버-클라이언트 방식으로 이루어져있다. 하지만 이는 확장성과 유지비용 측면에서 큰 단점을 갖는다. 이러한 단점들을 보완하기 위하여 Chord[2], CAN[3], Pastry[4]와 같은 DHT(Distributed Hash Table)를 이용하는 구조적 P2P(Peer-to-Peer)를 SIP와 결합하는 연구들이 최근 활발히 이루어지고 있다.

하지만 DHT를 이용한 방식은 기존 서버-클라이언트 방식에 비해 검색효율이 낮다는 큰 단점을 가지고 있다. 서버-클라이언트의 검색효율은 $O(1)$ 임에 비해 일반적인 DHT의 검색효율은, 전체 참여노드수가 N 일 때, $O(\log N)$ 을 갖는다. 게다가 Chord와 같은 DHT는 물리적인 위치와 논리적인 위치의 상이성 때문에 실제 논리적인 위치상 한 홉이라 할지라도 물리적인 위치상으로는 여러 홉 이상일 가능성이 높다. 따라서 DHT를 사용하는 구조적 P2P 오버레이 네트워크는 서버-클라이언트 방식에 비해 상당히 긴 검색지연을 가지고 있다. 이러한 검색지연은 SIP에서 QoS의 중요한 지표중 하나인 호 설정 지연에 60~80%

정도의 큰 영향을 미치게 되므로 해결해야 할 중요한 문제이다.

파일공유를 위한 전통적인 P2P에서는 검색시간이 길어도 사용자가 인내할 수 있지만 VoIP는 전화의 특성상 통화 연결음이 울리기까지 기다리는 시간은 굉장히 크게 느껴진다. P2P방식을 이용하는 가장 인기 있는 VoIP 프로그램인 Skype는 통화 연결음이 울리기 전까지의 시간이 3~8초의 상당히 긴 호 설정 지연을 가지고 있다.

이러한 지연을 해결하기 위하여 DHT 알고리즘의 개선, 효율적인 Overlay 구조 구성, 캐싱기법 등이 연구되었다.

기존의 캐싱 기법은 파일공유를 위하여 연구되었기 때문에 VoIP특성을 고려한 P2PSIP에서의 새로운 캐싱 기법이 필요하다. 현재까지는 P2PSIP에서의 캐싱 기법에 대한 연구가 거의 진행되고 있지 않다. 현재 P2PSIP에서 사용되는 캐싱 기법은 종단 노드에서 한번 검색한 사용자의 주소를 캐싱하여 같은 사용자와 재 통화 시 사용하는 단순한 방법을 취하고 있다.

대부분의 유저들이 DHCP(Dynamic Host Configuration Protocol)를 이용하여 유동적인 IP 주소를 할당 받는 것을

감안하면, 캐싱된 주소는 검색된 사용자가 오버레이 네트워크를 떠나기 전까지의 짧은 시간에만 이용 할 수 있다는 점과 한명의 사용자에게만 적용할 수 있다는 점에서 한계를 지닌다.

이에 본 논문에서는 계층적 오버레이 네트워크에서 프록시 서버 역할을 맡고 있는 슈퍼노드를 이용하여 같은 도메인에 속한 노드들의 검색 비용을 절약하고 호 설정 지연을 감소할 수 있는 효율적인 캐싱 방법을 제안한다.

2. 관련연구

SIP[1]는 시그널링 프로토콜로서 VoIP통신에서 호의 세션 수립에 사용된다. SIP의 주요 컴포넌트로는 User Agent(UA)와 Registrar서버, 그리고 Proxy서버, Redirect 서버 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. Redirect서버는 UA로부터 호를 받으면 목적지의 주소를 검색해서 UA에게 알려주는 역할을 수행한다.

미국의 MIT와 UC Berkeley 대학의 학생들에 의해 고안된 Chord[2]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 자원ID를 계산한다. 이 노드ID와 자원ID는 Chord-링이라는 가상의 공간의 원에 놓여지게 된다. 자원 ID와 자원의 URL(Uniform Resource Locator)은 현재 Chord-링에 참여중인 노드들 중 자신의 노드ID 보다 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 질의를 전송해야 할 노드의 위치를 알려준다. 질의를 받은 노드는 자신이 알고 있는 자원에 가장 근접한 노드를 알려주게 되고 이러한 질의를 반복하여 실제 자원을 가지고 있는 노드의 위치를 최종적으로 찾아낼 수 있다.

