



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Master's Thesis

Social Relation-aware P2P Resource Control Algorithm

Donghyeok Ho (호 동 혁)

Division of Electrical and Computer Engineering
(Computer Science and Engineering).

Graduate School of Pohang University of
Science and Technology

2012





소셜 관계 기반의
P2P 자원 제어 알고리즘

Social Relation-aware P2P
Resource Control Algorithm



Social Relation-aware P2P Resource Control Algorithm

by

Donghyeok Ho

Division of Electrical and Computer Engineering
(Computer Science and Engineering)
Pohang University of Science and Technology

A thesis submitted to the faculty of Pohang University of
Science and Technology in partial fulfillment of the
requirements for the degree of Master in the Division of
Electrical and Computer Engineering
(Computer Science and Engineering).

Pohang, Korea

11. 28. 2011

Approved by
Hwangjun Song (Signature)
Academic Advisor



Social Relation-aware P2P Resource Control Algorithm

Donghyeok Ho

The undersigned have examined this thesis and hereby certify
that it is worthy of acceptance for a master's degree from
POSTECH

11/28/2011

Committee Chair	Hwangjun Song
Member	Chee Ha Kim
Member	Young Joo Suh

(Seal)
(Seal)
(Seal)



MECE

호 동 혁 Donghyeok Ho,

20100727

Social Relation-aware P2P Resource Control Algorithm

소셜 관계 기반의 P2P 자원 제어 알고리즘

Department of Computer Science and Engineering, 2011, 46P,

Advisor: Prof. Hwangjun Song.

Text in Korea.

Abstract

In this thesis, we propose a social relation-aware P2P resource control algorithm to guarantee the social fairness and to satisfy the QoS of video streaming. To achieve this goal, we define friendship by analyzing online social networking websites (OSNs), and model the credit mechanism using friendship and resource contribution to the network. Based on this credit mechanism, we propose a social relation-aware P2P resource control algorithm. It is constantly adjusted to achieve an effective trade-off the QoS of video streaming and the user's total credit. Finally, we provide experimental results to show the performance of the proposed algorithm.

Keywords: Social network, Incentive mechanism, Video streaming, P2P network, Resource control



목차

I. 서론	1
II. 관련 연구	5
2.1 비트 토렌트	5
2.2 소셜 네트워크	6
2.3 Scalable Video Coding (SVC)	7
III. 제안하는 소셜 관계 기반의 P2P 자원 제어 알고리즘	10
3.1 시스템 구조	10
3.2 친밀도 모델링	13
3.3 신용도 메커니즘	17
3.4 비디오 퀄리티 모델링	19
3.5 문제 수식화	20
3.6 제안하는 알고리즘	22
IV. 실험 결과	28
4.1 파일 다운로드 서비스의 성능 평가	30
4.2 비디오 스트리밍 서비스의 성능 평가	33
4.3 기존 알고리즘과의 성능 비교	36
V. 결론	41
참고문헌	42



그림 목차

그림 1. SVC의 시간적, 공간적, 화질적 확장성.....	8
그림 2. SVC (레이어 기반 스트림 표현).....	8
그림 3. 소셜 관계 기반의 P2P 비디오 스트리밍 시스템구조.....	11
그림 4. 부트스트랩 과정.....	12
그림 5. 소셜 네트워크 분석 : (a) 소셜 관계 (b) 친구 네트워크 (c) 친밀도 기반의 클러스터링	15
그림 6. 상태 공간 트리 ($G^{Max} = 3$, $L^{Max} = 4$ 의 경우).....	24
그림 7. The branch and bound 상태 공간 트리 ($G^{Max} = 3$, $L^{Max} = 4$ 의 경우).....	26
그림 8. The branch and bound 알고리즘	27
그림 9. 목표 신용도와 초기 신용도에 따른 총 신용도 변화 : (a) 초기 신용도=0.5 (b) 목표 신용도=0.5.....	31
그림 10. λ 값 조절에 따른 비디오 퀄리티와 총 신용도 변화 : (a) 비디오 퀄리티 변화 (b) 총 신용도 변화.....	36
그림 11. 기존 알고리즘과의 각 그룹에 대한 평균 업로드 레이트 비교.....	38
그림 12. 기존 알고리즘과의 평균 다운로드 레이트 비교	39



표 목차

표 1.	시스템 파라미터.....	30
표 2.	평균 다운로드 완료 시간(sec)과 평균 다운로드 레이트 (kbps)33	
표 3.	SVC 비디오 데이터	34



I. 서론

기존의 클라이언트-서버 기반 네트워크와는 다르게 Peer-to-Peer (P2P) 네트워크는 개개인의 사용자들이 서버와 클라이언트의 역할을 동시에 수행할 수 있는 장점을 가진다. P2P 네트워크의 급격한 성장은 클라이언트-서버 기반 네트워크의 효율적인 대안으로 제시되어 대규모의 콘텐츠 공유 뿐만 아니라 최근에는 멀티미디어 스트리밍 서비스를 위한 각광받는 커뮤니케이션 기술로 널리 사용되고 있다[1]. P2P 네트워크에서는 수많은 익명 (anonymous)의 사용자들이 어떠한 중앙 집중적 장치 없이 자기조직적 (self-organizing)이고 자급자족 (self-sufficient)한 시스템을 구축한다. 이러한 시스템은 동적이고 분산적이며 규제를 받지 않는다는 특징을 가지기 때문에 P2P 네트워크의 성능은 피어들의 자발적이고 협력적인 참여에 크게 의존하고 있다. 따라서 시스템의 성능을 높이기 위해서는 피어들 사이의 협업을 장려하고 네트워크의 형평성을 공정하게 보장하기 위한 인센티브 메커니즘이 필요하다. 지금까지 많은 논문 [2], [3]에서 P2P 네트워크에 적용될 수 있는 다양한 인센티브 메커니즘을 제시해 왔다.

Tit-for-Tat은 가장 대표적인 인센티브 메커니즘 중 하나로써 비트 토



렌트[4]에서 피어 선택 알고리즘으로 채택되어 사용되고 있다. Tit-for-Tat은 자신에게 가장 좋은 다운로드 레이트(download rate)를 제공해주는 피어들을 선택하여 자신의 업로드 레이트(upload rate)를 제공하게 되는데 간단하지만 효율적인 장점들 때문에 널리 사용되고 있다. 특히, 비트 토렌트는 하루에도 수백만 명의 사용자들이 사용하고 있으며 비트 토렌트가 점유하는 인터넷 상의 트래픽 또한 급격한 속도로 증가하고 있다[5]. 그러나 최근 들어 이러한 비트 토렌트의 한계점을 지적하는 연구들이 많이 발표되고 있다 [6], [7], [8], [9]. 그 중에서도 대표적인 문제점을 살펴보면 다음과 같다. 첫째, 조작 (manipulation)에 취약하다. 비트 토렌트는 악의적인 피어들이 고의로 클라이언트 프로그램을 수정하여 무임 승차(free-riding) [10], [11]함으로써 기존의 협력적인 피어보다 더 좋은 다운로드 레이트를 제공받게 되는 단점이 있다. 둘째, 다른 피어에게 자원을 할당할 때 오직 현재 받고 있는 다운로드 레이트 만을 고려한다. 이러한 근시안적인 자원 정책은 자신이 현재 충분한 서비스를 제공할 수 있는 환경이 아니라면 좋은 서비스를 기대하기 어렵다. 셋째, 자신에게 가장 좋은 서비스를 제공해주는 상위 몇 명의 피어들을 선택하지만 그들 사이에 정확한 차별성을 두지 않기 때문에 협력의 구체적인 수준까지는 고려하고 있지 않다. 마지막으로 사용자들의 관계를 단순히 익명성 (anonymity), 무관계성 (unrelated entities) 등으로 취급하고 있으며 그들 사이에 어떠한 소셜 관계(social relation)도 고려하고 있지



않다.

본 논문에서는 기존의 인센티브 정책의 문제점을 개선하기 위해 최근 큰 이슈가 되고 있는 소셜 네트워크의 특성을 적용하고자 한다. 온라인상에서도 사용자들 사이에 의미 있는 관계를 형성하여 다양한 서비스를 제공하는 온라인 소셜 네트워킹 웹사이트 (online social networking websites)가 급증하고 있다[12]. Facebook, MySpace, Window Live Spaces 등은 3000 만 명 이상의 사용자들이 등록된 대표적인 온라인 소셜 네트워킹 웹사이트이며 실제 오프라인에서처럼 사용자들 사이의 친밀도를 기반으로 비슷한 취미나 흥미를 가진 사람들끼리 커뮤니티를 형성하기도 한다.

본 논문에서는 이러한 소셜 관계를 분석하여 P2P 네트워크에 적용한 새로운 형태의 신용도(credit) 메커니즘을 제안한다. 이를 위해 먼저 온라인 소셜 네트워킹 웹사이트에 나타나는 개인의 소셜 관계를 분석하여 피어 사이에 나타나는 친밀도 (friendship)를 정의한다. 여기서 정의된 친밀도와 피어가 네트워크에 제공한 자원 기여도를 동시에 고려하여 피어 고유의 신용도를 계산한다. 신용도 메커니즘은 소셜 관계를 분석하기 위해서 피어 개개인을 고유의 ID로 식별하기 때문에 조작에 근본적으로 대처할 수 있다. 또한 신용도에 따라 차별적으로 자원을 할당하기 때문에 정확한 인센티브를 제공할 수 있으며 무임 승차와 같은 문제도 효과적으로 방지할 수 있다. 그리고 이러한 신용도 메커니즘을



P2P 비디오 스트리밍의 자원 교환 정책에 에 적용함으로써 효율적인 인센티브를 보장함과 동시에 피어가 요구하는 Quality-of-Service (QoS) 를 최대한 만족시킬 수 있는 효과적인 자원 제어 기법을 제안한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 관련된 연구를 소개한 후, 3장에서 제안하는 소셜 관계 기반의 P2P 자원 제어 알고리즘에 대해 구체적으로 설명한다. 4장에서는 실험을 통해 제안하는 알고리즘의 성능을 평가하며 마지막으로 5장에서 결론을 맺는다.



II. 관련 연구

2.1 비트 토렌트

이 절에서는 비트 토렌트에서 쓰이는 용어와 동작 과정에 대해 설명한다. 비트 토렌트는 트래커 (tracker) 라고 하는 기반 구조 노드를 갖고 있으며 어떤 피어가 파일 다운로드 요청을 위해 트래커에 접속하면 트래커는 현재 참여하고 있는 피어들 집합에서 임의의 피어들(일반적으로 50)을 선택하여 그들의 주소가 담긴 목록을 요청 피어에게 알려준다. 요청 피어는 트래커로부터 받은 목록에 있는 피어들과 TCP 연결을 설정하고 성공적으로 연결된 피어들과 청크를 교환해가면서 파일 다운로드를 진행한다.

