플래쉬 디세미네이션을 위한 안정적이고 효과적인 오버레이 네트워크 기반 전송 시스템

(Reliable and Effective Overlay Network based Dissemination System for Flash Dissemination)

김경백*

(Kyung baek Kim*)

요 약

컴퓨터 네트워크의 말단에 위치한 개인형 머신들의 성능 향상과 백본 라우터들의 속도향상에 힘입어, 데이터 전송시스템을 사용자 머신들을 이용해서 구성하는 오버레이 네트워크 기반의 데이터 전송 시스템들에 대한 연구가 활발히 이루어지고 있다. 오버레이 네트워크 기반의 데이터 전송시스템들은 그 적용범위에 따라 알맞은 네트워크 구조와 전송 프로토콜을 사용하여야 한다. 이 논문에서는 동일한 데이터를 대규모 수신자들에게 짧은 시간 안에 안정적으로 전송하는 플래쉬 디세미네이션을 위한 오버레이 네트워크 기반의 전송 시스템들(ReCREW와 FaReCAST)을 소개하고, 플래쉬 디세미네이션을 위한 오버레이 네트워크 구조와 전송 프로토콜에 있어서 고려점들을 분석한다. 플래쉬 디세미네이션 상황 하에서의 실험 결과를 통해 제안된 오버레이 네트워크 기반의 전송 시스템들이 기존의 오버레이 네트워크 기반 멀티캐스팅 시스템보다 데이터 전송의 안정성 그리고 데이터 전송 딜레이 측면에서 모두 향상된 성능을 보이는 것을 확인한다. 또한 FaReCAST의 이론적 분석을 통한 데이터 전송의 안정성에 대해서 이론적으로 논의한다.

■ 중심어 : | 오버레이 네트워크 | 플래쉬 디세미네이션 | 컨텐트 디스트리뷰션 | 멀티캐스팅| 안정성

Abstract

The significant enhancement of the edge portion of computer networks including user-side machines and last mile network links encourages the research of the overlay network based data dissemination systems. Varieties of overlay network based data dissemination systems has distinct purposes, and each of them has a proper structure of an overlay network and a efficient communication protocol. In this paper, overlay network based data dissemination systems for Flash Dissemination, whose target is the distribution of relatively small size data to very large number of recipients within very short time, are explored. Mainly two systems, RECREW and FaReCAST, are introduced and analyzed in the aspects of design considerations for overlay networks and communication protocols. According to evaluations for flash dissemination scenarios, it is observed that the proposed overlay network based flash dissemination systems outperforms the previous overlay network based multicasting systems, in terms of the reliability and the dissemination delay. Moreover, the theoretical analysis of the reliability of data dissemination is provided by analysing FaReCAST.

■ keyword : | Overlay Network| Flash Dissemination| Content Distribution| Multicasting| Reliability

Ⅰ. 서 론

컴퓨터 네트워크에서 단일 데이터를 다수의 사용자들에게 전달하는 대표적인 방법은 멀티캐스팅이다. 효과적인 멀티캐스팅을 위해서 IP 계층 네트워크에서의 멀티캐스팅을 구현하는 방법에 대한 연구가 있었다. 하지만, 이러한 IP 계층 네트워크 멀티캐스팅은 IP 계층의 모든 다양한 장비들이 동일한 멀티캐스팅 프로토콜을 이해하고 지원해야 한다는 제한점에 따라 현실적으로 그 적용이 힘들다.

이러한 제한점을 해소하여 보다 쉽게 멀티캐스팅 서비스를 제공하기 위해서 IP 계층 네트워크 상위 계층, 즉 사용자 계층

* 정회원, 전남대학교 전자컴퓨터공학부

이 논문은 2012년도 전남대학교 학술연구비 지원에 의하여 연구되었음

접수일자 : 2012년 11월 05일
수정일자 : 2013년 02월 13일
게재완료일 : 2013년 2월 13일

교신저자 : 김경백 e-mail: kyungbaekkim@jnu.ac.kr

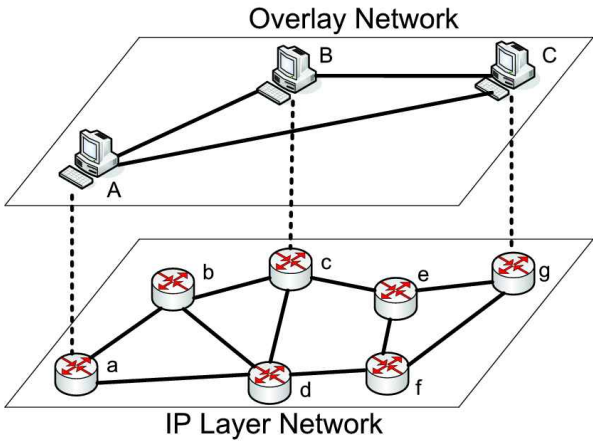


그림 1. 오버레이 네트워크와 IP 계층 네트워크

네트워크에서 멀티캐스팅을 지원하는 오버레이 네트워크 기반의 멀티캐스팅에 대한 연구가 있었다. [1][2] 특히 컴퓨터 네트워크의 대역폭의 증가, 말단 사용자 단말의 급격한 성능 향상, 컴퓨터 네트워크 백본 장비들의 단순화 등 다양한 요소들이 오버레이 네트워크 기반의 멀티캐스팅에 대한 연구들을 더욱 활발하게 해주는 역할을 하고 있다.

