

P2P 미디어 스트리밍 시스템의 현황

배인한*

◆ 목 차 ◆

- | | |
|-----------|-----------|
| 1. 서론 | 4. 동적 적응성 |
| 2. 시스템 구조 | 5. 결론 |
| 3. 견고성 | |

1. 서론

미디어 스트리밍이 인터넷 트래픽의 점점 더 큰 부분을 형성하는 것처럼, 다수의 수신자들에게 미디어 콘텐츠를 효율적으로 전달하기 위한 적절한 방법이 있어야 한다. IP 멀티캐스트는 좋은 해결책이 될 수 있으나, 인터넷 상에 그것의 배치는 아직 제한된다. 따라서 최근에 연구자들은 P2P 미디어 스트리밍 시스템이라 부르는 미디어 스트리밍에 P2P 기법들을 사용하는 것을 중점적으로 다루고 있다. 그러나 효율적인 P2P 미디어 스트리밍 시스템을 구축하는 것은 피어들 구성과 효율적인 분산 트리 구축을 어떻게 하는지, 예측할 수 없는 피어 고장을 어떻게 처리하는지, 그리고 네트워크 조건 변화에 어떻게 적응하는지를 포함하는 다수의 난제를 직면한다. 이 논문은 미디어 스트리밍 시스템의 최근 연구들, 상기 난제를 처리하는 개념과 기법에 대한 개요를 제공한다.

미디어 스트리밍 응용들은 인터넷 상에서 끊임없이 증가되고 있다. 여기서 용어 '미디어 스트리밍'은 소스들이 수신자들에게 미디어 콘텐츠를 동시에 분배하고 수신자들은 미디어 콘텐츠를 다운로드하면서 미디어 콘텐츠를 재생하는 방법을 말한다. 미디어 콘텐츠 자체는 라이브이거나 녹화 재생될 수 있다. 미디어 스트리밍을 실행하는 가장 간단한 방법은 소스가 각 수

신자에게 미디어 콘텐츠를 유니캐스트 하는 것이다. 그러나 미디어 스트림들이 기본적으로 고대역폭 응용들이므로, 이 방법은 엄청난 량의 값비싼 대역폭을 소비할 것이고 소스에서 병목현상을 일으킨다. 따라서 그것은 확대할 수 없다. IP Multicast[3, 7]는 하나의 스트림으로 다수의 수신자들에게 미디어 콘텐츠를 공급할 수 있으므로 이 단점을 극복하기 위한 최적 방법이 될 수 있다. 그러나 지금까지 그것은 인터넷 상에서 아직 폭넓게 개발되지 않았다.

그러므로 P2P 기법이 서비스를 다른 수신자들에게 제공하기 위하여 수신자의 대역폭을 이용할 수 있고 IP 유니캐스트에서 중계되기 때문에 최근에 연구자들은 미디어 스트리밍에 P2P(peer-to-peer) 개념의 응용을 중점적으로 다루기 시작했다. 그러나 효율적인 P2P 미디어 스트리밍 시스템을 구축하는 것은 다음과 같은 다수의 문제에 기인하여 도전적인 일이다.

- 피어 관리와 분배 트리 구축: 미디어 콘텐츠를 많은 수신자들에게 전달하기 위하여, 대부분 P2P 미디어 스트리밍 시스템은 응용 단계 멀티캐스트 분배 트리를 구축한다. 여기서 난제는 효율적인 트리를 어떻게 얻고 피어가 결합(join)되거나 이탈(leave)할 때 이 트리를 어떻게 관리할 것인가 하는 것이다.
- 견고성: P2P 미디어 스트리밍 시스템에서, 수신자들의 행위는 예측할 수 없다. 수신자들은 어떤 시점에 그 서비스를 결합하고 이탈하는 것이 자유롭다. 더욱이, 수신자들은 네트워크 단절, 호스트 파

* 대구가톨릭대학교 컴퓨터정보통신공학부 교수

피, 또는 다른 이유 때문에 고장이 될 수 있다. 즉, 고장이 발생하였을 때 빠르고 적절하게 복구하기 위하여 견고성 기법이 제공되어야 한다. 아니면, 서비스 중단이 발생할 것이다.

- 동적 적응성: 미디어 스트리밍 세션 동안에, 네트워크 조건은 변할 수 있고, 네트워크는 혼잡해 질 수 있고 분실율은 증가할 수도 있다. 따라서 네트워크 변화에 적응하기 위하여 미디어 스트리밍 시스템을 위한 어떤 기법이 필요하다.