P2P 네트워크는 예측 불가능한 특성을 가지기 때문에 시간에 따라 이웃 피어들의 목록과 피어가 가진 청크 집합이 불규칙하게 변동한다. 따라서 서로에게 어떤 청크를 요구할 지 선택하는 것이 중요하게 고려된다. 비트 토렌트에서는 희귀 청크 우선 선택 기법 (*rarest first chunk selection*)을 사용한다. 이는 자신의 피어 집합 안에서 피어들이 가장 드물게 가지고 있는 청크를 우선시하여 선택하는 방식으로 희귀한 청크를 빠르게 배포시킴으로써 전체 다운로드 속도를 향상시킬 수 있다.

또한 *choking* 알고리즘을 사용하여 주기적으로(일반적으로 10초)

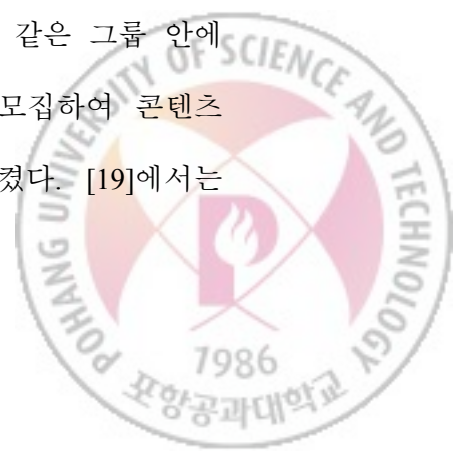


4명의 피어를 선택하여 업로드한다. 그리고 *optimistic unchoking* 알고리즘을 사용하여 매 30초마다 자신이 받은 서비스와는 상관없이 업로드를 요청한 피어 중 랜덤으로 하나를 선택한다. 이는 세션에 처음 참여한 피어가 협업을 활성화할 수 있도록 도와주고 피어 선택 시 로컬 미니멈 (local minimum)에 빠지지 않도록 방지하는 장점이 있다.

비트 토렌트는 위에서 언급한 기법 외에도 *pipelining*, *endgame mode*, *anti-snubbing* 등 효율적인 알고리즘을 많이 사용하고 있으며 실제로 널리 사용됨에 따라 처리율, 형평성 그리고 인센티브 메커니즘에 관해 많은 연구가 진행되었다[13], [14], [15].

2.2 소셜 네트워크

최근 들어 소셜 네트워크의 개념을 P2P 네트워크에 적용한 다양한 연구들이 진행되었다. F2F[16]는 P2P 저장 시스템을 구축하였으며 신뢰적이고 견고한 저장 노드를 찾기 위해 소셜 인센티브 정책을 사용한다. Sprout[17] 은 Chord DHT 라우팅 테이블의 성능을 높이기 위해서 추가적으로 온라인 상의 친구와의 신뢰적인 소셜 링크를 사용하여 쿼리 결과를 향상시키고 지연을 감소시켰다. Tribler[18]은 파일 공유 어플리케이션으로써, 소셜 그룹을 형성하고 같은 그룹 안에 참여하고 있는 자신의 친구들을 다운로드 도우미로 모집하여 콘텐츠 검색, 추천과 파일 다운로드 면에서 성능을 향상시켰다. [19]에서는

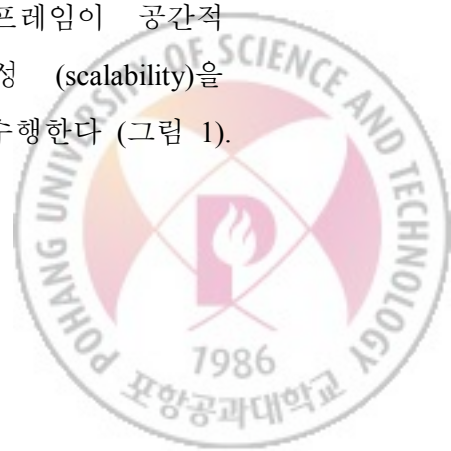


신뢰적인 피어들 사이의 검색 성능을 향상시키기 위해 비슷한 선호도, 관심도 또는 지식을 기반으로 소셜 P2P 네트워크를 조직하였다. [20]에서는 두 피어 사이의 친밀도를 P2P 네트워크 상에서의 방향성 있는 edge로 나타내었다. 각 edge는 신용도와 payment를 갖게 되고 피어 i 에서 피어 j 까지의 신용도는 피어 j 에서 피어 i 까지의 payment를 의미한다. 각 노드는 1-홉 이웃에 신용도와 payment를 할당하며 이러한 이웃들을 친구라고 부른다. 피어들 간의 전송된 데이터는 친밀도의 강함을 나타내는 데 사용되며 친밀도의 균형은 라우팅 경로를 결정하기 위한 결정 함수에 사용된다.

이와 같이 소셜 네트워크 기반의 P2P 시스템의 연구가 활발히 진행되어 왔지만 대부분 소셜 관계를 이용한 콘텐츠의 검색 및 등록 또는 파일 다운로드 서비스를 위한 성능 향상에 중점을 두었다. 본 논문에서는 기존의 연구와는 차별성을 두어 비디오 스트리밍 서비스를 제공하기 위해 피어들 사이의 자원 제어 기법에 소셜 관계를 적용한 새로운 알고리즘을 제안한다.

2.3 Scalable Video Coding (SVC)

SVC[21]는 영상 정보를 인코딩할 때 비디오 프레임이 공간적 (Spatial), 시간적 (Temporal), 화질적 (SNR) 확장성 (scalability)을 가지도록 하기 위해 계층형 부호화 (layered coding)를 수행한다 (그림 1).



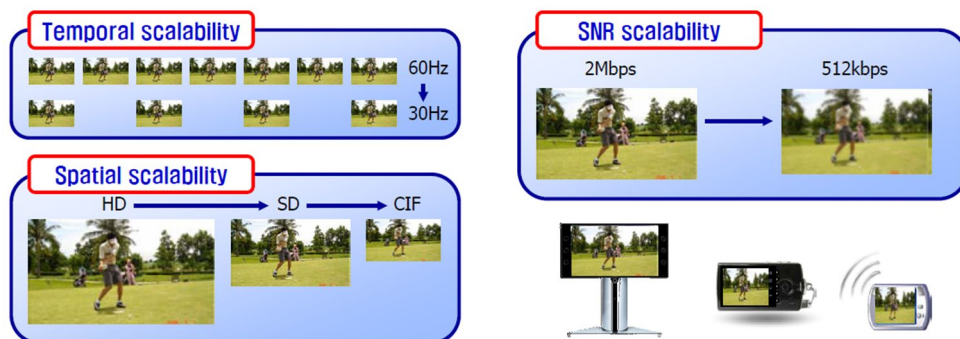


그림 1. SVC의 시간적, 공간적, 화질적 확장성

SVC 스트림은 여러 개의 세그먼트로 나누어지게 되는데 하나의 세그먼트는 시간적, 공간적, 화질적 확장성을 갖는 여러 개의 tier 를 포함하게 된다 (그림 2). Base tier 는 비디오의 가장 기본적 퀄리티를 제공하며 상위 tiers 는 전체적인 비디오의 퀄리티를 향상 시키는데 사용된다.

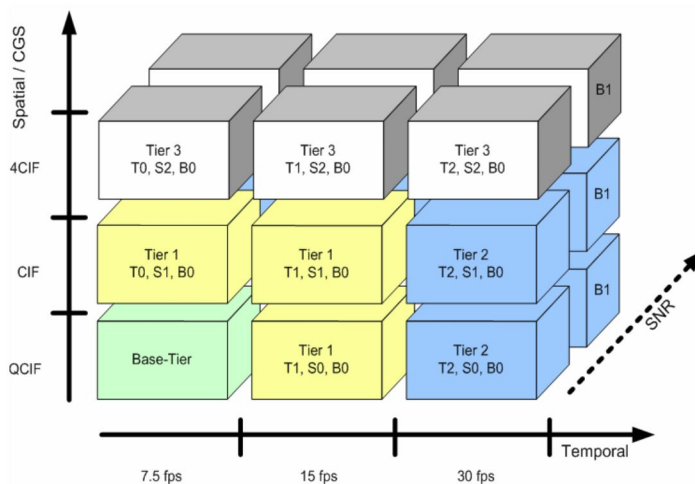


그림 2. SVC (레이어 기반 스트림 표현)



확장성을 지원하지 않는 부호화 방식은 채널의 대역폭이 증가하여도 이미 부호화된 비트스트림이 갖는 화질로만 서비스 되지만, SVC 로 압축된 비트스트림은 채널의 대역폭 변화에 맞추어 가변적인 화질로 전송 및 소비 될 수 있다. 따라서 각각의 피어는 자신의 환경에 따라 여러 단계의 퀄리티로 수신하는 것이 가능하다. 피어가 느끼는 비디오 퀄리티는 수신한 레이어 수에 비례하게 되는데 레이어 사이에는 내포적 의존성 (nested dependency)이 존재하기 때문에 상위 레이어는 반드시 그 밑의 레이어들을 성공적으로 복호화해야만 복호화가 가능하다.

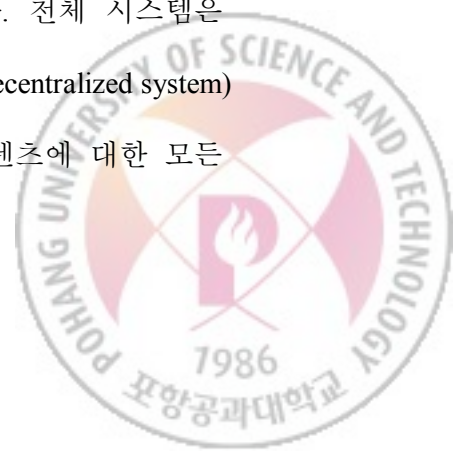


III. 제안하는 소셜 관계 기반의 P2P 자원 제어 알고리즘

제안하는 시스템의 목적은 소셜 P2P 네트워크의 형평성을 보장하면서 피어가 요구하는 비디오 퀄리티를 만족시킬 수 있는 효율적인 인센티브 메커니즘을 제공하는 것이다. 소셜 네트워크 상에서의 형평성이란 피어의 협력을 통한 자원 기여 뿐만 아니라 소셜 네트워크에서 나타나는 피어 사이의 친밀도를 함께 고려하여 공정한 서비스를 제공받는 것을 의미한다. 이를 위해 먼저 온라인 소셜 웹사이트를 통해 개인의 소셜 관계를 분석하여 피어 사이에 나타나는 친밀도를 정의한다. 그리고 여기서 정의된 친밀도와 함께 피어가 네트워크에 제공한 자원 기여도를 동시에 고려하여 피어 고유의 신용도를 계산한다. 마지막으로 이를 P2P 네트워크의 인센티브 정책에 적용함으로써 형평성을 보장함과 동시에 피어의 QoS를 만족시킬 수 있는 효과적인 자원 제어 기법을 제안한다.

