

# 멀티 사이트 인터넷 환경에서 효율적인 서버 선택

정회원 이 현 표\*, 이 균 하\*\*

## An Efficient Server Selection Over The Multi-site Internet Environments

Hyun Pyo Lee\*, Sung Sik Park\*\*, Kyoon Ha Lee\*\* *Regular Members*

### 요 약

인터넷 서비스 품질의 향상을 위해 고성능 서버의 증설 및 트래픽 분산 등 서비스 대역의 확장으로 인터넷 접속 장치인 서버 및 가입자망의 접속 속도는 급속히 향상되었으나, 인터넷의 접속 품질과 속도에 대한 만족도는 기대에 못 미치고 있다. 이러한 문제는 인터넷 접속장치의 증설로는 증가하는 트래픽을 수용하는 데에는 한계가 있고, 사용자측 노드에서 CP(Content Provider) 서버에 이르는 미들마일(Middle-mile) 구간이 개선되지 못하는 데 원인이 있다.

본 논문에서는 지역적으로 분산된 멀티 사이트 인터넷 환경에서 각 클라이언트의 위치에 따라 클러스터링된 서버들의 로드 밸런싱을 유지하면서 서버와 사용자 측 노드를 최소화하여 사용자에게 효율적인 서버를 선택하는 방법을 제안하였다. 제안된 방법에는 효율적인 서버를 선택하기 위해 네트워크 상태 감시자(NSP)와 콘텐츠 서버 관리자를 두어 각 서버의 상태와 각 분산된 네트워크의 상태를 파악 할 수 있도록 하였으며, 서버 선택 알고리즘과 알고리즘을 실현하기 위한 서비스 구조를 제시하였다. 또한, 효율적인 서비스 서버를 선택하기 위한 인자들과 측정 방법을 나타내었으며, 제안된 서비스 구조에서 실험을 통하여 타당성을 확인하였다.

### ABSTRACT

In order to providing more improved quality of Internet service, the access speed to a subscriber's network and a server which is the Internet access device was rapidly enhanced by traffic distribution and installation of high-performance server. But the quality of the Internet access and the content for a speed was remained out of satisfaction. With such a hazard, an extended node at Internet access device has a limitation for coping with a growing network traffic, and the root cause is located in the Middle-mile node between a CP(Content Provider) server and a user node. For such a problem, this paper proposes a new method to select a effective server to a client as minimizing the number of node between the server and the client while keeping the load balance among servers which is clustered by the client's location on the physically distributed multi-site environments. The proposed method use a NSP(Network Status Prober) and a contents server manager so as to get a status of each servers and distributed network, a new architecture will be suggested for the server selecting algorithm and the implementation for the algorithm. And also, this paper shows the parameters selecting a best service providing server for client and that the grantor will be confirmed by the experiment over the proposed architectures.

### I. 서 론

인터넷은 이미 정보화 시대의 필수사항으로 인식

되고 있고, 인터넷 사용자의 수는 기하급수적으로 증가하고 있으며, 인터넷의 서비스 요구는 텍스트 기반의 정적인 콘텐츠에서 동적인 멀티미디어 콘텐츠

\* 안산1대학 인터넷상거래과(hplee@mail.ansan.ac.kr),  
논문번호 : 010082-0427, 접수일자 : 2001년 4월 27일

\*\* 인하대학교 전자계산공학과

츠 서비스로 전 세계에 걸친 광대역 서비스를 요구하고 있다. 이러한 인터넷 사용자의 기하급수적 증가와 인터넷 서비스의 광대역화는 필연적으로 정보 제공 서버의 과부하를 초래하게 된다. 이러한 과부하를 해결하고 인터넷 서비스의 성능 향상을 위해 대부분의 인터넷 서비스 제공자들은 고성능 서버와 같은 고가의 네트워크 장비를 확장하고 있다. 그러나, 이와 같이 고성능 서버를 도입하는 데에는 많은 비용이 소요되고 있으며, 더욱이 단일 서버로 운영되는 경우에는 아무리 고성능 서버라 해도 폭증하는 서비스 요구를 처리하는데는 한계가 있다. 또한, 단일 서버 운영으로 인해 고성능 서버가 중지했을 경우 이에 적절히 대응할 수 없게 되어 신뢰성이 낮아지는 문제가 있다<sup>[2][3][5]</sup>.

낮은 비용과 신뢰성 높은 인터넷 서비스를 제공하기 위해, 인터넷 서비스 제공자들은 정보 제공 서버측에 여러대의 서버를 두어 클라이언트로부터 서비스 요구시 여러대의 서버로 고르게 분배하여 서비스 처리 능력을 향상시키는 클러스터링 방법을 사용하고 있다. 이것으로 여러 대의 인터넷 서버 중 하나의 서버가 다운되거나 한시적으로 엄청난 트래픽이 발생하더라도 지속적이고 보다 향상된 인터넷 서비스를 제공할 수 있게 되었다<sup>[5]</sup>.

이와 같이 서버의 증설 및 부하 분산 등 서비스 대역의 확장으로 인터넷 접속 장치인 서버 및 가입자망의 접속 속도는 급속히 향상되었으나, 인터넷의 접속 품질과 속도에 대한 만족도는 기대에 못 미치고 있다. 이러한 문제는 네트워크의 증설로 증가하는 트래픽을 수용하는 데에는 한계가 있고 사용자측 노드에서 CP(Content Provider) 서버에 이르는 미들마일(Middle-mile) 구간이 개선되지 못하는 데 원인이 있다. 이것을 개선하기 위해서는 지역적으로 분산된 네트워크마다 서버를 두어 사용자측 노드와 서버사이의 구간을 최소화하여 클라이언트 요청에 대한 응답 시간을 단축하여 네트워크 성능을 향상시키는 멀티 사이트 트래픽 관리 장치를 사용할 필요가 있다<sup>[10]</sup>.