P2P 미디어 스트리밍 시스템에 대한 다수의 연구들이 있었다. 그러한 연구들은 상기 난제들을 해결하기 위하여 다른 해결책들을 제공한다. 이 논문의 목표는 최근 연구에서 상기 난제를 처리하는 개념과 기법을 소개하기 위하여 P2P 미디어 스트리밍 시스템의 연구에 대한 개요를 제공하는 것이다. 이 논문은 다음 P2P 미디어 스트리밍 시스템들: SpreadIt[2], Narada[1], CoopNet[5, 6], ZigZag[9, 10] 그리고 PROMISE[4]를 포함한다.

2. 시스템 구조

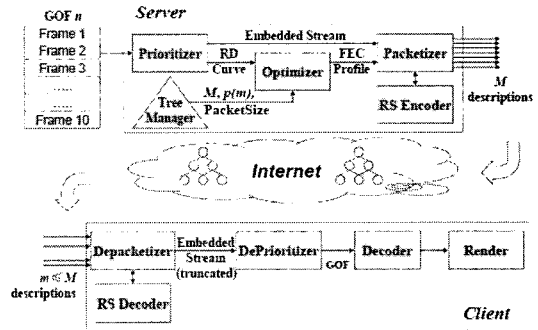
피어를 구성하고 분배 트리를 구축하는 방법에 따라, 위의 5개 P2P 미디어 스트리밍 시스템들은 다음 3 가지 범주: 중앙 서버 기반, 분산 기반, 계층 기반으로 분류할 수 있다. 여기서 중앙 서버 기반은 피어 관리와 분배 트리 구축에 책임이 있는 시스템내의 중앙 서버가 있다는 것을 의미한다. 분산 기반 시스템은 그러한 중앙 서버를 가지지 않고, 모든 피어 관리 연산과 트리 구축은 분산되어진다. 계층 기반 방식은 피어들을 다수의 계층 구조 클러스터로 구성한다. 따라서 시스템 확장성을 증가시킨다.

2.1 중앙 서버 기반 시스템

CoopNet은 이 범주에 속한다. CoopNet의 하나의 주요 목표는 복원력 있고 견고한 P2P 미디어 스트리밍을 지원하는 것이다. 그것은 네트워크 경로와 견고성을 제공하기 위하여 데이터에 중복성을 도입하였다.

CoopNet에서, 미디어 데이터 스트림 전송을 위한 대역폭이 피어들의 분산 집합에 의해 제공될 동안, 자원이 풍부한 서버는 분배 트리를 구축하고 관리에서 중심 역할을 한다. CoopNet은 소스와 모든 수신자들을 걸치는 다수의 분배 트리를 구축한다. 어떤 노드가 결함을 원할 때, 그것은 각 트리에서 지정된 부모 노드로 반응하는 중앙 서버와 접촉한다. 어떤 노드가 이탈할 때, 그것은 이탈 노드의 자식들을 위한 새로운 부모를 찾고 그것들의 새로운 부모의 식별자들을 자식들에게 공지하기 위하여 중앙 서버에게 알린다. 트리 관리는 중앙 서버에서 수행되고, 어떤 네트워크 통신을 포함하지 않는다. 따라서 이 방법은 빠른 결합과 빠른 이탈을 제공한다. 그러나 중앙 서버는 모든 분배 트리의 완전한 지식을 유지해야 하므로, 그것은 그 서버에 무거운 제어 부하를 줄 것이다. 그러므로 확장성이 매우 좋지 않다. 이 방법의 다른 문제는 중앙 서버가 고장의 하나의 포인트를 만든다는 것이다.

그림 1은 CoopNet MDC 시스템 구조를 보여준다. 입력 스트림은 계층화된 코덱으로부터 나온다. 연산 순서는 다음과 같다.



(그림 2) CoopNet 시스템 구조

- ① GOF 내의 프레임들은 율 변화 정보를 가지고 있는 데이터 단위 집합으로 분할된다. 우선순위는 신호 왜곡을 감소시키기 위하여 그것들의 기여도에 따라 그러한 데이터 단위들의 우선순위를 결정하고 정렬한다.
- ② 최적화기는 클라이언트로부터 받은 피드백에

기초하여 $p(m)$ 분포를 계산한다.

