

대용량 콘텐츠 전송을 위한 병렬전송 오버레이 멀티캐스트

(A Parallel Transmission Overlay Multicast Scheme for
Massive Contents Delivery)

박진홍[†] 윤미연^{**} 김선호^{***} 신용태^{****} 신석규^{*****}
(Jinhong Park) (Miyoon Yoon) (Seonho Kim) (Yongtae Shin) (Seokkyoo Shin)

요약 오버레이 멀티캐스트 전송방식은 멀티캐스트의 기능을 응용계층에 두어 인터넷상에서 멀티캐스트 전송이 가능한 새로운 방식이다. 그러나 오버레이 멀티캐스트 프로토콜들은 아직까지 표준화가 이루어지지 못하였고 대용량 콘텐츠 전송 시 많은 제약사항으로 인해 적용이 쉽지 않은 형편이다. 그러므로 오버레이 멀티캐스트 기반에 대용량 콘텐츠를 전송할 수 있는 새로운 전송기법의 연구가 요구되고 있다. 본 연구에서는 공통의 파일을 수신하고자 하는 호스트간의 그룹을 설정하여 기존 오버레이 멀티캐스트 기법의 그룹관리와 전송관리를 분리하여, 강건한 그룹관리가 가능한 프로토콜 스택을 구현하고, 그룹 멤버들간의 분산 다운로드 전송기법의 도입을 통해 기존 데이터 릴레이 기반의 오버레이 멀티캐스트 방식보다 빠른 병렬적 전송기법을 제안하여 전송속도 대비 대역폭 소모율에 대한 전송효율을 개선하였다.

키워드 : 오버레이 멀티캐스트, 분산 다운로드, Working Set

Abstract Overlay multicast delivery method is a new approach in which multicast functionality is implemented at the end-hosts application layer in the timing of sparse deployment of IP multicast. However, existing overlay multicast protocols are not being standardization and many restrictions occur when delivering high capacity contents. Therefore, new delivery mechanism is required for the overlay multicast based high capacity contents delivery. In this paper, we separate group management and delivery management of overlay multicast and describe a capable group management. We also defined high speed delivery method better than that of existing overlay multicast through use of collaborated distribute downloading. This improved efficiency of massive contents transmission.

Key words : Overlay Multicast, Distribute Download, Working Set

1. 서론

멀티캐스트 통신은 유니캐스트 전송과는 달리 하나의 메시지 전송으로 그룹에 속한 모든 멤버들에게 메시지 전송이 가능한 효율적인 전송방식이다. 그러나 이러한

멀티캐스트의 도입은 라우터에서 멀티캐스트 기능을 지원할 수 있도록 하드웨어적인 수정이 불가피하다. 따라서 IP 멀티캐스트는 이상적인 효율성에도 불구하고, 범용적인 적용이 이루어지지 못하였다.

응용계층 또는 오버레이 멀티캐스트라 불리는 새로운 멀티캐스트 기법은 IP 멀티캐스트에서 라우터가 담당하던 메시지 복제 및 분배의 기능을 그룹에 속한 각 노드 자체에서 수행하는 형태로 동작한다. 비록 기존 IP 멀티캐스트의 이상적인 효율에는 미치지 못하지만, 멀티캐스트를 범용 적으로 적용시킬 수 있는 방식으로 최근 각광받고 있다. 현재 응용계층 멀티캐스트 프로토콜에 관한 다양한 연구가 진행 중에 있으나, 현재까지 표준으로 정해지지 못한 상태이며, 다양한 애플리케이션을 위한 여러 프로토콜들이 제안되고 있다.

인터넷상에서 대용량의 콘텐츠를 전송하는데 있어서 고려해야 할 문제는 첫째, 대용량 콘텐츠 전송 시 많은

· 본 연구는 한국과학재단 특장기초(과제번호 : R01-2001-000-00362-0(2003)) 연구비 지원에 의해 수행되었음

† 비 회 원 : 한국정보통신기술협회 SW시험인증센터 연구원
elzk@tta.or.kr

** 비 회 원 : 한국정보보호진흥원 정보보호기술단 연구원
myoon@kisa.or.kr

*** 비 회 원 : 서울시소방방재본부 전산개발팀 전문위원
shkim2005@fire.seoul.kr

**** 종신회원 : 숭실대학교 컴퓨터학과 교수
shin@comp.ssu.ac.kr

***** 비 회 원 : 한국정보통신기술협회 SW시험인증센터 센터장
skshin@tta.or.kr

논문접수 : 2004년 6월 1일

심사완료 : 2005년 5월 14일

대역폭 소모와 전송 오버헤드를 해결할 수 있는 효율적인 전송 메커니즘, 둘째, 전송에 따르는 신뢰성 보장 문제 등을 들 수 있다고 선행 연구에서 밝히고 있다[1,2].

효율적인 전송기법을 위한 방안으로 오버레이 멀티캐스트 프로토콜이 사용된다. 가장 대표적인 프로토콜로는 그룹관리를 위한 메시지를 먼저 생성하는 메시지 우선기반 프로토콜인 NARADA 프로토콜[3], 직접 전송 트리를 먼저 생성하는 트리우선기반 프로토콜인 YOID 프로토콜[4,6], 확장성을 제고한 메시지 트리 혼합기반 프로토콜인 NICE 프로토콜[5] 등이 있다. 이러한 프로토콜들[3-6]은 단순히 멤버간의 데이터 전송을 위해 멀티캐스트 트리를 전송하는 기법들을 제안하고 있다. 대용량 콘텐츠를 효과적으로 전송하려면 특정 링크의 대역폭의 소모가 심하지 않도록, 또한 파일 전송 시간이 길어지지 않도록 해야 한다. 그러나 기존 연구에서 제안하고 있는 단순한 릴레이 기반의 멀티캐스트 트리를 이용한 전송기법을 대용량 콘텐츠 전송에 그대로 적용한다면 대역폭의 소모도 심할 뿐 아니라, 전송 트리의 마지막 노드가 콘텐츠를 모두 수신하기까지의 시간이 길어질 수밖에 없다. 따라서, 대용량 콘텐츠 전송에 적합한 개선된 전송기법이 필요하다.

본 연구에서는 공통의 파일을 수신하고자 하는 호스트간의 그룹을 설정하여 효율적인 전송메커니즘을 위해 기존 프로토콜을 응용한 적용성 있는 오버레이 멀티캐스트 인프라 기반에, 오버레이 멀티캐스트 그룹 멤버들 간의 개선된 P2P형 분산 다운로드 기법을 적용하여 병렬적 전송기법을 제안하였다. 효과적인 인코딩 기법으로 많이 연구되고 있는 Digital Fountain 접근방식의 인코딩 기법[7]을 적용함으로써 콘텐츠 유사성(Resemblance) 검사와 함께 어느 정도의 신뢰성을 보장하도록 하였다. 궁극적으로 기존 P2P 기반 분산 다운로드의 대역폭 소모를 방지하기 위해 유효 Working Set의 적절한 규모를 정의하여 확장성 있는 동적 그룹관리 및 대역폭을 소모를 고려한 빠른 전송기법을 제안하도록 한다.