그림 1은 오버레이 네트워크의 개략적인 개념을 보여준다. 오버레이 네트워크는 IP 계층 위의 사용자 계층에 존재하는 머신들로 이루어진 네트워크로써, 오버레이 네트워크 그래프의 하나의 에지(Edge)는 하위 IP 계층 네트워크 그래프의 하나 또는 그 이상의 에지들로 이루어져 있다. 이렇게 구성된 오버레이 네트워크를 기반으로 퍼플리쉬-서브스크라이브 시스템 (Publish-Subscribe System)과 같은 단순 메시지 멀티캐스팅[3], 스트리밍 미디어 멀티캐스팅[4], 대용량 데이터 멀티캐스팅[5] 와 같은 다양한 목적을 가지는 오버레이 네트워크 기반 멀티캐스팅 서비스들이 연구 되었다. 특히, 갈수록 커지는 사용자 데이터를 전송하기 위해서 보다 효과적으로 네트워크의 대역폭을 사용하는 연구가 오버레이 네트워크 기반 멀티캐스팅에서의 주 이슈였다.[6][7][8]

최근 이러한 오버레이 네트워크 기반의 멀티캐스팅을 플래쉬 디세미네이션(Flash Dissemination) 상황에 적용하는 연구들이 있었다 [9][10][11]. 플래쉬 디세미네이션이란 기존의 오버레이 네트워크 기반의 멀티캐스팅에서 주로 다루는 데이터 보다 상대적으로 작은 크기의 데이터를 보다 많은 사용자들에게 아주 짧은 주어진 시간 안에 전송시키는 것을 목적으로 한다. 이와 같은 플래쉬 디세미네이션 관련 어플리케이션으로는 재난 정보 예보, 경보, 관리 시스템을 생각할 수 있다. 그 한 가지 예로, 미국의 Advanced National Seismic System (http://www.anss.org)에서 지원하는 “Share-Cast"라는 서비스를 생각 할수 있다. 이 서비스는 지진파의 정보와 GIS (Geographic Information System)정보와 같은 지리적 정보를

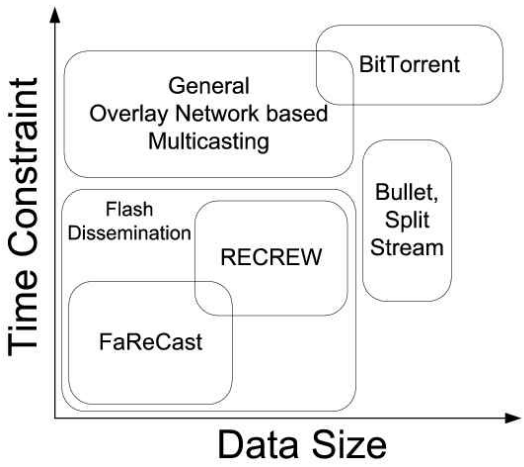


그림 2. 데이터 크기와 시간제한에 따른 오버레이 네트워크 기반의 멀티캐스팅 분류

종합해서, 지진의 영향권에 있는 사용자들에게 현재 지진의 영향에 대한 정보를 200K정도의 크기를 가지는 지도 형태의 데이터에 담아 전송하는 서비스이다. 또 다른 예로는, 지진 경보 서비스를 생각 할 수 있다. 즉, 대지를 흔들어 건물과 인명 피해를 부르는 지진충격파(S-wave)가 오기 수십 초전에, 가 지진파(P-wave)를 인지하여 예보 메시지를 수초 안에 관련 지역 사람들에게 전달함으로써 인명피해 및 산업의 피해를 최소화 할 수 있는 서비스도 이러한 플래쉬 디세미네이션에 관련된 응용 서비스로 고려 할 수 있다.

이러한 플래쉬 디세미네이션의 가장 큰 특징은 데이터 전송에 있어서 **엄격한 시간제한**을 둔다는 것이다. 즉 정해진 시간 내에 도착하지 못한 경보 관련 데이터는 그 경보에 관련된 이벤트가 발생한 후에는 더 이상 사용자에게 필요 없게 된다는 것이다. 단적인 예로, 지진 경보 시스템의 경보 메시지가 지진의 충격파가 도달 한 후에 사용자에게 도착한다면, 이 시스템은 제 역할을 하지 못했다고 말할 수 있다.

또 다른 플래쉬 디세미네이션의 특징은 **적은 전송 데이터의 크기** 이다. 즉, 전송하는 데이터의 크기가 다른 오버레이 네트워크 기반 멀티캐스팅에서 사용하는 데이터의 크기보다 적다는 것이다. 수십 메가 또는 수 기가 바이트의 크기를 가진 미디어 파일들을 전송하는 것이 그 주 목적인 기존의 오버레이 네트워크 기반 멀티캐스팅에 비해, "Shake-Cast"서비스의 경우 최대 수백 킬로 바이트를 가지는 데이터(e.g. Shake-Map)를 전송한다. 특히 경보 시스템과 같은 경우 그 데이터의 양은 수백 바이트 단위까지 줄여서 생각 할 수 있다. 이와 같은 점들을 종합하여 오버레이 네트워크 기반의 멀티캐스팅을 구분해보면 그림 2와 같이 생각 할 수 있다.

이 논문에서는 이러한 플래쉬 디세미네이션을 보다 효과적으로 지원하기 위한 오버레이 네트워크 기반의 멀티캐스팅 시스

템들에 대해 소개하고 각 시스템에서의 고려사항들에 대해서 살펴본다.크게 두가지 시스템들(RECREW와 FaReCAST)을 중심으로 플래쉬 디세미네이션 상황에서 오버레이 네트워크 의 구성 시 고려해야 할 사항과 오버레이 네트워크 구성 후 통신프로토콜의 개발시 고려해야 할 사항에 대해서 알아본다. 또한 FaReCAST에 대한 이론적 분석을 통해, 플래쉬 디세미네이션 상황에서의 오버레이 네트워크의 전송 안정성의 이론적 분석을 제공한다.

앞으로의 논문은 다음과 같이 이루어져 있다. 2장에서는 중간 정도 크기의 데이터의 플래쉬 디세미네이션을 위해 사용자의 불특정 장애에도 안정적으로 동작하는 메쉬 기반의 오버레이 네트워크를 구성하고 Gossip기반의 통신프로토콜을 사용해서 보다 짧은 데이터 전송시간을 보장할 수 있는 RECREW에 대해서 설명한다. 3장에서는 예측할 수 없는 대규모 오버레이 네트워크 장애에도 불구하고 작은 데이터의 플래쉬 디세미네이션을 안정적으로 지원하기 위해 M2M (Multiple Parents to Multiple Children) 오버레이 네트워크를 구성하고 Multidirectional Multicasting을 사용하는 FaReCAST 시스템에 대해서 설명한다. 마지막으로 4장에서 오버레이 네트워크 기반의 플래쉬 디세미네이션 시스템에 대한 분석과 더불어 이 논문의 결론을 말한다.