멀티 사이트 트래픽 관리 장치는 네트워크 성능과 네트워크의 신뢰성을 향상시키는 것을 목적으로 하고 있으며, 단일 데이터 센터에서의 트래픽 관리 뿐만 아니라 지리적으로 분산된 데이터 센터의 트래픽을 관리한다. 멀티 사이트 트래픽 관리 장치는 도처에 흩어져있는 클라이언트에 대한 서비스 질을 향상시키는 역할을 하며, 클라이언트 요청에 대한 응답 시간을 줄임으로서 네트워크 성능을 향상시킬

뿐만 아니라 갑작스런 데이터 센터 불량에 대한 예방도 제공한다. 비록 클라이언트가 인접한 데이터 센터를 사용할 수 없는 상황이 발생하더라도 멀티 사이트 트래픽 관리 장치는 효과적으로 네트워크 자원들을 액세스할 수 있게 보장해주는 역할을 한다.

본 논문에서는 클라이언트의 요구에 따라 지리적으로 분산된 멀티 사이트 인터넷 서비스 환경에서 클라이언트에 대한 서비스 질을 향상시키고, 클라이언트 요청에 대한 응답시간을 줄여 인터넷 서비스 성능을 향상시킬 수 있는 시스템 구조와 서버를 선택하는 방법을 제시하였다. 클라이언트에게 효율적인 서버를 제공하기 위해 클라이언트와 서버 사이의 네트워크 상태와 지리적으로 분산된 서버들의 성능 상태를 고려하였다. 클라이언트에게 가장 효율적인 서버를 선택하기 위해 클라이언트와 각 사이트에 대한 지연시간과 주기적으로 수집된 네트워크 상태 정보(RTT, Packet loss, 네트워크 혼잡도)로 최적의 사이트를 선택하고 선택된 사이트에 위치한 콘텐츠 서버들의 상태 정보(디스크 부하량, CPU 사용률, 트래픽 량, TCP 연결 설정 수)로 최적의 콘텐츠 서버가 결정되는 2단계 서버 선택 알고리즘을 사용하였으며, 로컬 네트워크 서버 클러스터링에서 각 서버간의 로드 밸런싱이 유지되도록 하여 서비스 신뢰도 및 확장성을 제공하였고 실험을 통하여 그 타당성을 검증하였다.

## II. 기존 로드 밸런싱을 위한 부하 분산 시스템

클라이언트의 서비스 요구에 따라 여러 대의 서버로 향하는 트래픽을 효과적으로 분산하는 방법으로는 DNS(Domain Name System)를 사용한 방법과 IP-디스패치를 사용한 방법, HTTP-재설정을 사용한 방법으로 나눌 수 있다. 기존의 방법들은 모두 단일 네트워크 환경에서 클러스터링 기술을 사용해서 클라이언트에 대한 부하를 분산시켰다는 제한을 갖는다.

DNS를 사용한 방법<sup>[1][2]</sup>은 그림1에서와 같이 클라이언트는 잘 알려진 도메인 네임으로 DNS 서버에 해당하는 IP주소를 요청하고, DNS 서버는 클러스터링 되어진 여러 대의 서버들 중에 특정 서버의 IP 주소를 할당하여 클라이언트의 요구에 응답하여 이루어진다. 이러한 DNS를 사용한 접근 방법은 낮은 비용과 높은 가용성을 유지하면서 증가하는 네트워크의 요청에 효과적으로 대처하고 트래픽을 분

산시키기 위한 방법으로 클라이언트의 요구에 대해 순차적으로 응답하는 RR-DNS(Round Robin DNS)<sup>[1]</sup>를 많이 사용하고 있다.

RR-DNS는 인터넷 서비스를 제공하는 여러 대의 클러스터링 되어진 서버가 모두 동일한 성능을 가지고 있을 때 효과적인 방법이며 다음과 같은 문제점들이 있다. 특정 도메인에 대한 정보가 변경되었을 때 사용자가 변경된 사실을 알기 위해서는 여러 대의 DNS 서버를 거쳐야 하기 때문에 많은 시간이 소요되며, 전파 지연 시간 동안 일정한 비율의 사용자들은 서비스 제공을 받지 못하는 경우가 발생할 수 있다. 또한, 클라이언트와 RR-DNS 사이에 있는 중간 DNS 서버에서 클라이언트의 요청을 좀더 빠르게 처리하기 위해 한 번 요청 받았던 도메인 네임에 대한 IP 주소를 캐시에 저장해 두고 그것이 만기할 때까지 동일한 도메인에 대한 요청은 캐시의 내용을 참조해 응답한다. 이것은 부하 분산의 효과를 저하시키는 요인이 되며, RR-DNS 서비스 큐에 들어오는 요구들을 단순하게 분산시키므로 선택된 서버에 존재하는 부하나 성능을 고려하지 않았다는 문제점이 있다<sup>[2]</sup>.

IP-디스패처<sup>[3]</sup>를 사용한 방법은 NAT(Network Address Translation) 방식이라고도 하며, 패킷 내의 IP 주소를 변경해 부하분산을 수행하는 방법이다. 그림2는 기본적인 NAT 동작 방식을 나타낸 것으로 기본적으로 클라이언트에게는 로드 밸런서의 도메인 네임 또는 IP주소가 알려져 있다.