2. 관련연구

기존의 멀티캐스트 데이터 전송을 위한 오버레이 멀티캐스트 기법에 대해 간략히 알아보고, 각 기법의 장단점을 기술한다. 각 기법을 대용량 콘텐츠 전송에 적용했을 때의 한계점을 분석하도록 한다.

2.1 응용계층 멀티캐스트

IP 멀티캐스트의 여러 가지 문제로 인해 멀티캐스트의 기능을 응용계층으로 올린 응용계층 멀티캐스트에 대한 연구가 활발히 진행되고 있다. 응용계층 멀티캐스트는 IP 멀티캐스트 상에서 라우터가 수행하는 기능을 각 종단 호스트에서 수행하도록 하여 어떠한 하드웨어의 변경 없이 멀티캐스트 전송이 가능한 기법이다. 논리적으로

종단 호스트들은 오버레이 네트워크를 형성하고, 응용계층 멀티캐스트는 이러한 오버레이를 효율적인 전송을 위하여 구축하고 관리한다.

기본적인 응용계층 멀티캐스트의 동작방식은 그림 1과 같다. 첫 번째 그림의 IP 멀티캐스트가 각 라우터에서 패킷을 복제하는 것과는 달리 나머지 응용계층 멀티캐스트들은 각 종단 호스트에서 데이터의 복제 및 전송을 수행하고 있음을 알 수 있다.

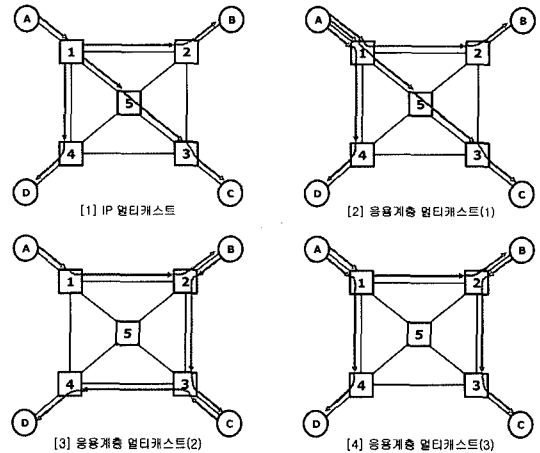


그림 1 응용계층 멀티캐스트 기본 토폴로지

2.1.1 NARADA 응용계층 프로토콜[3]

NARADA 프로토콜은 데이터 전송 트리를 구성하기 전에 각 노드 간 메시지를 구성하는 대표적인 메시지 우선기반 응용계층 멀티캐스트 프로토콜이다. 멀티캐스트 그룹에 참여하는 모든 노드들에 대해 모든 노드들이 연결되어 있는 메시지를 형성하며, 생성된 메시지는 가입 탈퇴 절차를 통해 형성되고 세분화된다.

2.1.1.1 전송관리

NARADA 프로토콜의 트리 생성은 메시지 생성 후 RSP (Reverse Shortest Path) 트리를 생성하는 2단계 절차를 갖는다. 멤버들은 메시지 상에서 거리벡터 프로토콜을 이용하여 다른 멤버들의 경로와 라우팅 값을 함께 유지하며, 이러한 정보를 바탕으로 RSP 경로를 측정하여 트리를 구축한다. 이때 거리 벡터를 측정하는 메트릭으로 지연시간이 사용된다. 구축된 트리는 전송경로를 결정지며 트리구축을 통해 전송관리가 수행된다.

2.1.1.2 그룹관리

메시지를 통한 NARADA 프로토콜의 그룹관리는 멤버의 가입 및 탈퇴 과정을 반복하며 수행된다. NARADA 프로토콜 그룹 내에서 멤버가 탈퇴하고자 할 경우 탈퇴를 원하는 멤버는 이웃 노드들에게 탈퇴 메시지를 보냄으로써 노드정보를 변경하도록 한다. 메시지 상의 임의의

노드가 결함 발생시 인접 이웃 노드들은 결함이 발생한 노드의 주기적인 갱신 메시지를 받지 못하게 된다. 이러한 경우 인접 노드들은 결함 발생노드에게 상태 확인 신호를 보내고, 상태 확인신호에도 결함 노드가 응답하지 않을 경우 해당 노드가 더 이상 동작하지 않는 것으로 간주한다. 즉시 다른 이웃노드들에게 이러한 메시지 변화 정보를 전달하여 메시지의 상태를 변경하여 강건한 그룹관리가 가능하도록 한다.

2.1.2 YOIX(Your Own Internet Distribution) 프로토콜[4,6]

YOIX 프로토콜은 직접적으로 데이터 전송트리를 구축하는 트리우선 방식의 대표적인 프로토콜이다. YOIX는 곧바로 데이터 전송 트리를 생성하므로 트리 생성에 대한 여러 조건들을 직접 제어해야한다. 직접적으로 제어해야 하는 조건들로는 트리 이웃 선택, 자식노드 허용임계 값 등이 있다.

2.1.3 NICE(Nice Is Cooperative Environment) 프로토콜[5]

NICE 프로토콜은 인터넷상에서 확장성 있는 분산 응용을 위한 프레임워크로서 메시지 우선 방식과 트리 우선 방식 두 가지 범주에 속하지 않는 혼합형 응용계층 멀티캐스트 프로토콜이다. 평면상의 모든 호스트들을 일정 규모로 클러스터링 하며 각 클러스터의 리더들로 다시 클러스터링 하여 계층적 구조를 가진다. 멤버들은 자신이 속한 클러스터 내에서 클러스터 리더의 정보만을 유지함으로써 인터넷 규모의 확장성을 갖는다.

2.1.4 대용량 콘텐츠 전송시의 한계점 분석

각 프로토콜의 특성을 살펴보면 현존하는 응용계층 프로토콜들이 많은 제약을 가지며, 그에 따라 특정 응용에만 적합함을 알 수 있다. NARADA 프로토콜은 강건한 그룹관리와 실시간 전송을 지원하는 반면 그룹의 크기가 증가할 경우 심각한 성능저하가 예상되고[3], YOIX 프로토콜의 경우 상대적으로 적은 정보만을 유지하지만 대역폭 보장이 어렵고 실시간 전송 또한 적합하지 않아 범용적인 사용이 어려움을 알 수 있으며[4], NICE 프로토콜은 그룹관리의 확장성을 제고하여 인터넷 환경에서 사용을 유도하였으나 역시 대역폭을 보장하지 못하므로 사용상의 제한이 따름을 알 수 있다[5]. 따라서 보다 범용적이고 통합적인 대용량 콘텐츠 전송 방식에 대한 연구가 필요함을 인지할 수 있다.

