

공유 네트워크에서 최대 요구대역폭 트리 구축을 위한 효율적인 알고리즘

정 균 락*

An Efficient Algorithm for Constructing a Maximal Request Bandwidth Tree on Public-shared Network

Kyun-Rak Chong*

요 약

최근에 사용자 스스로가 네트워크를 구축하여 자신이 소유한 AP의 일부 대역폭을 다른 사람과 공유하는 방법이 대두되었는데 이러한 네트워크를 공유 네트워크라 한다. 응용 애플리케이션으로 공유 네트워크에서 SVC 기술을 사용하는 비디오 스트리밍 전송 시스템을 구축하는 방안이 제안되었는데, 서버로부터 클라이언트에게 비디오 스트림을 보내기 위해서는 루트는 서버이고 내부노드는 공유 AP이며 리프는 클라이언트인 트리 구조를 만들게 된다. 기존의 연구들은 공유 AP의 공유대역폭의 합을 최소로 사용해서 모든 클라이언트를 서비스하는 최소 공유대역폭 트리를 구축하는데 주안점을 두고 있다. 본 논문에서는 공유 AP들의 집합이 주어졌을 때 클라이언트의 비디오 스트림 요구를 최대 만족시키는 최대 요구대역폭 트리를 구축하는 문제가 NP-하드임을 증명하였다. 또 이 문제를 해결하기 위한 효율적인 휴리스틱 알고리즘을 개발하고, 실험을 통해 성능을 평가하였다.

▶ Keywords : Wi-Fi 커뮤니티 네트워크, 공유 네트워크, 대역폭 공유, 비디오 스트리밍, NP-하드

Abstract

Recently, an idea has been suggested in which members construct the network by sharing their surplus bandwidth of their own access point. This kind of network is called public-shared network. As an application, SVC video streaming delivery system on public-shared network has been proposed. To send video stream from the stream server to clients, a tree structure is constructed where the root is a stream server, internal nodes are sharable access points, and leafs are clients. The previous researches have focused on constructing the minimal sharable-bandwidth tree which can serve all video streaming requests

•제1저자 : 정균락 •교신저자 : 정균락

•투고일 : 2015. 1. 7, 심사일 : 2015. 2. 5, 게재확정일 : 2015. 4. 6.

* 홍익대학교 컴퓨터공학과 (Dept. of Computer Engineering, HongIk University)

※ 이 논문은 홍익대학교 교수 연구년 기간 (2014.9-2015.8) 중 연구되었음.

using the minimal sharable bandwidth. In this paper, we have shown that the problem of constructing a tree structure with given sharable access points to serve maximal video streaming requests is NP-hard. We also have developed an efficient heuristic algorithm for this problem and evaluated experimentally our algorithm.

▶ Keywords : Wi-Fi community network, public-shared network, bandwidth sharing, video streaming, NP-hard

I. 서 론

최근에 사용자가 자신이 소유한 AP로 네트워크를 구축하여 여분의 대역폭을 다른 사람과 공유하는 방법이 대두되었다. 이러한 공유 네트워크의 예로는 현재 가장 큰 Wi-Fi 커뮤니티인 FON[1]이 있는데, 회원들은 인터넷에 연결된 FON 라우터를 소유하고 있으며, 대역폭 공유를 통해 어디에서든지 인터넷에 무료로 접속할 수 있고, 파일을 업로드하거나 다운로드할 수 있다[2, 3, 4].

공유 네트워크를 이용한 응용 애플리케이션으로 최근에 SVC(scalable video coding) 방식으로 코딩된 비디오 스트림을 전송하는 시스템을 구축하는 방법이 제안되었다[5, 6]. SVC는 영상 콘텐츠를 시간적, 공간적, 화질적 스케일러 빌리티를 모두 제공하는 하나의 비트스트림을 구성하여 여러 이질적인 단말기에서 자신의 성능에 맞는 비트스트림을 받아 복원할 수 있게 하는 비디오 코딩 기술이다[7, 8].