3.1 시스템 구조

제안하는 시스템의 전체적인 구조는 그림 3과 같다. 전체 시스템은 오버레이 하이브리드 비집중적 시스템 (overlay hybrid decentralized system) 모델로써 서버는 현재 등록된 피어와 이용 가능한 콘텐츠에 대한 모든



정보를 관리하고 있다고 가정한다. 시스템의 목표 서비스는 Video-on-Demand (VOD) 이고 비디오는 SVC 를 이용해 부호화 되었다고 가정한다. 서버와 각 피어의 동작은 다음과 같다.

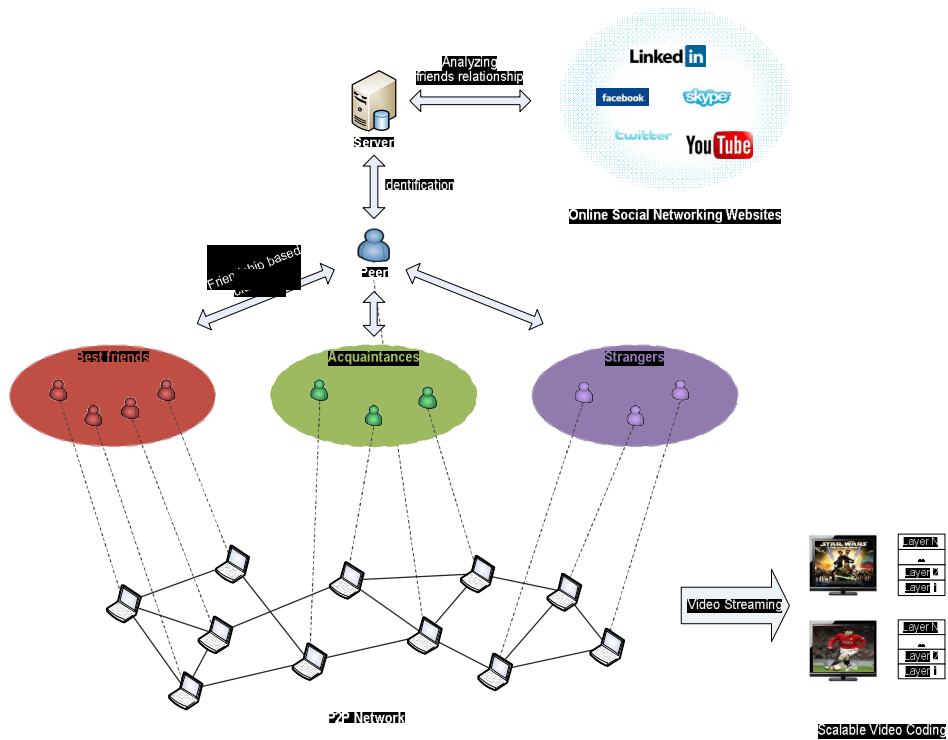


그림 3. 소셜 관계 기반의 P2P 비디오 스트리밍 시스템구조

서버는 크게 두 가지 역할을 한다. 첫째, 비트 토렌트의 트랙커와 같이 새롭게 참여한 피어들에게 요청한 콘텐츠와 배포자 (seeder)의 위치를 알려주고 공유집단 (coalition)에 조인할 수 있도록 도와준다. 둘째, 피어의 ID를 이용해 소셜 네트워크에 연결하여 피어가 형성하고

있는 소셜 관계에 대한 정보를 얻는다. 여기서 말하는 소셜 네트워크는 현재 널리 상용화 되고 있는 Facebook, Twitter, Skype 등의 온라인 소셜 네트워킹 웹사이트를 말한다. 그리고 이 정보를 분석하여 피어가 친밀도에 따라 자신의 이웃 피어들을 클러스터링 하는 것을 도와준다.

피어는 각 그룹의 피어들과 VoD 서비스를 위한 P2P 오버레이를 형성하고 비트 토렌트와 마찬가지로 자신이 원하는 청크를 주고 받으며 스트리밍을 진행한다. 기존의 비트 토렌트는 기본적으로 파일 다운로드 서비스를 대상으로 하기 때문에 이를 비디오 스트리밍 서비스에 그대로 적용하기에는 적합하지 않다. 파일 다운로드 서비스는 청크 선택 방식에서 순서를 고려하지 않고 희귀 청크 우선 선택 기법을 주로 사용하지만 비디오 스트리밍에서는 재생 시간에 임박한 청크가 보다 중요한 우선 순위를 갖게 되므로 기존의 청크 선택 방식보다 좀 더 세밀한 과정이 필요하다. 비디오 스트리밍을 위한 청크 선택 및 교환 프로토콜에 관한 내용은 [22], [23]에 자세히 설명되어 있다.

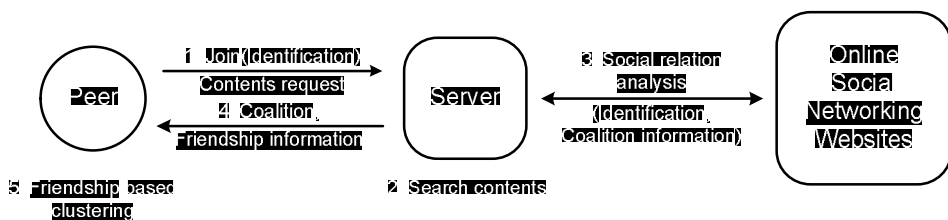


그림 4. 부트스트랩 과정

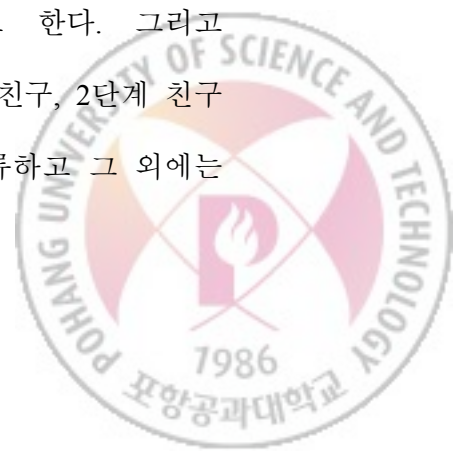
그림 4는 부트스트랩 과정을 나타낸다. 먼저 피어는 고유의 ID를 이



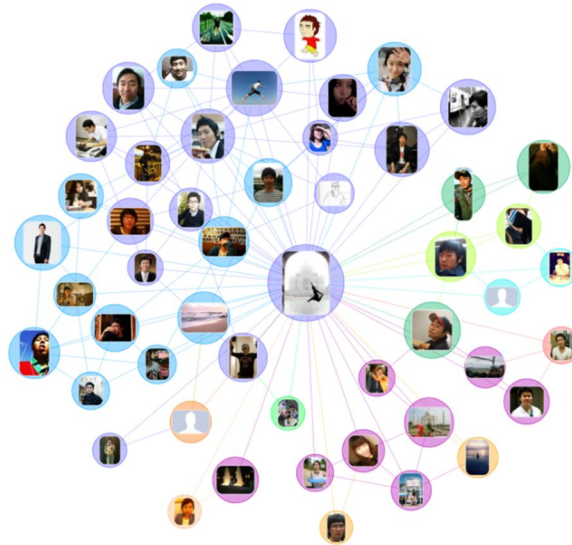
용해서 서버에 조인하며, 이 때 자신이 가진 콘텐츠의 메타데이터와 네트워크 정보를 함께 서버에 등록한다. 그리고 자신이 원하는 콘텐츠를 서버에게 요청한다. 요청을 받은 서버는 먼저 등록된 콘텐츠를 검색하여 배포자 (seeder)와 공유집단 정보를 얻는다. 그리고 온라인 소셜 네트워킹 웹사이트에 접근하여 요청 피어와 공유집단에 속한 피어 사이의 소셜 관계를 분석한다. 서버는 공유집단과 친밀도 정보를 피어에게 전달하고 피어는 친밀도를 기반으로 공유집단 내의 피어들을 클러스터링 한다. 위와 같은 과정 후 성공적으로 세션에 참여한 피어는 주기적으로 서버에 접속하여 피어 목록을 새롭게 갱신 한다.

3.2 친밀도 모델링

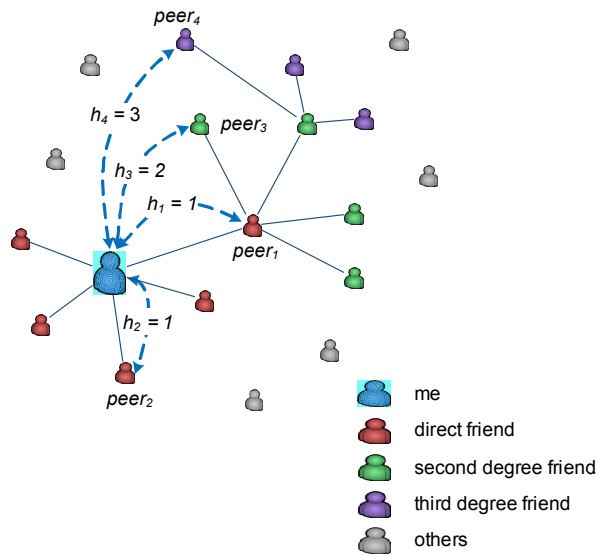
대부분의 온라인 소셜 네트워킹 웹사이트에서 사용자들은 친밀도, 신뢰성 (trust), 유사성 (similarity) 등 다양한 형태의 척도를 기준으로 서로 간에 의미 있는 관계를 형성하고 있다. 그리고 이러한 관계를 기반으로 각각의 목적에 맞는 서비스를 제공하고 있다. 예를 들면 YouTube나 Flickr에서는 관심 있는 콘텐츠의 유사성으로 사용자들을 분류하여 동영상이나 사진 서비스를 제공한다. Twitter에서는 지역, 학교, 직장 등의 유사성으로 사람을 검색하거나 정보를 공유하기도 한다. 그리고 LinkedIn과 Facebook에서는 개인의 소셜 관계를 1단계 친구, 2단계 친구 (친구의 친구), 3단계 친구 (2단계 친구의 친구)로 분류하고 그 외에는



다른 사람으로 구분 지어 친구 관계에 따라 서비스를 제공하는 범위를 제한할 수 있다.