이 방식은 로드 밸런서에서 다양한 스케줄링 방법을 사용할 수 있다. 또한 로드밸런서에게 적절한 페일오버 기능을 부여할 경우, 클러스터 내의 서버가 다운되더라도 이것을 감지해 스케줄링 대상에서 제외하므로 네트워크 서비스가 중단되는 단점을 보완할 수 있다. 실제 서버는 사실 IP주소를 사용해도 되며 오직 부하 분산 서버만 실제 IP주소를 사용하면 된다. 그러나 로드 밸런서는 서버로 오가는 모든 패킷을 재작성 해야하므로 병목현상이 발생할 수

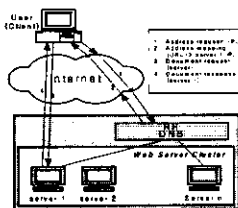


그림 1. RR-DNS를 사용한 부하분산 시스템

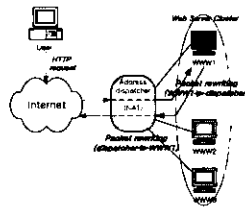


그림 2. 기본적인 NAT 동작방식

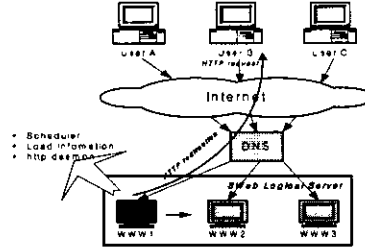


그림 3. Redirection을 이용한 SWEB에서의 부하분산

있으며, 클러스터를 구성할 수 있는 서버의 개수에 제한을 받는다. 또한, 로드 밸런서가 다운될 경우 전체서비스가 중단되는 문제가 발생한다<sup>[3]</sup>.

HTTP-재설정 방식의 대표적인 방식인 SWEB(the Scalable Web server) 시스템<sup>[5]</sup>은 그림3과 같이 두 레벨의 분산 스케줄러를 사용한다. 처음에 DNS를 통해서 웹서버에게 할당된 클라이언트의 요구는 HTTP 재설정으로 다른 서버에게 재할당된다.

HTTP-재설정 방법은 각각의 재전송되는 요구가 새로운 클라이언트와 서버 연결을 요구하기 때문에 응답시간이 증가된다. 또한, 클러스터링된 로컬 서버들의 상태를 고려하여 재설정을 수행하고 있으나 클라이언트와 서버측에 이르는 미들미일 구간은 고려하지 않았다는 문제점을 가지고 있다.

### III. 멀티 사이트 인터넷 환경에서 제안된 서비스 구조

클라이언트의 요구에 따라 지리적으로 분산된 멀티 사이트 인터넷 서비스 환경에서 클라이언트에 대한 서비스 질을 향상시키고, 클라이언트 요청에 대한 응답시간을 줄여 인터넷 서비스 성능을 향상시킬 수 있는 시스템 구조와 서버를 선택하는 방법을 제시하였다.

#### 1. 효율적인 서버 선택 목표

본 논문에서는 멀티 사이트 콘텐츠 서버들 중에 클라이언트를 위한 가장 효율적인 서버를 선택하는 기준은 다음과 같은 목표를 가지고 설정하였다.

- (1) 같은 지역 네트워크 사이트에서 콘텐츠 서버들의 로드 밸런싱은 유지되어야 한다.
- (2) 각 콘텐츠 서버의 처리 능력을 초과하지 않아야 한다.
- (3) 클라이언트의 요구에 빠르게 응답할 수 있어야 한다.

위와 같은 목표를 달성하기 위하여 다음과 같은

정보를 수집하고 분석하였으며, 효율적인 콘텐츠 서버 선택 기준으로 사용되었다.

- (1) 각 콘텐츠 서버의 상태는 TCP 연결 설정수, 디스크 부하, CPU 사용율, 이더넷 이용률 등이 측정되었다. 이는 위에 설정된 목표(1), (2)를 위한 것으로 서버 상태를 위한 측정값은 모두 같은 계수를 갖고 반영하였다.
- (2) 클라이언트와 각 지역 사이트 사이의 네트워크 상태를 알기 위해 RTT(Round Trip Time), 패킷 손실률, 네트워크 트래픽 량이 측정되었다. 이는 위에 설정된 목표(3)을 위한 것으로 서버 선정시 RTT 측정값이 상대적으로 높은 계수를 갖고 반영되었다. RTT 측정 평균값이 실험오차 범위내에서 같을 경우 패킷 손실률과 네트워크 트래픽 량이 반영되도록 하였다.

2. 원형 시스템(Prototype System)의 형태  
본 논문에서 제안하는 시스템의 구성요소는 다음과 같다.

- (1) 콘텐츠 목록 서버
- (2) 콘텐츠 서버
- (3) 네트워크 상태 감시자
- (4) 콘텐츠 서버 관리자

사용자에게 단순히 서비스 목록만을 제공하는 콘텐츠 목록 서버(CIS:Contents Index Server)와 분산된 지역 네트워크에 위치하여 데이터를 저장하고 있는 콘텐츠 서버(CS:Contents Server), 콘텐츠 서버가 위치한 네트워크의 상태를 감시하는 네트워크 상태 감시자(NSP:Network Status Prober), 네트워크 상태 감시자로부터 상태 정보를 전달받아 전체 콘텐츠 서버 및 네트워크의 상태를 감시하는 콘텐츠 서버 관리자(CSM:Contents Server Manager)로 구성되어 있다. 네트워크 상태 감시자는 지역 사이트에 있는 콘텐츠 서버의 상태와 트래픽 정보를 갱신하기 위해 사이트내의 각 콘텐츠 서버와 주기적으로 통신을 한다. 각 콘텐츠 서버는 네트워크 상태 감시자가 요구한 상태정보에 대하여 응답하거나 콘텐츠 서버의 이상 발생 즉시 네트워크 상태 감시자에게 이상 상황을 공지한다. 네트워크 상태 감시자는 이상 발생 공지를 받은 즉시 콘텐츠 서버 관리자에게 그 사실을 알려 다음 클라이언트의 요구에 참조할 수 있도록 한다.