제안된 시스템은 비디오 스트림을 제공하는 비디오 스트리밍 소스, 전체 시스템을 관리하는 서버, 클라이언트에게 비디오 스트림을 전달하는 공유 AP들로 이루어져 있다. 비디오 스트림은 하향 링크를 따라 공유 AP로 전달되고, 여기서 대역폭 요구에 따라 다중 복사되어 다른 공유 AP나 클라이언트 로들로 전달된다. 이와 같이 대역폭 확장을 통해 비디오 스트리밍 소스는 비디오 스트림을 시스템에 전달하는 데 작은 양의 대역폭만 필요하게 되고, 다수의 클라이언트는 비디오 스트림을 동시에 받아 볼 수 있게 된다. 공유 네트워크에서 서버로부터 모든 클라이언트에게 비디오 스트림을 보내기 위해서는 트리 구조를 만드는데, 기존의 연구들은 공유 AP의 공유대역폭을 최소로 사용해서 모든 비디오 스트림 요구를 서비스할 수 있는 최소 공유대역폭 트리 구축 문제에 주안점을 두

고 있다[5, 9, 10].

최소 공유대역폭 트리 구축 문제는 문제 특성상 공유 AP의 공유대역폭이 클라이언트의 요구대역폭에 비해 충분히 크다고 가정할 때만 의미가 있다. 그러나 실제로는 이러한 가정이 성립되지 않는 경우가 많고, 이런 경우에는 클라이언트의 요구를 모두 만족시키는 트리는 구축할 수 없으므로, 대신에 가능한 많은 클라이언트의 요구를 만족시키는 트리를 구축하는 것이 바람직하다.

본 연구에서는 공유 AP의 집합이 주어졌을 때, 클라이언트의 요구대역폭의 합을 최대로 서비스하는 최대 요구대역폭 트리 구축 문제가 NP-하드임을 증명하였고, 근사해를 구하기 위한 효율적인 알고리즘을 개발하였다. 또 실험을 통해 제안된 알고리즘의 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 자세히 기술하고 3장에서는 최대 요구대역폭 트리 구축 문제에 대해 정의하고, 이 문제가 NP-하드임을 증명하며, 이 문제를 해결하기 위한 제안된 알고리즘에 대해 기술한다. 4장에서는 제안 알고리즘의 성능을 실험을 통해 평가하고, 5장에서 결론을 맺는다.

II. 관련 연구

최근에 공유 네트워크에서 SVC 방식으로 코딩된 비디오 스트림을 전송하는 방안들이 연구되었다. [5]에서는 전체 공유대역폭을 최소로 사용해서 모든 클라이언트의 요구를 서비스하는 트리를 구축 문제가 NP-하드임을 증명하였다. 근사해를 구하기 위한 알고리즘도 제안되었는데, 트리를 구축할 때, 공유대역폭이 큰 공유 AP부터 요구대역폭이 큰 클라이언트를 먼저 할당하기 때문에 공유대역폭이 큰 공유 AP가 일찍 소모되어 트리를 구축할 수 있는 경우에도 트리를 구축하지

못하는 비율이 상당히 높은 단점을 가지고 있다.

이 단점을 개선하여 해를 찾는 성공률이 높은 알고리즘이 [9]에서 제안되었는데, 공유대역폭이 작은 공유 AP부터 요구대역폭이 작은 클라이언트를 먼저 할당한다. 이 방법은 공유대역폭이 큰 공유 AP를 나중에 사용하므로 해를 찾는 성공률은 높으나 구축된 공유대역폭 트리가 사용하는 전체 공유대역폭이 다소 커지는 단점을 가지고 있다. [10]에서는 공유대역폭이 큰 공유 AP부터 요구대역폭이 큰 클라이언트를 먼저 할당하는데 일정 비율의 공유 AP를 트리 구축 시 나중 단계에서 사용할 수 있게 예비해둔다. 이 방법은 해를 찾는 비율이 앞에 기술한 두 방법보다 우수하였고, 전체 공유 대역폭도 [9]에서 제안한 방법보다 향상되었다.