(a)



(b)

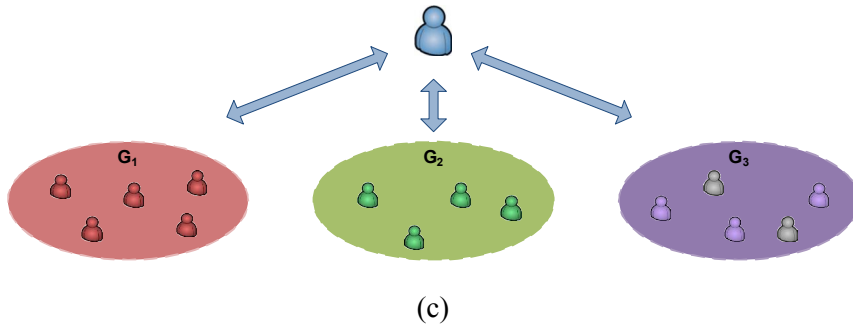


그림 5. 소셜 네트워크 분석 : (a) 소셜 관계 (b) 친구 네트워크 (c)
친밀도 기반의 클러스터링

제안하는 시스템에서는 LinkedIn과 Facebook에서와 같이 개인의 친구 네트워크를 분석하여 친밀도를 정의한다. 친구 네트워크란 개인의 친구 관계를 무방향성의 그래프로 나타낸 것으로써 각각의 노드는 피어를 의미하고 노드 간의 링크는 두 노드가 친구 관계라는 것을 의미한다. 온라인 소셜 네트워킹 웹사이트에 접속하여 개개인의 소셜 관계를 분석하는 어플리케이션[24]은 현재 많이 상용화되어 있기 때문에 제안하는 시스템에서도 충분히 적용 가능하다. 그림 5 (a)는 소셜 네트워크 상의 한 피어가 형성하는 소셜 관계를 보여준다. 이러한 개인의 소셜 관계는 그림 5 (b)와 같은 형태의 친구 네트워크로 나타낼 수 있다. 여기서 $h_i (1 \leq h_i \leq H^{Max})$ 는 친구 네트워크 상에서 자신과 $peer_i$ 사이의 거리를 의미한다. H^{Max} 는 두 피어 사이의 친구 관계가 가질 수 있는 최대 거리를 의미한다. 그림 5 (b)에서 $peer_1$ 과 $peer_2$ 는 나의 1

단계 친구가 되며 따라서 h_1 과 h_2 는 1이 된다. 마찬가지로 $peer_3$ 은 자신의 1단계 친구의 친구이기 때문에 h_3 는 2가 된다.

모든 피어는 그림 5 (c) 와 같이 동일한 거리만큼 떨어진 피어들을 묶어서 그룹을 형성하고 총 G^{Max} 개의 그룹을 갖게 된다. 이를 식으로 나타내면 다음과 같다.

$$G_k = \{peer_i | h_i = k\} \text{ for } \forall i, \quad (1)$$

$$G = \bigcup_{k=1}^{G^{Max}} G_k, \quad G_k \cap G_l = \emptyset \text{ for } \forall k, l \ (k \neq l), \quad (2)$$

G_k 는 h_i 가 k 인 $peer_i$ 들의 집합이고 G 는 피어가 가지는 모든 그룹의 집합이다. 같은 그룹에 속하는 피어는 같은 친밀도를 가지며 각 그룹의 피어는 서로 중복되지 않는다. 이를 바탕으로 피어가 G_k 에 가지는 친밀도 f_{G_k} 를 정의하면 다음과 같다.

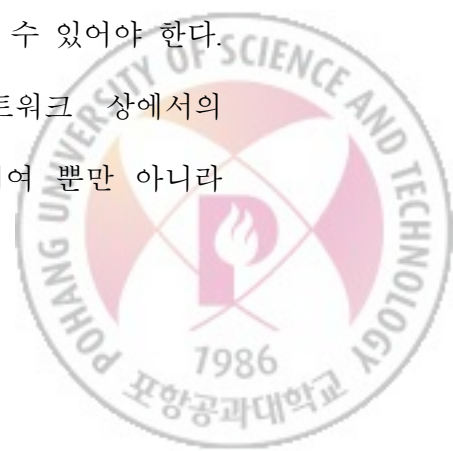
$$f_{G_k} = \frac{1}{h_i} \cdot \alpha \text{ for } peer_i \in G_k, \quad (3)$$

위 수식에서 α ($0 \leq \alpha \leq 1$) 는 상수이며, f_{G_k} 가 1 에 가까울수록 더 친밀한 그룹이 되고 0 에 가까울수록 덜 친밀한 그룹이 된다. 모든 피어는 자신만의 유일한 친구 네트워크를 형성하고 있기 때문에 제안하는 시스템에서 각각의 피어는 친밀도에 따라 고유한 그룹들을 갖게 된다.



3.3 신용도 메커니즘

P2P 네트워크에 참여하는 모든 피어는 시스템에 제공하는 자신의 자원을 자유롭게 결정할 수 있다. P2P 네트워크에서 사용되는 자원이란 업로드 레이트, 계산 능력 (computational power), 저장 공간 (storage space) 등 여러 가지가 있지만 본 논문에서는 VOD 서비스를 대상으로 하고 있기 때문에 피어가 제공할 수 있는 업로드 레이트라고 가정하겠다. 피어의 자원 기여 수준은 기기의 성능과 네트워크 환경과 같은 물리적 요인 뿐만 아니라 개개인의 이타적인 (altruistic) 성향에 따라 다르게 나타난다. 따라서 이를 공정하게 반영하여 서비스를 받을 수 있는 메커니즘이 필요하다. 기존의 P2P 네트워크에서 말하는 형평성은 일반적으로 시스템에 기여한 총 자원의 양에 따라서 자신이 받을 수 있는 서비스의 질이 결정되는 것으로써 제공한 자원이 많을수록 더 좋은 서비스를 보장함을 의미한다. 그러나 앞에서 설명하였듯이 인터넷 상의 무수히 많은 피어들은 서로 간에 크고 작은 의미 있는 관계를 형성하고 있다. 이러한 관계는 피어 간의 상호 작용이 중요한 P2P 기반 서비스에서 반드시 고려되어야 한다. 예를 들어 내가 현재 제공할 수 있는 자원이 충분치 않은 상황이라도 나와 친한 친구로부터 서비스를 받을 수도 있고 그 반대의 경우 또한 성립할 수 있어야 한다. 따라서 이러한 현상을 반영하기 위해 소셜 네트워크 상에서의 형평성을 새롭게 정의한다. 즉, 협력을 통한 자원 기여 뿐만 아니라



소셜 네트워크에서 나타나는 피어 사이의 친밀도를 함께 고려하여 개인의 신용도를 나타내고 이에 따라 서비스를 제공받음으로써 소셜 P2P 네트워크의 형평성을 보장한다. 제안하는 시스템에서 모든 피어는 각각의 그룹에 대해 신용도를 가지며 다음과 같이 정의된다.

$$c_{G_k}(t) = f_{G_k} \cdot c_{G_k}(t - \Delta) + (1 - f_{G_k}) \cdot \frac{u_{G_k}(t)}{u_{G_k}(t) + d_{G_k}(t)}, \quad (4)$$

위 식에서 $c_{G_k}(t)$ 는 t 시간일 때 G_k 에 가지는 신용도를 의미하고 $u_{G_k}(t)$ 와 $d_{G_k}(t)$ 는 피어가 t 시간에 G_k 에 제공한 업로드 레이트와 다운로드 레이트를 나타낸다. Δ 는 신용도가 갱신되는 기본 단위 시간이다. 업로드 레이트가 높을수록 피어의 기여도가 높기 때문에 신용도가 증가하고 반대로 다운로드 레이트가 높을수록 신용도가 감소한다. 또한 신용도는 과거 신용도와 현재 주고받는 자원 양 사이에 친밀도를 가중치로 갖게 되는데 이는 친밀한 그룹일수록 과거에 가지고 있던 신용도를 더 신뢰할 수 있기 때문에 과거 신용도에 비중을 두고 덜 친밀한 그룹일수록 과거에 가지고 있던 신용도보다는 현재 주고받는 자원 양에 더 비중을 두겠다는 의미이다.

모든 그룹에 대해 가지는 총 신용도는 각 그룹에 대한 신용도의 가중 합 (weighted sum)이 되며 다음과 같다.

$$c^{Total}(t) = \sum_{k=1}^{G^{Max}} \omega_{G_k} \cdot c_{G_k}(t), \quad (5)$$



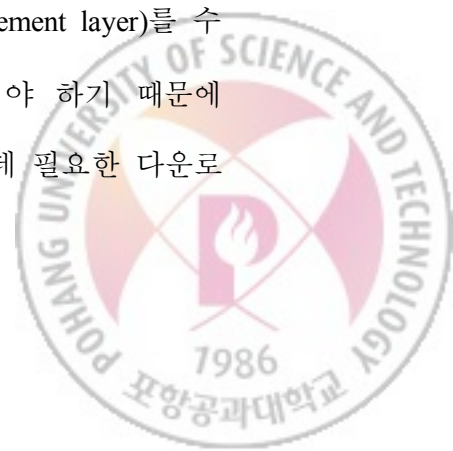
위 식에서 $c^{Total}(t)$ 은 t 시간에 갖는 총 신용도를 나타내며

$\omega_{G_k} = f_{G_k} / \sum_j^{G^{Max}} f_{G_j}$ 는 그룹 가중치 값 이다.

3.4 비디오 퀄리티 모델링

이 장에서는 SVC를 사용한 비디오의 퀄리티를 정의하는 방법에 대해 설명한다. [25]에 따르면 비트율 r 로 부호화된 비디오의 PSNR은 로그 함수 $\beta \cdot \log(r)$ 로 근사 (approximation) 될 수 있다. β 는 비디오 특성에 관련된 상수으로써 높은 β 는 많은 양의 움직임 (large amount of motion)이 있는 비디오에 사용되고 낮은 β 는 정지 장면 (still scenes)이 많은 비디오에 사용된다. 이 값은 소스 피어에서 비디오 부호화를 수행하여 얻을 수 있다. 그리고 이러한 근사는 SVC 경우에도 적용될 수 있다[16].

소스 피어가 총 L^{Max} 개의 레이어로 부호화한 SVC 비디오가 있다고 가정하자. $L = \{l_1, l_2, \dots, l_{L^{Max}}\}$ 은 레이어의 이진 벡터로 l_n 은 n 번째 레이어를 의미한다. 피어가 l_n 을 수신하였다면 1, l_n 을 수신하지 않았다면 0 이 되며 모든 피어는 기본 레이어 (base layer)를 기본적으로 수신하기 때문에 l_1 은 항상 1이다. SVC에서 상위 레이어 (enhancement layer)를 수신하기 위해서는 반드시 하위 모든 레이어를 수신해야 하기 때문에 $l_{n+1} \leq l_n$ 이 된다. r_n 을 n 번째 레이어를 스트리밍 하는데 필요한 다운로



드 레이트라고 한다면 비디오 퀄리티 $q_{l^{Max}}$ 는 다음과 같다.

$$q_{l^{Max}} = \beta \cdot \log \left(\sum_{n=1}^{l^{Max}} r_n \cdot l_n \right), \quad (6)$$

위 식에서 $l^{Max} (1 \leq l^{Max} \leq L^{Max})$ 는 피어가 다운로드한 가장 높은 레이어 인덱스 (maximum download layer index)를 의미한다. l^{Max} 가 클수록 수신하는 레이어의 수가 증가하기 때문에 비디오 퀄리티도 증가하게 된다.

3.5 문제 수식화

본 논문에서는 앞서 정의한 신용도와 비디오 퀄리티를 고려하여 소셜 P2P 네트워크의 형평성을 보장함과 동시에 피어가 요구하는 비디오 퀄리티를 만족시킬 수 있는 효율적인 자원 제어 기법을 제안한다.

문제 정의 :

다음 효용함수 $\Psi(t)$ 을 최대화 하는 l^{Max} 와 $\vec{g} = (g_1, g_2, \dots, g_{l^{Max}})$,

$\vec{U}(t) = (u_{G_1}(t), u_{G_2}(t), \dots, u_{G_{l^{Max}}}(t))$ 을 결정한다.

$$\Psi(t) = \lambda \cdot q_{l^{Max}} + (1 - \lambda) \cdot c^{Total}(t), \quad (7)$$

$$\text{subject to} \quad r_n \leq d_{g_n}(t) \text{ for } 1 \leq n \leq l^{Max}, \quad (8)$$



$$\sum_{n=1}^{I^{Max}} d_{g_n}(t) \leq D^{Max}, \quad (9)$$

$$\sum_{k=1}^{G^{Max}} u_{G_k}(t) \leq U^{Max}, \quad (10)$$

$$c^{Total}(t) \geq c^{Target}, \quad (11)$$

위 식에서 $\lambda(0 < \lambda < 1)$ 는 상수이며, $\vec{U}(t)$ 는 각 그룹에 제공하는 업로드 레이트를 나타내는 벡터이다. $\vec{g} = (g_1, g_2, \dots, g_{I^{Max}})$ 는 각각의 레이어를 다운로드 받는 그룹을 나타내는 벡터이며 여기서 $g_n (g_n \in G)$ 은 n 번째 레이어를 다운로드 받는 그룹을 의미한다. $d_{g_n}(t)$ 는 t 시간에 g_n 으로부터 제공받는 다운로드 레이트를 말한다. D^{Max} 와 U^{Max} 는 각각 피어의 최대 다운로드 용량 (maximum download capacity), 최대 업로드 용량 (maximum upload capacity)을 의미하며 $c^{Target} (0 \leq c^{Target} < 1)$ 은 피어가 원하는 목표 신용도 (target credit)를 의미한다. 이것은 최소한의 신용도 수준으로써 미래에 자신이 원하는 또 다른 서비스를 원활하게 제공받기 위해 현재 일정 수준의 신용도를 유지시켜 놓는 역할 (savings)을 한다고 볼 수 있다.

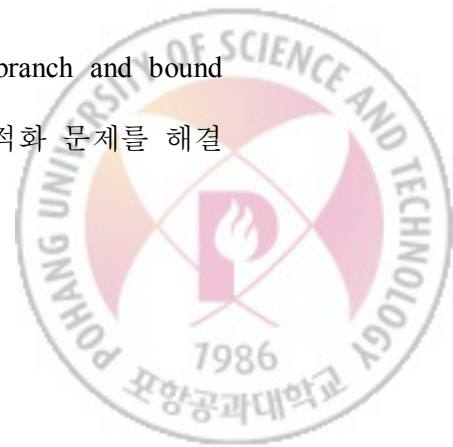
비디오 퀄리티는 I^{Max} 가 높을수록 증가하지만 많은 다운로드 레이트를 요구하기 때문에 상대적으로 총 신용도는 감소한다. 따라서



비디오 퀄리티와 총 신용도는 상관 관계 (trade-off)에 놓여 있으며 이를 효과적으로 조절하기 위해 다음 세가지 중요한 컨트롤 파라미터를 결정하는 것이 중요하다. 첫째, 비디오 퀄리티와 총 신용도 사이의 상관 관계를 고려하였을 때 얼마나 많은 레이어를 받아야 하는지(l^{Max}) 결정한다. 둘째, 각 그룹에 대한 친밀도와 신용도를 고려하였을 때 어느 그룹으로부터 각각의 레이어를 받을 것인지 ($\vec{g}=(g_1, g_2, \dots, g_{l^{Max}})$) 결정한다. 마지막으로, 비디오 퀄리티를 안정적으로 보장받고 목표 신용도를 유지하기 위해 각 그룹에 제공해야 하는 업로드 레이트를 ($\vec{U}(t)=(u_{G_1}(t), u_{G_2}(t), \dots, u_{G_{Max}}(t))$) 결정한다. 각 제약조건의 의미는 다음과 같다. (8)은 하나의 레이어는 한 그룹으로부터 다운로드하며 이때 원활한 비디오 재생을 위해 다운로드 레이트는 비디오 스트리밍을 위해 요구되는 비트 레이트를 만족시켜야 함을 의미한다. (9)과 (10)은 자신의 최대 용량을 초과하지 않는 범위 내에서 업로드와 다운로드가 이루어져야 한다는 것이다. (11)은 피어가 가지는 총 신용도를 목표 신용도 이상으로 유지시켜야 함을 의미한다.

3.6 제안하는 알고리즘

이번 장에서는 앞서 제시한 문제를 해결하기 위해 branch and bound 알고리즘을 설계한다. Branch and bound 알고리즘은 최적화 문제를 해결



하는 대표적인 기법 중의 하나로 언제나 최적 해를 보장하는 것이 증명되었다[26]. Branch and bound algorithm을 사용하기 위해서는 먼저 탐색할 상태 공간 트리(state space tree)를 정의해야 한다. 상태 공간 트리는 최적 해의 후보들을 트리 형식으로 표현한 것으로써 그림 6은 구축된 상태 공간 트리의 예를 나타낸다. 트리의 깊이는 레이어 인덱스를 나타내며 깊이가 깊어질수록 더 높은 레이어 인덱스를 의미한다. 루트 노드에서 1번 노드로 확장하는 것은 베이스 레이어를 그룹 1로부터 받는다는 의미이고 2번 노드로 확장하면 그룹 2로부터 받는다는 의미이다. 이런 식으로 계속 진행하여 상태 공간 트리를 구축하면, 루트 노드부터 리프 노드까지의 모든 경로는 각각 하나의 후보해를 나타내게 된다. 그러나 전체 해 공간을 탐색하는 것은 최적해가 될 가능성이 전혀 없는 노드들도 모두 탐색해야 하기 때문에 비효율적이다.



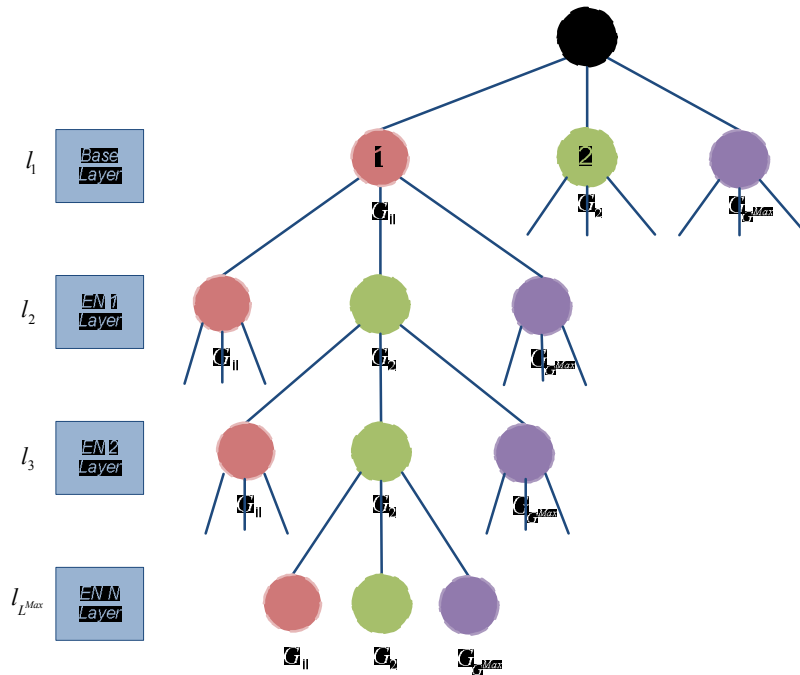


그림 6. 상태 공간 트리 ($G^{Max} = 3$, $L^{Max} = 4$ 의 경우)

Branch and bound 알고리즘의 기본 전략은 전체 해를 갱신해 가며 노드의 한계값(bound)과 비교하여 해 공간의 불필요한 부분 트리를 제거함으로써 탐색 공간을 최소화 하는 것이다. 따라서 각 노드를 탐색할 때마다, 그 노드가 유망한지의(promising) 여부를 결정하기 위해서 한계값을 계산한다. 여기서 말하는 한계값이란 그 노드로부터 가치를 뺀어 나가서 얻을 수 있는 최대 이익을 의미한다. 만약 한계값이 지금까지 찾은 최적의 해보다 좋지 않은 경우는 더 이상 가치를 뺀어서 탐색을 계속할 필요가 없으므로, 해당 서브 트리를 제거(pruning)한다. 노드의

유망성을 판단하기 위해서 다음의 조건식을 사용한다. 현재 트리의 깊이를 l , 탐색한 노드를 i 라고 가정한다.

$$\text{조건 1. } \sum_{n=1}^{l+1} d_{g_n}(t) \leq D^{Max}$$

만약 상위($l + 1$) 레이어까지 받을 경우 요구되는 다운로드 레이트가 피어의 최대 다운로드 용량보다 작거나 같다면 해당 노드는 유망하다고 할 수 있다.

$$\text{조건 2. } \sum_{k=1}^{G^{Max}} u_{G_k}(t) \leq U^{Max}$$

조건 1로부터 다운로드 레이트가 결정된다면 $c^{Total}(t) \geq c^{Target}$ 을 유지시키기 위해 각 그룹에 제공해야 하는 $\vec{U}(t) = (u_{G_1}(t), u_{G_2}(t), \dots, u_{G_{G^{Max}}}(t))$ 를 결정하게 된다. 이 때, 계산된 업로드 레이트가 피어의 최대 업로드 용량을 초과하지 않는다면 총 신용도가 목표 신용도를 보장하는 것을 의미한다. 이 경우 해당 노드는 유망하다고 판단한다.

$$\text{조건 3. } \Psi^{current}(t) < bound_i$$

$\Psi_k(t)$ 를 노드 k 에서 얻은 해라고 하였을 때 $\Psi^{current}(t) = \max_k \Psi_k(t)$ 는 현재까지 찾은 해 중 가장 큰 값을 의미한다. 한계값 $bound_i$ 는 현재 노드 i 를 루트 노드로 갖는 부분 트리의 노드 k 중 조건 1과 2를 만족하는 가장 큰 $\Psi_k(t)$ 를 의미한다. 만약 $\Psi^{current}(t)$ 가 $bound_i$ 보다 작다면 노드 i 는 유망하다고 할 수 있다.



(unexpanded) 노드를 살펴보고 그 중에서 가장 좋은 한계값을 가진 노드부터 확장한다. 트리의 정점에 도달하면 현재까지의 최적 해와 비교하여 더 좋은 경우에 갱신한다. 이와 같은 과정을 우선 순위 큐에 아무 것도 남지 않을 때까지 반복해 나가면 최적해를 얻을 수 있다. 제안하는 알고리즘의 세부적인 과정은 그림 8과 같다.

The branch and bound algorithm

1. priority queue of node PQ;
2. node u, v ;
3. initialize(PQ);
4. v = root of state space tree T;
5. insert(PQ, v);
6. current best $\Psi^{current}(t) = 0$;
7. **WHILE** Q is not empty **DO**
8. remove the node v with the highest bound (PQ, v);
9. **IF** promising(v) **THEN**
10. **FOR** each child u of v **DO**
11. calculate $\Psi_u(t)$;
12. **IF** $\Psi_u(t)$ is better than $\Psi^{current}(t)$ **THEN**
13. $\Psi^{current}(t) = \Psi(t)$;
14. **ENDIF**
15. **IF** promising(u) **THEN**
16. insert(PQ, u);
17. **ENDIF**
18. **ENDFOR**
19. **ENDIF**
20. **ENDWHILE**

그림 8. The branch and bound 알고리즘



IV. 실험 결과

본 실험은 NS-2 시뮬레이터 [27]를 사용하여 진행하였다. 피어의 참여는 λ 울의 푸아송 분포를 따른다고 가정하였다. 이러한 가정은 [14], [28], [29], [30], [31]에서도 널리 사용되고 있다. 본 실험에서는 $\lambda=30$ 으로 설정하였고 매 1분마다 평균 30명의 피어가 참여한다고 볼 수 있다. 피어의 네트워크 환경은 Cable/Ethernet 망을 가정하였고 각 피어의 최대 업로드 용량은 768 kbps, 최대 다운로드 용량은 3000 kbps로 설정하였다. 본 논문에서는 자원 할당 및 분배에 중점을 두고 있기 때문에 성능을 보다 명확하게 측정하기 위해 청크 선택 알고리즘은 제시하지 않고 서로에게 원하는 청크가 항상 있다고 가정하였다. 실험은 다음 두 가지 이슈를 중심으로 진행하였다. 첫째, 파일 다운로드 서비스와 비디오 스트리밍 서비스에 대하여 제안하는 시스템의 성능을 검증한다. 둘째, 기존의 다른 알고리즘과의 성능 비교를 통해 제안하는 알고리즘의 우수성을 증명한다.

친구 네트워크를 구성하기 위해 Facebook 사용자들을 대상으로 수집한 실제 dataset[32]을 사용하였다. Dataset은 New Orleans의 Facebook 사용자들을 분석한 것으로써 총 60,290명의 피어와 이들 사이의 1,565,686



개 친구 관계로 구성되어 있으며 평균 1 단계 친구 수는 25.3명이다. 이러한 모든 피어와 관계를 반영하기에는 복잡도 측면에서 제한이 있기 때문에 평균 1 단계 친구 수를 그대로 유지시킨 500명의 피어를 추출해서 친구 네트워크를 구성하였다. 그리고 친구 네트워크의 각 피어를 P2P 네트워크에 랜덤으로 맵핑시켜서 실험을 진행하였다. 실험을 위한 구체적인 파라미터는 표 1과 같다.

[33]에서는 무작위로 선택된 두 사람 간의 거리를 알기 위해 얼마나 많은 아는 사람들이 중간에 필요한가에 대해 실험하였으며 평균적으로 여섯 개의 링크를 거치면 어느 누구와도 연결될 수 있다는 것을 밝혀내었다. 이를 여섯 단계의 분리 (six degree of separation)라 한다. 따라서 일반적인 경우에 H^{Max} 는 6이라고 할 수 있다. 일반적으로 H^{Max} 가 6이기 때문에 G^{Max} 도 마찬가지로 6이 될 수 있지만 제안하는 시스템에서는 [33]의 실험에서처럼 무작위로 선택된 피어를 고려하기 보다는 어느 정도의 소셜 관계가 형성되어 있는 피어들을 대상으로 하고 있으며 Facebook과 같이 현재 상용화 되고 있는 서비스에서도 3 단계 친구 관계까지만 다루고 있기 때문에 G^{Max} 를 3으로 가정한다. 그리고 h_i 가 G^{Max} 보다 큰 $peer_i$ 에 대해서는 마지막 그룹에 함께 포함시키도록 한다. 그리고 한 그룹 내에 있는 모든 피어는 서로 동일한 친구 관계를 가지고 있다고 가정한다.



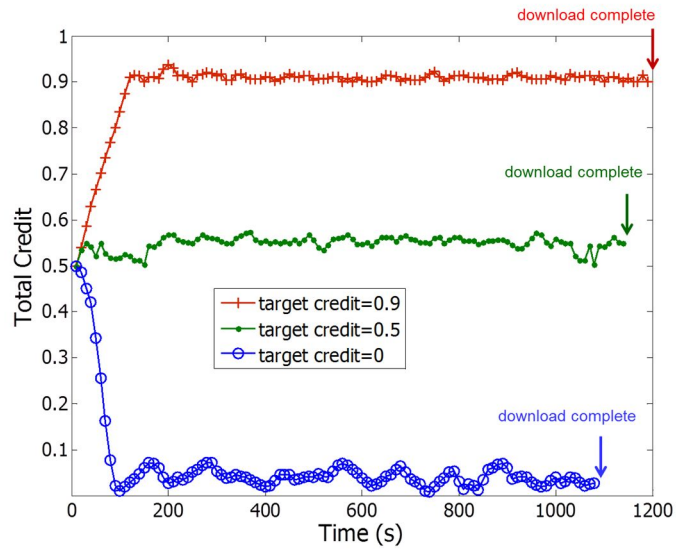
표 1. 시스템 파라미터

System parameter	Value
H^{Max}	6
α	0.9
G^{Max}	3
$f_{G_1} / f_{G_2} / f_{G_3}$	0.9 / 0.45 / 0.3
$\omega_{G_1} / \omega_{G_2} / \omega_{G_3}$	0.5 / 0.3 / 0.2
Δ	10 seconds
U^{Max}	768 kbps
D^{Max}	3000 kbps

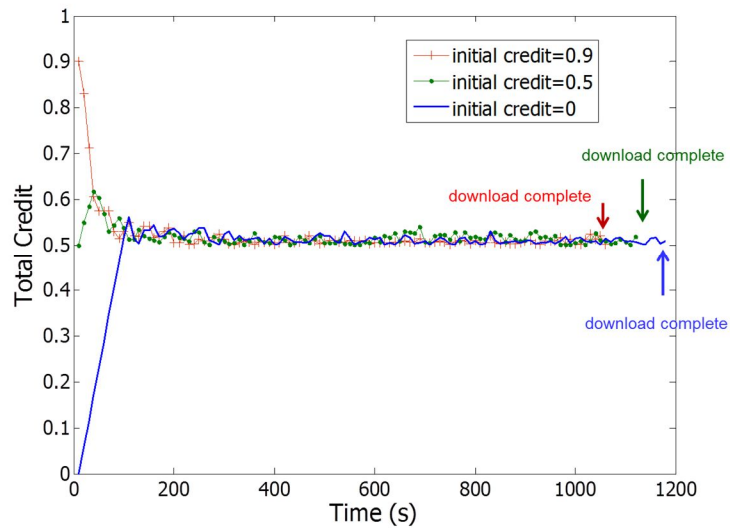
4.1 파일 다운로드 서비스의 성능 평가

이 절에서는 초기 신용도와 목표 신용도의 변화에 따른 신용도 메커니즘의 성능을 살펴본다. 실험을 위해 사용된 파일 크기는 100 Mbytes 이고 청크 크기는 256 Kbytes 로 설정하였다.





(a)



(b)

그림 9. 목표 신용도와 초기 신용도에 따른 총 신용도 변화 : (a) 초기
신용도=0.5 (b) 목표 신용도=0.5



그림 9 (a)는 모든 피어의 초기 신용도를 0.5로 설정하고 목표 신용도를 0 / 0.5 / 0.9 로 균일하게 설정하였을 때, 총 신용도의 변화와 파일 다운로드 완료 시간을 보여준다. 시간이 지남에 따라 피어가 가지는 총 신용도는 목표 신용도에 수렴하는 경향을 보인다. 또한, 목표 신용도의 값에 따라 피어의 다운로드 완료 시간이 조금씩 차이가 있는 것을 확인 할 수 있다. 이것은 목표 신용도가 높은 피어는 자신의 현재 신용도를 많이 사용하여 좋은 서비스를 제공받는 것보다 미래에 제공받을 서비스를 위해 신용도를 높이려는 성향을 가지기 때문이다. 반대로 목표 신용도가 낮은 피어는 신용도를 많이 사용함으로써 현재의 서비스를 최선의 상태로 제공 받는 성향이 더 크다고 할 수 있다. 따라서 목표 신용도가 높은 피어는 목표 신용도가 낮은 피어보다 다운로드 완료 시간이 더 늦어지게 된다. 그림 9 (b)는 모든 피어의 목표 신용도를 0.5로 설정하고 초기 신용도를 0 / 0.5 / 0.9 로 균일하게 설정하였을 때, 총 신용도의 변화와 파일 다운로드 완료 시간을 보여준다. 초기 신용도는 피어가 처음 세션에 참여하거나 또는 탈퇴 후 재 참여할 때 가지고 있는 신용도이다. 따라서 목표 신용도가 같더라도 초기 신용도가 높은 피어가 다운로드 완료 시간이 더 빠른 것을 볼 수 있다. 표 2에서 각각의 경우에 따른 평균 다운로드 완료 시간과 평균 다운로드 레이트를 정리하였다.



표 2. 평균 다운로드 완료 시간(seconds)과 평균 다운로드 레이트 (kbps)

	(a) Initial credit 0.5			(b) Target credit 0.5		
	Target credit 0.9	Target credit 0.5	Target credit 0	Initial credit 0.9	Initial credit 0.5	Initial credit 0
Avg. download completion time (sec)	1195.3	1146.2	1087.6	1063.4	1142.7	1187.0
Avg. download rate (kbps)	685.4	714.7	753.2	770.4	716.7	690.1

4.2 비디오 스트리밍 서비스의 성능 평가

이 절에서는 식 (7)의 λ 값 조절에 따라 비디오 퀄리티와 총 신용도가 어떻게 변화하는지 실험하였다. [34]에 따르면 현재 대부분의 VOD 시스템에서 사용되고 있는 소스 비디오 스트리밍 레이트는 381 kbps 에서 450 kbps이다. 본 실험에서는 소스 비디오 스트리밍 레이트를 400 kbps로 정하고 JSVM 소프트웨어를 이용하여 SNR 확장성을 갖는 1 개의 기본 레이어와 9 개의 상위 레이어들로 계층적 부호화하였다. 비디오 테이터는 352 X 288의 FOREMAN을 사용하였다. 구체적 파라미터들은 프레임 레이트 15, GOP 크기 16, QP 값은 30-40으로 설정하였다. 초기 신용도는 0.2 / 0.5 / 0.8 일 때로 나누어 실험하였고 목표 신용도는 0 으로 설정하였다.



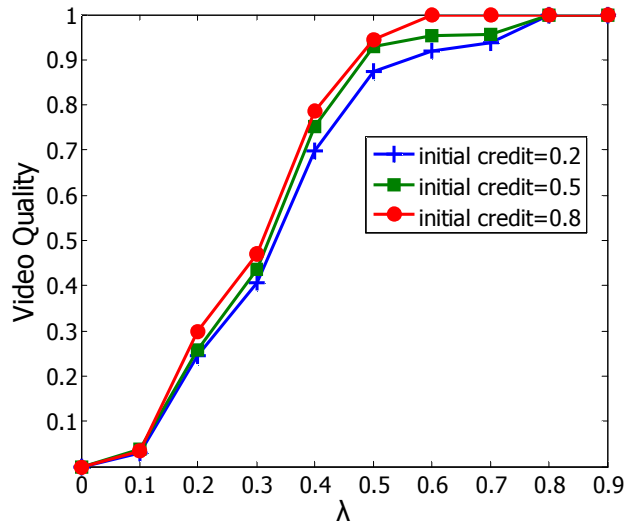
표 3. SVC 비디오 데이터

Layer	PSNR	Rate (Kbps)	Video Quality
1	32.3	70	0
2	33.8	170	0.24
3	35	240	0.44
4	36	290	0.60
5	36.3	310	0.65
6	37	330	0.76
7	37.5	360	0.84
8	38	370	0.92
9	38.3	390	0.97
10	38.5	400	1.00

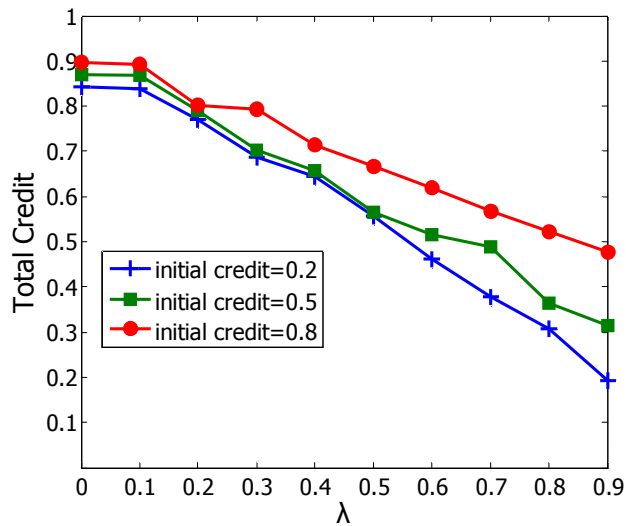
그림 10의 (a)와 (b)는 λ 값 조절에 따른 비디오 퀄리티와 총 신용도의 변화를 나타낸다. λ 가 0에서 0.9로 증가함에 따라 비디오 퀄리티가 증가하는 반면 총 신용도가 감소하는 것을 볼 수 있다. 이것은 λ 가 클수록 비디오 퀄리티를 더 중점적으로 고려하여 현재 가지고 있는 신용도를 많이 소모하고, λ 가 작을수록 현재 제공받는 비디오 퀄리티 보다는 신용도에 더 가중치를 두어 총 신용도를 쌓는 것에 중점을 두기 때문이다. 또한 같은 λ 값이라도 초기 신용도가 클수록 비디오 퀄리티가



더 빨리 증가하는 것을 볼 수 있는데 더 많은 신용도를 사용하여 더 좋은 퀄리티의 비디오 서비스를 제공받기 때문이다.



(a)



(b)



그림 10. λ 값 조절에 따른 비디오 퀄리티와 총 신용도 변화 : (a)

비디오 퀄리티 변화 (b) 총 신용도 변화

4.3 기존 알고리즘과의 성능 비교

이 절에서는 제안하는 알고리즘과 기존 알고리즘과의 성능을 비교한다. 비교하는 기존 알고리즘은 비트 토렌트에서 사용하는 Tit-for-Tat과 Maze[35]에서 사용하는 Point 기반 방식이다. 비교 알고리즘의 선택은 현재까지도 실제 어플리케이션에서 널리 상용화 되어 있고 그 특징과 장단점이 명확하게 나타나는가를 우선시하여 고려하였다. 각 알고리즘들의 특징은 다음에 요약되어 있다.

- Tit-for-Tat

각 피어는 자신에게 제공하는 업로드 레이트가 가장 높은 순서대로 n 명(일반적인 경우 4명)의 이웃 피어를 선택하여 업로드하며 이를 *unchoking* 이라 한다. 그리고 *unchoking* 되지 못한 나머지 이웃 피어들을 자신에게 제공하는 서비스가 충분하지 못하다고 판단하여 *choking* 하게 되는데 이처럼 너무 잦은 *choking*이 일어나면 세션을 다시 연결하는 데 걸리는 시간 때문에 오히려 업로드 / 다운로드 레이트를 충분히 활용하지 못하는 문제가 발생할 수 있다. 또한 *optimistic unchoking* 정책을 사용하여 주기적으로 자신이 받은 서비스와는 상관없이 업로드를 요청한 피어 중 랜덤으로 하나를 선택하여 체크를

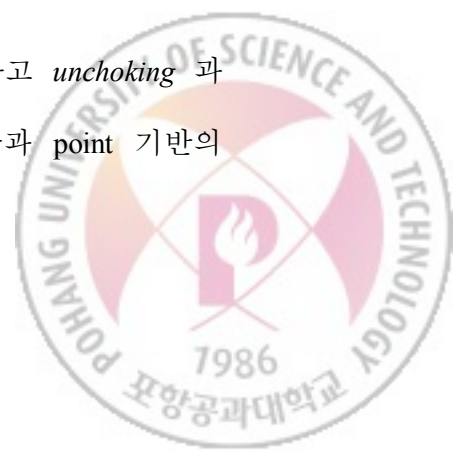


제공하기도 한다. 이는 세션에 처음 참여한 피어가 협업을 활성화할 수 있도록 도와주고 피어 선택 시 로컬 미니멈 (local minimum)에 빠지지 않도록 방지하는 장점이 있다. 하지만 특히 배포자의 경우 네트워크에 대한 자원 기여도를 고려하지 않고 다운로드 레이트가 높은 피어만을 우선시하여 *unchoking* 하기 때문에 무임 승차자들이 고의로 *optimistic unchoking* 할 수 있는 기회를 많이 만들어서 악의적으로 네트워크의 자원을 소비하는 문제점이 발생할 수 있다.

- Point-based

Point 기반의 인센티브 정책을 사용한 대표적인 상용화 시스템으로 Maze가 있다. 처음 세션에 참여한 피어는 시스템으로부터 일정량의 초기 point를 지급받고 이 후 자신이 업로드한 양 만큼을 point로 보상받고 자신이 다운로드 한 양에 대해서는 point로 지불하게 된다. 따라서 피어의 네트워크에 대한 기여도는 현재 가지고 있는 point로 나타나게 된다. Point 기반 방식은 자신이 가진 point의 범위 내에서 다운로드를 받기 때문에 비트 토렌트에 비해 상대적으로 무임 승차 문제를 방지할 수 있는 장점을 가진다. 그러나 자신이 현재 가진 point가 부족하다면 다운로드 서비스를 받을 수 없기 때문에 다운로드 레이트를 충분히 활용하지 못하는 단점이 있다.

본 실험에서는 희귀 청크 우선 선택 기법을 제외하고 *unchoking* 과 *optimistic unchoking* 정책을 사용한 비트 토렌트 모듈과 point 기반의



인센티브 모듈을 직접 구현하였다. 그리고 각 피어 마다 피어 수를 비슷하게 분포시킨 3 개의 그룹을 갖게 하였다. 각 그룹은 기존의 알고리즘이 피어 간의 소셜 관계를 고려하지 않는 특성을 그대로 유지하기 위해 친밀도를 0으로 설정하였다. 제안하는 알고리즘에서 총 신용도와 초기 신용도는 0.5로 설정하였다.

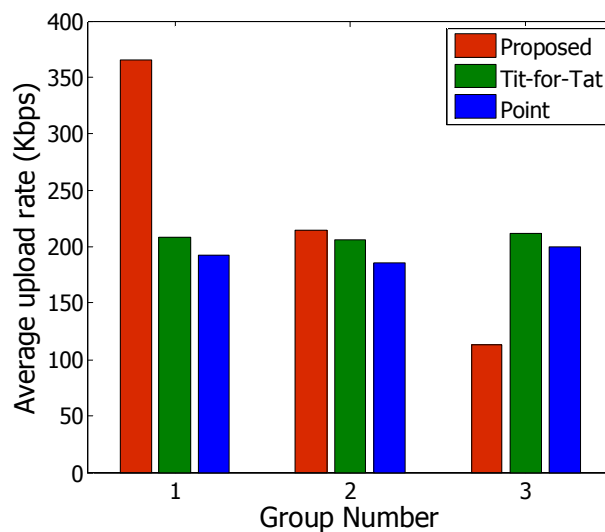


그림 11. 기존 알고리즘과의 각 그룹에 대한 평균 업로드 레이트 비교

그림 11에서는 각 그룹에 대한 평균 업로드 레이트를 비교하였다. 제안하는 알고리즘은 친밀도가 높은 그룹에 업로드 레이트를 많이 제공하고 친밀도가 낮은 그룹에는 상대적으로 낮은 업로드 레이트를 제공하게 된다. 그러나 다른 알고리즘은 소셜 관계를 전혀 고려하지 않은 자원 분배가 이루어지기 때문에 모든 그룹에 비슷한 업로드



레이트를 제공하게 된다.

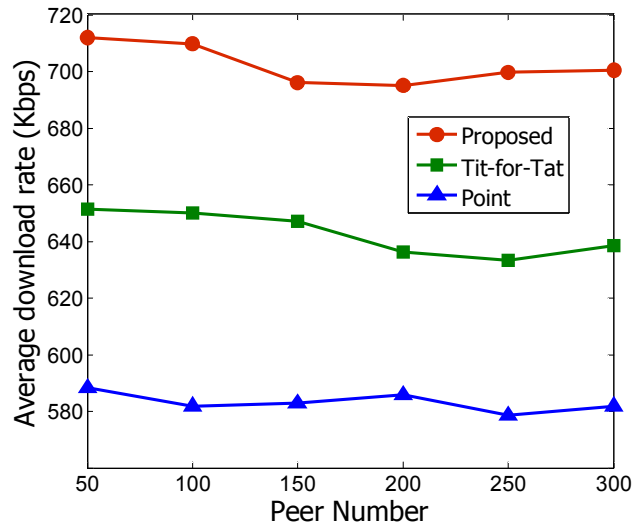
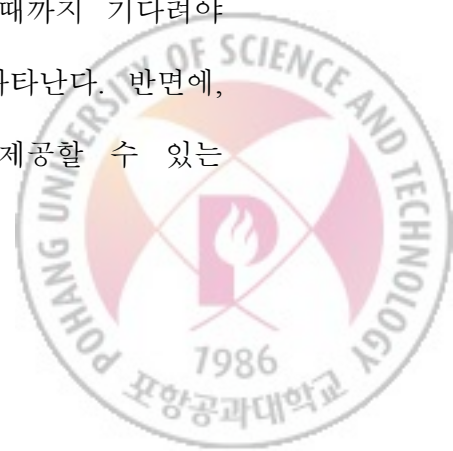


그림 12. 기존 알고리즘과의 평균 다운로드 레이트 비교

그림 12에서는 피어의 수를 점점 늘려가면서 각 알고리즘의 평균 다운로드 레이트를 비교하였다. 제안하는 알고리즘이 다른 알고리즘에 비해 더 좋은 성능을 보이는 것을 알 수 있다. 이것은 비트토렌트가 잦은 chocking으로 인해 피어의 다운로드 레이트를 충분히 활용하지 못하기 때문이다. 또한, point 기반 시스템에서는 일단 point를 모두 사용하면 업로드를 통해 다시 충분한 point를 쌓을 때까지 기다려야 하기 때문에 평균적으로 다운로드 레이트가 낮게 나타난다. 반면에, 제안하는 신용도 기반 인센티브 메커니즘은 현재 제공할 수 있는



업로드 레이트가 충분하지 않은 상황이라도 친밀도가 높은 그룹으로부터 상대적으로 적은 신용도를 소모하여 다운로드 받을 수 있기 때문에 다른 알고리즘에 비해 높은 성능을 보인다.

따라서 제안하는 알고리즘은 기존의 알고리즘과 비교했을 때 더 많은 다운로드 레이트를 제공받기 때문에 더 높은 QoS를 만족할 수 있다. 그리고 친밀도에 따라 업로드 레이트를 차별적으로 제공하기 때문에 소셜 네트워크의 특성을 충분히 반영한 자원 분배 정책을 사용한다고 할 수 있다.



V. 결론

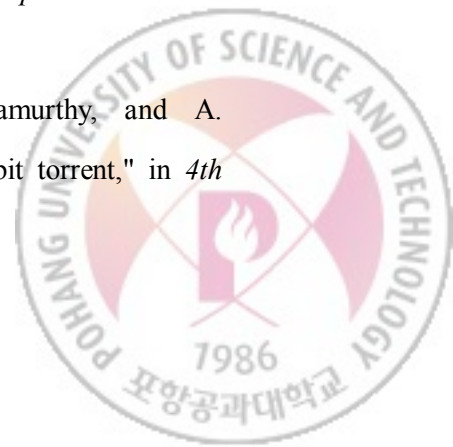
본 논문에서는 기존의 P2P 인센티브 정책의 단점을 개선하기 위하여 소셜 관계 기반의 P2P 자원 제어 알고리즘을 제안하였다. 제안하는 알고리즘은 친밀도를 고려한 신용도 메커니즘을 통해서 소셜 네트워크 상에서의 형평성을 보장 할 뿐 아니라 사용자가 요구하는 비디오 품질을 보장 할 수 있는 자원 할당 정책을 사용한다. 실험을 통해 제안하는 알고리즘이 기존의 알고리즘에 비해 다양한 측면에서 향상된 성능을 보이는 것을 확인할 수 있다.

최근 무선 네트워크 기술의 발달과 함께 전세계적으로 스마트폰의 개발과 확산이 급속도로 증가하고 있다. 더욱이 LTE (Long Term Evolution) 기술이 상용화되면서 앞으로도 사용자들의 관심과 이에 발맞추어 무선 사업의 발전 가능성 또한 무궁무진 할 것이라 예상된다. 향후에는 제안하는 시스템을 모바일 기기로 확장하여 소셜 네트워크 기반의 다양한 서비스를 실제 무선 환경에서 제공할 수 있도록 연구할 계획이다.



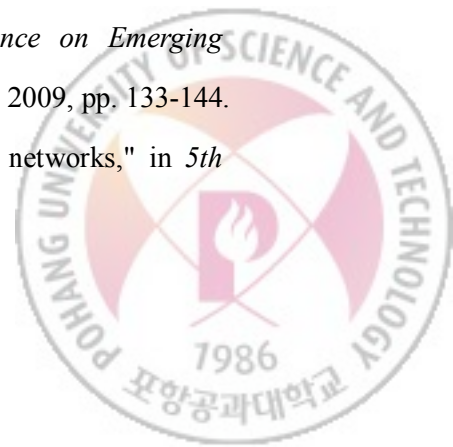
참고문헌

- [1] Y. Liu, Y. Guo, and C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-peer Networking and Applications*, vol. 1, no.1, pp. 18-28, 2008.
- [2] M. R. Rahman, "A survey of incentive mechanisms in peer-to-peer systems," Technical Report, cs-2009-22.
- [3] M. Karakaya, I. Korpeoglu, and O. Ulusoy, "Free Riding in Peer-to-Peer Networks," *IEEE Internet Computing*, vol. 13, no.2, pp. 92-98, 2009.
- [4] B. Cohen, "Incentives build robustness in BitTorrent," in *1st Workshop on Economics of Peer-to-Peer Systems*, 2003, pp. 68-72.
- [5] *Mininova's Torrent Downloads Double to 7 Billion in a Year*. Available: <http://torrentfreak.com/mininovas-torrent-downloads-doubled-in-a-year-090105>
- [6] S. Jun and M. Ahamad, "Incentives in BitTorrent induce free riding," in *2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, Philadelphia, Pennsylvania, USA, 2005, pp. 116-121.
- [7] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free riding in bittorrent is cheap," in *2006 ACM Workshop on Hot Topics in Networks (HotNets'06)*, 2006.
- [8] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bit torrent," in *4th*



USENIX conference on Networked systems design & implementation (NSDI'07), Cambridge, MA, 2007.

- [9] M. Sirivianos, J. H. Park, R. Chen, and X. Yang, "Free-riding in BitTorrent Networks with the Large View Exploit," in *6th International workshop on Peer-To-Peer Systems (IPTPS07)*, 2007.
- [10] E. Adar and B. A. Huberman, "Free riding on Gnutella," *First Monday*, vol. 5, no.10, 2000.
- [11] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Multimedia Computing and Networking 2002*, San Jose, CA, 2002, pp. 156-170.
- [12] T. Bhuiyan, "A Survey on the Relationship between Trust and Interest Similarity in Online Social Networks," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no.4, pp. 291-299, 2010.
- [13] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in BitTorrent systems," in *the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, San Diego, California, USA, 2007, pp. 301-312.
- [14] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no.4, pp. 367-378, 2004.
- [15] A. Sherman, J. Nieh, and C. Stein, "FairTorrent: bringing fairness to peer-to-peer systems," in *the 5th international conference on Emerging networking experiments and technologies*, Rome, Italy, 2009, pp. 133-144.
- [16] J. Li and F. Dabek, "F2F: Reliable storage in open networks," in *5th*

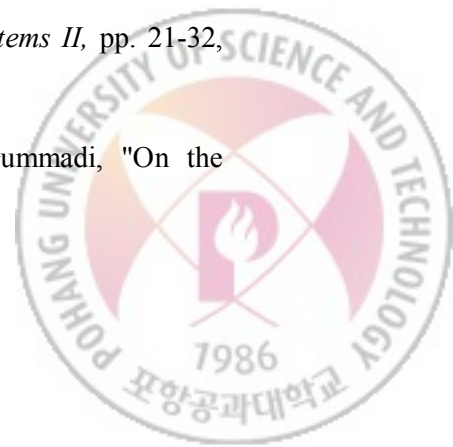


International Workshop on Peer-to-Peer Systems (IPTPS' 06), 2006.

- [17] S. Marti, P. Ganesan, and H. Garcia-Molina, "SPROUT: P2P Routing with Social Networks," in *Current Trends in Database Technology - EDBT 2004 Workshops*, 2005, pp. 425-435.
- [18] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. Van Steen, and H. J. Sips, "TRIBLER: a social-based peer-to-peer system," *Concurrency and Computation: Practice and Experience*, vol. 20, no.2, pp. 127-138, 2008.
- [19] S. J. H. Yang, J. Zhang, L. Lin, and J. J. P. Tsai, "Improving peer-to-peer search performance through intelligent social search," *Expert Systems with Applications*, vol. 36, no.7, pp. 10312-10324, 2009.
- [20] W. Wenyu, Z. Li, and Y. Ruixi, "Improving cooperation in peer-to-peer systems using social networks," in *20th International Parallel and Distributed Processing Symposium (IPDPS' 06)*, 2006.
- [21] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no.9, pp. 1103-1120, 2007.
- [22] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing BitTorrent for Supporting Streaming Applications," in *Proceedings of 25th IEEE International Conference on Computer Communications*, 2006, pp. 1-6.
- [23] P. Shah and J. F. Paris, "Peer-to-Peer Multimedia Streaming Using BitTorrent," in *IEEE International Performance, Computing, and*



- Communications Conference*, 2007, pp. 340-347.
- [24] *TouchGraph*. Available: <http://www.touchgraph.com/>
 - [25] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no.1, pp. 169-180, 2008.
 - [26] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Dover Pubns, 1998.
 - [27] S. McCanne and S. Floyd. (1995). *NS network simulator*. Available: <http://www.isi.edu/nsnam/ns>
 - [28] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of peer-to-peer systems," in *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, Monterey, California, 2002, pp. 233-242.
 - [29] T. Li, M. Chen, D.-M. Chiu, and M. Chen, "Queuing models for peer-to-peer systems," in *Proceedings of the 8th international conference on Peer-to-peer systems*, Boston, MA, 2009.
 - [30] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the role of helpers in peer-to-peer file download systems: Design, analysis and simulation," in *International Workshop on Peer-to-Peer Systems (IPTPS'07)* 2007.
 - [31] R. Mahajan, M. Castro, and A. Rowstron, "Controlling the cost of reliability in peer-to-peer overlays," *Peer-to-Peer Systems II*, pp. 21-32, 2003.
 - [32] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the



- evolution of user interaction in Facebook," in *Proceedings of the 2nd ACM workshop on Online social networks*, Barcelona, Spain, 2009, pp. 37-42.
- [33] S. Milgram, "The small world problem," *Psychology today*, vol. 2, no.1, pp. 60-67, 1967.
- [34] Y. Huang, T. Z. J. Fu, D.-M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, Seattle, WA, USA, 2008, pp. 375-388.
- [35] L. Qiao, Z. Zheng, Y. Mao, B. Y. Zhao, D. Yafei, and L. Xiaoming, "An Empirical Study of Collusion Behavior in the Maze P2P File-Sharing System," in *27th International Conference on Distributed Computing Systems (ICDCS '07)*, 2007.



Acknowledgements



Curriculum Vitae

Name Donghyeok Ho

Education

2003 – 2010 Department of Media,
Soongsil university (B.S.)

2010 – 2012 Department of Computer science and Engineering,
POSTECH (M.S.)

Publications

- 호동혁, 송황준, "소셜 네트워크 기반의 P2P 자원 할당 기법"
2011 한국통신학회 하계학술대회 (KICS 2011)