일반적인 파일공유 목적의 P2P 시스템에서 자원은 음악파일과 같은 자원의 키워드에 해당되지만, P2PSIP의 경우 자원은 UA의 AoR에 해당 된다.

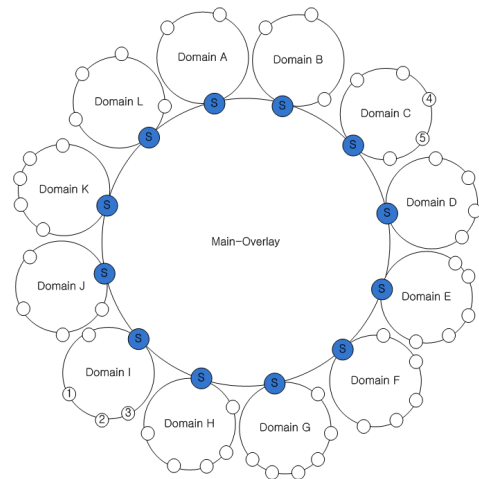
Chord는 실제 네트워크의 물리적인 위치가 고려되지 않았기 때문에 논리적인 링에서 한 홉일지라도 실제 물리적인 노드간의 위치는 멀리 있게 되어 큰 검색지연이 발생할 수 있는 단점이 존재한다. [5]에서는 물리적인 노드의 위치가 가까울수록 논리적인 위치도 가까운 곳에 위치시키도록 기존의 Chord 알고리즘을 수정하여 검색시간을

단축시켰다.

[6]에서는 검색지연 단축 등의 장점을 지닌 계층적인 P2P 시스템을 제안하였다. 이 논문에서 일반노드들은 여러 개의 그룹을 형성하고 각 그룹에서 성능이나 대역폭이 좋은 노드들은 슈퍼노드로 선정이 된다. 선정된 슈퍼노드들이 모여서 상위 오버레이를 구성한다. 각 하위 그룹은 각각 자신만의 검색 프로토콜(e.g., Chord, CAN, Pastry, Tapestry)을 운영할 수 있다. 다른 하위 그룹에 속한 노드의 검색은 슈퍼노드를 통해 이루어진다.

3. 제안 알고리즘

(그림1)과 같이 도메인으로 구분된 계층적 P2PSIP 네트워크에서 슈퍼노드는 프록시 서버의 역할을 한다. 다시 말하자면, 각 도메인에 속한 노드들은 슈퍼 노드를 통해 다른 도메인에 있는 노드의 위치를 검색하게 된다.



(그림 1) 계층적 구조의 P2PSIP 네트워크

이러한 구조에서는 각 오버레이마다 DHT를 운영하고 있기 때문에 총 3번의 $O(\log N)$ 의 검색비용이 발생한다. 예를 들어, 도메인 I에 있는 1번 노드가 도메인 C에 있는 5번 노드에게 전화를 한다고 가정한다면 도메인 I내에서 슈퍼노드를 검색할 때, 메인 오버레이에서 도메인 C의 슈퍼노드를 검색할 때, 마지막으로 도메인C에서 노드 5번을 검색할 때 총 3번의 검색비용이 발생한다. 이러한 계층적 구조에서는 도메인수가 많아질수록 메인 오버레이에서 각 도메인을 검색하는 시간이 길어진다[7].

기존 캐싱 알고리즘의 대부분은 P2P 파일 공유를 기반으로 연구되었기 VoIP의 특성을 갖는 P2PSIP에서는 적용되기 어렵다. 현재 P2PSIP에서의 캐싱 알고리즘에 대한 연구는 거의 이루어지고 있지 않고 있다.

사람들의 전화패턴을 조사해보면 자주 전화를 거는 일부 대상이 전체 전화통화량의 상당부분을 차지하는 것을 볼 수 있다[8]. 이를 근거로 우리는 각 노드가 자주 전화

를 거는 도메인의 상위 20%가 전체 통화량의 80%를 차지 한다고 가정하였다.

각 도메인에 속한 노드들은 간단한 랭킹 알고리즘을 운영하여 자신이 자주 전화를 거는 도메인 상위 20%에 해당하는 리스트를 유지하고 있다. 우리는 노드가 자주 전화를 거는 도메인을 Preference 도메인이라 한다.

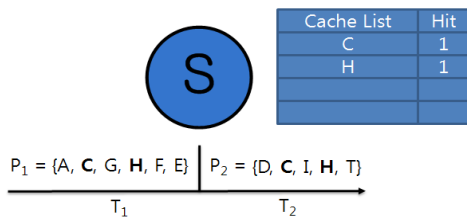
노드가 Preference도메인에 위치한 노드와 통화 시 슈퍼노드에게 이를 알려줄 수 있도록 아래와 같은 SIP헤더를 정의 하였다.