II. RECREW

RECREW (REliable Concurrent Random Expanding Walkers) 는 플래쉬 디세미네이션중 중간 크기의 데이터를 보다 빠른 시간 내에 전송하기 위한 오버레이 네트워크 기반 멀티캐스팅 시스템이다. RECREW는 inter/intra node concurrency 개념을 적용한 보다 효과적인 gossip 프로토콜을 사용한다. 이 gossip 프로토콜은 이전에 전송받은 데이터를 식별할 수 있는 메타데이터와 Expanding Random Network상에서 Random Walk을 사용하여, 불필요한 데이터 전송 횟수를 줄이고 필요한 데이터를 찾을 확률을 높임으로써, 전체적인 데이터 전송 시간을 단축시킬 수 있도록 한다. 이때, 사용되는 Expanding Random Network가 사용자들의 머신으로 이루어진 오버레이 네트워크이다. RECREW는 예기치 못한 사용자 머신의 장애에도 안정적으로 오버레이 네트워크를 유지하기 위해 Bounce Protocol을 사용한다.

1. 메타데이터 기반 pull 방식 gossip 프로토콜

일반적인 gossip 프로토콜의 문제점은 데이터를 주고받는 과정에서 불필요한 데이터를 교환하는 오버헤드이다. 이러한 오버헤드는 gossip 프로토콜이 일반적으로 push-based protocol 이기 때문이다. 즉 자신이 받은 데이터를 자신의 주변 노드들에

```
INITIALIZE:
    RecvdChunksIds ← {∅}
    RecvdChunks ← {∅}
    ChunksToGet ← {c1.id, c2.id, ...cM.id}
BEGIN
1) While |ChunksToGet| > 0
2)    Node X ← get next random node
3)    Chunk ck ← RPCa (X, GossipPull, RecvdChunksIds)
4)    RecvdChunks ← RecvdChunks ∪ ck
5)    RecvdChunksIds ← RecvdChunksIds ∪ ck.id
6)    ChunksToGet ← ChunksToGet − ck.id
END
```

그림 3. RECREW 기본 프로토콜

게 전송하는 단순한 push-based 방식은 데이터 전송 초기에는 효과적이지만, gossip이 진행될 수록 데이터를 전송받은 노드의 개수가 증가함에 따라, 불필요한 오버헤드를 발생시킨다. 이러한 문제를 해결하기 위해서 RECREW에서는 메타데이터 기반의 pull 방식의 gossip 프로토콜을 사용한다. 전송하고자 하는 데이터의 내용을 여러 개의 chunk들로 나누고 각각의 chunk는 유일한 chunk-id를 할당 받는다. 모든 chunk-id의 리스트를 RECREW 메타데이터라 하고, 이 정보는 본격적인 gossip protocol이 시작되기 이전에 모든 노드에게 전송되어야 한다. 이를 위해서는 구성된 오버레이 네트워크에서 단순한 브로드캐스팅 기법을 사용해서 메타데이터를 전송한다.

RECREW의 gossip프로토콜은 메타데이터 기반의 pull 방식을 사용한다. 이 기본적인 알고리즘은 그림 3과 같다. 새로운 chunk를 얻기 위한 gossip 프로토콜을 시작하기 위해서 RECREW의 오버레이 네트워크의 각 노드는 오버레이 네트워크상에서 Random Walk방식을 통해 타겟 노드를 선택하고, 현재 자신이 가지고 있는 chunk들의 정보를 기록한 메타데이터를 타겟 노드로 전송한다. 메타데이터를 받은 타겟 노드는 gossip중인 노드가 가지고 있지 않은 chunk들중 하나를 임의로 선택하여 전송 한다. 만약 타겟 노드가 gossip중인 노드를 위한 새로운 chunk를 가지고 있지 않다면, 에러 메시지를 전송한다. 따라서, 일반적인 gossip프로토콜과는 달리, RECREW 시스템에서 오버레이 멀티캐스팅을 수행하는 동안 불필요한 데이터 전송이 발생하지 않는다.

임의의 노드가 필요한 데이터를 모두 받게 되면, 이 노드는 즉시 gossiping을 멈춘다. 즉 더 이상 새로운 chunk를 위해서 타겟 노드를 선택하지 않는다. 이에 따라, 일반적인 gossip protocol는 확률적으로 데이터 전송을 멈추게 되는 점과는 다르게, RECREW는 deterministic termination delivery특성을 가지고 있다고 할 수 있다.

2. 이형 오버레이 네트워크 지원

RECREW에서 구성되는 오버레이 네트워크는 일반사용자 머신이기 때문에, 각 노드들의 네트워크 딜레이와 대역폭이 아

```
BEGIN
1) While |ChunksToGet| > 0
2)    While Spare bandwidth exists
3)       Node X ← get next random node
4)       Do Concurrently With Main Thread:
5)          ChunkId id ← RPC(X, IntentToPull, RecvdChunksIds)
6)          Acquire Mutex Lock
7)          If (id ∈ RecvdChunksIds)
8)             Release Mutex Lock
9)          Else
10)            RecvdChunksIds ← RecvdChunksIds ∪ id
11)            ChunksToGet ← ChunksToGet − id
12)            Release Mutex Lock
13)            Chunk ck ← RPC (X, GetChunk, id)
14)            RecvdChunks ← RecvdChunks ∪ ck
END
```

그림 4. 이형 오버레이 네트워크를 고려한 RECREW프로토콜

주 다양하게 존재한다. 만약 한 오버레이 노드가 gossip 프로토콜을 시작했다면, 이 노드는 타겟 노드로부터 응답을 받기 전까지 최소한 RTT (Round Trip Time)을 기다려야 한다. 이 RTT는 오버레이 네트워크 노드의 각 노드별로 다르게 존재하기 때문에, 이 값이 큰 노드에서 gossip 프로토콜이 수행 될때, 이 RTT기간 동안은 전체 대역폭을 낭비하게 된다. 이전의 오버레이 네트워크 기반의 데이터 전송 시스템들에서는 이와 같은 RTT에 의한 불필요한 대역폭 낭비를 줄이기 위해서 하나의 오버레이 커넥션 상에서 여러 개의 chunk들을 전송한다. 하지만 이와 같은 방식은 RECREW의 기본 gossip 프로토콜을 위배하는 방식이다.