클라이언트들이 인터넷상에 널리 퍼져있는 경우에는 비디오 스트림이 다수의 링크를 따라 보내질 수 있다. 그러면 네트워크 대역폭을 더 많이 점유하게 되고 전송시간도 길어져서 클라이언트들이 비디오 스트림을 제시간에 전송받지 못하게 된다. 이 문제를 해결하기 위해 유사한 추적 경로를 갖는 공유 AP와 클라이언트를 같은 그룹에 할당하여 트리를 구축하고, 이러한 그룹들을 모아서 상위 그룹으로 반복적으로 만들어 계층적 트리를 구축하는 방법이 [6]에서 제시되었다.

공유 네트워크와 관련된 연구들을 보면 [11]에서는 공유 네트워크에서 무선 공유 기술을 사용해서 스트리밍 대역폭을 감소시키는 방안을 제안하였다. 유선과 무선이 혼합된 트리 구조에서 각 노드는 공유 AP이고 각 공유 AP 근처에는 무선 링크들이 클러스터를 이루고 있는데 이 무선 링크들이 라이브 스트림을 운반하게 된다.

공유 네트워크에서 클라이언트들이 스마트폰 같은 휴대기 기일 때 라이브 스트림을 전송하기 위한 효율적인 대역폭 관리 방법이 [12]에서 제안되었다. 이 경우 SVC 방식은 부호화와 복호화에 많은 부하가 걸리고, 따라서 휴대기들이 SVC 연산을 수행하기 어려우므로, 클라이언트들의 비트 전송률을 만족시키기 위해 분리-합병(split-merge) 방법을 사용하였다. 이 방법에서 전송할 스트림은 n개의 서브스트림으로 분리되고, 이 서브스트림들은 공유 AP로 이루어진 n개의 서브트리를 통해 운반되는데 각 서브트리가 1개의 서브스트림을 자신에게 속한 각 클라이언트로 보내게 된다. 모든 클라이언트들은 버퍼를 사용하여 전송받은 n개의 서브 스트림을 원래 스트림으로 합병하게 된다. [13]에서는 FON과 같은 네트워크에서 공유 IP를 사용할 때 생기는 보안 문제에 대해 연구하였다

III. 문제 정의와 알고리즘

1. 최대 요구대역폭 트리 구축 문제

이 장에서는 최대 요구대역폭 트리 구축 문제를 정의하고자 한다. 트리의 루트는 서버이고, 내부노드는 공유 AP이고, 리프는 클라이언트이다. 먼저 서버, 공유 AP와 클라이언트가 갖는 특성을 보면 다음과 같다.

각 클라이언트 c는 요구대역폭 RB(c)를 가지며, 클라이언트는 자식은 없고 부모는 공유 AP이거나 서버이다. 공유 AP s의 자식은 클라이언트나 자식들이 할당된 다른 공유 AP이며, 자식들에게 제공할 수 있는 공유대역폭 SB(s)를 가지는데 자식들의 요구대역폭의 합은 SB(s)를 초과하지 못한다. 자식들이 할당된 공유 AP s의 요구대역폭 RB(s)는 자식들의 요구대역폭 중 최대 요구대역폭으로 정의되고, 자식들의 대역폭 요구들을 만족시키기 위해 부모로부터 비디오 스트림을 받는데 필요한 대역폭이다. 서버의 자식은 클라이언트나 자식이 있는 공유 AP가 될 수 있는데 자식들의 요구대역폭의 합은 주어진 서버의 용량을 초과하지 못한다[9]. 트리 T의 공유대역폭 SB(T)는 트리를 구성하고 있는 모든 공유 AP들의 공유대역폭의 합이고, 요구대역폭 RB(T)는 트리를 구성하고 있는 모든 클라이언트들의 요구대역폭의 합이다.

최대 요구대역폭 트리 구축 문제는 공유 AP들의 집합 S, 클라이언트의 집합 C와 서버의 용량이 주어졌을 때, RB(T)가 최대인 트리 T를 구하는 문제이다.