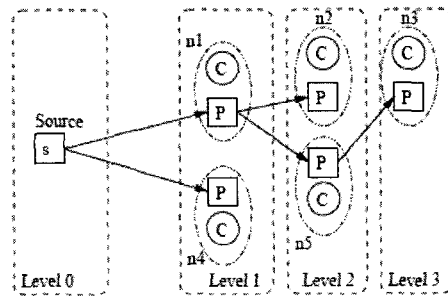
- ③ description의 개수(M), 패킷 크기(P), $p(m)$ 분포, 그리고 RD 곡선을 사용하여, 최적화기는 최적 패킷을 위한 우선순위 부호화 프로파일, R_1, \dots, R_M 을 생성한다.
- ④ 패킷화기 FEC는 최적화기로부터의 FEC 프로파일에 따라 최적화기에 의해 생성된 임베디드 스트림을 부호화하고, M 패킷을 생성한다. GOF 번호는 그것의 헤더에 포함된다. 스트리밍 서버는 그러한 패킷들을 그것의 트리에 분산한다.
- ⑤ M 패킷들은 다수의 CoopNet 트리를 운영하고 어떤 클라이언트 도착에 있어서 다른 지연을 경험할 수도 있다. 어떤 클라이언트에서 수신된 패킷들은 그것들의 헤더에 포함된 GOF 번호(n)을 사용하여 동기화 되어진다.
- ⑥ 어떤 GOF 내의 M 패킷의 부분집합을 수신하면, 클라이언트 FEC에서 역패킷화기는 받은 패킷들을 해독하고, 임베디드 스트림을 조립한다.
- ⑦ 역패킷화기는 임베디드 스트림으로부터 각 데이터 단위들을 검색하고 그것들을 미디어 해독 시점에 따라 정렬한다. 재구축된 GOF의 품질은 수신된 description의 개수에 의존한다.
- ⑧ 마지막으로, 미디어 재생기는 그 GOF를 해독하고 클라이언트에서 그것을 재생한다.

2.2 분산 기반 시스템

SpreadIt, Narada, 그리고 PROMISE는 분산 기반 시스템에 속한다. 그러나 그것들은 피어를 구성하고 분배 트리를 구축하기 위하여 다른 방법을 사용한다.

- SpreadIt: SpreadIt은 피어들의 집합 위에 하나의 멀티캐스트 분배 트리를 구축한다. 각 피어는 멀티캐스트 트리를 설립하고 관리하기 위하여 다른 피어들과 조정하는 피어링 계층을 실행한다. 즉, 그것은 분산 방식으로 그 트리를 구축하고 관리한다. 어떤 새로운 수신자가 결합을 원할 때, 그것은 불포화된

대역폭을 갖는 노드를 찾을 때 까지 소스로부터 하향으로 그 트리 노드들을 운행한다. 어떤 수신자가 이탈을 원할 때, 만일 그것이 자식을 가지고 있다면, 그것은 그 자식들을 부모 또는 소스와 연결을 요구하는 메시지를 전송할 것이다. SpreadIt의 문제는 어떤 고장이 발생한 경우에 SpreadIt은 포함된 소스를 가지고 있어야 한다는 것이다. 따라서 소스에서 심한 병목현상에 기인하여 혼잡에 취약하다.



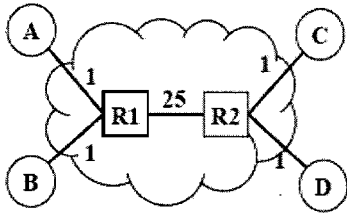
N = {n1, n2, n3, n4, n5}, P: Peering Layer, C: Application Client

(그림 3) 피어 상에 구축된 응용 단계 멀티캐스트 트리

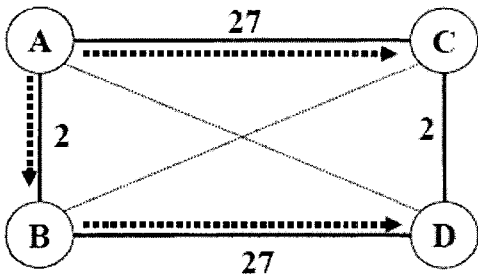
그림 2에서 보여 지는 s에서 루트 된 멀티캐스트 트리는 노드들의 집합 N 상에 구축된다. 노드들은 다른 단계들로 구성된다. 단계 $l+1, l=0, 1, 2, \dots$,에서 각 노드 n에 대해, 단계 l에서 그것의 부모 p라 부르는 하나의 노드가 있다. n은 p의 자식이라 부른다. p에서 루트 된 부 트리의 모든 노드들을 그것의 자손이라 한다. 데이터 전송 세션은 어떤 노드와 그것의 각 자식들 간에 설립되어진다. 단계 l에서 각 노드 n은 그 스트림을 단계 l+1의 그것의 모든 자식들에게 전송한다.