하나의 커넥션에서 다수의 chunk를 전송하는 대신, RECREW는 다수의 Thread를 사용하여 여러 노드들에서 chunk를 동시에 받는 방식으로 이형 네트워크에서 발생할 수 있는 문제를 해결하였다. 이와 같은 다수 Thread사용은 전송 딜레이를 줄이는 역할 뿐만 아니라, 큰 대역폭을 가진 노드들을 보다 효과적으로 사용할 수 있는 가능성을 열어두는 역할을 한다. 메타데이터 기반의 gossip 프로토콜을 사용하는 것을 inter concurrency 개념을 사용한다고 본다면, 각 노드의 네트워크 대역폭에 따라 여러 chunk를 동시에 주고받는 방식을 사용하는 것을 intra concurrency 개념을 사용한다고 할 수 있다.

하지만 다수의 Thread를 사용할 경우, 여러 노드에서 동시에 같은 chunk를 전송할 가능성이 생기게 된다. 이와 같은 문제를 해결하기 위해서 RECREW는 gossip 프로토콜을 그림 4와 같이 두 단계의 스텝으로 나눈다. 그 첫 번째 스텝은 “intent to pull” 메시지를 타겟 노드에 전송하는 것으로 (그림 4의 5번 줄), 타겟 노드는 전송 가능한 chunk의 id를 가지고 응답한다. 이렇게 얻은 chunk-id가 현재 전송 중에 있지 않다면 노드는 해당 chunk를 타겟 노드로 부터 전송 받는다 (그림 4의 7-14 번 줄).

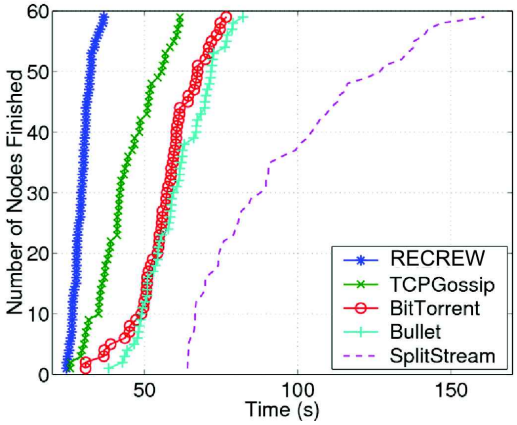


그림 5. 동형 오버레이 네트워크에서 RECREW 성능평가

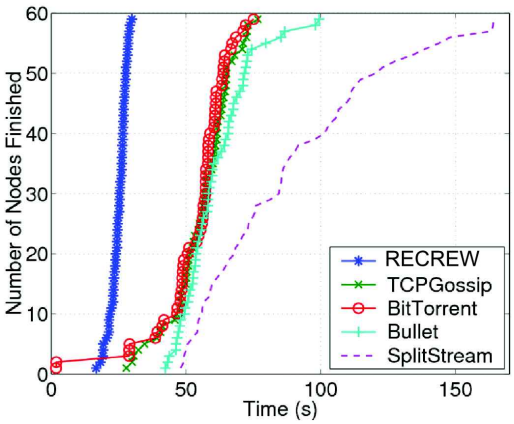


그림 6. 이형 오버레이 네트워크에서 RECREW 성능평가

3. Reliable Overlay Network 관리

RECREW의 메타데이터 기반의 gossip프로토콜은 메쉬 기반의 오버레이 네트워크를 필요로 한다. 구성된 오버레이 네트워크는 전송하고자 하는 데이터의 메타데이터를 전체 오버레이 네트워크에 전송하는 역할뿐만 아니라, gossip프로토콜을 수행하는 동안 타겟 노드를 유니폼리 랜덤하게 선택할 수 있는 Random Walker를 동작시키기 위한 수단으로 사용 된다.

이러한 역할을 잘 수행하기 위해서는 구성된 오버레이 네트워크는 Expanding Random Network의 특성을 가져야 하고, 이러한 네트워크의 구성하기 위해서는 각 오버레이 노드는 균등한 수의 오버레이 네트워크 네이버 노드를 가져야 한다.

이와 같은 조건의 오버레이 네트워크를 장애를 예측하기 힘든 일반 사용자 머신들을 사용하여 구축하기 위해서 RECREW는 Bounce 프로토콜을 사용한다. Bounce 프로토콜을 통해 각 노드는 구축된 오버레이를 따라 Random Walk을 수행하면서 후보 노드를 선택하고 네이버 노드 요청을 보낸다. 네이버 노드 요청을 받은 후보노드는 현재 자신의 네이버 노드의 개수(d)와,

요청을 보내는 노드의 요청 시도 횟수(t)를 기반으로 기준값 P_{Accept}를 계산하고 임의의 랜덤 값 x를 생성한다. 이때, x가 P_{Accept}보다 작을 경우, 네이버 요청을 받아들이다.사용하는 기준 값 P_{Accept}는 다음의 식(1) 을 사용해 계산한다.

$$P_{Accept} = \frac{1}{d} + rand(0,1) * Log(t) \quad (1)$$

4. RECREW의 성능평가

N개의 오버레이 노드에 M개의 chunk로 이루어진 데이터를 전송하고자 할 때, gossip프로토콜을 사용하여 한 번의 스텝에 하나의 chunk를 주고받는다고 가정하면, 이상적인 경우 $Log(N)+2M-1$의 스텝을 거치면 모든 노드들이 모든 데이터를 받게 된다. RECREW의 경우 필요한 전송 스텝은 $O(M + Log(N)*Log(M))$의 코스트를 가지게 된다. 이 값은 이상적인 경우의 대략적인 코스트인 $O(Log(N) + M)$값에 근사한 값이라 할 수 있다.

보다 실제적인 비교를 위해서, 다양한 오버레이 네트워크 멀티캐스팅 기법들(SplitStream, Bullet, BitTorrent, Basic TCP Gossip)과 비교한 결과를 그림 5와 그림 6에서 보여준다. 60개의 오버레이 노드를 사용해서 200KB의 파일을 동종네트워크 기반에서 실험한 결과를 그림 5에서 나타내고 있다. 그 결과를 살펴보면, 큰 사이즈 데이터를 전송하기 위해서 개발된 BitTorrent나 Bullet과 같은 경우 상대적으로 적은 사이즈인 200KB 파일을 전공하는데 있어서 기본적인 gossip프로토콜보다 성능이 좋지 않음을 확인 할 수 있었다. 반면, RECREW의 경우 40초 내에 모든 노드들이 파일을 전송을 받게 된다. 즉, RECREW가 TCP Gossip보다 빠른 전송 속도를 보이는 것을 알 수 있다. 또한 이형의 네트워크에서의 성능을 확인하기 위해서 같은 실험을 이형의 네트워크에서 수행한 결과를 그림 6에서 보여준다. 그 결과는 그림 5에서와 비슷하게 나타나는 것을 확인 할 수 있다.