예를 들어, 공유 AP의 공유대역폭이 $SB(s_i) = (16, 8, 8, 8)$ 이고, 클라이언트의 요구대역폭이 $RB(c_i) = (8, 8, 8, 4, 4, 4, 4)$ 이며, 서버 용량이 16일 때, 최대 요구대역폭 트리가 그림 1에 나타나 있다. 편의상 64kbs를 1단위로 사용하였다.

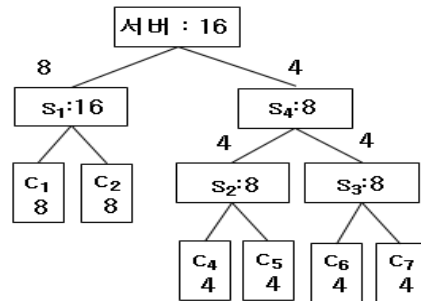


그림 1. 최대 요구대역폭 트리의 예
Fig. 1. An Example of Maximal Request Bandwidth Tree

그림 1에서 리프 안에 있는 숫자는 클라이언트의 요구대역폭을 나타내고, 내부노드 안에 있는 숫자는 공유 AP의 공유대역폭을, 내부 노드 위에 있는 숫자는 공유 AP의 요구대역폭을 나타낸다. 이 트리가 서비스하는 클라이언트는 $c_1, c_2, c_4, c_5, c_6, c_7$ 이고, 이 트리의 요구대역폭은 32이다.

2. NP-하드 증명

먼저 최대 요구대역폭 트리 구축 문제가 NP-하드임을 보이기 위해 이미 NP-하드임이 증명되어 있는 분할 문제를 정의하기로 한다 [14].

정의 1 : 분할(partition) 문제

n 개의 양의 정수 a_1, a_2, \dots, a_n 이 주어졌을 때 다음을 만족하는 분할 P 가 존재하는가?

$$\sum_{i \in P} a_i = \sum_{i \notin P} a_i \quad (1)$$

(1)은 집합 P 에 속한 수들의 합과 집합 P 에 속하지 않은 수들의 합이 같은 P 가 존재하는가를 의미한다.

정리 1. 최대 요구대역폭 트리 구축 문제는 NP-하드이다.

(증명) 분할 문제가 최대 요구대역폭 트리 구축 문제로 방향 시간 변환됨을 보이는데, 공유 AP의 수가 2일 때 이 문제가 NP-하드임을 보인다.

n 개의 양의 정수 a_1, a_2, \dots, a_n 이 주어졌을 때 이 수들의 합 $\sum_{1 \leq i \leq n} a_i = 2u$ 라 하면, 분할 문제로부터 최대 요구대역폭 트리 구축 문제의 예(instance)를 다음과 같이 만든다. 서버의 용량은 $2u+2$ 라 하고, 공유 AP s_1 과 s_2 의 공유대역폭은 모두 $2u+1$ 이라 한다. 클라이언트의 수는 $n+2$ 라 하고 클라이언트를 c_1, c_2, \dots, c_{n+2} 라 하면, 클라이언트의 요구대역폭은 $RB(c_i) = a_i, 1 \leq i \leq n, RB(c_{n+1}) = u+1, RB(c_{n+2}) = u+1$ 로 정의한다. 공유 AP의 공유대역폭이 $2u+1$ 이므로 클라이언트 c_{n+1} 과 클라이언트 c_{n+2} 는 요구대역폭의 합이 $2u+2$ 이므로 같은 공유 AP에 할당 될 수 없고 서로 다른 공유 AP에 할당 되어야 한다.

그림 2에 가능한 트리가 나타나 있다. 그림 2에서 분할 P 에 속한 클라이언트들과 클라이언트 c_{n+1} 은 공유 AP s_1 에 할당되고, 분할 P 에 속하지 않은 클라이언트들과 클라이언트 c_{n+2} 은 공유 AP s_2 에 할당되었다고 가정한다.