- Narada: Narada는 다중-송신자, 다중-수신자 스트리밍 응용을 중점적으로 다룬다. 그것은 2-단계 프로세스로 분배 트리를 구축한다. 첫째, 그것은 그 노드 쌍들 간의 연결에 의해 형성된 더 풍부한 연결 응용-단계 망사(mesh)를 구축한다. 그 망사에서 링크는 그 망사의 품질을 향상시키기 위하여 주기적으로 감시되어진다. 두 번째 단계에서, 어떤 송신자가 수신자들의 집합에 미디어 콘텐츠를 전송하기를 원할 때, Narada는 효율적인 멀티캐스트 분배 트리

를 구축하기 위하여 그 망사 상에서 역 경로 전송 알고리즘을 실행한다. Narada에서, 모든 멤버는 그 그룹 내의 모든 다른 멤버들의 리스트를 관리한다. 그리고 그 망사 내의 그것의 이웃들과 그룹 멤버쉽 지식을 주기적으로 교환한다. 어떤 노드가 그 그룹에 결합하기를 원할 때, 그것은 out-of-band bootstrap 메커니즘으로 그룹 멤버의 리스트를 얻어야 하고, 그것은 어떤 이웃으로 추가되기 위하여 요청하는 메시지를 전송한다. 어떤 노드가 그 그룹을 이탈하기를 원한다면, 그것은 그것의 이웃에게 공지하고, 그리고 이 정보는 그 망사에 따라 그 그룹의 다른 멤버들에게 전달된다. 이것 때문에, Narada는 대형 네트워크를 지원할 수 없고, 그것은 단지 작은 P2P 네트워크에 역점을 둔다.



(그림 3) 물리적 위상



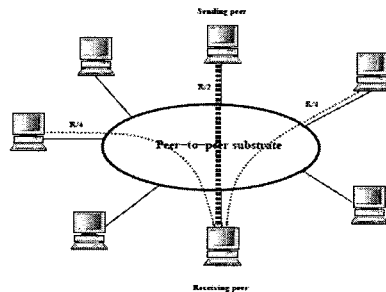
(그림 5) Narada에서 망사 기반 방법의 예

그림 4는 Narada가 그림 3에서 보여 지는 물리적 위상을 위하여 구축한 망사와 노드 A에서 루트된 스페닝 트리를 보여준다.

- PROMISE: PROMISE는 “다수의 송신자로부터 미디어 스트림 데이터를 모으는 하나의 수신자”의 패턴에서 다른 4가지 P2P 미디어 스트리밍 시스템과

아주 다르다. 각 미디어 스트리밍 세션에 대해, 다수의 전송 피어들은 요청하는 피어를 서비스하기 위하여 협력한다. PROMISE는 피어들 간의 연결성 관리, 피어 멤버쉽 관리, 그리고 객체 검색 수행을 책임지는 P2P 기판(substrate)의 상단에 배치된다. 다른 미디어 스트리밍 시스템과 달리, PROMISE는 미디어 콘텐츠를 전달하기 위하여 멀티캐스트 분배 트리를 구축하지 않는다. 대신, 어떤 피어가 어떤 미디어 데이터를 요청할 때, 그것은 먼저 하부 P2P 기판에 검색 요청을 제출하고, 그 미디어 데이터를 갖는 후보 피어들의 집합이 반환될 것이다. 그리고 그것은 스트리밍 세션 동안 최적 품질을 생성할 것 같은 후보들의 최적 부분집합인 활동적인 송신자 집합을 결정하기 위하여 위상-인식 선택 알고리즘(topology-aware selection algorithm)을 사용한다. 그 다음, 그 수신자는 그 활동적인 집합 내의 모든 피어들과 병렬 연결을 설립한다. 연결이 설립되자마자, 그 수신자는 각 활동적인 송신자들에게 송신율과 데이터 할당을 배정한다. 그리고 미디어 스트리밍을 시작한다. PROMISE에서, 노드 결합과 이탈은 하부 P2P 기판에 의해 처리된다.