2.2 Parallel 다운로드

인터넷상에서 과부하의 서버, 병목 구간을 갖는 회선 등의 조건은 대용량 콘텐츠를 다운로드함에 있어서 큰 장애요소로 알려져 왔다[9,11]. 단일 서버로부터의 다운로드 방식은 이러한 장애요소를 극복하는데 큰 어려움을 갖는다. 최근 인터넷상에서 파일을 다운로드 하는 기법 중 Parallel 다운로드 기법이 많은 애플리케이션에 적용되어 사용되고 있다[9-12]. 콘텐츠 분산 네트워크와 P2P

네트워크가 널리 확산되어있는 현 시점에서 Parallel 다운로드는 단일 서버 다운로드보다 더 좋은 성능을 보장함을 선행 연구 사례에서 나타내고 있다[11].

이러한 Parallel 다운로드의 기본 스킴은 인터넷상에 혼재하는 동일 콘텐츠를 가지는 여러 복사본 서버로부터 다중 전송받음으로 인해 보다 빠른 전송시간을 보장하는 것이다.

본 논문에서도 대용량 콘텐츠의 빠른 전송을 위한 협동적 분산 다운로드를 적용한 전송기법을 통해 콘텐츠 전송 성능을 보장하도록 하였다.

2.2.1 협동적 분산 다운로드 기법

응용계층 멀티캐스트 트리를 사용한 전송방식은 콘텐츠 수신을 원하는 멤버들 간의 수신시간 및 수신용량의 현저한 차이를 보인다.

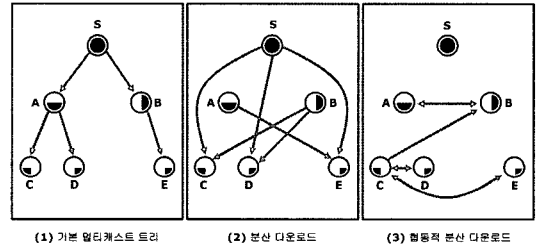


그림 2 콘텐츠 다운로드 기법

그림 2는 세 가지 콘텐츠 다운로드 기법의 예를 나타내고 있다[1]. 첫 번째 기본 멀티캐스트 트리를 따라 콘텐츠 전송이 이루어질 경우 A, B 멤버가 수신하는 콘텐츠의 양과 C, D, E 멤버가 수신하는 콘텐츠의 양은 A, B 두 멤버가 수신과 동시에 중계를 한다 하여도 시간적인 차이와 용량상의 차이를 발생시키게 된다.

기본 멀티캐스트 트리에 추가적으로 연결을 보강한 분산 다운로드 기법은 기본적인 멀티캐스트 트리 상에서 다중 소스로부터 전송 받음으로써 어느 정도의 시간차 및 용량상의 차이를 극복할 수 있다. 그러나 분산 다운로드 또한 기본적인 멀티캐스트 트리를 따라 하향식으로 전송하기 때문에 개선의 여지가 남아있는 상태이다.

협동적 분산 다운로드 방식은 분산 다운로드 방식에 부가적으로 수평적인 연결을 추가하여 기본적인 멀티캐스트 전송트리에 영향을 받지 않으며, 각 멤버들 간 자유로운 전송이 이루어지기 때문에 보다 빠르고 동일하게 전송받을 수 있는 장점을 가진다.

3. 제안하는 대용량 콘텐츠 전송기법

3.1 NARADA 기반 그룹관리

모든 응용계층 멀티캐스트 프로토콜은 메시지, 트리 형성을 통한 그룹관리와 전송관리로 그 기능이 분리되어진다. 메시지 우선 방식인 NARADA 프로토콜은 메시지 구축

을 통해 그룹관리를 실시하며, 그룹에 가입되어있는 모든 호스트들은 모든 그룹멤버들의 상태정보를 유지하고 주기적인 갱신을 실시한다. 이와 같은 형태의 동작방식으로 인해 새로운 멤버의 가입이나 멤버의 탈퇴 또는 장애나 고장으로 인한 메시지의 분리 등의 상황에서 즉각적이고 동적인 그룹관리가 가능하다.

3.2 협동적 분산 다운로드 기반 전송관리

3.2.1 협동적 분산 다운로드 고려요소

(1) Working Set 정의

임의의 파일 f 의 일부를 가지는 호스트들을 파일 f 에 대한 Working Set 이라 정의한다[1].

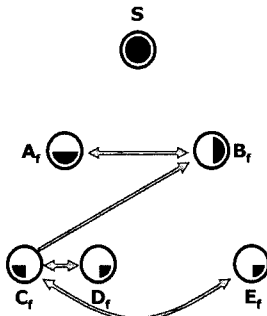


그림 3 Working Set 형성

그림 3에서 보이는 바와 같이 온전한 검은색 원이 파일 f 를 의미하며, A, B, C, D, E 호스트는 각각 파일 f 의 일부분을 가지고 있다. 이때 각 호스트 A, B, C, D, E를 파일 f 의 Working Set으로 정의하고 A_f, B_f, C_f, D_f, E_f 로 표현한다.

(2) 유사성 검사

동일한 콘텐츠의 일부분을 가지는 Working Set 내에서 협동적 분산 다운로드가 실행되기 위해서는 Working Set 내에서의 규제가 필요하다. 두 Working Set A_f, B_f 가 가지는 파일 f 의 부분이 동일한 경우 A와 B는 콘텐츠 전송 시 서로에게 전혀 도움이 되지 못하기 때문이다. 따라서 콘텐츠 전송 전에 콘텐츠 유사성을 검사하여 보유하고 있는 콘텐츠의 부분이 일치하지 않는 멤버들 간의 전송을 유도하도록 한다.

두 Working Set A_f, B_f 의 유사성은 수식 (1)과 같이 표현될 수 있다[8].

$$\frac{A_f \cap B_f}{A_f \cup B_f} \quad (1)$$

이 값은 0에서 1 사이의 값으로 표현되며 그 값이 0에 가까울수록 파일 f 에 대해 서로 중복되는 부분이 적음을 나타낸다. 이것은 그만큼 두 Working Set간 콘텐츠를 교환할 경우 서로에게 많은 이익이 됨을 뜻한다. 반대로

그 값이 1에 가까울 경우 두 Working Set은 파일 f 의 많은 부분을 중복되게 가지고 있음을 의미한다. 이러한 경우에는 서로에게 유익한 부분이 매우 적음을 알 수 있다. 하지만 이러한 경우에도 적은 부분이지만 서로에게 유익한 파일의 일부를 가질 수 있기 때문에 간과할 수 없다.

본 연구에서는 유사성 검사를 실시하여 유사성이 완전히 중복되는 형태, 즉 1값이 나올 경우 해당 Working Set간의 파일교환 대상에서 제외시키고 1 외의 어떠한 값이 나온다면, 유사성의 높고 낮음에 관계없이 콘텐츠 교환을 실시하도록 가정한다.

유사성 검사 방법으로는 선행연구에 사용된 Random Sampling에 의한 key 정보 교환 기법을 사용하였다[1,13]. 온전한 콘텐츠인 파일 f 를 균등한 크기로 분할하여 각 조각을 식별하기 위한 고유의 정수형 key를 부여하도록 한다. Random Sampling 기법을 통해 각 멤버들은 자신의 Key 정보의 샘플을 추출하여 서로의 key값 비교를 통해 유사성을 검사하도록 한다. Working Set의 각 멤버들은 이러한 key 값을 비교하여 동일하지 않는 key 정보가 발견 될 경우 일치하지 않는 부분을 가지는 것으로 간주하여 콘텐츠 전송을 시도하도록 한다.

본 연구에서는 유사성 검사 시 일치, 불일치 2가지 요소만을 검사하여 유사성 검사의 정밀도를 낮추고, Sampling된 key값을 비교하여 유사성 검사 오버헤드 발생을 줄이도록 하였다.

(3) 콘텐츠 인코딩 기법

멀티캐스트 상에서 신뢰적인 전송을 보장하기 위한 다양한 연구가 진행되어왔다. NACK 억제 방식, 트리 기반 방식, FEC 기반 방식이 가장 대표적인 신뢰적 멀티캐스트 접근 기법들이라 할 수 있다. 본 연구에서는 콘텐츠의 유사성을 비교하고, 전송상의 신뢰성을 추구하기 위하여 FEC 기반 인코딩 기법을 사용하여 콘텐츠를 인코딩 한 후 전송하도록 가정하였다. 본 연구에서는 선행연구 결과인 FEC 기반방식의 Digital Fountain[7] 인코딩 기법을 적용하도록 하였다.

3.2.2 Working Set의 범위

본 연구에서 제안하는 협동적 분산 다운로드기법은 동일한 파일을 가진 멤버가 많을수록 콘텐츠 다운로드에 걸리는 시간이 적게 소모된다. 이러한 전송방식은 현재 다수 P2P기반 응용에서 많이 사용되고 있으나 이러한 기법의 사용은 많은 대역폭 소모를 유발하게 된다. 본 연구에서 제안하는 전송기법에서는 대역폭 소모를 줄이면서 기존의 전송방식보다 빠른 속도를 보장할 수 있도록 협동적 분산 다운로드의 유효 Working Set의 범위를 정의하고자 한다.

콘텐츠 교환을 원하는 호스트는 교환의 대상을 필요로 한다. 파일 f 의 다운로드를 원하는 호스트는 각자 유효한

Working Set의 범위를 가질 필요가 있다. Working Set의 범위 지정은 홉(hop)카운트 메트릭 기반방식과 지연시간 메트릭 기반방식으로 구분할 수 있으나, 본 연구에서는 두 가지 방식을 동시에 고려한 다중 메트릭 기법을 적용하여 두 메트릭이 가지는 단점을 상쇄하도록 하였다.

홉 카운트(hopcount) 기반의 검색에서 고려할 사항은 1 홉이 최적의 서비스를 의미하지 않는다는 점이다. 1 홉 내의 멤버 중 심각한 오버로드를 가지는 호스트가 존재할 수 있음을 의미한다. 반면 지연시간 메트릭 기반방식의 경우 물리적인 근접도를 만족시킬 수 없다. 이것은 동일 지연의 경우 물리적으로 가까운 멤버와 콘텐츠를 교환하는 것에 대해 고려하지 못함을 의미한다. 따라서 본 연구에서는 특정 방식에 의존하지 않는 새로운 알고리즘을 적용한 혼합 메트릭 기법을 적용하였다.

3.2.2.1 다중 메트릭 알고리즘

```

Input :
delay(i) : host latency value, hopcount : physical distance
Output :
target node to compare resemblance

MAX_DISTANCE = 1;
for each host i
    if(delay(i) < T and hopcount == MAX_DISTANCE)
        forward request to host i to compare resemblance;
        MAX_DISTANCE++;
endfor
    
```

그림 4 Working Set 선택 다중 메트릭 알고리즘

그림 4의 제안하는 다중 메트릭 알고리즘은 1 홉 내의 멤버들 중에서 지연시간 임계 값 T 를 넘지 않는 멤버들, 2 홉 내의 멤버들 중에서 $T*2$ 의 지연시간을 넘지 않는 멤버들, 계속해서 홉 수를 증가시키면서 구성되는 멤버들과 콘텐츠 유사성 검사를 실시하여 동일하지 않은 부분을 가질 경우 콘텐츠 교환을 실시하도록 하는 알고리즘이다. 이때 지연시간 임계 값 T 는 최대 지연시간 max_L 값과, 최소 지연시간 min_L 값을 제외한 나머지 임계 값의 평균을 기본값으로 정의하며, 다양한 조건하의 성능평가를 통해 상황에 따른 합리적인 임계 값 T 를 산출하도록 한다.

파일 f 의 일부를 가지는 새로운 멤버의 출현은 언제나 가능한 상황이다. 새로운 Working Set의 멤버가 그룹에 합류할 경우 그룹관리를 담당하는 RP 노드는 멤버를 가입시키고 메시지를 갱신하여 그룹관리를 실시한다. 이렇게 새로운 그룹으로 갱신되는 시점에 전송관리를 위한 다중 메트릭 알고리즘이 동작하여 새로운 홉 멤버 정의가 이루어지며 새로 정의된 각 홉 멤버들로부터 계속해서 파일 다운로드를 실시하게 된다.

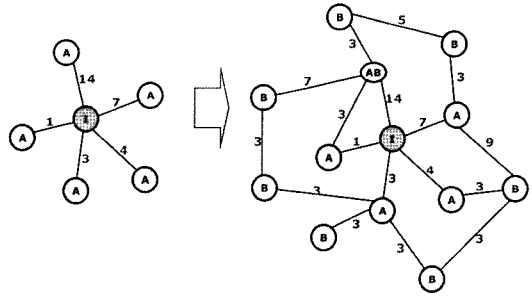


그림 5 다중 메트릭 알고리즘 동작 절차

그림 5은 지연시간 임계 값 $T=5$ 로 가정한 상태에서 알고리즘의 동작 절차를 나타낸다. 최초 1홉 멤버 A 중에서 $T < 5$ 를 만족하는 3개의 멤버가 선택되었다. 이때 선택된 멤버들과 유사성 검사를 실시하고, 보유하고 있는 콘텐츠의 일부가 완전히 중복되지 않을 경우 콘텐츠 교환을 실시한다. 유사성 검사와 동시에 2홉 멤버 B 중에서 $T*2 < 10$ 을 만족하는 5개의 멤버가 선택되었다. 2홉에서 선택된 멤버에서 A와 B의 성질을 모두 만족시키는 AB 노드는 1홉에서는 배제되었으나 2홉에서 다른 경로를 통해 전송 받을 수 있음을 보여주고 있다.

본 연구에서는 한번 제외된 멤버는 많은 홉 수의 멤버보다 지연시간이 적다하여도 콘텐츠 교환을 하지 않는 것으로 규정한다. 이것은 비교적 높은 지연시간을 갖는 제외된 멤버가 콘텐츠 교환을 하지 않는 대신 그 시간동안 오버로드에서 벗어날 수 있는 시간적 여유를 주기 위함이다.

3.2.2.2 Working Set의 범위

3.3절에서 제안한 협동적 분산 다운로드 기법을 통한 콘텐츠 전송은 대용량 파일을 빠른 시간에 다운로드 하는데 유리한 기법이다. 하지만 이와 같은 기법의 사용은 동일 파일을 가지는 Working Set의 모든 멤버들과 콘텐츠 교환 및 중계로 인해 발생하는 많은 중복 패킷 및 제어 패킷을 발생시켜 심각한 대역폭의 소모를 발생시킨다.

본 연구에서는 홉 카운트 제한을 통해 궁극적으로 기존 분산다운로드의 심각한 대역폭 소모를 줄이며 개선된 전송속도를 보장할 수 있는 새로운 전송기법을 제안하며, 성능평가를 통해 제안한 방식의 전송효율을 증명하고자 한다. 성능평가를 위한 주요한 고려요소는 다음과 같다.

1. 콘텐츠의 유형 : 스트리밍 콘텐츠, 텍스트 형 콘텐츠
2. 콘텐츠의 크기 : 대용량 콘텐츠, 중용량 콘텐츠, 소용량 콘텐츠
3. 그룹의 규모 : 소규모 그룹, 중규모 그룹, 대규모 그룹
4. 네트워크 규모 : 지역 네트워크, 인터넷

4. 성능분석

4.1 실험 토폴로지 구성

본 연구에서 제안한 다중 메트릭 알고리즘 적용을 위해 그림 6과 같은 실험 환경의 기본 토폴로지를 정의하고 각 노드의 지연시간 값에 대한 초기화를 실시하였으며, 지연시간 측정에 사용된 값은 각 노드의 RTT (Round Trip Time)값을 측정하여 설정하였다. 기본적인 토폴로지는 인터넷상에서 파일 j 의 일부를 가지는 95개의 멤버를 대상으로 구성하였으며, 현실적인 성능 분석을 위해 국내 120여 대학의 웹 서버 중에서 정상적인 RTT 패턴을 보이는 95개의 서버를 선택하였다. 그림 7은 2003년 11월부터 2004년 3월까지 숭실대학교를 기준으로 전국 각 대학 웹 서버를 대상으로 1000개의 질의를 요청하여 산출한 RTT 질의 결과 값의 평균 분포도로써, 실 RTT 평균값의 분포는 20msec 이하 구역에 집중 분포되어있으며, 최대 180msec를 넘지 않는 것으로 나타났다.

본 연구에서는 네트워크상의 RTT값이 지수분포를 따른다고 가정하며, 산출된 평균값 $\frac{1}{\lambda}$ 은 9.02ms 이다. 그림 6과 같은 토폴로지상의 각 종단간의 지연시간을 얻기 위하여 실제 RTT값의 평균 ($\frac{1}{\lambda}=9.02$)을 기반으로 하여 지수분포를 따르는 임의의 RTT값(δ_i)을 수식 (2)를 적용하여 95개의 지연 값을 생성하도록 하였다. 생성된 지연 값은 그림 7의 측정 RTT값과 유사한 분포를 갖는다.

$$\delta_i = \frac{\ln D}{\lambda} \quad (2)$$

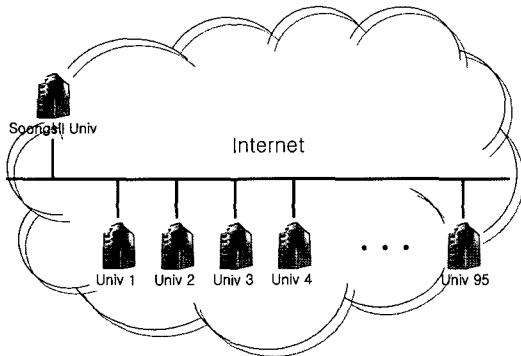


그림 6 실험 토폴로지

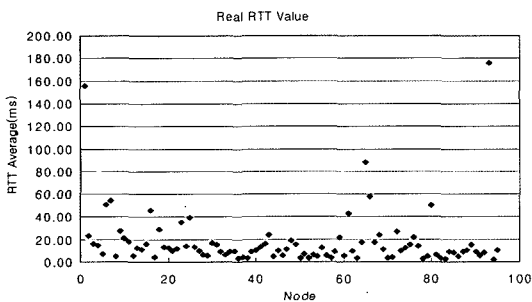


그림 7 측정 RTT값 분포도

수식 (2)에서 D 는 난수로써 0~1 사이의 값을 가지며, 95개의 난수를 적용하여 각 노드의 지연시간을 구할 수 있다.

4.2 지연 임계값 T 설정

본 연구에서 제안한 다중 메트릭 Working Set의 선택 알고리즘에서 무엇보다 중요한 것은 지연시간 임계 값 T를 설정하는 것이다. 지연시간 임계 값 T를 너무 크게 선택할 경우 본 알고리즘의 도입이 무의미 하게 되며, 너무 작은 값을 선택할 경우 적절한 성능을 보장하기 위해 넓은 지역까지 Working Set을 구축해야 하는 단점이 있다. 따라서 본 절에서는 국내 95개 대학의 실 지연 값을 구하여 가상의 토폴로지에 적용한 후 현실성 있는 범위 내에서 가장 적절한 T를 구하도록 하였으며, 선택한 T의 적용을 통해 각 노드의 전송 성능의 측정과 그에 따르는 대역폭의 소비율을 측정하여 분석 하였다.

$$T = \alpha \cdot T_{prev} + (1-\alpha) \cdot T_{new} \quad (3)$$

인터넷상의 지연시간은 시간의 경과에 따라 동적으로 변화하기 때문에 각 노드의 RTT 값 또한 유동적이다. 따라서 지연 임계 값 T 또한 고정적일 수 없으며 해당 그룹의 RP는 지연 임계 값 T를 주기적으로 공지하여야 한다. 본 연구에서는 지연 임계 값 T의 초기값을 전체 RTT값의 평균값으로 정의하며, 보다 유연한 T값 산출을 위해 수식 (3)과 같은 Weighted average 기법을 적용하였다. T_{prev} 는 이전 T값으로 α 비율로 반영을 하고 있으며, 새로 측정된 평균 지연 임계 값 T_{new} 값을 $1-\alpha$ 비율로 반영하고 있다. 본 연구에서 α 의 비율은 2:8로 정의 하여, 이전 T값을 20% 반영하도록 하였다. 실험에 사용하기 위한 최초 T값은 수식 (2)를 100회 수행하여 얻은 값을 T_{prev} , 101회 수행하여 얻은 값을 T_{new} 로 적용하여 산출한 후 수식 (3)에 적용하였다.

4.3 1 홉 내의 전송 분석

본 절에서는 지연 임계 값 T에 대한 성능평가 및 분석을 위하여 콘텐츠 수신노드의 1홉 내에서 T 값 적용에 따른 전송 속도와 대역폭 소모를 분석하였다. 본 실험에 사용되는 가상 토폴로지상의 1홉은 실제 적용한 노드의 수를 감안하여 10개 노드로 제한하였으며, 실 데이터를 가상의 토폴로지에 적용하기 위해 지연 값에 혼합 정렬을 30회 수행하여 300개의 1홉 지연 값을 산출하였다. 또한 실험의 공정성을 기하기 위하여 각 가상 링크의 가용 대역폭을 동일하게 설정하였다.

본 실험을 통해 지연 임계 값 도입 시 성능상의 효율을 증명할 수 있으며, 실험의 본질에 접근하기 위해 다음의 가정사항을 둔다.

- 가정 1. 전송 콘텐츠는 60Mbyte의 대용량 콘텐츠이다.
- 가정 2. 60Mbyte의 콘텐츠는 1500byte의 단위패킷으로 분할 전송된다.
- 가정 3. 각 가상 링크상의 가용 대역폭은 20Kbps로 동

일하다.

가정 4. 모든 멤버들은 전송 콘텐츠의 Working Set으로서 해당 파일의 일부를 가진다.

가정 5. 각 수신자 R은 동시에 전송을 시작한다.

가정 6. 전송에 따르는 패킷 손실률은 없다.

수식 (4)는 각 노드에서의 전송시간에 대한 관계를 나타낸 행렬식으로써 t_i 는 i 노드의 지연 값을 나타내고, x_i 는 i 노드가 전송하는 패킷의 양을 나타낸다. 결과적으로 N 은 전체 전송 패킷의 양을 나타낸다. 본 실험에서 i 는 지연시간 순서로 오름차순 정렬을 취하여 i 값이

적을수록 낮은 지연 값을 갖는 노드로 정의한다. 따라서 t_i 는 최소 지연 값을 나타내며, $x_1 \times t_1$ 은 최소 요구 전송시간을 나타낸다.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ t_1/t_2 - 1 & 0 & 0 & 0 & \dots & 0 \\ t_1/t_3 & 0 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_1/t_n & 0 & 0 & 0 & \dots & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} N \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4)$$

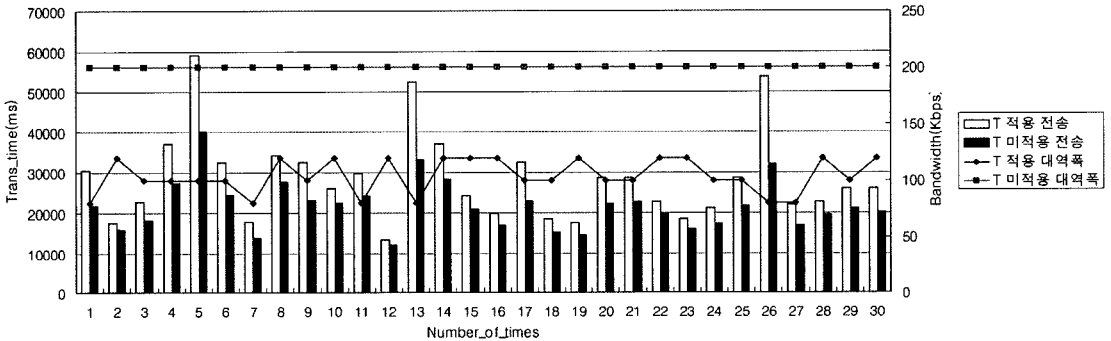


그림 8 최소지연 그래프

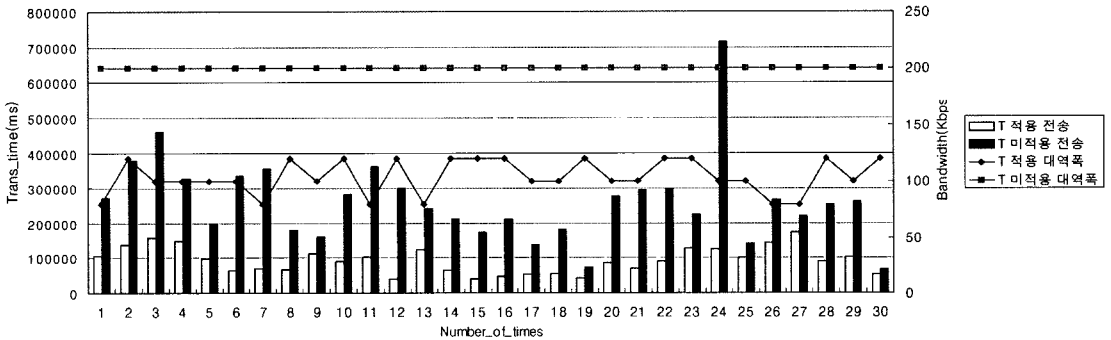


그림 9 최대지연 그래프

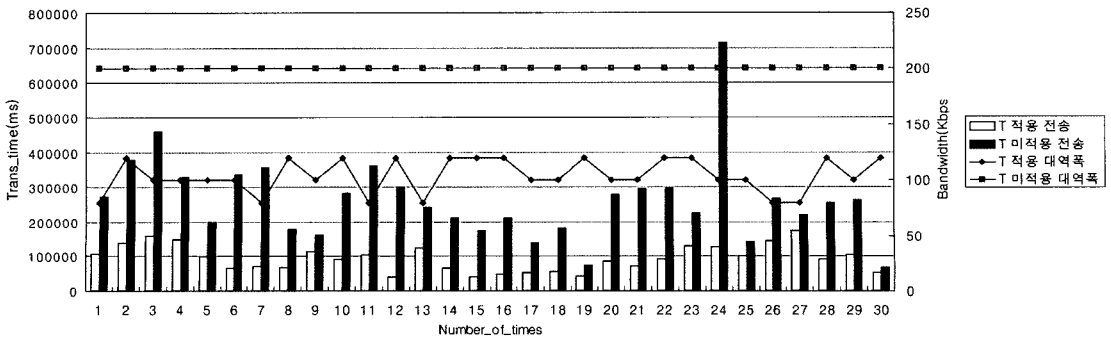


그림 10 평균지연 그래프

그림 8에서 각 1홉 데이터들은 수식 (3)을 적용한 각각의 지연 임계 값 T를 가지며 T값 적용시의 전송성과 대역폭 소비율, T값 미적용 시의 전송성과 대역폭 소비율을 나타내고 있다. 이때 T값을 적용하지 않을 경우 10개의 1홉 송신노드로부터 콘텐츠를 전송받기 때문에 T값을 적용하였을 때 보다 빠른 전송속도를 보임을 알 수 있다. 그러나 T 값을 적용하지 않을 경우 모든 링크를 사용하여 전송하기 때문에 전체 대역폭을 모두 소비하므로 자원 활용의 효율성이 낮음을 알 수 있다.

그림 9는 T값 적용 시 및 미적용 시의 경우에서 최악의 상황에 대한 성능분석 결과이다. 이러한 경우는 최대 지연시간을 갖는 노드의 전송에 의존하기 때문에 T값 적용 시 평균 2배 이상의 전송속도를 확보할 수 있으며, 대역폭 또한 미적용 시의 전송보다 대역폭의 손실을 방지할 수 있음을 보여주고 있다.

그림 10은 그림 8과 그림 9의 평균을 나타내는 그래프로 제안한 지연 임계 값 T 적용 시, 보다 빠른 전송과 대역폭 소모를 방지함으로써 궁극적으로 전송속도 대비 대역폭 소비 측면에서 전송효율이 향상됨을 알 수 있다.

각 그래프의 양 세로축은 각각 총 전송시간(ms)과 대역폭(Kbps)을 나타내며, 가로축은 1홉 데이터를 30회 추출한 시도 횟수를 나타내고 있다.

그림 11에서 그림 14까지의 그래프들은 제안하는 지연 임계 값 T를 적용하였을 때 전송효율을 나타내는 그래프로써 주어진 지연시간 범위 내에서 T값을 증가시켜 각 상황별 효율을 보여주고 있다. 30번의 1홉 데이터를 대상으로 실험한 결과 본 연구에서 제안한 Weighted average 기법을 사용하여 얻은 T값을 사용할 경우 전송속도는 T값 미적용 시 보다 평균적으로 1.2배 정도의 낮은 전송속도를 보이나, 대역폭 소비율은 대략 50% 이하로 감소함을 볼 수 있다. 따라서 본 연구에서 각 멤버들 간 분산 다운로드를 통해 대용량 콘텐츠를 전송할 경우 제안한 지연 임계 값 T를 적용하여 보다 효율적인 전송을 보장함을 알 수 있다.

4.4 Max_Distance 설정에 따른 분석

본 절에서는 1홉 내의 실험 결과를 바탕으로 전체 토폴로지 상에서 지연 임계 값 T를 적용하고, 홉 카운트 값인 Max_Distance를 점차적으로 증가시켜 전체 토폴로지 상의 전송효율을 실험 하였다. 그림 15 Max_Distance 그래프는 x 축의 Max_Distance 값을 점차적으로 늘려가면서 각 상황별 전송효율을 분석한 그래프로써 적용하지 않을 경우 T 그래프와 마찬가지로 전송속도는 빠르지만 대역폭의 소모가 심각함을 알 수 있다. 제안한 Max_Distance 값의 적용 시 전송 속도에서는 약 0.2배의 지연손실을 초래하지만 대역폭 측면에서는 50%의 절감 효과를 보임을 알 수 있다. 결과적으로 제안하는 유효 Working Set

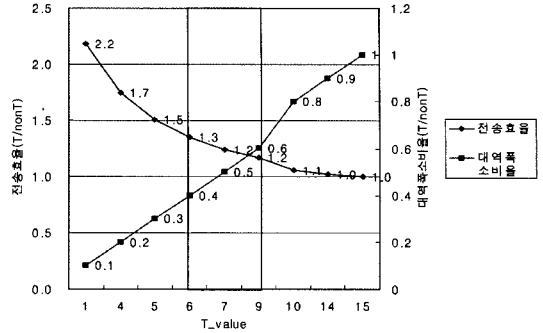


그림 11 T 적용 전송효율 그래프 1단계(T=7)

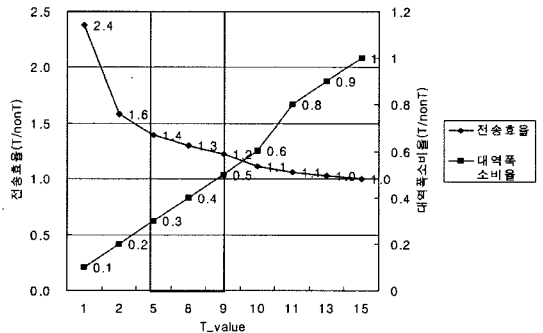


그림 12 T 적용 전송효율 그래프 2단계(T=8)

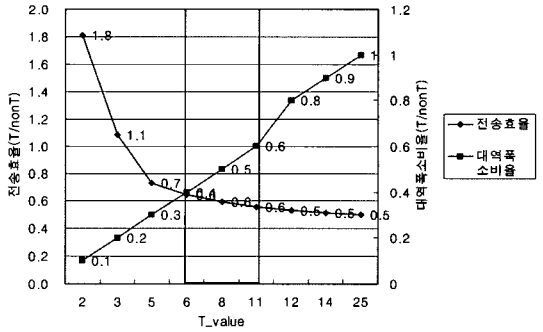


그림 13 T 적용 전송효율 그래프 3단계(T=9)

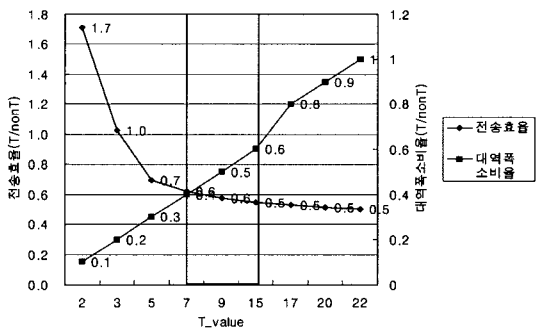


그림 14 T 적용 전송효율 그래프 4단계(T=11)

참고 문헌

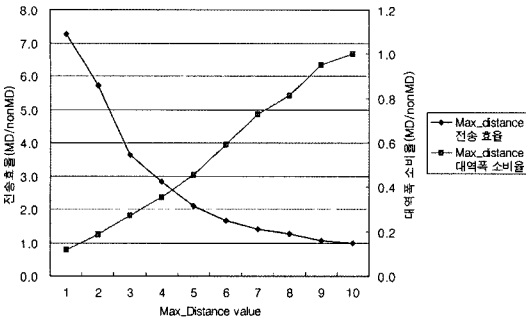


그림 15 Max_Distance 전송효율 그래프

을 제한시킴으로써 전송상의 성능 향상과 대역폭의 효율을 보장함을 보일 수 있다.

5. 결론

최근 인터넷 및 정보통신 기술의 발전으로 인한 인터넷 사용자의 폭발적 증가와 P2P 서비스의 확산으로 인한 대용량 콘텐츠 전송이 급격히 증가하고 있는 추세이다. 이러한 현상으로 인해 급격히 증가하는 네트워크 트래픽을 수용하고 처리하기 위해 다양한 연구들이 진행되고 있다. 응용계층 멀티캐스트는 기존 IP 멀티캐스트의 장점을 수용하면서 인터넷상에 보급이 가능하도록 멀티캐스트의 기능을 응용계층으로 올린 신 개념의 서비스로, 기존의 네트워크 구조, 설비의 변경 없이 곧바로 적용할 수 있기 때문에 새로운 네트워크 서비스의 주축으로 성장할 것이다.

본 연구에서는 새로운 네트워크 서비스의 중심으로 등장할 응용계층 멀티캐스트 환경에서 대용량 콘텐츠를 전송하기 위한 방안으로 협동적 분산 다운로드 기법을 제안하였다. 강력한 그룹관리를 제공하는 NARADA 프로토콜의 그룹관리와, 기존 P2P 방식의 분산 다운로드 기법에서의 대역폭 소모를 해결하기 위해 유효 Working Set의 규모를 정의 하여, 응용계층 멀티캐스트 기반 대용량 콘텐츠 전송 기법을 제안하였다. 성능평가를 통해 얻어진 유효 Working Set 규모의 적용을 통해 제안한 전송기법이 일반적인 멀티캐스트 트리상의 전송보다 빠른 전송속도를 보장하면서 대역폭의 소모를 줄일 수 있음을 증명하였다. 본 연구 결과를 바탕으로 향후, 전송 콘텐츠의 중복전송으로 인한 대역폭 손실 문제에 대한 연구와 실시간성 데이터 전송 및 유사성 검사를 통한 유사도 값을 콘텐츠 교환 시 우선순위 요소로 적용하는 연구를 통해, 더욱 효과적이고 범용적인 전송기법으로 유용하게 사용될 수 있을 것이다.

[1] John Byers, Jeffery Considine, Michael Mitzenmacher, and Stanislav Rost, "Informed Content Delivery Across Adaptive Overlay Networks," In *Proc. ACM SIGCOMM*, August 2002.

[2] Suman Banerjee, and Bobby Bhattacharjee, "A Comparative Study of Application Layer Multicast Protocols," In *Submitted for review*, 2002.

[3] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan, and Hui Zhang, "A Case for End System Multicast," In *Proceedings of ACM SIGMETRICS*, June 2000.

[4] Paul Francis, "Yoid: Extending the Multicast Internet Architecture," 1999. White paper <http://www.aciri.org/yoid/>

[5] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy, "Scalable Application Layer Multicast," In *Proceedings of ACM SIGCOMM*, August 2002.

[6] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris, "The Case for Resilient Overlay Networks," In *Proceedings of HotOS VIII*, May2001.

[7] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," In *Proc. ACM SIGCOMM*, September 1998.

[8] Andrei Z. Broder, "On the Resemblance and Containment of Documents," In *Compression and Complexity of Sequences(SEQUENCES'97)*, 1997.

[9] John W. Byers, Michael Luby, and Michael Mitzenmacher "Accessing multiple mirror sites in parallel: Using Tornado codes to speed up downloads," In *Proc. IEEE INFOCOM*, vol. 1, pp. 275-283, New York, NY, March 1999.

[10] Pablo Rodriguez and Ernst W. Biersack, "Dynamic Parallel Access to Replicated Content in the Internet," *IEEE/ACM Transactions on Networking*, 10(4), Aug. 2002.

[11] Simon G. M. Koo, Catherine Rosenberg, and Dongyan Xu, "Analysis of Parallel Downloading for Large File Distribution," In *Proc. of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, San Juan, Puerto Rico, 2003.

[12] Y. Liu, W. Gong, and P. Shenoy. "On the Impact of Concurrent Downloads," In *Proc. of INFOCOM*, volume 1, pp. 304-312, 1999.

[13] VITTER, J. S., "Random sampling with a reservoir," *ACM Trans. on Math. Software* 11, (1985), 37-57.



박진홍

2002년 배재대학교 컴퓨터공학과 공학사
 2004년 숭실대학교 컴퓨터학과 공학석사
 2004년~현재 TTA SW시험인증센터 전
 임연구원. 관심분야는 멀티캐스트, SW
 품질평가, 벤치마크테스트



윤미연

2000년 가톨릭대학교 수학과/컴퓨터학과
 이학사. 2002년 숭실대학교 컴퓨터학과
 공학석사. 2005년 숭실대학교 컴퓨터학과
 공학박사. 2005년~현재 KISA 정보보호
 기술단 기술기획팀 선임연구원. 관심분야
 는 센서네트워크, 애드혹 네트워크, 멀티

캐스트, 정보보호



김선호

1987년 이화여자대학교 사범대학 수학교
 육전공 이학사. 1992년 이화여자대학교
 교육대학원 전자계산교육전공 교육학석
 사. 2004년 숭실대학교 컴퓨터학과 컴퓨
 터통신전공 공학박사. 1987년~1989년
 대우전자부품(주) 전산실. 1990년~1993

년 한국생산성본부 정보화사업부 선임연구원. 1998년~2004
 년 동덕여자대학교 정보과학대학 강의전임교수. 2004년~현
 재 서울시 소방방재본부 전산개발팀 전문위원. 관심분야는
 Internet Protocol, Mobile IP, CDN, DRM



신용태

1985년 한양대학교 산업공학과 졸업(학
 사). 1990년 Univ. of IOWA 전산학과
 (석사). 1994년 Univ. of IOWA 전산학
 과(박사). 1994년~1995년 Michigan
 State Univ. 전산학과 객원교수. 1995

년~현재 숭실대학교 컴퓨터학과 부교수
 관심분야는 멀티캐스트, 정보보호, DRM, USN, 그룹통신,
 CDN



신석규

1998년 충남대학교 컴퓨터학과 석사
 1995년 시스템공학연구소(SERI) 경영정
 보과장. 1998년 한국전자통신연구원
 (ETRI) 시스템통합연구팀장. 2000년 한
 국전자통신연구원(ETRI) SW시험운영팀
 장. 2001년~현재 TTA SW시험인증센

터 센터장. 관심분야는 SW품질평가, SW컴포넌트, DBMS,
 SW품질표준화