분할 P 가 존재하면 $\sum_{i \in P} a_i = \sum_{i \notin P} a_i = u$ 가 되므로 공유 AP s_1 과 공유 AP s_2 에 할당된 클라이언트의 요구대역폭의 합은 각각 $2u+1$ 이 되어 할당된 모든 클라이언트의 요구

를 만족시킬 수 있다. (분할 P 가 존재하지 않으면 $\sum_{i \in P} a_i > u$ 또는 $\sum_{i \notin P} a_i > u$ 가 되므로 공유 AP s_1 과 공유 AP s_2 중 하나는 할당된 클라이언트의 요구대역폭의 합이 $2u+1$ 보다 커져 할당된 클라이언트의 요구를 만족시킬 수 없게 된다.)

역으로 클라이언트의 모든 요구대역폭을 만족시키는 트리가 존재하면 각 공유 AP에는 요구대역폭이 $u+1$ 인 클라이언트(c_{n+1} 또는 c_{n+2})가 이미 하나씩 할당되어 있어야 하므로, 공유 AP s_1 에 할당된 나머지 클라이언트의 요구대역폭의 합과 공유 AP s_2 에 할당된 나머지 클라이언트의 요구대역폭의 합은 모두 u 가 되어야 하고, 따라서 분할이 존재하게 된다. □

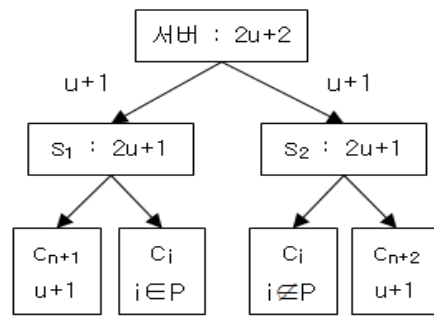


그림 2. 가능한 최대 요구대역폭 트리
Fig. 2. Possible Maximal Request Bandwidth Tree

3. 제안 알고리즘

최대 공유대역폭 트리는 클라이언트를 공유 AP에 단계적 할당하면서 구축된다. 클라이언트가 할당된 공유 AP는 다시 클라이언트처럼 취급되어 서버나 다른 공유 AP에 할당된다. 이러한 공유 AP와 클라이언트를 합해서 확장 클라이언트라 칭하기로 한다. c 가 확장 클라이언트일 때 $TB(c)$ 는 c 의 서브트리에 속해 있는 클라이언트의 요구대역폭의 합으로 정의한다.

3.1 클라이언트 할당

먼저 qS 와 qC 는 각각 공유 AP와 클라이언트의 큐(queue)로 초기에는 각각 공유대역폭과 요구대역폭의 내림차순으로 정렬되어 있다. 첫번째 단계에서는 요구대역폭이 큰 클라이언트부터 공유대역폭이 큰 공유 AP에 할당하는데 클라이언트의 요구대역폭의 합이 공유 AP의 공유대역폭을 넘지 않을 때까지 클라이언트를 순서대로 공유 AP에 할당한다. 클라이언트의 할당이 완료된 공유 AP는 클라이언트로 변환시켜 qSC 큐에 저장해 놓는다.

3.2 공유 AP 비축

클라이언트 할당 시 공유대역폭이 큰 공유 AP부터 사용하기 때문에 공유대역폭이 큰 공유 AP가 먼저 소모된다. 그런데 트리를 성공적으로 구축하기 위해서는 나중에도 공유대역폭이 큰 공유 AP가 필요하므로 공유 AP를 비축할 필요가 있게 된다[10]. 새로 자식이 할당된 공유 AP들의 요구대역폭의 합이 다음 공유 AP의 공유대역폭보다 크거나 같게 되면 다음 공유 AP를 qRS 큐에 저장해 놓는다.

```

Maximal Request Bandwidth Tree Algorithm
Tree = ∅;
initialize qS, qC, qRS, qSC;
tot_rb = sum of the requested bandwidth of
clients in C;
ci = deleteC();
sj = deleteS();
while (tot_rb > server capacity) {
    sum = 0;
    while (sum + RB(ci) ≤ SB(sj)) {
        sum = sum + RB(ci);
        Tree = Tree ∪ {(sj, ci)};
        tot_rb = tot_rb - RB(ci);
        if (qC ≠ ∅) ci = deleteC();
        else {
            qCempty = true; break;
        }
    }
    if (sum > 0) {
        max_rb = find_max_rb(sj);
        tot_rb = tot_rb + max_rb;
        insertSC(sj);
        reserve_snode();
        if (qS ≠ ∅) sj = deleteS();
        else qSempty = true;
    }
    if (qCempty) {
        qS ← qS ∪ qRS;
        qC ← qSC;
        if (qC ≠ ∅) ci = deleteC(); else break;
        if (qS ≠ ∅) sj = deleteS(); else break;
    }
    else if (qSempty) {
        qS ← qRS;
        if (qS ≠ ∅) sj = deleteS(); else break;
    }
}
qC ← qC ∪ qSC;
if (tot_rb > server capacity) sort qC by TB value
ci = deleteC();
sum = 0;
while (sum + RB(ci) ≤ server capacity) {
    sum = sum + RB(ci);
    Tree = Tree ∪ {(sever, ci)};
    if (qC ≠ ∅) ci = deleteC(); else break;
}
return Tree, RB(Tree)