결과적으로, RECREW는 중간 사이즈의 데이터를 위한 플래쉬 디세미네이션을 지원하기 위한 적절한 오버레이 네트워크 기반의 멀티캐스팅 시스템이라는 것을 확인 할 수 있다.

III. FaReCAST

RECREW가 중간 사이즈 데이터를 위한 오버레이 네트워크 기반 플래쉬 디세미네이션 시스템이라면, FaReCAST (Fast and Reliable application layer multiCAST) 는 보다 더 작은 데이터를 보다 더 짧은 시간 내에 전송 하기위한 오버레이 네트워크 기반의 플래쉬 디세미네이션 시스템이다. RECREW또는 다른 오버레이 네트워크 기반 멀티캐스팅 시스템들이 데이터를 전송하는 동안 오버레이 네트워크의 장애를 인지하고 네트워크를 재구성하는 시간적 여유를 고려하는 것과 달리, 제한 시간이 십 수초에 불과한 플래쉬 디세미네이션을 목표로 하는 FaReCAST에서는 데이터를 전송하는 순간의 오버레이 네트워크 재구성하는데 사용할 시간적 여유가 없다. 즉, FaReCAST는 데이터 전송시 발생할 수 있는 오버레이 네트워크의 장애를 동적으로 인지하여 현재 운용 가능한 오버레이 네트워크의 부분들만을 사용하여 최상의 전송 안정성을 보장하고자 한다.

1. M2M (Multiple parents to Multiple children)

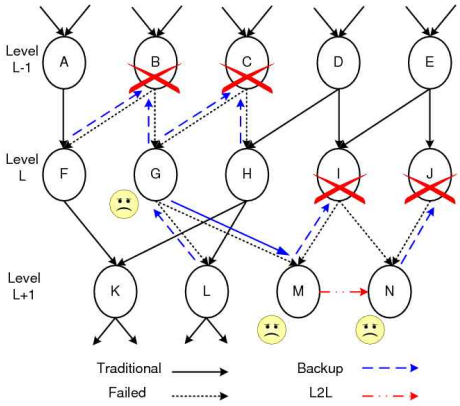
작은 크기의 데이터를 전송하는데 있어서 가장 대표적으로 사용되는 네트워크 구조는 Tree구조이다. Tree구조는 각 노드가 다수의 자식 노드들을 관리함으로서, 데이터 전송 시 inter node concurrency개념을 사용해 그 전송 속도를 크게 향상 시킨다. 하지만, Tree구조는 오버레이 네트워크 장애 발생시, 장애가 발생한 노드가 위치한 부분을 복구하기 전에는, 그 노드의 하위 부분에 위치한 노드들에게 데이터를 전송할 수 없다. 이는 각 자식 노드들이 하나의 부모 노드만을 가지고 있는 구조적 한계 때문으로, 이와 같은 문제를 single point of failure라고 부른다.

FaReCAST에서는 이와 같은 single point of failure를 해소하고, 문제 발생 시 네트워크 복구 없이 데이터를 전송할 수 있는 오버레이 네트워크를 구성하기 위해 M2M (Multiple parents to Multiple children) 구조를 사용한다. 그림 7에서와 같이 각 노드는 다수의 자식 노드를 가질 뿐만 아니라, 다수의 부모 노드를 가지게 된다. 이에 따라, 각 노드가 부모 노드로부터 받은 데이터를 자식 노드들에 전달하는 단수한 프로세스를 생각할 때, 각 노드는 동일한 메시지를 부모 노드의 개수 만큼 받게 된다. 따라서, 모든 부모 노드에 동시에 장애가 발생하지 않는 한, 데이터의 전송이 이루어진다. 이와 같은 다수의 부모 노드들과 데이터의 redundancy를 사용한 전송 개념을 Path Diversity라고 한다.

M2M의 구조는 기존의 Forest (Multiple Tree)또는 Mesh와 비슷한 Path Diversity특성을 가지고 있지만, 다음의 특성들에 의해서 차별화 된다.

· 노드의 레벨 제한 : M2M 구조는 하나의 root 노드를 가지고 있고, root부터 한 노드까지의 오버레이 네트워크 경로에 포함된 노드의 개수가 그 노드의 레벨이 된다. root노드의 레벨은 0이다. 이 레벨정보는 부모, 자식 노드를 결정할 때 사용되는 것으로, 레벨 L에 위치한 노드의 부모 노드들은 레벨 $L-1$ 노드들로 구성되고, 자식 노드들은 레벨 $L+1$ 노드들로 구성된다.

· Loop-free 특성 : M2M 구조를 사용한 데이터의 전송 경로에서 Loop가 발생하지 않는다. 이는 노드의 레벨 제한에 따른 부모/자식 노드의 선택의 결과로서, 만약 평균적으로 오버레



Pseudo Code of Multidirectional Multicasting

```
MessageListener()
  while(true)
    MSG = Listen()
    p = GetSender(MSG)
    if MSGManager.isFirstMessage(MSG)
      PM = MSGManager.InitParentMask(MSG,p)
      LP = MSGManager.InitLeaflinkPool(MSG,p)
      run MD_Multicast(p,MSG,PM,LP)
    else
      MSGManager.setParentMast(MSG,p)
      // set all leaves sharing parent p
      MSGManager.setLeaflinkinkPool(MSG,p)

Backup(MSG,PM,T) // runable
  Wait(T)
  for p in PM
    if PM.isUnset(p)
      PM.set(p)
      SendMessage(MSG,p)
```

```
MD_Multicast(p,MSG,PM,LP) //runable
  if PM.contains(p)
    if Children exist
      SendMessage(MSG,Children)
    run Backup(MSG,PM,Tw)
    run L2L(MSG,LP,Random(seed)*Tw)
  else
    if Children exist
      SendMessage(MSG,Children except p)
    run Backup(MSG,PM,0)
    run L2L(MSG,LP,Random(seed)*Tw)

L2L(MSG,LP,T) // runable
  Wait(T)
  for l in LP
    if LP.isUnset(l)
      LP.set(l)
      SendMessage(MSG,l)
```

그림 7. FaReCAST의 M2M 오버레이 네트워크 구조와 MultiDirectional Multicasting

이 노드간의 전송 딜레이가 비슷하다고 가정하면, M2M 구조에서 같은 레벨에 있는 노드들은 데이터를 비슷한 시간대에 받을 수 있게 된다.

· 부모 노드 셋의 유일성 : M2M구조에서는 임의의 노드의 부모 노드 셋이 유일할 수 있도록 한다. 즉 서로 다른 두 노드는 적어도 한 개 이상의 서로 다른 부모 노드를 가진다. 이는 부모 노드의 장애에 의한 영향 범위의 최소화를 위해 사용 된다. 이 유일성을 만족하기 위해서는 노드의 레벨 L, 부모 노드의 개수 F_i, 자식 노드의 개수 F_o가 주어질때, 한 레벨의 노드의 개수 N이 다음의 식 (2)를 만족해야 한다.

$$N=(F_o)^L+F_o+F_i-2, where\ F_o>1, F_i>1, L>0 \quad (2)$$

2. Multidirectional Multicasting

M2M구조에서 Path Diversity개념을 사용해 데이터 전송을 보다 안정적으로 할 수 있다. 하지만 이러한 Path Diversity를 사용하더라도 데이터를 받지 못하는 노드들이 발생하는데, 이와 같은 노드들을 Bypassed 노드라고 부른다. 이러한 Bypassed 노드들이 발생하게 되는 주 원인은 전통적 데이터 전송의 방향이 부모노드에서 자식노드로 향한다는 것이다. 그림 7에서 실선으로 되어있는 링크들이 전통적 데이터 전송 방식을 나타내고 있다.

Bypassed노드는 크게 세가지 경우로 나누어 생각 할 수 있다. 첫 번째는 부모 노드들 모두 장애가 생겼지만 자식노드들은 데이터를 받는 경우이다. 그림 7에서 노드 G가 이 같은 경우라 할 수 있다. 두 번째는 부모 노드가 Bypassed노드와 장애 노드들로 구성되는 경우이다. 그림 7에서 노드 M과 같은 경우이다. 세 번째 경우는 leaf 노드의 부모 노드들이 모두 장애가 생긴 경우이다. 그림 7에서 노드 N과 같은 경우이다.

이러한 Bypassed노드들에 데이터를 전송하기 위해서 FaReCAST는 Multidirectional Multicasting방식을 사용한다. 이 Multidirectional Multicasting은 자식노드에서 부모노드 쪽으로 데이터를 전송하는 Backup 전송 방식과 같은 레벨에 있는 Sibling 노드들에게 데이터를 전송하는 Leaf to Leaf 전송 방식을 의미한다.

Backup 전송 방식은 그림 7에서 긴 대쉬 화살표로 그 진행 방향을 보여주고 있다. 즉 노드 G는 전통적 데이터 전송방식을 사용할 경우 Bypassed되었지만, Backup전송방식을 사용해서 노드 L로부터 데이터를 전송받을 수 있게 된다. 노드 G가 데이터를 받을 수 있게 됨으로써 이전에 Bypasssed되었던 노드 M도 데이터를 받을 수 있게 된다.

하지만 Backup 전송방식은 자식노드가 없는 Leaf노드에는 적용할 수 없다. 일반적인 Tree구조의 경우 Leaf 노드의 개수가 전체 참여 노드의 절반에 해당하기 때문에, Bypassed Leaf 노드들에 데이터 전송은 전체 시스템의 안정성향상에 있어 아주 중요하다. 이 같은 문제를 해결하기 위해, L2L (Leaf to Leaf)전송방식을 사용할 수 있다. 그림 7에서는 노드 M에서 노드 N으로 향하는 화살표가 L2L전송 방식의 경우를 나타내 주고 있다. 즉, Leaf 노드중 자신과 같은 부모노드를 가지는 주변 노드에게 메시지를 전송하는 것이 L2L전송 방식이다.

이와 같은 Multidirectional Multicasting을 사용함으로써 예측하지 못한 장애에도 안정적인 데이터 전송이 가능해진다. 하지만, 장애가 없는 일반적인 경우에는 Multidirectional Multicasting은 불필요한 오버헤드로 생각 할 수 있다. 따라서, FaReCAST에서는 각 노드가 자신의 부모노드가 Bypassed가 되었는지를 예측하여 가능한 한 꼭 필요한 경우에만 Multidirectional Multicasting을 수행할 수 있도록 한다.

FaReCAST에서 각 노드는 M2M구조의 특성을 고려해서 Bypassed노드를 예측한다. Loop-free특성에 따라 root노드부터 임의의 노드까지의 데이터 경로들의 길이가 같기 때문에, 만약 한 노드가 새로운 데이터를 부모노드들 중 하나에서 받을 경

우, 일정 시간내에 다른 부모 노드들로부터 같은 데이터를 받게 될 것이라 예측하게 된다. 만약, 새로운 데이터를 받은 후 동일 데이터를 일정 시간내에 보내지 않은 부모 노드가 있다면, 그 부모노드는 Bypassed되었을 가능성이 있다. 그림 7의 노드 L의 경우 부모 노드 H로부터 데이터를 전송 받았지만, 또 다른 부모 노드 G로부터 데이터를 전송받지 못했다. 이와 같은 경우 노드 L은 Backup 전송방식을 시작한다. 또한 노드 M의 경우 부모노드 G로부터 데이터를 받았음에도 노드 I에서 데이터가 오지 않을 경우, 자신이 Leaf 노드이기 때문에 Backup전송 뿐만 아니라 L2L전송도 시작한다. 이처럼 FaReCast는 Multidirectional Multicasting을 장애의 정도에 따라서 Adaptive하게 시작함으로써 예측이 불가능한 장애에도 안정적이고 효과적으로 대처할 수 있다.

3. FaReCAST의 성능평가

M2M구조와 같은 Path Diversity를 이용할 경우 임의의 노드의 부모노드의 개수는 F_i가 된다. 이때, 일반적인 데이터 전송에 있어서, 한 노드의 장애가 발생할 확률이 P_f라 하면, 임의의 하나의 노드가 Bypassed될 확률은 다음의 식(3)과 같다.

$$P_{bypass} = (P_f)^{F_i} \tag{3}$$

하지만 Flash Dissemination의 경우와 같이 노드나 링크에 장애가 발생하였을 때 이를 복구할 시간이 부족한 경우에는, 한 노드가 Bypassed될 확률은 M2M구조의 root 노드부터 이 노드까지의 모든 path에서 fail이 나지 않을 확률을 고려해서 계산해야 한다. 즉, 임의의 레벨 L에 있는 노드는 루트 노드부터 현재의 레벨 L까지의 모든 노드들이 fail이 날 확률을 고려해서 메시지가 Bypass될 확률을 고려해야 한다. 만약 모든 노드의 부모 노드가 모두 다르다고 가정하면, 레벨 L노드가 Bypassed될 확률은 다음 식(4)와 같이 된다.

$$\begin{aligned} P_{bypass}^{L} &= (P_f)^{F_i} + \binom{1}{F_i}(P_f)^{F_i-1}(1-P_f)(P_{bypass}^{L-1}) \\ &\quad + \binom{2}{F_i}(P_f)^{F_i-2}(1-P_f)^2(P_{bypass}^{L-1})^2 \\ &\quad + \ldots + (1-P_f)^{F_i}(P_{bypass}^{L-1})^{F_i} \\ &= (P_f)^{F_i} + \sum_{i=1}^{F_i}\binom{i}{F_i}(P_f)^{F_i-i}(1-P_f)^i(P_{bypass}^{L-1})^i \end{aligned} \tag{4}$$

일반적인 M2M구조에서 부모노드 셋의 유일성을 보장하는 반면, 모든 노드들의 부모노드가 다르다는 점을 보장하지 않으므로, 식 (4)의 값은 노드가 Bypassed될 확률의 Low-Bound가 된다.

이때, Backup Dissemination을 사용할 경우, 자식노드의 장애가 없을 경우 자식노드로부터 데이터를 전송 받을 수 있기 때문에, 부모노드의 개수가 F_i이고 자식노드의 개수가 F_o인 노드가 Bypassed될 확률은 다음의 식(5)와 같이 정의 될 수 있다.

$$\begin{aligned} P_{bypass}^{L} &= (P_f)^{F_i+F_o} \\ &\quad + \sum_{i=1}^{F_i+F_o}\binom{i}{F_i+F_o}(P_f)^{F_i+F_o-i}(1-P_f)^i(P_{bypass}^{L-1})^i \end{aligned} \tag{5}$$

Backup Dissemination의 효과를 사용할 수 없는 Leaf 노드들을 위한 L2L전송 방식에서는, 각 Leaf 노드들의 Leafset, 즉 같은 부모노드를 가지는 Leaf 노드들의 집합을 고려해서 Leaf 노드가 Bypassed될 확률을 계산하여야 한다. 부모노드의 개수가 F_i이고 자식노드의 개수가 F_o로 설정된 M2M 구조에서는 임의의 한 Intermediate노드가 가지게 되는 평균적인 자식노드의 개수는 F_iF_o로 생각할 수 있고, 이에 따라 일반적인 Leafset의 크기는 $(F_iF_o-1)F_i$로 생각할 수 있다. 이를 고려하면 L2L전송방식에서 Leaf 노드가 Bypassed될 확률은 다음의 식 (6)과 같이 정의 된다. 이때, F_l은 $(F_iF_o-1)F_i$이다.

$$\begin{aligned} P_{bypass}^{L} &= (P_f)^{F_i+F_l} \\ &\quad + \sum_{i=1}^{F_i+F_l}\binom{i}{F_i+F_l}(P_f)^{F_i+F_l-i}(1-P_f)^i(P_{bypass}^{L-1})^i \end{aligned} \tag{6}$$

결과적으로 Multidirectional Multicasting을 사용할 경우 기존의 오버레이 기반 멀티캐스팅 시스템보다 데이터 전송의 안정성을 향상 시킬 수 있고, 부모노드의 개수 뿐만 아니라 자식노드의 개수가 데이터 전송의 안정성에 큰 영향을 미친다는 것을 알 수 있다.

실제적인 비교를 위해, 네트워크 에뮬레이션을 이용해 100,000개의 노드에 데이터를 전송할 경우, 다양한 시스템과 FaReCAST의 안정성(Reliability)와 오버헤드를 그림 8과 9에서 나타낸다. 공정한 비교를 위해, 모든 시스템에서 부모노드의 개수와 자식노드의 개수를 각각 3으로 설정하고 있다. 안정성이란 데이터 전송 시 장애가 발생했다 가정하고 오버레이 네트워크 재구성 없이 남아있는 현재 네트워크 상태를 가지고 데이터를 전송할 경우, 데이터를 받은 노드의 현재 장애상태가 아닌 노드에 대한 비율을 의미한다. 그리고 오버헤드란, 네트워크에서 사용된 메시지의 개수의 현재 장애상태가 아닌 노드에 대한 비율을 의미한다.

그림 8에서는 안정성 측면에서 같은 조건을 가진 오버레이 구조를 가지는 다양한 오버레이 네트워크 기반 스트리밍 시스템들이 20% 이상의 오버레이 장애가 발생할 경우 급격하게 그 안정성이 떨어지는 반면, FaReCAST는 40%의 오버레이 장애가 발생하더라도 그 안정성을 100%로 유지 하는 것을 확인 할 수 있다. 즉 Multidirectional Multicasting의 역할이 중요함을 알 수 있다.

그림 9에서는 시스템 오버헤드를 보여준다. 부모노드의 개수가 3일 경우 일반적인 Forest기반의 오버레이 네트워크 구조를

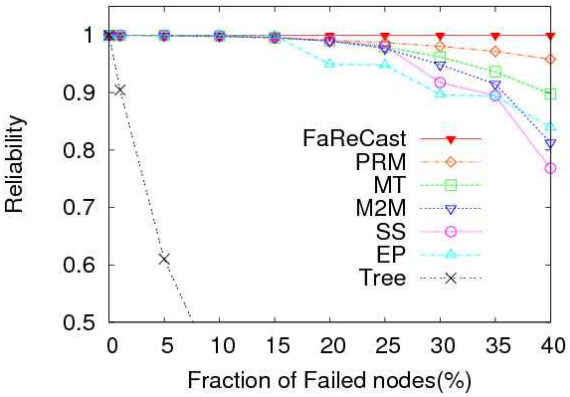


그림 8. FaReCAST와 기존 시스템 간의 안정성비교

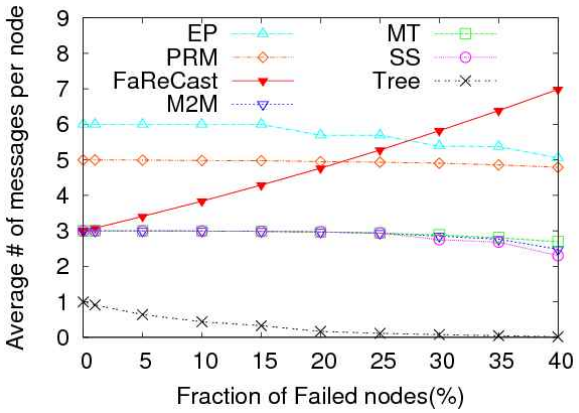


그림 9. FaReCAST와 기존 시스템 간의 오버헤드 비교

사용하는 시스템의 오버헤드는 3이 된다. 하지만 Multidirectional Multicasting은 장애를 인지하면서 적절한 전송방식이 발동하게 된다. 이에 따라 그림 9에서와 같이 장애의 정도가 약할 경우에는 메시지를 많이 사용하지 않는 반면, 심해질수록 보다 많은 메시지를 사용함으로써, 예측 불가한 다양한 장애에도 불구하고 FaReCast는 높은 시스템 안전성을 유지한다.

또한, 주목할 점은 Mesh 구조의 시스템에서 브로드캐스팅을 하는 경우는 평균적으로 높은 데이터 오버헤드를 가지지만 이러한 오버헤드가 시스템 안정성을 확실히 보장하지 못한다는 것이다. 즉 예측 불가능한 장애가 존재하는 플래쉬 디세미네이션에서는 정적인 Randomness에 의존하는 전송 시스템으로는 효과적으로 높은 시스템 안정성을 유지하기 힘들다는 것이다. 반면, FaReCAST는 발생하는 장애의 결과에 따라 필요한 경우 오버헤드를 동적으로 증가시킴으로써 보다 효과적으로 높은 안정성을 유지 할 수 있다.

IV. 결 론

시간 제한이 엄격하면서도 아주 높은 전송 안정성을 요구하는 플래쉬 디세미네이션상황에 적합한 오버레이 네트워크 기반의 멀티캐스팅 시스템을 구현하기 위해서는 오버레이 네트워크의 구조뿐만 아니라 그 전송 방식에 있어서도 새로운 개념이 필요하다. 즉, RECREW의 경우 메타데이터의 사전 전송, 메타데이터 기반의 Push-Pull Gossip프로토콜 전송방식 적용 그리고 Bounce 프로토콜을 이용한 적절한 오버레이 구성을 생각할 수 있고, FaReCAST의 경우 Path Diversity를 위한 M2M 오버레이 구조 사용과 예측되는 Bypassed 노드를 위한 추가적인 Multidirectional Multicasting 데이터 전송을 생각 할 수 있다. 이와 같은 플래쉬 디세미네이션을 위한 오버레이 네트워크 기반 멀티캐스팅 시스템들은 재해 경보/정보 관리 시스템, 긴급 정보 전송 시스템 등에서 활용될 수 있다. 또한 오버레이 네트워크기반의 시스템은 사용자들의 머신들의 리소스를 이용함으로써, 이와 같은 플래쉬 디세미네이션 관련 서비스들의 초기 구축 비용을 절감하는 효과를 기대 할 수 있다.

참 고 문 헌

[1] Y. Chu, S.G. Rao, S. Seshan, H. Zhang, "A Case for End System Multicast," *Proc. of ACM Sigmetrics*, 2000.

[2] S. Banerjee, B. Bhattacharjee, C. Kommareddy, "Scalable Application Layer Multicast," *Proc. of SIGCOMM*, 2002.

[3] A. Rowstron, A. Kermarrec, M. Castro, P. Druschel, "Scribe: The design of a large-scale event notification infrastructure," *Networked Group Communication*, pp. 30~43, 2001.

[4] D.A. Tran, K.A. Hua, T.T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," *Proc. of INFOCOM*, 2003.

[5] B. Cohen, "BitTorrent", http://www.bitconjurer.org/BitTorrent/, 2001.

[6] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron, A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment," *Proc. of SOSP*, 2003.

[7] C.D. Kosti, A. Rodriguez, J. Albrecht, A. Vahdat, "Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh," *Proc. of SOSP*, 2003.

[8] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, A.E. Mohr, "Chainsaw: Eliminating Trees from Overlay Multicast," *Proc. of IPTPS*, pp. 127~140, 2005.

[9] M. Deshpande, K. Kim, B. Hore, S. Mehrotra, N. Venkatasubramanian, "ReCREW: A Reliable Flash-Dissemination System," *IEEE Transaction on Computers, PrePrint*, 2012.

[10] K. Kim, S. Mehrotra, N. Venkatasubramanian, "FaReCast: Fast, Reliable Application Layer Multicast for Flash Dissemination," *Proc. of ACM Middleware*, 2010.
[11] K. Kim, N. Venkatasubramanian, "Assessing the impact of geographically correlated failures on overlay-based data dissemination," *Proc. of IEEE Globecom*, 2010.

저 자 소 개

김 경 백(정회원)

1999년 2월 한국과학기술원 전자전산 학사 졸업.
2001년 2월 한국과학기술원 전자전산 석사 졸업
2007년 2월 한국과학기술원 전자전산 박사 졸업
2007년~2011년 University of California, Irvine, 박사후연구원
2012년~현재 전남대학교 전자컴퓨터공학부 조교수
<주 관심분야 : 분산시스템, 미들웨어, 피어투피어 네트워크, 오버레이 네트워크, 소셜 네트워크, 모바일 클라우드 시스템.>