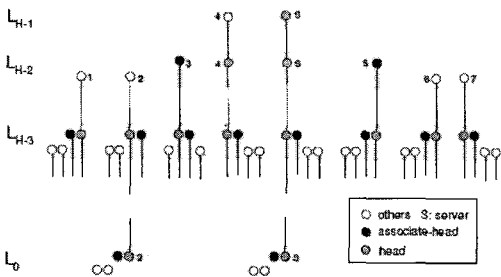
PROMISE 구조는 그림 5와 같이 P2P 기판을 통하여 상호 연결된 피어들의 집합으로 구성된다. P2P 기판은 피어들 간의 연결성을 유지하고, 피어 멤버쉽을 관리하고, 그리고 객체 검색을 수행한다. PROMISE 연산은 하부 P2P 기판에 독립적이다. 따라서 PROMISE는 P2P 기판의 상부에 배치될 수 있다. 각 P2P 기판들은 어떤 객체 검색 요청에 대해 단지 하나의 피어를 반환한다.



(그림 6) PROMISE 구조

2.3 계층 기반 시스템

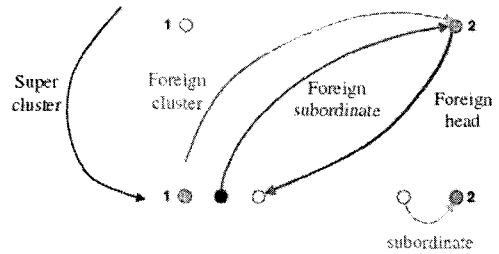
ZIGZAG는 계층 기반 시스템이다. 그것은 인터넷 상의 많은 수신자들을 지원할 수 있는 단일-소스 미디어 스트리밍을 중점적으로 다룬다. 좋은 확장성을 얻기 위하여, ZIGZAG는 피어들을 클러스터들의 다중-계층구조로 재귀적으로 구성하고, 클러스터의 크기는 $3k$ 로 한정되어진다. 여기서 k 는 상수이다. 각 하위 계층 클러스터는 그 클러스터의 헤드로 어떤 피어를 선택하고, 이 헤드는 상위 계층 클러스터의 멤버가 된다. 그 소스는 그것이 거주하는 클러스터의 헤드로 항상 선택된다. 따라서 그것은 항상 최상위 계층에 있다. ZIGZAG는 이 다중-계층구조를 ‘관리 구조 (administrative organization)’라 부른다. 그러나 이 관리 구조는 데이터 전달 위상을 의미하지 않는다. 대신, ZIGZAG는 미디어 스트림 전달을 실행하기 위하여 관리 구조에 기초하여 멀티캐스트 분배 트리를 구축한다. 이 멀티캐스트 트리 구축의 핵심은 그 콘텐츠를 그 클러스터 내의 다른 멤버들에게 전송하기 위하여 어떤 클러스터의 헤드를 제외한 외부 헤드(foreign head)의 사용이다. 어떤 클러스터의 외부 헤드는 그 클러스터의 헤드의 헤드인 동료 클러스터이다. ZIGZAG는 멀티캐스트 트리가 높이 $O(\log_k N)$ 와 노드 차수 $O(k^2)$ 를 가진다. 여기서 N 은 수신자들의 개수이고 k 는 상수이다. 따라서 ZIGZAG는 좋은 확장성을 가지고 많은 수신자들을 지원할 수 있다.



(그림 7) 피어 관리 구조

그림 6의 예에서처럼, 최상위 계층(계층 H-1)에서, 서버 S는 헤드와 부 헤드를 가진다. 피어 3은 계층

H-2에서 어떤 클러스터의 부 헤드이고 계층 H-3에서 어떤 클러스터의 헤드이다. 피어 4는 계층 H-2에서 어떤 클러스터의 헤드이고, 역시 계층 H-1에 속한다.



(그림 8) 클러스터와 피어 간의 관계

그림 7은 클러스터와 피어 간의 관계를 보여준다.

- 종속자(Subordinate): 어떤 피어 X가 헤드인 어떤 클러스터의 헤드가 아닌 피어들을 X의 종속자라 한다. 그림 6에서 피어 1, 2, 3은 피어 4의 종속자이고, 피어 4, 5, 6, 7은 그 서버의 종속자들이다.
- 외부 헤드(Foreign head): 계층 $j > 0$ 에서 어떤 피어 X의 헤드가 아닌 동료 클러스터 Y를 X의 계층 $-(j-1)$ 종속자들의 외부 헤드라 한다. 그림 6에서 피어 4는 피어 5, 6, 7의 외부 헤드이다.
- 외부 종속자(Foreign subordinate): X의 계층 $-(j-1)$ 부 헤드를 위에서 언급한 Y의 외부 종속자라 한다. 그림 6에서, 피어 5는 피어 4의 외부 종속자이다.
- 외부 클러스터(Foreign cluster): X의 계층 $-(j-1)$ 클러스터를 위에서 언급한 Y의 외부 클러스터라 한다. 그림 6에서 헤드가 피어 1인 계층 $-(H-3)$ 클러스터는 피어 2, 3, 4의 외부 클러스터이다.
- 슈퍼 클러스터(Super head): 클러스터 U의 헤드가 다음 상위 단계에서 클러스터 V에 나타나고 가정하자. V를 U의 슈퍼 클러스터라 한다.

멀티캐스트 트리와 관리 구조에서 위치와 연결을 관리하기 위하여, ZIGZAG내의 모든 피어는 그것의 동료 클러스터들, 그 멀티캐스트 트리상의 그것의 자

식들과 부모와 주기적으로 통신한다. 이 주기적인 메시지는 피어 고장을 탐지하는데 사용될 수 있다. 어떤 피어가 결합을 원할 때, 그것은 소스에 요청을 제출한다. 그리고 이 결합 요청은 결합을 위한 적절한 피어를 찾을 때 까지 그 소스로부터 하향으로 멀티캐스트 트리를 따라 운행된다. ZIGZAG는 결합 오버헤드가 $O(\log_k N)$ 이라는 것을 보장한다. 어떤 피어가 이탈을 원할 때, 그것은 영향을 받는 피어들을 위해 새로운 피어를 찾는다. 그리고 이것은 $O(k^2)$ 에 실행된다는 것을 보장한다.

3. 견고성

P2P 미디어 스트리밍 시스템에서, 피어는 어떤 시점에 고장일 수 있다. 만일 어떤 고장된 피어가 현재 다른 피어들에게 미디어 콘텐츠를 전송 중이면, 이 미디어 콘텐츠를 수신하는 그 피어들은 중단될 것이다. 따라서 어떤 복구 메커니즘이 예측할 수 없는 피어 고장을 처리하기 위하여 필요하다. 이 논문에서 논의된 5가지 미디어 스트리밍 시스템 모두는 복구 메커니즘을 제공한다.

3.1 피어 고장 탐지와 복구

피어 고장을 처리하기 위한 첫 번째 일은 가능한 한 빨리 고장을 탐지하는 것이고, 그리고 영향을 받는 서비스를 복구한다. SpreadIt, Narada, 그리고 ZIGZAG는 피어 고장을 탐지하기 위하여 약간의 "heart-beat" 메커니즘을 사용한다. 더 자세하게, SpreadIt에서 어떤 피어는 그것의 부모와 자식들에게 alive 메시지를 주기적으로 전송한다. 만일 어떤 피어가 어떤 자식이 잠시 동안 alive 메시지를 전송하지 않았다는 것을 탐지하면, 그것은 그 자식이 고장된 것으로 가정하고, 데이터-전송 세션을 종료한다. 만일 어떤 피어가 그것의 부모가 잠시 동안 alive 메시지를 전송하지 않았다는 것을 탐지하면, 그것은 그 부모가 고장된 것으로 가정하고, 그것의 새로운 부모로 소스로부터 하향으로 불포화된 대역폭을 갖는 어떤 피어를 찾아본다. Narada와 ZIGZAG에서, alive 메시지는 요구되지 않는다. 그

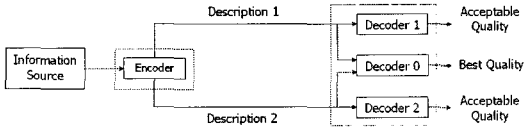
러므로 피어들은 다른 피어들에게 refresh/exchange 메시지를 주기적으로 전송할 것이다. 어떤 피어가 다소의 시간 동안에 다른 피어로부터 refresh/exchange 메시지를 수신하지 못했다면, 그것은 그 피어가 고장된 것으로 가정한다. 피어 고장이 탐지되었을 때, Narada는 망사를 통해 고장 정보를 전달할 것이고 가능한 망사 분할을 수리해 본다. ZIGZAG에서, 그것은 영향을 받는 피어들을 위해 새로운 부모를 찾아본다. ZIGZAG는 최악 경우에, 고장에 기인하여 재연결을 위하여 필요한 피어의 개수는 $O(k^2)$ 이라는 것을 보장할 수 있다.

CoopNet과 PROMISE는 피어 고장을 탐지하기 위하여 다른 방법을 사용한다. 피어들은 입력 스트림의 패킷 분실율 또는 비트율을 감시하고, 패킷 분실율이 임계치를 초과하거나 비트율이 아주 떨어지면 그 부모가 고장된 것으로 가정한다. 고장이 탐지되면, CoopNet의 피어는 새로운 부모를 요청하기 위하여 중앙 서버와 접촉할 것이다. 반면에, PROMISE에서 피어는 새로운 피어를 선택하기 위하여 위상-인식 선택 알고리즘을 실행할 것이고 새로운 피어를 활동적인 송신자 집합에 넣는다.

3.2 MDC와 FEC

피어 고장 탐지와 복구 이외에, CoopNet과 PROMISE는 견고성을 향상시키기 위하여 여분의 방법들을 제공한다. CoopNet은 데이터에서 중복성을 제공하기 위하여 MDC(multiple description coding)과 네트워크 경로에서 중복성을 제공하기 위하여 다수의 다양한 분배 트리들을 사용한다. CoopNet은 미디어 콘텐츠를 위해 MDC 방법을 사용한다. 이 방법에서, 미디어 신호는 다수의 분리된 스트림들 또는 description으로 부호화되므로 그것들의 모든 부분집합은 복호화 할 수 있다. CoopNet은 소스와 모든 수신자들을 걸치는 다중 분배 트리들을 구축하고, 각 트리는 미디어 신호의 분리된 description을 전송한다. 그러므로 수신자는 최적 경우에 모든 description을 수신할 수 있다. 어떤 피어 고장은 그것의 후손 피어들이 일부 description 분실만을 일으킨다. 그 후손들은 소스

부담 없이 품질 희생으로 그것들의 서비스를 여전히 지속할 것이다.



(그림 9) MDC 구조

PROMISE는 미디어 파일을 부호화하기 위하여 FEC(Forward Error Correction)를 사용한다. 이것은 제한된 피어 고장을 감내하기 위한 다소의 기능을 제공한다. 더 자세하게, 미디어 파일은 같은 길이 데이터 세그먼트들로 나누어진다. 각 세그먼트는 FEC를 사용하여 별개로 보호된다. 그러므로 피어 고장에 기인하여 다소의 패킷이 분실될지라도, 그 수신자는 나머지 패킷을 사용하여 원래 미디어 신호를 재구성할 수 있다.

4. 동적 적응성

스트리밍 세션 동안에, 네트워크 환경은 변할 수 있다. 네트워크 경로들은 혼잡 될 수 있고 분실율은 증가할 수도 있다. 수신측에서 좋은 스트리밍 품질을 유지하기 위하여, 동적 적응성 방법이 필요하다.

4.1 동적 적응성 방법

SpreadIt의 논문은 그것이 어떤 동적 적응성 방법이 있는지 없는지를 언급하지 않았다. Narada에 대해, 그것은 네트워크 역학관계에 따라 오버레이 링크들의 추가하고 폐기함으로써 망사 품질을 점차적으로 향상시키는 것을 허용한다. 그러므로 Narada에서 멀티캐스트 분배 트리가 그 망사에 기초하여 구축되어진다. 정제된 망사는 미디어 스트리밍 품질을 증가시킬 것이다. CoopNet에서, 그것은 스트리밍 서버가 클라이언트 description 정보를 주기적으로 수집하고, 네트워크 역학관계와 클라이언트 인기도에 적응하기 위하여 이 정보를 MDC 최적화기에 제공하는 MDC에 기초한 동적 적응성 메커니즘을 사용한다. ZIGZAG에서, 관리

구조와 멀티캐스트 트리는 클라이언트들에게 더 좋은 서비스 품질을 제공하기 위하여 주기적으로 재구성될 수 있다. 더 자세하게, 만일 어떤 피어가 많은 지식들을 바쁘게 서비스하고 있다면, 그것은 다소의 지식들의 그것의 친자관계를 덜 바쁜 다른 헤드가 아닌 동료 클러스터로 교체할 수 있다. 그러므로 스트리밍 품질을 증가시킨다. PROMISE는 2단계 동적 적응성 메커니즘을 제공한다. 어떤 수신자가 네트워크가 현재 분실 허용 수준 보다 많은 패킷들이 폐기된다는 것이 인지되면, 그것은 새로운 분실 허용 수준을 계산한다. 만일 이 새로운 분실 허용 수준이 어떤 임계치를 초과하지 않는다면, 그것은 각 활동적인 송신자를 위해 새로운 전송률과 데이터 할당을 계산하고 그것들에게 새로운 전송율에서 새로운 데이터 할당을 전송하는 것을 요구한다. 아니면, 새로운 활동적인 송신자 집합이 위상-인식 선택 알고리즘을 사용하여 선택되어지는 동적 교체되는 활동적인 송신자들이 발생할 것이다.