그림 4는 지역적으로 분산된 인터넷 환경에서 클라이언트에 효율적인 서버를 선택하기 위한 시스템

구조 및 동작과정을 보여준다. 클라이언트는 웹 브라우저를 통하여 콘텐츠 목록 서버의 웹 페이지에 접속하여 원하는 콘텐츠 서비스를 요청한다(①). 요청된 콘텐츠 서비스에 대해 최적의 콘텐츠 서비스를 제공할 수 있는 콘텐츠 서버를 발견하는 알고리즘을 수행한다(②,③). 최적의 서버를 선택한 후 콘텐츠 목록 서버는 HTTP-재설정지의 형태로 최적의 콘텐츠 서버 주소를 적당한 상태코드를 포함하여 클라이언트에게 응답한다(④). 클라이언트는 응답 받은 상태코드를 참조하여 콘텐츠 서버 주소로 TCP 재설정하고, 콘텐츠 서버에게 서비스를 요청한다(⑤). 콘텐츠 서버는 해당 콘텐츠를 서비스 응답한다(⑥).

사용자와 콘텐츠 서버가 새로운 접속이 이루어져 서비스가 개시될 때 콘텐츠 서버는 네트워크 상태 감시자에게 현재의 콘텐츠 서버의 상태를 공지한다(⑦). 네트워크 상태 감시자는 해당 콘텐츠 서버의 상태정보를 콘텐츠 서버 관리자에게 중계하여 다음의 클라이언트 요구에 참조할 수 있도록 한다(⑧). 또한, 콘텐츠 서버 관리자는 네트워크 상태 감시자에게 일정 주기로 콘텐츠 서버의 상태정보와 네트워크 트래픽 정보를 요청한다. 네트워크 상태 감시자는 요청된 정보를 콘텐츠 서버 및 네트워크로부터 얻은 후 콘텐츠 서버 관리자에게 보낸다. 콘텐츠 서버 관리자는 네트워크 상태 감시자로부터 모여진 상태정보를 관리 객체 정보로 변환하여 갱신하며 다음의 클라이언트 요구시 참조할 수 있도록 한다.

1) 네트워크 상태 감시자  
(NSP:Network Status Prober)

분산된 지역 사이트의 네트워크 상태정보 및 콘텐츠 서버의 상태를 측정한다. 네트워크 상태 정보(표1)는 각 사이트로 유입되고 유출되는 패킷의 트래픽 양을 측정한다. 이것은 각 지역 사이트의 혼잡도를 측정하기 위함이다. 콘텐츠 서버의 상태 정보

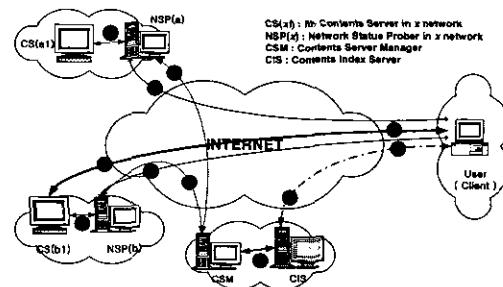


그림 4. 원형(Prototype) 시스템 서비스 구조 및 동작

(표2)를 위해 각 서버로부터 TCP 연결 설정의 수와 디스크 부하량, CPU 사용률을 수집한다<sup>[8]</sup>. 또한, 사용자로부터 서비스 요구시 사이트와 클라이언트 사이의 RTT(Round Trip Time) 값을 실시간으로 측정한다.

하나의 네트워크 안에 다수의 콘텐츠 서버가 위치할 수 있으며, 각 서버들의 로드 밸런싱은 유지되어야 한다. 동일한 네트워크에 위치하는 서버들과 클라이언트 사이의 RTT 값은 같기 때문에 네트워크 상태 감시자에서 측정하도록 하였다. 이는 콘텐츠 서버가 위치한 네트워크와 해당 클라이언트까지의 지연 시간을 측정하여 콘텐츠 서버로부터의 서비스가 지연 제한에 적합한지를 알기 위해서 이다. RTT 값 측정은 클라이언트의 IP주소로 ICMP의 "ping" 명령으로 수행하였으며, 서버와 클라이언트 사이의 이용 가능한 네트워크 대역폭이 고려된 것이므로 5회의 측정 중 최소 값이 사용되었다<sup>[8]</sup>.

네트워크 혼잡도 계산은 네트워크 상태 감시자에서 네트워크 내의 모든 패킷을 모니터링 하여 지속적으로 검사하도록 하였으며, 식(1)은 네트워크 혼잡도 계산식이다.

Network\_traffic(%)

$$\approx \frac{\sum_{t_1}^{t_2} (size\ of\ total\ packets\ on\ network)_i}{\Delta t \times network\ bandwidth} \times 100$$

$$\Delta t = t_2 - t_1 \quad (1)$$

표 1. 네트워크 상태 정보 측정 항목

| 측정항목                           | 측정단위 | 명령어                        | 측정시간     |
|--------------------------------|------|----------------------------|----------|
| RTT                            | ms   | ping                       | dynamic  |
| Packet loss                    | %    | ping                       | dynamic  |
| Transmitting/Receiving traffic | bps  | application packet monitor | periodic |

네트워크 상태 감시자가 각 서버의 상태 정보를 위하여 각 서버에 SNMP 에이전트를 설치하여 서버의 디스크 부하량, CPU 사용률을 주기적으로 수집하였고 각 서버에서 네트워크 인터페이스를 통한 트래픽 양은 일정시간 동안 인터페이스를 통해 유입/유출되는 Octects의 크기와 인터페이스의 전송 속도를 통하여 이더넷 이용률은 SNMP의 ifInOctects, ifOutOctects, sysUpTime, ifSpeed를 이용하여 주기적으로 식(2)에 의하여 계산되었다<sup>[10][12][14]</sup>.

표 2. 콘텐츠 서버 상태 측정 항목

| 측정항목                      | 측정단위  | 명령어  | 측정시간     |
|---------------------------|-------|------|----------|
| Number of TCP connections | count | snmp | dynamic  |
| Disk load                 | 전송수   | snmp | periodic |
| CPU Utilization           | %     | snmp | periodic |
| Ethernet Utilization      | %     | snmp | periodic |

Ethernet Utilization(%)

$$\approx (total\ bits\ sent + total\ bits\ received) / bandwidth$$

$$= \frac{[(ifInOctects_{t_2+t_1} - ifInOctects_t) + (ifOutOctects_{t_2+t_1} - ifOutOctects_t)] \times 8}{(sysUpTime_{t_2+t_1} - sysUpTime_t) \times ifSpeed} \times 100 \quad (2)$$

2) 콘텐츠 서버 관리자(CSM)

콘텐츠 서버 관리자는 네트워크 상태 감시자로부터 상태 정보를 전달받아 전체 콘텐츠 서버 및 네트워크의 상태를 관리하는 역할과 서비스 요구에 대해 최적의 콘텐츠 서버를 선정하여 클라이언트에게 전송해 주는 역할을 한다. 콘텐츠 서버 관리자에는 세 가지 형태의 파일이 존재한다. 첫 번째, 콘텐츠 데이터 정보에 관련된 파일로 콘텐츠 데이터의 파일명, 콘텐츠 파일의 경로, 콘텐츠 파일을 저장하고 있는 콘텐츠 서버들의 IP 주소, 콘텐츠 파일의 크기와 타입에 관한 정보와 해당 콘텐츠 파일의 요청 빈도에 대한 정보를 수록한다. 해당 콘텐츠 파일의 요청 수를 통하여 차후에 콘텐츠 파일의 분산 저장 시 유용한 자료로서 사용되어 질 수 있도록 하였다. 두 번째는 네트워크 상태 정보 파일로 네트워크 상태 감시자의 주소와 그에 속해 있는 콘텐츠 서버들의 주소 및 주기적인 네트워크의 혼잡도 정보를 수록한다. 마지막으로 콘텐츠 서버 정보 파일로 네트워크 상태 감시자로부터 전송된 각 콘텐츠 서버의 상태를 기록하여 클라이언트의 콘텐츠 요구 시 참조하게 된다.

3. 서버 선택 알고리즘

최적의 콘텐츠 서버를 선정하는 알고리즘에는 콘텐츠 정보 파일에서 사용자가 요청한 서비스를 제공할 수 있는 콘텐츠 서버의 IP 주소를 알아내고 각 콘텐츠 서버가 위치한 네트워크 상태 감시자에게 클라이언트까지의 지연시간을 요청하는 것이 포함되어 있다. 콘텐츠 서버의 선정은 2단계 알고리즘에 의해 결정된다. 첫 번째 단계는 측정된 지연시간과 주기적으로 수집된 네트워크 상태 정보를 통하

여 최적의 사이트가 결정된다. 두 번째 단계는 선택된 사이트에 위치한 콘텐츠 서버들의 정보로부터 최적의 콘텐츠 서버가 결정된다.

1) 효과적인 사이트 선택

사이트 선정에 있어 네트워크 상태 값은 현재 네트워크 상태에 대한 인자들에 계수를 곱해서 계산하였으며 식(3)과 같다.

$$\text{네트워크 상태 값} = RTT \times \alpha + PL \times \beta + NSPT \times \gamma$$

- RTT : 클라이언트에서 NSP까지의 RTT
- PL : 패킷손실율
- NSPT : NSP의 트래픽
- $\alpha, \beta, \gamma$  : coefficient

(3)

2) 효과적인 서버 선택

식(3)에 의해서 사이트가 선정이 되면 네트워크 상태 감시자에 의해서 수집된 정보를 가지고 콘텐츠 서버를 선택한다. 콘텐츠 서버의 선택은 서버 로드에 대한 인자에 적당한 계수를 곱해서 계산되었으며 식(4)와 같다.

$$\text{서버 상태 값} = LINK \times \alpha + IO \times \beta + CPU \times \gamma + CST \times \delta$$

- LINK : TCP연결수
- IO : 디스크로드
- CPU : CPU사용량
- CST : 콘텐츠서버의 트래픽
- $\alpha, \beta, \gamma, \delta$  : coefficient

(4)

4. 제안 시스템의 동작

본 논문에서 클라이언트의 위치에 따라 사용자에게 효율적인 서버를 제공하기 위한 제안된 시스템의 동작은 그림 5와 같으며, 각 시스템의 동작은 아래와 같다.

1) 네트워크 상태 감시자(NSP)

네트워크 상태 감시자에 있는 SNMP 매니저는 각 네트워크 내의 모든 콘텐츠 서버에 있는 SNMP 에이전트에게 Crontab 파일에 지정한 시간마다 컨

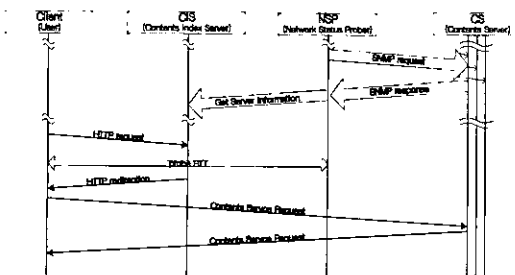


그림 5. 제안 시스템의 동작

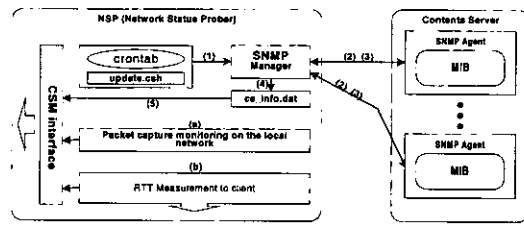


그림 6. 네트워크 상태 감시자의 동작 과정

텐츠 서버의 특정 객체 정보를 요구한다. SNMP 에이전트는 SNMP 매니저의 특정 객체(OID)에 대한 요청 값을 리턴 한다. SNMP 매니저는 리턴된 정보들을 적용하여 계산하며, 트래픽 정보 필드에 리턴된 값을 업데이트한다. 콘텐츠 서버의 실행능력을 나타내는 다른 값들도 업데이트한다. 네트워크 상태 감시자의 동작 과정은 그림 6과 같다.

- (1) 주기적으로 crontab에서 쉘 프로그램 실행.
- (2) SNMP 에이전트에게 서버의 상태 정보 요청.
- (3) SNMP 에이전트로부터 서버의 상태 정보 응답.
- (4) 콘텐츠 서버들의 상태 정보 저장.
- (5) 콘텐츠 서버 관리자로 공지.
  - (a) 네트워크의 트래픽 상황 감지
  - (b) 클라이언트와의 RTT 값 측정

2) 콘텐츠 서버 관리자(CSM)

콘텐츠 서버 관리자의 동작은 그림 7과 같고 동작과정은 아래와 같다.

- (1) 콘텐츠 목록 서버로부터 사용자 요구 콘텐츠 항목 수신
- (2) 콘텐츠 정보 파일(contents\_info.dat)로부터 해당 콘텐츠에 대한 경로 및 콘텐츠 서버 목록 읽기
- (3) 네트워크 상태 정보 파일(network\_status\_info.dat)로부터 콘텐츠 서버의 네트워크 상태 감시자 선정
- (4) 선정된 네트워크 상태 감시자에게 클라이언트로 RTT 측정 수행

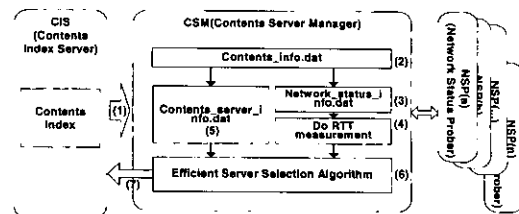


그림 7. 콘텐츠 서버 관리자의 동작 과정

- (5) 주기적으로 갱신된 콘텐츠 서버 정보 파일 (contents\_server\_info.dat) 읽기 수행
- (6) 효율적인 콘텐츠 서버 선택 알고리즘 수행
- (7) 선택된 콘텐츠 서버 전송

#### IV. 실험환경 및 구현

콘텐츠 서버 관리자의 운영체제는 SUN OS 5.7 버전을 사용하였으며, 웹 서버 대문은 아파치 를 사용하였다. 서로 다른 위치에 있는 3개의 망(A, B, C)에 네트워크 상태 감시자와 각각의 콘텐츠 서버 를 설치하여 실험하였다.

SNMP 에이전트는 Linux 환경에서 CMU-SNMP 보다 기능이 확장된 UCD(University California at Davis)-SNMP Agent를 이용하였고<sup>[6]</sup>, Windows 환경에서는 확장 SNMP agent인 perfmib.dll를 사용하여 구성하였다<sup>[7]</sup>. 네트워크 상태 감시자의 SNMP 매니저는 응용 프로그램으로 구현하여 Crontab에서 매 5분마다 실행되도록 하였으며, 콘텐츠 서버에 탑재된 SNMP 에이전트로부터 OID 정보를 받아 갱신하도록 하였다<sup>[4][13][14]</sup>. 또한, 지역 네트워크의 트래픽은 네트워크 상태 감시자에서 패킷 모니터 응용프로그램을 구현하여 모든 패킷의 헤더정보를 검사하여 측정하였다. 네트워크 상태 감시자에서 측정된 항목은 모두 콘텐츠 서버 관리자로 전송하여 서버 선택 알고리즘에 참조되도록 하였다.

클라이언트는 미리 콘텐츠 목록 서버의 주소를 알고 있어야 하며, 콘텐츠 목록 서버에 접속한 클라이언트는 자신이 서비스 받고자 원하는 콘텐츠 목록의 파일 이름이나 이미지를 클릭하기만 함으로서 투명하게 클라이언트의 위치에서 지속적인 서비스가 가능한 콘텐츠 서버들 중에 최적의 콘텐츠 서버에 접속하여 원활한 서비스를 받을 수 있다. 클라이언트에서 요청된 콘텐츠 서비스 요구 항목은 콘텐츠 목록 서버에 작성된 PHP<sup>[11]</sup> 코드로 실행되도록 하였으며, 콘텐츠 서버 관리자에서 요청된 콘텐츠 서비스 요구 정보를 파일로 저장하여 기록하도록 하였다. 콘텐츠 서버 관리자는 콘텐츠 서버들의 상태 및 트래픽 정보를 가지고 있는 네트워크 상태 감시자와의 상호 작용으로 클라이언트의 서비스 요청이 들어오면 2단계 서버 선정 알고리즘을 통하여 클라이언트의 위치에 가장 적합한 콘텐츠 서버 IP주소를 선정한다. 최적의 콘텐츠 서버 IP주소와 콘텐츠 서비스가 사용하는 프로토콜, 요청한 콘텐츠 파일명을 메타파일에 포함하여 클라이언트로 HTTP 재 설

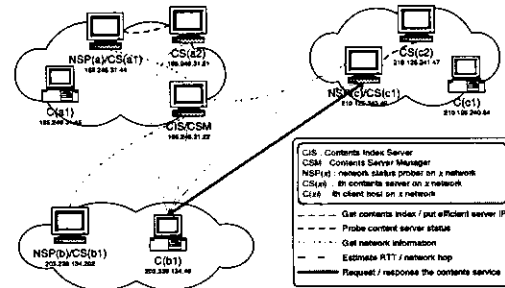
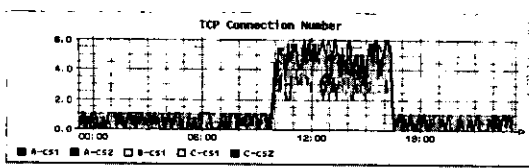


그림 8. 제안 시스템 실험 환경

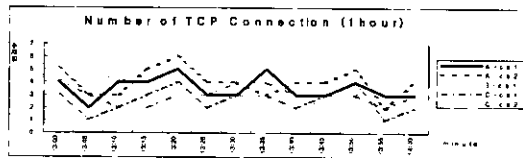
정(리턴코드 302)의 형태로 전송되도록 구현하였다. 그림8은 제안 시스템의 실험 환경을 보여주고 있다.

#### V. 실험 결과

지역적으로 분산된 사이트에서 콘텐츠 서버의 부하 분산을 보이기 위해 매5분마다 각 콘텐츠 서버의 TCP 연결 수를 RRDtool(Round Robin Database tool)<sup>[9]</sup>을 사용하여 그래프로 나타내었다. MRTG가 단일 시스템에 적용되는 틀이라고 한다면, RRDtool은 멀티 시스템의 연속적인 데이터(예 네트워크 대역폭, 기계실 온도, 서버의 부하)를 저장하고 그래프로 보여주는 틀이다. 그림 9는 (a)24시간과 (b)1시간 동안의 분산된 각 콘텐츠 서버의 TCP 연결 수를 보여주고 있으며, 지역 네트워크 안에서 분산되어 동작되고 있음을 보여주고 있다. 그림 9의 (b)에서 “A-cs1”과 “A-cs2”의 의미는 네트워크 A에 콘텐츠 서버가 2개로 클러스터링 되어 있음을 나타내고, 네트워크 B에는 1개, 네트워크 C에는 2개가 클러스터링 되어 동작됨을 나타낸다. 여기서 클라이언트의 위치에 따라 지역적으로 분산된 네트워크 A, B, C로 구분되고, 각 네트워크 안에 클러스터링된 서버간에 TCP 연결이 고르게 분포되어 부하 분산이 이루어 짐을 확인하였다. 그림 10은 콘텐츠 서버 관리자에서 클라이언트의 서비스 요청에 대해 서비스할 수 있는 콘텐츠 서버들 가운데 최적의 콘텐츠 서버가 선택되는 과정을 기록한 로그 파일을 보여주고 있다. 그림 10에서 “IP = 165.246.31.22”는 클라이언트의 위치를 나타내고, “Count NSP = 3”은 사용자가 선택한 서비스에 대해 서비스 가능한 서버가 위치한 지역적으로 분산된 네트워크의 수가 3개 임을 의미한다. 또한, “Count CAS = 2”는 선택된 네트워크에서 서비스 가능한 서버의 수가 2개 임을 나타낸다. 결과 로그 파일에서 선택된 네트워크(0)의 RTT 측정값은 0.7 msec이고, 최고



(a) TCP 연결 수(24 시간)



(b) TCP 연결 수(1 시간)

그림 9. 콘텐츠 서버의 TCP 연결수

```

.....
IP = 165.246.31.22 Date/Time = 02/18/01 13:55:35
Request File = "musicvideo_mv_001_kb_002"

Count MSP = 3
IP : [RTT] [NSPH] [PL] [NSPT] [NetworkStatusValue]
(0) IP="165.246.31.44" = 0.7, 1, 0, 3.4 = 15.7
(1) IP="203.238.134.200" = 3.8, 10, 0, 6.2 = 57.6
(2) IP="210.129.243.45" = 5.8, 14, 0, 2.5 = 84.5

Select MSP = http://165.246.31.44

Count CAS = 2
IP : [LINK] [IO] [CPU] [CST] [ServerStatusValue]
(0) IP="165.246.31.22" = 4, 0.5, 0.12, 5.2 = 36.22
(1) IP="165.246.31.45" = 3, 0.4, 0.13, 4.8 = 39.93

Select CAS = http://165.246.31.45/musicvideo_mv_001_kb_002.png
.....
    
```

그림 10. 결과 로그 파일

늦은 네트워크(3)의 RTT 측정 값은 5.8 msec 이다. 제안한 시스템에서 사용자에게 서비스되는 콘텐츠의 서비스 속도는 사용자와 선택된 네트워크 사이의 트래픽 변화에 따라 지속적인 서비스를 유지할 수는 없지만, RTT 측정값의 결과만으로도 약 9 배의 성능 향상을 가져올 수 있음을 볼 수 있다. 또한, 본 논문에서 제안한 네트워크 상태 값 계산의 결과는 약 4 배의 성능 향상을 가져올 수 있었다.