```

그림 3. 최대 요구대역폭 트리 구축 알고리즘

Fig. 3. Maximal Request Bandwidth Tree Construction Algorithm

3.3 확장 클라이언트의 할당

첫 번째 단계가 끝나고 qS나 qC가 비게(empty) 되면 다음 단계가 시작된다. qS가 비어 있으면 qRS에 있는 비축된 공유 AP를 qS로 이동시킨 후, 클라이언트들을 비축된 공유 AP에 할당한다. qC가 비어 있으면 qRS에 있는 비축된 공유 AP를 qS로 이동시키고, qSC에 있는 확장 클라이언트를 qC로 이동시킨 후 qS에 있는 공유 AP에 계속 할당한다. 이 과정을 다음 절의 조건을 만족할 때까지 반복한다.

3.4 알고리즘의 종료 조건

다음의 경우에 알고리즘이 종료된다.

(1) qC와 qSC에 있는 확장 클라이언트의 요구대역폭의 합이 서버의 용량보다 작거나 같을 때

이 경우에는 서버가 확장 클라이언트들을 직접 서비스하게 되고 알고리즘은 종료된다.

(2) qS나 qC가 비어 있을 때

이 경우에는 qC와 qSC에 있는 확장 클라이언트들을 TB 값에 대한 내림차순으로 정렬하고 순서대로 요구대역폭의 합이 서버의 용량을 넘지 않을 때까지 선택하여 트리를 구축한다.

제안된 알고리즘이 그림 3에 나타나 있다. deleteC와 deleteS는 각각 qC와 qS에서 노드를 삭제하는 함수이고, insertSC는 qSC에 노드를 삽입하는 함수이다. find_max_rb(s)는 공유 AP s에 할당된 자식들 중에서 최대 요구대역폭을 찾아 공유 AP s의 요구대역폭으로 할당하는 함수이다. tot_rb는 남아있는 요구대역폭의 총합으로, 이 값이 서버의 용량보다 작거나 같으면 서버가 qC에 있는 노드들을 직접 서비스하게 된다. tot_rb의 초기값은 모든 클라이언트의 요구대역폭의 합이 된다. reserve_snode()는 공유 AP를 비축하는 함수이고, Tree는 구축된 트리의 에지들의 집합이다.

IV. 성능 평가

제안된 알고리즘은 C 언어를 사용해서 Intel(R) Core(TM) i5 CPU를 가진 PC에서 구현되고 실험되었는데 성능 평가는 공유 AP의 수와 클라이언트의 수의 합과 서버의 용량을 고려하였다.