5. 결론

효율적인 미디어 스트리밍 시스템을 구축하는 것은 피어 구성과 효율적인 분배 트리 구축을 어떻게 하는지, 피어 고장을 어떻게 처리하는지, 그리고 네트워크 역학관계에 어떻게 적응하는지를 포함하는 다수의 난제를 직면한다. 이 논문은 5 개의 최근 P2P 미디어 스트리밍 시스템을 조사하였다.

피어를 구성하고 분배 트리를 구축하는 방법에 따라, 5개 P2P 미디어 스트리밍 시스템들은 3가지 범주: 중앙 서버 기반, 분산 기반, 그리고 계층 기반으로 분류될 수 있다. 일반적으로, 계층 기반 시스템은 좋은 확장성을 갖는 경향이 있다. 반면에, 중앙 서버 기반 방법은 서버에 무거운 제어 부하를 주기 때문에 좋은 확장성을 가질 수 없다. 대형 P2P 기판에 기초하지 않는 분산 방식에 대해, 주기적 메시지에 기인한 네트워크 오버헤드 때문에 확장성은 좋지 않다. 중앙 서버 기반 방법은 피어 결합과 이탈이 빠르다는 장점이 있다. 그러나 중앙 서버는 고장의 하나의 포인트를 만든다.

피어 고장을 탐지하기 위하여, 2가지 방법이 사용

될 수 있다. 하나는 피어들에 의해 전송되는 주기적 메시지에 기초한 *heart-beat* 메커니즘이다. 다른 하나는 입력 스트림의 패킷 분실을 또는 비트율을 감시하는 것이다. 피어 고장이 탐지되면, 시스템은 영향을 받는 서비스를 복구해야 한다. 복구 방법은 특정 시스템에 의존한다. MDC와 FEC는 견고성을 제공하기 위하여 사용될 수 있다. 그러나 그것들은 스트리밍 전에 미디어 데이터를 부호화해야 한다. 그러므로 그것들은 실시간 미디어 스트리밍에 매우 적합하지 못하다.

동적 적응성 메커니즘에 대해, 기본 개념은 네트워크 상황을 감시하고 수신자 피드백을 수집하고, 네트워크 역학관계에 적응하는 것이다.

참고 문헌

- [1] Yang-Hua Chu, Sanjay G. Rao, and Hui Zhang, "A Case for End Systems Multicast," In *ACMSIGMETRICS*, 2000, pp. 1-12.
- [2] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming Live Media over a Peer-to-peer Network," Technical report, Stanford University, 2001.CPSC538 Computer Systems Course Project Report 8
- [3] S. Deering. Host Extension for IP Multicasting, RFC1112. August, 1989.
- [4] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: Peer-to-Peer Media Streaming Using CollectCast," In Proc. of the 11th ACM international conference on Multimedia, Berkeley, CA, USA, November, 2003.
- [5] V. Padmanabhan, H. Wang, and P. Chou, "Resilient Peer-to-Peer Streaming," 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, November, 2003.
- [6] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," In Proc. of NOSSDAV'02, Miami Beach, FL, USA, May 2002.
- [7] B. Quinn and K. Almeroth. IP Multicast Applications: Challenges and Solutions, Internet Engineering Task Force (IETF) Internet Draft. March 2001.
- [8] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," In Proc. of 18 IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany, November 2001.
- [9] D. Tran, K.Hua, and T. Do, "A Peer-to-Peer Architecture for Media Streaming," IEEE Journal of Selected Areas in Communications, Vol. 22, No. 11, pp. 121-133, Jan. 2004.
- [10] D. Tran, K.Hua, and T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," In Proc. of IEEE INFOCOM'03, San Francisco, CA, USA, April 2003.
- [11] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava, "On Peer-to-Peer Media Streaming," In Proc. of IEEE ICDCS'02, Vienna, Austria, July 2002.

◎ 저 자 소개 ◎



배 인 한

1986년 경남대학교 전자계산학과(공학사)

1986년 중앙대학교 대학원 전자계산학과(이학석사)

1990년 중앙대학교 대학원 전자계산학과(공학박사)

1996년~1997년 Department of Computer Science and Engineering, The Ohio State University(Post-Doc)

2002년~2003년 Department of Computer Science, Old Dominion University (Visiting Professor)

1989~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수

관심분야: 모바일 멀티미디어, 모바일 컨버전스, 모바일 컴퓨팅, 무선 인터넷 등

E-mail: ihbae@cu.ac.kr