Preference = "Preference" HCOLON Preference-value
 Preference-value = "P"

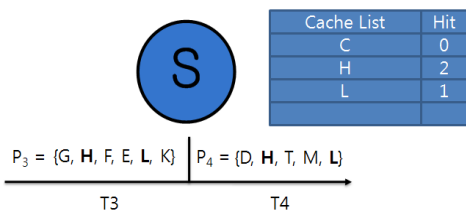
노드들은 다른 도메인에 있는 노드와 통화 시 전화 대상이 Preference 도메인에 속한 노드라면 INVITE 메시지에 P플래그를 설정하여 보낸다.

일정 시간 T1동안 슈퍼노드가 받은 Preference 도메인의 집합을 P1이라 하고, 일정시간 T2동안 받은 Preference 도메인의 집합을 P2라고 하자.

(그림2)와 같이 슈퍼노드는 P1과 P2 두 집합의 교집합을 구해 결과를 캐시 리스트에 저장한다. Hit 카운트는 캐시 리스트에 저장 될 때 마다 1씩 증가한다.



(그림 2) 캐시 리스트 생성 1



(그림 3) 캐시 리스트 생성 2

다음 캐시 리스트를 업데이트 할 때 Hit 카운트가 올라가지 않는 도메인은 1을 감소시킨다. Hit 카운트가 0보다 작아지는 도메인은 리스트에서 삭제된다.

이 알고리즘에서 가장 중요한 것은 어떠한 시간 간격으로 캐시 리스트를 업데이트 하느냐이다. 우리는 이 문제를 해결하기 위해 아래의 조건을 만들었다.

- 1) 교집합의 결과가 공집합이 되어서는 안 된다.
- 2) A1은 노드가 자주 전화하는 Preference 도메인의 집

합이고, A2는 슈퍼노드에 캐시된 도메인의 집합일 때 Identical Ration를 아래와 같이 정의 하였다.

$$\text{Identical Ratio} = \frac{A1 \cap A2}{A1}$$

Identical Ratio가 최대값을 갖는 시간간격을 구한다.

4. 향후 연구계획

본 논문에서는 구조적 P2PSIP에서의 효율적인 캐싱 알고리즘을 제안하였다. 중간노드인 슈퍼노드에서 캐시를 함으로서 한 노드가 전화를 걸어서 발생한 검색비용을 다른 노드가 다시 같은 도메인으로 전화를 걸 때 사용할 수 있으므로 검색비용을 절감할 수 있고 검색 지연을 감소시킬 수 있다. 향후 시뮬레이션을 통해 위에서 제시한 두 가지 조건을 만족하는 시간간격을 찾아 낼 것이다. 또한 제안된 캐싱 알고리즘을 적용하여 도메인 간 통화시의 호 설정 딜레이를 구하여 캐싱 알고리즘이 적용되지 않았을 때와 비교하여 평균 지연시간이 줄어든 것을 입증할 것이다..

참고문헌

- [1] J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.
- [2] 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.
- [3] S. Ratnasamy et al., "A Scalable Content Addressable Network," Proc. ACM SIGCOMM, 2001, pp. 161 - 172.
- [4] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-peer Systems," Proc. Middleware, 2001.
- [5] Xiong, J.; Zhang, Y.; Hong, P. & Li, J. "Chord6: IPv6 Based Topology-Aware Chord" Proc. ICAS-ICNS 2005, 2005, 4-4
- [6] Garces-Erice, L.; Biersack, E.; Felber, P.; Ross, K. & Urvoy-Keller, "Hierarchical Peer-to-Peer Systems" G Euro-Par 2003 Parallel Processing, pp. 1230-1239 , 2003
- [7] Zhang, C.; Shi, J.; Li, L.; Lin, W.; Wang, Y.; Gu, L.; Ji, Y. & Feng, Z. "Signaling Latency Analysis of Peer-to-Peer SIP Systems," Proc. 5th IEEE Consumer Communications and Networking Conference CCNC 2008, 2008, 505-509
- [8] Seshadri, M., Machiraju, S., Sridharan, A., Bolot, J., Faloutsos, C., and Leskove, J. 2008. "Mobile call graphs: beyond power-law and lognormal distributions." In Proceeding of the 14th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining