

특집논문 (Special Paper)

방송공학회논문지 제23권 제1호, 2018년 1월 (JBE Vol. 23, No. 1, January 2018)

<https://doi.org/10.5909/JBE.2018.23.1.26>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

미디어 서버와 위치-인지 오버레이 네트워크를 연계한 효율적 콘텐츠 공유 및 서비스 방법

정 원 호^{a)†}, 이 승 연^{a)}

A Scheme of Efficient Contents Service and Sharing By Associating Media Server with Location-Aware Overlay Network

Won-Ho Chung^{a)†} and Seung Yeon Lee^{a)}

요 약

최근의 오버레이 네트워크 기술은 다양한 유형의 콘텐츠 분산 공유를 가능하게 하고 있다. 비록 오버레이 네트워크가 분산 공유의 거대한 콘텐츠 저장소로서 훌륭한 장점을 가지고 있지만, 스트리밍과 같은 인터넷 서비스는 현실적으로 용이하지 않다. 반면에 콘텐츠 서비스에 특화되어 있는 미디어 서버는 훌륭한 서비스 능력은 보유하고 있지만, 끊임없이 생성되며 지속적으로 서버와 스토리지의 성능 개선과 증설을 요구하는 콘텐츠들에 대해서 효율적 관리를 크게 필요로 하고 있다. 그러므로 오버레이 네트워크는 미디어 서버를 위한 대규모 콘텐츠 관리를 지원하고, 미디어 서버는 오버레이 네트워크를 위해 고성능 콘텐츠 서비스를 제공한다면, 두 시스템의 효율적 연관은 훌륭한 시너지 효과를 낼 수 있을 것이다. 본 논문에서는 위치-인지 개념을 기반으로 한 오버레이 네트워크의 구성 방법과 미디어 서버와의 연계를 위한 서비스 정책 및 캐시 기반의 콘텐츠 관리 기법, 그리고 그들을 기반으로 하는 효율적인 콘텐츠 서비스 방법이 제안된다. 그리고 스트리밍 서비스에 대해 제안된 시스템의 성능 분석이 이루어진다.

ABSTRACT

The recent development of overlay network technology enables distributed sharing of various types of contents. Although overlay network has great advantages as a huge content repository, it is practically difficult to directly provide such Internet service as streaming of contents. On the other hand, the media server, which is specialized in content services, has excellent service capabilities, but it suffers from the huge contents that are constantly created and requires large expansion of servers and storages, and thus requires much effort for efficient management of the huge repository. Hence, the association of an overlay network of huge storage with a media server of high performance content service will show a great synergy effect. In this paper, a location-aware scheme of constructing overlay networks and associating it with media server is proposed, and then cache-based contents management and service policy are proposed for efficient content service. The performance is analysed for one of the content services, streaming service.

Keywords : Location-Awareness, Media server, Ordinary-Peer, Overlay Network, Super-Peer

Copyright © 2017 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

I. 서론

고속 전송 장비와 네트워킹 기술의 커다란 발전으로 과거에는 힘들었던 영상 혹은 음악 등 대용량 콘텐츠 서비스가 지금은 클라우드와 같은 미디어 서버 기반으로 어렵지 않게 제공되고 있으며, 이를 기반으로 하는 스트리밍 서비스, 전송 네트워크 등 그에 관한 연구가 활발히 이루어지고 있다^[1,2]. 더불어 개인 콘텐츠 제작이 폭증하면서 이들을 위한 미디어 서비스 및 공유가 더욱 활성화되고 있다^[3-5]. 끊임없이 생성되는 콘텐츠의 저장과 관리를 위해서는 서버의 지속적 성능 개선과 스토리지 증설이 필요하다. 그리하여 최근에는 서버에 대한 의존성을 감소시키면서 사용자 근접한 영역에서 서비스를 제공하기 위한 포그 컴퓨팅에 관한 연구가 활발히 진행되고 있다^[6,7]. 이 문제가 해결되지 않으면 인터넷 서비스의 실시간성은 점점 약해질 수밖에 없다. 콘텐츠 스트리밍 서비스를 제공하는 미디어 서버와 같은 경우에도 시간이 흐를수록 동일한 현상을 나타낼 것이다.

오버레이 네트워크는 다양한 유형의 자원을 공유할 수 있는 유망한 방법 중 하나로 스토리지 미디어와 콘텐츠 같은 대용량 자원의 분산 공유라는 특성으로 최근 크게 발전해 왔다. 그러나 그것이 콘텐츠의 분산 공유라는 좋은 특성을 가지고 있지만, 수많은 사용자로부터 요청되는 스트리밍과 같은 서비스를 직접 감당하기에는 현실적인 문제를 가지고 있다. 반면에 미디어 서버와 같은 스트리밍 전문 시스템은 훌륭한 서비스 능력을 보유하고는 있지만, 끊임없이 생성되면서 지속적인 미디어 성능 개선과 스토리지 증설을 요구하는 수많은 콘텐츠들에 대한 효율적 관리를 크게 필요로 하고 있다. 그러므로 오버레이 네트워크는 미디어 서버를 위해 분산 공유 방식의 대형 콘텐츠 저장소를 제공하고, 미디어 서버는 그에게 스트리밍 서비스를 제공한다면 두 시스템의 결합은 훌륭한 시너지 효과를 낼 수

있을 것이다.

본 논문은 콘텐츠 저장소 관리와 서비스를 분리하여, 콘텐츠 저장소를 담당하는 오버레이 네트워크를 미디어 서버를 위한 포그 계층으로 연계하여 효율적인 콘텐츠 서비스 시스템을 구성하는 방법을 제안한다. 각 개인이 생성하는 다양한 유형의 콘텐츠를 오버레이 네트워크를 통해 서로 분산 공유하면서 이를 미디어 서버와 연계하여 콘텐츠 서비스를 효율적으로 수행할 수 있는 구조를 가지고 있다. 특히 콘텐츠 전송 시간이 성능에 중요한 요소가 되므로 이를 최소화하기 위해 새로운 위치 정보를 사용하는 위치-인지 기반의 오버레이 네트워크의 구성 방법과 서비스 정책 및 캐시를 사용하는 콘텐츠 관리 기법, 그리고 그들을 기반으로 하는 효율적인 콘텐츠 서비스 방법이 제안된다. 생성되는 모든 콘텐츠를 서버로 업로드하지 않고 오버레이에 그대로 두고, 일반피어의 해당 콘텐츠에 대한 서비스 요구가 있을 시점에 서버로 업로딩 하여 서비스하는 JiTUL(Just-in-Time UpLoad) 탑재 정책을 사용한다. 그리고 탑재된 콘텐츠라도 항상 서버에 존재하는 것은 아니며, MLRU(Modified Least Recently Used) 정책 기반으로 서비스 없이 오랜 시간 공간만 차지하는 콘텐츠는 서버에서 다시 제거된다. 그렇게 함으로써 서버에 존재하는 콘텐츠의 수를 설정된 범위 수준으로 유지하도록 하여 스토리지 관리의 효율성을 높인다.

II. 관련연구 동향 및 제안 시스템의 특징

미디어 서버에 관한 기존 연구는 인터넷 상에서의 방송 시스템 구축을 위한 효율적인 하드웨어 및 소프트웨어 설비에 관한 연구로 다중 채널 방송을 위한 시스템 구축에 관한 것이 주를 이루어 왔다. 서비스 품질 개선을 위해서는 서버 성능 및 네트워크 대역폭과 같은 고비용 자원의 지속적인 확보가 필수적이며, 적은 예산 범위 내에서 운영되는 개인 혹은 소규모 조직에게는 큰 부담이 아닐 수 없다. 그러나 지금은 운영 주체가 모든 하드웨어 자원을 지원하고 있으며, 사용자를 위한 소프트웨어도 API 등으로 대부분 공개되어 있어, 각자 콘텐츠를 제작하여 서버에 탑재시키면 완료되는 환경으로 바뀌어 왔다^[4,5,8]. 이러한 통합 시스템이

a) 덕성여자대학교 디지털미디어학과(Dept. of Digital Media, Duksung Women's University)

‡ Corresponding Author : 정원호(Won-Ho Chung)

E-mail: whchung@duksung.ac.kr

Tel: +82-2-901-8343

ORCID:https://orcid.org/0000-0001-7157-9935

* 본 논문은 2017년도 덕성여대 연구비 지원에 의해 이루어졌음.

· Manuscript received November 10, 2017; Revised November 23, 2017; Accepted November 23, 2017.

라 할지라도 성능 개선을 위해 하드웨어의 지속적 증설 및 소프트웨어 개선을 필요로 하며, 또한 이를 위한 데이터센터의 증설은 필수적인 것이다. IoT 서비스와 방송단말 응용 사이에 포그 애플릿을 두어 임베디드 응용의 서비스 접근요구를 감소시키고 클라우드 기반 IoT 서비스와의 유연한 구성을 위한 하이브리드 방송용 애플릿 구조도 제안되었다⁷⁾.

오버레이 기술은 컴퓨터 자원을 효율적으로 분산 공유할 수 있는 유망한 방법 중 하나이나, 콘텐츠 서비스와는 서로 독립적으로 연구가 이루어지고 있다. 오버레이 기술은 자원의 분산 공유를 중심으로 발전해 왔으며 스트리밍과 같은 서비스 지원은 현실적 문제점을 지니고 있어, 효율적 콘텐츠 서비스를 위한 연구가 필요한 실정이다. 최근의 오버레이 네트워크는 2계층 오버레이로 구성되고 있는데, 하나는 일반피어(Ordinary-Peer, OP)들로 구성되는 계층이며, 다른 하나는 슈퍼피어(Super-Peer, SP)들로 구성되는 계층인데 이를 슈퍼피어 오버레이라고 한다. 이러한 슈퍼피어 오버레이 패러다임은 이제 오버레이 네트워크를 구축하는 일반적인 접근 방식으로 간주되고 있으나 그 구성 세부 사항에 대한 표준이 존재하지 않으며 응용 의존적인 경우가 보통이다. 그래서 효율적인 슈퍼피어 오버레이 설계를 위한 성능의 trade-off와 현실적 가이드라인이 제안되었으며⁸⁾, Gossip 프로토콜과 같은 정보 교환 방식을 이용한 슈퍼피어 선정 및 슈퍼피어 오버레이 구성에 관한 많은 연구가 이루어졌다¹⁰⁻¹²⁾. 두 피어 간의 물리적 거리는 콘텐츠 유사성¹³⁾ 등과 같은 다른 척도와는 달리 성능 향상을 위한 공통적 상시적 특성이라고 할 수 있다. 피어간의 거리를 추정하는 여러 방법이 많이 제안되었으나¹⁴⁾, 주로 복잡한 연산 혹은 시뮬레이션 결과를 기반으로 하고 있어, 현실적인 어려움이 존재한다. 이후, 슈퍼피어는 SP, 일반피어는 OP 등으로 가능한 앞에서 정의된 모든 용어는 영어 약어를 사용하기로 하지만, 장이나 절의 제목에는 사용하지 않는 것으로 한다.

III. 위치-인지 기반의 오버레이 네트워크 구성

각 OP는 반드시 하나의 SP와 연관되어야 하며 자신에게 연관된 SP에게 필요한 서비스를 요청한다. 요청받은 SP는

OP를 대신하여 그 서비스를 대행하여 그 결과를 OP에게 전달한다. 이것이 오버레이 네트워크의 간략한 동작의 요약이다.

1. 위치-인지 기반의 근접성 정의

SP와 OP, 혹은 OP들 사이의 전송 시간은 성능과 관련된 중요한 요소 중의 하나이다. 그러므로 OP와의 효율적 연관을 위한 SP를 어떻게 선택하느냐 하는 것이 중요한 문제가 된다. 또한 이러한 SP의 선택 문제는 SP 오버레이를 구성하는데 있어서도 동일하게 발생한다. 그러므로 복잡한 연산이나 많은 정보 교환을 사용하지 않는 근접성 추정은 오버레이 네트워크를 구성하는데 매우 필요한 연구라 할 수 있다. 피어 간 전송을 위해서는 연결된 피어들의 ID 혹은 주소를 기반으로 하지만 그들만으로 근접성을 찾아내는 것은 용이하지 않은 문제이다. 본 연구에서는 위치-인지 기반으로 실제 거리 상 가장 좋은 근접성을 나타내는 SP를 선택하여 OP를 연관시키기 위해서 위치-인지를 위한 근접성을 다음과 같이 4 단계로 정의한다.

[정의-1] S 는 k 피어들의 집합이며 각 피어의 위치는 3-튜플로 표현된다고 하면, $S = \{s_i = (p_n^i, p_c^i, p_r^i) \mid i = 0, 1, 2, \dots, k-1\}$ 이다. 임의의 두 피어 $s_i = (p_n^i, p_c^i, p_r^i) \in S$ 와 $s_j = (p_n^j, p_c^j, p_r^j) \in S$ 에 대하여,

- 1) $p_n^i = p_n^j$ 를 만족하면, s_i 와 s_j 는 최소근접성 관계에 있다고 하며,
- 2) $(p_n^i = p_n^j) \& (p_c^i = p_c^j)$ 를 만족하면, s_i 와 s_j 는 중간근접성 관계에 있다고 하며,
- 3) $(p_n^i = p_n^j) \& (p_c^i = p_c^j) \& (p_r^i = p_r^j)$ 를 만족하면, s_i 와 s_j 는 최대근접성 관계에 있다고 한다.
- 4) 어떤 것도 일치하지 않는 경우 s_i 와 s_j 는 임의근접성 관계에 있다고 한다. ■

[정의-1]과 일치하는 3-튜플 정보로서 지금은 스마트폰 때문에 주목을 받고 있지 못하지만 과거 오랫동안 글로벌 표준으로 사용되어 왔던 국제전화번호 체계의 일부를 이용한다¹⁵⁾. 이 체계가 [정의-1]을 만족한다는 것은 잘 알 수

있다. 왜냐하면 이는 본질적으로 위치 정보를 내포하고 있기 때문이다. 그리하여 3개의 튜플이 모두 일치하면 두 피어가 가장 근접하다는 것을 의미하고 있으며, 2개의 튜플만 일치하는 경우 그리고 1개의 튜플만 일치하는 경우, 일치하는 튜플이 없는 경우, 이렇게 4 단계로 근접성을 정의하고 있다.

2. 슈퍼피어의 선택

[정의-1]의 근접성 단계를 기반으로 OP에 가장 근접한 SP를 선택하여 연관시키는 알고리즘을 기술한 것이 Algorithm SPSelect 이다. 오버레이에 참가하려는 OP를 p 라 하고 현재 SP 오버레이를 구성하고 있는 SP들의 집합을 S 라고 하자.

예를 들어, $S = \{(82,51,7574), (1,221,2754), (82,2,5280), (1,421,213), (1,907,432), (82,2,573), (61, 7, 453), (82, 2, 548)\}$ 라고 가정하자. 어떤 OP, $p = (82, 2, 5280)$ 가 이 오버레이 네트워크에 참여 하려고 할 때, `classify_sp()`로부터 $S_{max} = \{(82, 2, 5280)\}$, $S_{med} = \{(82, 2, 573), (82, 2, 548)\}$, $S_{min} = \{(82, 51, 7574)\}$ 그리고 $S_{na} = \{(1,221,2754), (1,421,213), (1,907,432), (61, 7, 453)\}$ 을 얻는다. 따라서 p 는 가장 근접한 SP = (82, 2, 5280)에 연관된다. 만약에 S_{max} 이 존재하지 않으면, 다음으로 근접한 S_{med} 중 하나의 SP를 연관시키며, 만약에 S_{med} 도 존재하지 않으면 S_{min} 중 하나의 SP에 연관시키며 그것도 존재하지 않으면 S_{na} 중 하나를 선택하여 연관시킨다. 그 결과를 보여주고 있는 것이 [그림 1]이다.

Algorithm SPSelect(SPSet, OP)

For $S = \{s_i = (p_n^i, p_c^i, p_r^i) \mid i = 0, 1, 2, \dots, k-1\}$ and $p = (p_n, p_c, p_r)$,

classify_sp(S, p) ;

if($S_{max} \neq \phi$) return (any $s \in S_{max}$) ;

if($S_{med} \neq \phi$) return (any $s \in S_{med}$) ;

if($S_{min} \neq \phi$) return (any $s \in S_{min}$) ;

return (any $s \in S_{na}$) ;

classify_sp(S, p) {

for(i=0; i<k ; i++) { // Find a set of SP's in *min, med and max proximity*

if ($p_n^i = p_n$) // p and s_i are in the minimum-proximity relation

if ($p_c^i = p_c$) // p and s_i are in the medium-proximity relation

if ($p_r^i = p_r$) // p and s_i are in the maximum-proximity relation

append(S_{max} , p) ; // closest proximity neighborhood relation

else append(S_{med} , p) ; // closer proximity neighborhood relation

else append(S_{min} , p) ; // close proximity neighborhood relation

else append(S_{na} , p) ; // no neighborhood relation

}

}



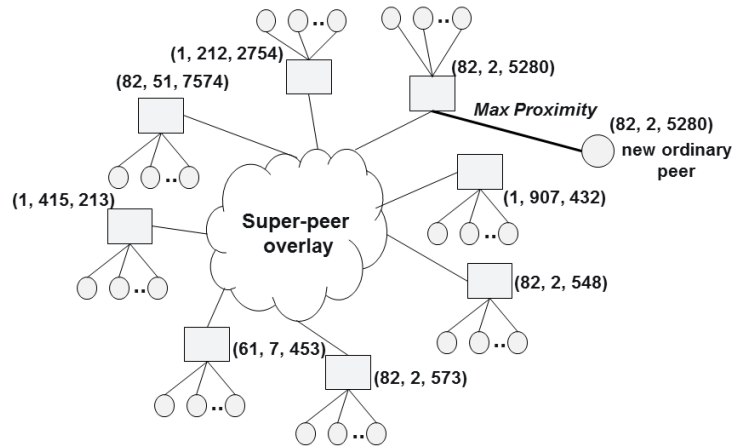


그림 1. 주어진 예에서 OP에 대한 최대근접 SP의 연관
 Fig. 1. Association an OP with the SP of maximum proximity in given example

3. 슈퍼피어 오버레이의 구성

새로 선발된 SP가 기존 SP 오버레이에 참여하려면, 현재 SP들 중에서 적절한 것들을 찾아서 그들과 연관이 이루어져야 한다. 각 SP가 자신에게 가장 근접한 SP들과 연관되면 SP 오버레이 상의 임의의 두 SP에 대해서도 최대근접성을 유지할 수 있다. 또한 SP 간의 메시지 중복을 줄이고 전송 효율을 높이기 위해서도 최대근접성을 유지하는 것이 바람직하다. [정의-1]의 근접성 단계를 기반으로 SP 오버레이를 구성하는 과정은 앞의 Algorithm SPSelect를 이용하면 매우 간단하다. 그것이 다음의 Algorithm SPOrganize이다. 여기서 S 는 현존하는 SP들의 집합이며, s 는 현재 SP 오버레이에 참여하기 위해 새롭게 선발된 SP이다.

앞에서 주어진 예에 대해 최대근접성으로 참여하는 경우 s 는 S_{max} 의 SP들과 연관되며, 중간근접성인 경우 S_{med} 의 SP들과, 최소근접성으로 참여한다면 S_{min} 의 SP들과 연

결된다. 다른 점이 있다면 OP의 연관은 하나의 SP에만 연

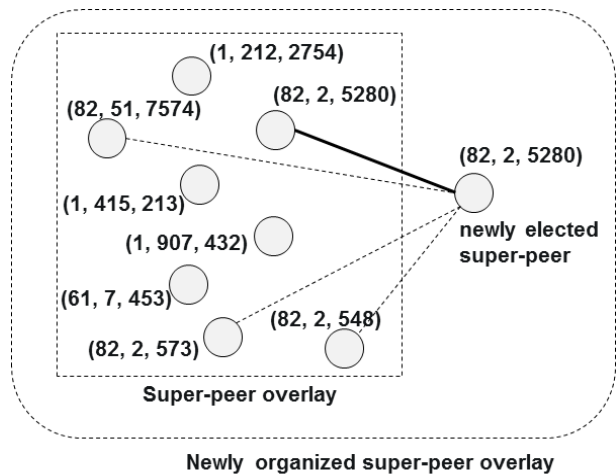


그림 2. 최대근접성(실선) 혹은 최소근접성(점선)을 가진 s 의 연관
 Fig. 2. Association s with SP's of either maximum proximity(bold line) or minimum proximity(dotted line)

Algorithm SPOrganize($SP, SPSet$)

For $S = \{s_i = (p_n^i, p_c^i, p_r^i) \mid i = 0, 1, 2, \dots, k-1\}$ and $s = (p_n, p_c, p_r)$,

classify_sp(S, p) ;

if ($(S_{max} \neq \phi) \ \& \ (\text{MinConnect} = \text{TRUE})$) return S_{max} ;

if ($(S_{med} \neq \phi) \ \& \ (\text{MedConnect} = \text{TRUE})$) return S_{med} ;

if ($(S_{min} \neq \phi) \ \& \ (\text{MaxConnect} = \text{TRUE})$) return S_{min} ;

return S_{na} ;



관되지만 SP 오버레이 구성은 해당 근접성을 가진 모든 SP들과 연관된다는 것이다. [그림 2]는 새로 진입한 SP, s 가 최대근접성으로 연관되는 경우(실선)와 최소근접성으로 연관되는 경우(점선)를 보여주고 있다.

가지 유형의 피어들로 구성되며, 이들은 모두 미디어 서버와 연결될 수 있다. [그림 3]에서 보여준 SP 오버레이는 미디어 서버에 대한 포그 계층 역할을 한다.

IV. 미디어 서버와의 연계

본 연구에서의 오버레이 네트워크는 [그림 3]에 보여준 바와 같이, OP, SP, 그리고 게이트피어(Gate-Peer, GP), 3

1. 슈퍼피어 블록과 일반피어 블록

각 SP가 관리하면서 동작의 근간이 되는 주요한 정보 블록으로 SP-블록(SPBlock)과 OP-블록(OPBlock)이 있다. OPBlock은 IP 주소, 위치-인지 정보와 각 OP의 특정 폴더에 저장된 콘텐츠 리스트 정보 등, 해당 SP에 연관된 모든 OP에 대한 각종

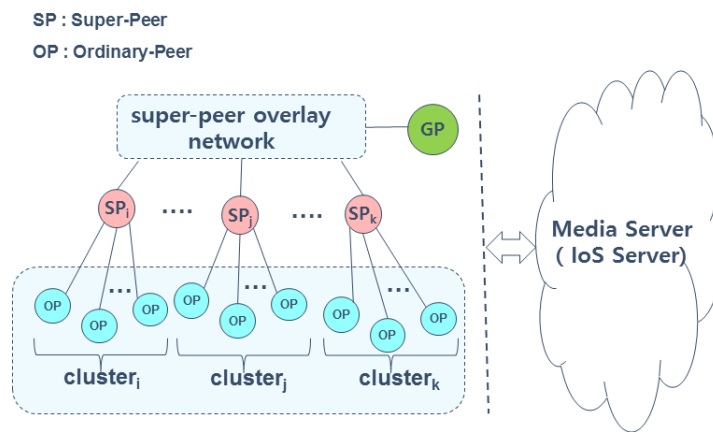


그림 3. 제안된 오버레이 네트워크 구조
 Fig. 3. Architecture of the proposed overlay network

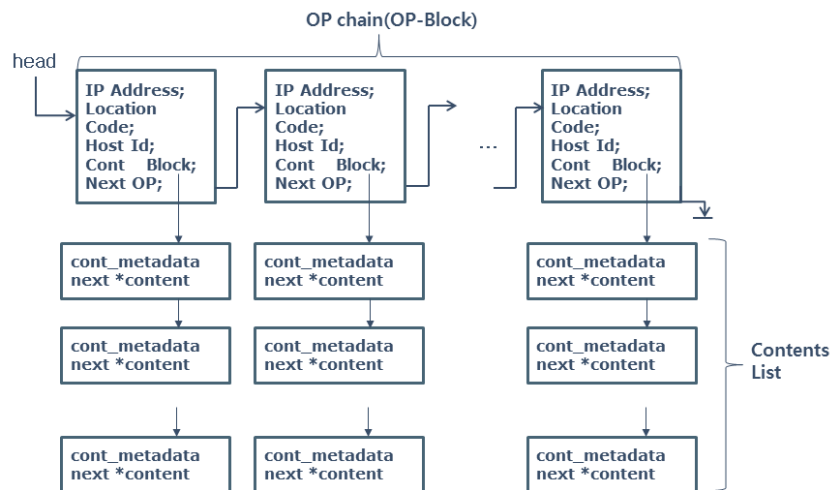


그림 4. 2차원 연결리스트를 가지는 OPB 구조
 Fig. 4. OPB structure with 2D linked-list

정보를 내포하는 블록으로, [그림 4]에 보여준 바와 같이 각 OP 레코드들이 연결리스트 구조를 이루고 있으며, 각 OP의 특정 폴더에 저장되어 있는 콘텐츠 리스트 정보들이 또한 연결리스트 구조로 추가되는 2차원 구조이다.

OP의 참여와 탈퇴가 일어날 때 해당 SP의 OPB에서 대응하는 OP 레코드의 추가 혹은 삭제 등의 갱신이 일어나며, 또한 특정 폴더에서 콘텐츠의 추가 삭제 등이 발생할 때에도 해당 OP 레코드의 콘텐츠 리스트 갱신이 일어난다. 연관될 SP의 선택은 SPSelect 알고리즘을 기반으로 이루어지며, 등록 수락 여부를 판단한 후, 등록 수락이 이루어져 자신의 IP 주소, 위치 코드, 호스트 id, 저장 중인 콘텐츠 리스트 등을 SP로 보내면 SP는 OP 레코드를 생성하여 OPB를 갱신한 후, 해당 OP에게 등록 완료를 알려준다. 이러한 과정을 거치면서 OPB는 생성된다.

SPB는 IP주소, 위치 코드 등, 현재 존재하는 각 SP에 대한 각종 정보를 내포하는 블록이며, OPB와 동일한 구조를 가지지만, 콘텐츠 정보를 위한 연결리스트가 없는 1차원 연결리스트라는 것만 다르다. 자주 발생하지는 않지만, SP들의 참여와 탈퇴 발생 시에 각 SPB는 GP를 통해 갱신이 일어나며, GP도 동일한 유형의 SPB를 관리하고 있다.

2. 콘텐츠 관리 및 서비스 정책

GP가 저장하여 관리하는 주요 정보 블록은 2가지가 있다. 하나는 SP와 마찬가지로 SPB이며, 다른 하나는 GP만이 보유하면서 관리하는 콘텐츠 캐시가 있다. 콘텐츠 캐시에는 현재 미디어 서버에 탑재되어 있는 콘텐츠들에 관한 정보가 저장된 유한 크기의 연결리스트이며, 리스트의 크기에 따라 미디어 서버에 탑재될 수 있는 콘텐츠의 수가, 예를 들면 N 으로 제한을 받는다. 요청 콘텐츠가 콘텐츠 캐시 존재하면 히트라고 하고 존재하지 않으면 폴트라고 한다. 이러한 콘텐츠 캐시 관리를 위한 주요 정보로 해당 콘텐츠의 탑재 시간(t_L), 서비스 시간(t_s) 그리고 서비스 횟수(cnt_s)가 있다. 이들은 GP가 사용하는 콘텐츠 관리 정책인 MLRU(Modified Least Recently Used) 정책을 위해 필요한 정보이다. MLRU 정책은 콘텐츠 캐시에 존재하는 콘텐츠들 중에서 서비스 횟수가 가장 작은 콘텐츠를 찾아 제거한다. 서비스 횟수가 동일한 경우에는, 그 중 가장 오랫동안

서비스되지 않았던 콘텐츠를 찾아 제거한다. 탑재시간 t_L 과 서비스 시간 t_s 는 처음에는 모두 초기 탑재시간으로 설정되고 서비스 횟수 $cnt_s=1$ 로 설정되지만, 그 후에는 해당 콘텐츠 서비스가 재요청 될 때마다 그 시점으로 t_s 는 변경되며, 서비스 횟수의 증가가 일어난다. 이들을 사용하여 콘텐츠가 관리되며, 또한 콘텐츠 캐시에 존재하는 콘텐츠들의 서비스 빈도수와 서비스 주기를 얻어낼 수 있다. 즉 $f_s = cnt_s / (t_{current} - t_L)$ 이며 $T_s = 1/f_s$ 이기 때문이다.

콘텐츠 서비스 정책으로서 MLRU 콘텐츠 관리 정책을 기반으로 하는 JiTUL 탑재 정책을 사용하고 있다. 이는 미디어 스토리지의 효율적 관리를 위해서 도입하는 정책이다. 즉 콘텐츠 서비스 요청이 있을 시에 해당 콘텐츠를 미디어 서버에 탑재하고 서비스를 제공하는 정책이다. 그러나 요청한 콘텐츠가 콘텐츠 캐시에 존재하는 경우에는 기존의 방식을 따른다. 그러므로 본 연구의 서비스 정책은, 한번 탑재하여 반영구적으로 서비스하는 기존의 서비스 정책에 미디어 스토리지의 효율적 관리를 위해 JiTUL 탑재 정책을 결합한 새로운 방식인 것이다. 콘텐츠 히트의 경우에는 해당 콘텐츠의 URL을 OP에게 전달하여 OP가 미디어 서버를 통해 서비스를 받도록 하며, 폴트가 발생하였을 경우에는 SP 오버레이를 통해 해당 콘텐츠를 보유한 OP들의 검색을 수행하여 미디어 서버에 최대 근접한 OP와 연관된 SP를 찾는다. 그리하여 그 SP는 최대근접 OP로 하여금 해당 콘텐츠를 미디어 서버로 업로드 시킨 후, 콘텐츠 URL을 통해 미디어 서버로부터 OP가 직접 서비스를 받도록 하는 것이다.

V. 성능 분석

어떤 OP가 스트리밍 서비스를 연관된 SP에게 요청하면 먼저, 그 서비스를 접수한 SP는 해당 콘텐츠가 미디어 서버에 존재하는지 여부를 판단한다. 여기서 콘텐츠 히트율을 h 라고 하면, 폴트율은 $(1-h)$ 이다. 히트인 경우에는 미디어 서버에 의해 서비스가 실행된다. 그러나 폴트가 발생한 경우, 즉 요청 콘텐츠가 미디어 서버에 존재하지 않는 경우에는 오버레이 네트워크 내에서 직접 담당하는 직접서비스와 미디어 서버와 연계하여 서비스하는 간접서비스로 나누어

성능을 비교 분석하기로 한다.

먼저 오버레이 네트워크 내에서의 직접서비스 경우에 있어서, T 를 유효 서비스시간(effective service time)이라 하고, T_c 를 미디어 서버의 서비스시간, T_o 를 오버레이 네트워크의 직접서비스시간, 그리고 h 를 콘텐츠 히트율 이라고 하면,

$$T = h \cdot T_c + (1-h) \cdot T_o \quad (1)$$

이다. 이때 $\epsilon = T/T$ 를 서비스 효율(service efficiency)이라고 하며, (1)로부터 다음과 같이 (2)가 유도된다.

$$\epsilon = T/T = 1/(h + (1-h) \cdot \alpha) \quad (2)$$

여기서 $\alpha = T_o/T_c$ 를 서비스 시간비(service time ratio)라고 한다. 그런데 요청 수가 증가할수록 $T_o \geq T_c$ 이며 (실제로는 $T_o \gg T_c$ 인 경우가 보통), $\alpha \geq 1$ 이라고 가정해도 무리가 없을 것이다. 그러므로 $\alpha = 1$ 인 경우 서비스 효율은 100%가 되어 유효 서비스 시간은 미디어 서버에서의 (혹은 오버레이에서) 서비스 시간과 동일하게 된다. 이 경우가 최적이지만 현실적이지 않으며, 앞에서 언급한 것처럼 $\alpha \gg 1$ 이라고 보는 것이 타당할 것이다. 여기서 알 수 있는 것은 α 가 증가할수록 서비스 효율을 일정 수준 유지하려면 콘텐츠 히트율을 높게 유지해야 한다. 예를 들면,

α 가 10 일 경우 서비스 효율을 0.8로 유지하기 위해서는 $h \geq 0.972$ 를 유지해야 한다. 그러므로 미디어 서비스를 오버레이 네트워크가 직접 담당하는 경우에는 서비스 효율을 높이기가 매우 힘들다는 것을 알 수 있다. 그것을 보여주고 있는 것이 [그림 5]이다.

다음은 오버레이에서 요청 서비스를 직접 담당하지 않고, JiTUL 정책에 따라 해당 콘텐츠를 미디어 서버로 업로딩하여 미디어 서버가 서비스하도록 연계하는 간접서비스의 경우이다. 여기서 T_{oc} 를 미디어 서버와 연계한 간접서비스 시간이라고 하고, T_{up} 을 OP로부터 미디어 서버로의 업로딩에 필요한 시간, T_{uo} 를 업로딩에 필요한 오버헤드라고 하면,

$$T = h \cdot T_c + (1-h) \cdot T_{oc} \quad (3)$$

이며 여기서 $T_{oc} = T_c + T_{up} + T_{uo}$ 이다.

이 경우 서비스 효율

$$\epsilon = T/T = 1/(1+(1-h) \cdot \beta) \quad (4)$$

이며, 여기서 $\beta = T_{up}/T_c$ 를 업로드 시간비(upload time ratio)라고 하며, (4)의 도출시 $T_{up} \gg T_{uo}$ 를 가정하였다. 여기서 $T_c = T_{up}$, 즉 $\beta = 1$ 인 경우가 최악의 조건이 될 것이며, $T_c \geq T_{up}$ 인 경우가 보편적이므로 $\beta \leq 1$ 이라 할 수 있

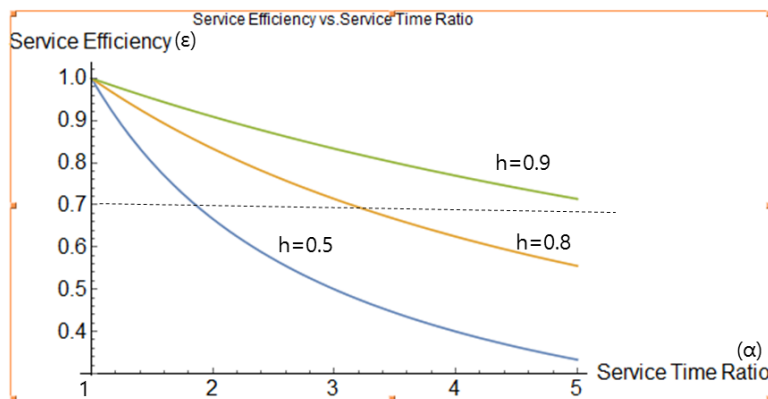


그림 5. 히트율 0.9, 0.8, 0.5에 대한 서비스 시간비에 대한 서비스 효율 그래프
 Fig. 5. Service Efficiency(ϵ) vs. Service Time Ratio(α) for $h=0.9, 0.8, 0.5$

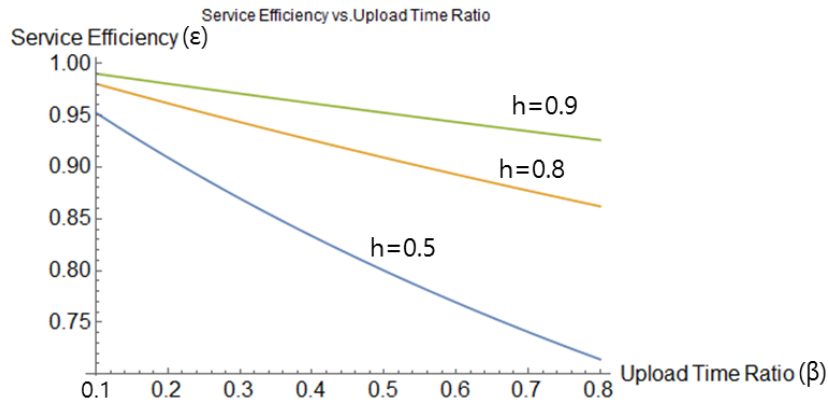


그림 6. 히트율 0.9, 0.8, 0.5에 대한 업로딩 시간비에 대한 서비스 효율 그래프
 Fig. 6. Service Efficiency(ε) vs. Upload Time Ratio(β) for h=0.9, 0.8, 0.5

다. 또한 동시 서비스의 수가 증가할수록 T_c 가 증가하는 것이 일반적이지만 업로딩도 동일한 현상을 나타낼 것이다. 그러므로 β 는 동시 업로딩의 수 그리고 서비스의 수에 따라 birth-death 프로세스 모델을 나타내므로 M/M/1 성능모델을 나타내는 것으로 알려져 있다^[6]. 오버레이가 서비스를 담당하는 경우의 서비스 시간 비율 α 에 비해 β 는 적은 값을 가지며, 그 변화 폭도 적은 것이 일반적이다. 즉 $\alpha \geq \beta$ 인 경우가 일반적이다. 3가지 히트율에 대해서, 업로드 시간 비율에 대한 서비스 효율 관계를 보여주고 있는 것이 [그림 6]이다.

예를 들어 $h=0.8$ 인 경우, $\epsilon \geq 0.990$ 을 유지하려면 $\beta \leq 0.051$ 을 유지해야 한다. 만약에 $\epsilon \geq 0.900$ 이라면 $\beta \leq 0.505$ 를 유지해야 한다는 것을 알 수 있다. 그러므로 업로드 시간을 줄이면 서비스 효율은 증가한다는 사실을 보여주고 있으며, 오버레이에서 직접 서비스 하는 경우보다는 매우 효율적이라 할 수 있다. 이러한 분석을 바탕으로 하면 β 를 더욱 줄이기 위해서는 업로드 시간을 줄여야 한다는 결론에 이르고, 그러므로 업로드 시에는 미디어 서버로부터 가장 근접한 OP로부터 업로드를 하도록 하는 것이 서비스 효율을 높일 수 있다는 사실을 알 수 있다. 그러므로 미디어 서버로부터 가장 근접한 OP를 선택할 필요가 있다는 결론에 도달한다. 그리고 OP가 요청하는 콘텐츠 서비스는 위치-인지 기반 오버레이 네트워크가 직접 담당하는 것보다, 미디어 서버와 연계하여 서비스 하는 것이 효율적이며 바람직하다는 것을 알 수 있다.

VI. 결론 및 향후 연구

오버레이 네트워크는 미디어 서버를 위한 분산 공유 방식의 대용량 스토리지 역할을 하고, 미디어 서버는 오버레이 네트워크를 위해 고성능의 콘텐츠 서비스를 제공한다면, 두 시스템의 효율적 연관은 훌륭한 시너지 효과를 낼 수 있을 것이다. 본 논문에서는 위치-인지 개념을 기반으로 한 오버레이 네트워크의 구성 방법이 제안되고, 그것과 미디어 서버를 연계한 효율적인 콘텐츠 서비스 방법과 그를 위한 콘텐츠 관리 및 서비스 정책이 제안되었다. 그리고 스트리밍 서비스를 기준으로 제안된 시스템의 성능이 직접서비스 혹은 미디어서버만 사용하는 경우에 비해 효율적이라는 분석이 이루어졌다. 좀 더 개선된 오버레이 구성을 위해 3-튜플 위치 정보를, 세계를 대륙별로 적절하게 분류하여 특정 코드를 부여한 4-튜플 정보로 확장하는 것이 바람직하다. 그리고 미디어 서버의 효율적 운영을 위한 개선된 서비스 정책에 관한 연구가 함께 이루어져야 할 것이다.

참고 문헌 (References)

- [1] S. J. Koh, J. Y. Park, E. S. Kim and S. G. Kang, "Overview of Multicasting for Internet Broadcasting," *Electronics and Telecommunications Trends*, Vol.17, No.3, pp.1-14, 2002
- [2] W.-H. Chung, "Design of Component Based Web-casting System for Real-Time Internet Broadcasting," *J. of Korea Multimedia Society*, Vol.12 No.1, pp.69-84, Jan. 2009
- [3] W.-H. Chung and Y.-M. Lim, "Movie Contents Design of One-Person

Production Using IP Camera,” *J. of Korea Contents Association*, Vol.11, No.2, pp.1-12, 2011

[4] YouTube, <http://www.youtube.com>

[5] PandoraTV, <http://www.pandora.tv>

[6] Cisco, “Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are,” White Paper, Cisco Public, 2015

[7] S. W. Keum, T. B. Lim and J. I. Park, “Fog Computing Architecture for IoT Service under Hybrid Broadcasting Environment,” *J. of Korean Institute of Broadcasting and Media Engineers*, Vol.22, No.1, pp.107-117, 2017

[8] AfreecaTV, <http://www.afreecatv.com>

[9] B. Yang and H. Garcia-Molina, Designing a Super-Peer Network, *Proc. of Int’l Conf. on Data Engineering*, Bangalore, India, 2003

[10] P. Eugster, R. Guerraoui, A.-M. Kermarrec and L. Massoulie, “From Epidemics to Distributed Computing”. *IEEE Computer*, Vol. 21, No. 4, pp.341 - 374, 2003

[11] GP. Jesi, A. Montresor, O. Babaoglu, “Proximity-aware Superpeer Overlay Topologies,” *Self-Managed Networks, Systems, and Services*, Vol. 3996, LNCS, pp.43 - 57, 2006

[12] M. Liu, E. Harjula and M. Ylianttila, “An Efficient Selection Algorithm for Building a Super-Peer Overlay,” *Journal of Internet Services and Applications*, Vol. 4, No. 4, 2013, pp.4-16

[13] S. -H. Min, J. Holiday, and D. -S. Cho, “Optimal Super-Peer Selection for Large-Scale P2P System,” *Proc. of 2006 Int’l Conf. on Hybrid Information Technology (ICHIT’06)*, 2006

[14] T. E. Ng, and H. Zhang, “Predicting Internet Network Distance with Coordinates-based Approach,” *Proc. of INFOCOM*, 2002

[15] W. H. Chung, “Proximity-based Efficient Super-Peer Selection for Clustering in Peer-to-Peer Systems,” *INFORMATION*, Vol. 19, No. 11(B), Nov. pp.5483-5490, 2016

[16] L. Kleinrock, *Queueing Systems Vol.1: Theory*, John Wiley&Sons, NY, 1975

저 자 소 개



정 원 호

- 1984년 3월 ~ 1984년 8월 : KAIST 위촉 연구원
- 1989년 3월 ~ 2010년 2월 : 덕성여자대학교 컴퓨터공학부 교수
- 1992년 3월 ~ 1992년 7월 : IBM TJ Watson연구소 방문연구원
- 2010년 3월 ~ 현재 : 덕성여자대학교 디지털미디어학과 교수
- ORCID : <https://orcid.org/0000-0001-7157-9935>
- 주관심분야 : 분산 및 모바일 컴퓨팅



이 승 연

- 현재 : 덕성여자대학교 디지털미디어공학과 학사과정
- 주관심분야 : 모바일 컴퓨팅