### VI. 결론

인터넷은 정보화 시대에 필수적인 요소로 인식되고 있으며, 인터넷 서비스 사용자의 수와 서비스 품질의 기대치는 기하급수적으로 증가하고 있다. 안정화된 인터넷 서비스를 제공하기 위한 노력의 일환으로 패일오버에 중점을 둔 클러스터링 기술에서부터 하나의 클러스터 안에 여러 대의 서버를 배치하여 부하를 분산시키는 로드 밸런싱에 관심을 두고 있다. 본 논문에서 지역적으로 분산된 멀티 사이트 인터넷 환경에서 로드 밸런싱을 유지하면서 각 클라이언트의 위치에 따라 가장 효율적인 서비스를

제공할 수 있는 서버를 선택하는 방안을 제안하였다. 클라이언트에게 효율적인 서버를 제공하기 위해 클라이언트와 서버 사이의 네트워크 상태와 지리적으로 분산된 서버들의 성능 상태를 고려하였다. 클라이언트에게 가장 효율적인 서버를 선택하기 위해 클라이언트와 각 사이트에 대한 지연시간과 주기적으로 수집된 네트워크 상태 정보(RTT, Packet loss, 네트워크 혼잡도)로 클라이언트에게 서비스 가능한 최적의 사이트를 선택하고, 선택된 사이트에 위치한 콘텐츠 서버들의 상태 정보(디스크 부하량, CPU 사용률, 트래픽 량, TCP 연결 설정 수)로 클라이언트의 서비스 이용 관점에서 지속적인 서비스가 제공될 수 있는 콘텐츠 서버가 결정되는 2단계 서버 선택 알고리즘을 사용하였다. 로컬 네트워크 서버 클러스터링에서 각 서버들 사이의 로드 밸런싱이 유지되도록 하여 서비스 신뢰도 및 확장성을 제공하였고, 실험을 통하여 가능성을 검증하였다. 실험 결과 로그 파일에서 RTT 측정값의 결과는 약 9 배의 성능 향상과 제안한 시스템의 네트워크 상태 값 결과로는 약 4 배의 성능 향상이 있었음을 보였다. 본 논문에서 시험적인 환경으로 3개의 네트워크에 각 콘텐츠 서버를 설치하여 실험을 하였고, 콘텐츠 서버가 설치된 동일한 네트워크내의 클라이언트에서 시험을 한 결과를 제시하였으나, 향후에 접속빈도가 높은 콘텐츠에 대하여 지역적인 네트워크를 구성하여 일반적인 성능을 평가하는 연구가 필요하다.

### 참고 문헌

- [1] T. Brisco. *DNS Support for load balancing*, RFC 1794, April 1995.
- [2] V. Cardellini, M. Colajanni, Philip S. Yu, "Dynamic Load Balancing on Web-server Systems," *IEEE Internet Computing*, Vol. 3, No. 3, May/June 1999.
- [3] G.D.H. Hunt et al., "Network Dispatcher: A Connection Router for Scalable Internet Services," *Computer Networks and ISDN Systems*, Vol. 30 N.1-7, Apr. 1998.
- [4] Stevens, W. R., *UNIX NETWORK PROGRAMMING*, FRENCH HALL SOFTWARE SERIES. p258~340. 1996.
- [5] Daniel Andesen, Tao Yang, Vegard Holmedahl, Oscar H. Ibarra, "SWEB : Towards a Scalable World Wide Web Server



on Multicomputers," Proceedings of International Conference on Parallel Processing, pp. 15-19, April 1996.

[6] UCD-SNMP, <http://ucd-snmp.ucdavis.edu>.

[7] Monitoring Windows 2000 with SNMP, <http://snmpboy.rte.microsoft.com/asp/perfmibhack.asp>.

[8] Robert L. Carter, Mark E. Crovella, "Server Selection Using Dynamic Path Characterization in Wide-Area Networks," Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Vol. 3, p1014~1021, Apr, 1997.

[9] RRD TOOL, <http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool>.

[10] Jon-Tae Park, James W.Hong, Sung-Uk Park and Young-Min Kang, "Enterprise Network Traffic Monitoring, Analysis, and Reporting Using Web Technology," Journal of Network and Systems Management, Vol. 9, p89~111, Mar, 2001.

[11] PHP:Hypertext Preprocessor, <http://www.php.net>.

[12] Gilbert Held, *LAN Management with SNMP and RMON*, Wiley Computer publishing, p27~101. 1996.

[13] Gail Anderson, Paul Anderson, *The UNIX C Shell Field Guide*, Prentice-Hall, p267~336. 1992.

[14] W. Richard Stevens, *TCP/IP Illustrated, Volume1: The Protocols*. Addison-Wesley Publishing Company, p359~388. 1995.

이 현 표(Hyun-Pyo Lee)

정회원



1988년: 인하대학교  
전자계산학과(학사)  
1990년: 인하대학교  
전자계산학과(석사)  
1989년~1995년: (주)나우정밀  
중앙연구소  
1996년~현재: 안산1대학  
인터넷상거래과 조교수  
1995년~현재: 인하대학교 전자계산공학과 박사과정  
<주관심 분야> 네트워크 관리, CDN, SLA, 무선통신

이 균 하(Kyoon-Ha Lee)

중신회원



1970년: 인하대학교 전기공학과  
(학사)  
1976년: 인하대학교 전자공학과  
(석사)  
1981년: 인하대학교 전자공학과  
(박사)  
1997년~1981년: 광운대학교 전자계산공학과 조교수  
1981년~현재: 인하대학교 전자계산공학과 교수  
<주관심 분야> 지능통신망, 한국어 정보처리, 패턴  
인식