공유 AP의 수와 클라이언트의 수의 합에 따라 세 그룹의 데이터를 생성하였는데, 노드의 수가 S 그룹은 30에서 70개, M 그룹은 80에서 120개, L 그룹은 180에서 220개를 사용

하였고 각 그룹별로 50개의 데이터를 생성하였다. 공유 AP의 공유대역폭의 종류는 6가지로 1,024, 512, 256, 128, 64, 32(단위 kbs)를 사용하였고, 클라이언트의 요구대역폭의 종류도 6가지로 512, 256, 128, 64, 32, 16을 사용하였다. 같은 공유대역폭을 갖는 공유 AP의 수와 같은 요구대역폭을 갖는 클라이언트의 수는 1부터 50사이에서 임의로 생성되었고, 클라이언트의 요구대역폭의 합이 공유 AP의 공유대역폭과 서버의 용량의 합보다 크지 않도록 생성되었다. 서버의 용량은 타입 1은 1024, 타입 2는 2048, 타입 3은 4,096을 사용하였다.

표 1 제안 알고리즘과 기존 알고리즘의 서비스 비율 비교
Table1. Comparison of the service ratio between the proposed algorithm and the previous algorithms

	D-algo	A-algo	M-algo	제안
타입 1	0.23	0.34	0.25	0.63
타입 2	0.31	0.41	0.38	0.77
타입 3	0.42	0.49	0.52	0.86

표 1에 기존 알고리즘과 제안 알고리즘의 결과 비교가 나타나 있다. 표 안의 숫자는 생성된 모든 데이터의 클라이언트의 요구대역폭의 합을 1이라 하였을 때, 각 알고리즘이 트리를 구축하여 서비스한 클라이언트의 요구대역폭의 합을 의미한다. 여기서 D 알고리즘은 [5]에서, A 알고리즘은 [9]에서, M 알고리즘은 [10]에서 제안된 알고리즘이다. 제안 알고리즘은 모든 클라이언트의 요구를 서비스 할 수 있는 트리를 구축하지 못할 경우에는 최대한 서비스할 수 있는 트리를 구축하므로 기존 알고리즘에 비해 서비스 비율이 좋아지게 된다.

그림 4에는 네트워크의 크기에 따라 제안 알고리즘이 서비스한 비율이 나타나 있다. 실험 결과를 보면 서버의 용량이 클수록 서비스 비율이 향상되는 데 타입 1에 비해 타입 2는

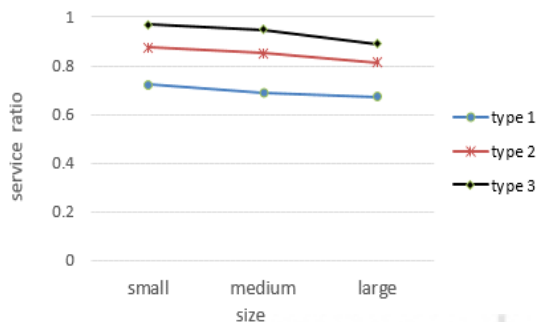


그림 4. 그룹 크기와 서버 용량에 따른 서비스 비율
Fig. 4. Service Ratio on Group Size and Server Capacity

평균 21.9% 향상되었고, 타입 3은 평균 34.7% 향상되었다. 공유 AP와 클라이언트의 수에 따른 결과를 보면 공유 AP의 수와 클라이언트의 수의 합이 작을수록 서비스되는 클라이언트의 요구대역폭의 비율이 증가하는 것을 알 수 있다.

V. 결론

최근에 사용자 스스로가 네트워크 시스템을 구성하여 자신이 소유한 AP의 일부 대역폭을 다른 사람과 공유하는 방법이 대두되었는데, 이러한 공유 AP를 통해 어디서나 인터넷에 접근할 수 있고 인터넷에 연결된 외장 하드 같은 장비에 Wi-Fi 접속이 가능하다. 또 공유 네트워크에서 응용 애플리케이션으로 SVC 기술을 사용하는 비디오 스트리밍 전송 시스템을 구축하는 방안이 제안되었다. 이 시스템은 서버로부터 모든 클라이언트에게 비디오 스트림을 보내기 위해서는 서버, 공유 AP, 클라이언트로 구성된 트리 구조를 만드는 것이 필요하다.

본 연구에서는 공유 AP들의 집합이 주어졌을 때 클라이언트들의 요구를 최대로 서비스하는 최대 요구대역폭 트리 구축 문제가 NP-하드임을 증명하였고, 이 문제를 해결하기 위한 휴리스틱 알고리즘을 개발하였다. 또 실험을 통해 네트워크 크기에 따라 제안된 알고리즘의 성능을 분석하였다.

본 연구의 결과는 공유 네트워크상에서 비디오 스트리밍을 전송할 때 클라이언트의 요구를 최대로 만족시키는데 유용하게 사용될 수 있다.

REFERENCES

- [1] FON official website: <http://www.fon.com>
- [2] G. Camponovo and A. Picco-Schwendener, "Motivations of Hybrid Wireless Community Participants: A Qualitative Analysis of Swiss FON Members," 2011 Tenth International Conference on Mobile Business (ICMB), pp. 253-262, 2011
- [3] C. Su, Y. Hwang, C. Yeh, "A Study on the Willingness of Using FON in the Domain of Wireless Communication," 4th Intl. Conf. on Network Computing and Advanced Information Management, pp. 159-164, 2008
- [4] A. Asheralieva, T. Erke, and K. Killki, "Traffic Characterization and Service Performance in

- FON Network," 1st Intl. Conf. on Future Information Networks, 2009
- [5] N.F. Huang, H.Y. Chang, Y.W. Lin, K.S. Hsu, and H.C. Liu, "On the Complexity of the Bandwidth Management Problem for scalable Coding Video Streaming on a Public-Shared Network," IEEE Communication Letters, vol. 13, no. 1, pp. 61-63, 2009
- [6] N.F. Huang, H.Y. Chang, T.C. Wang, Y.S. Lin, Y.W. Lin, S.Y. Cheng, and J.J. Lin, "An Efficient and Locality-aware Resource Management Scheme for SVC-based Video Streaming System on a Public-Shared Network," Asia-Pacific Conf. on Communication, pp. 682-685, 2009
- [7] H. Schwarz, D. Marpe, and T. Wieg, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103-1120, 2007
- [8] G. Auwera and M. Reisslein, "Implication of Smoothing Multiplexing of H.264/AVC and SVC Video Streams," IEEE Trans. on Broadcasting, vol. 55 no. 3, Sep. 2009
- [9] K. Chong, "An Efficient Bandwidth Management Algorithm for SVC Video Streaming on Public-shared Networks," Journal of KIISE : Information Networking, vol. 38, no. 3, pp.243-247, June 2011
- [10] K. Chong, "A Linear Time Algorithm for Constructing a Sharable-Bandwidth Tree in Public-shared Networks," Journal of the Korea Society of Computer Information, vol. 17, no. 6, pp. 93-100, 2012
- [11] H.-Y. Chang, N.-F. Huang, K.-B. Chen, Y.-J. Tzang, Y.-W. Lin, "Saving streaming bandwidth via wireless sharing for a tree-based live streaming system on Public-Shared Network," 2012 8th International Conference on Computing Technology and Information Management (ICCM), pp. 88-94, 2012
- [12] H.-Y. Chang, N.-F. Huang, Y.-W. Lin, Y.-J. Tzang, "A Novel Bandwidth Management System for Live Video Streaming on a Public-Shared Network," IEEE Transactions on Vehicular Technology, vol. 62, no. 8, pp. 3848-3862, 2013
- [13] N. Vratonjic, K. Huguenin, V. Bindschaedler, J.-P. Hubaux, "A Location-Privacy Threat Stemming from the Use of Shared Public IP Addresses," IEEE Transactions on Mobile Computing, vol. 13, Issue 11, pp. 2445 - 2457, 2014
- [14] E. Horowitz, S. Sahni, and D. Metha, "Fundamentals of Data Structures in C++", Compute Science Press, 1995

저자 소개



정 균 략

1980년 2월 : 한국과학기술원
전자계산학 석사

1991년 2월 : 미네소타대학교
컴퓨터공학 박사

1991년 ~ 현재 : 홍익대학교
컴퓨터공학과 교수

관심분야 : 네트워크 알고리즘,
이동 통신,
무선 센서 네트워크,
머신 스케줄링 알고리즘,
VLSI 알고리즘

Email : chong@cs.hongik.ac.kr