네트워크 자원의 효율적인 사용을 위한 ISP와 P2P 어플리케이션의 협력 방안

변해선*, 이미정*, 이종화**, 강신각**
*이화여자대학교 컴퓨터공학과
**한국전자통신연구원
e-mail:*ladybhs@ewhain.net, *lmj@ewha.ac.kr
**jhyiee@etri.re.kr, **sgkang@etri.re.kr

The Cooperation Approach of Internet Service Provider and P2P applications for Effective Network Resources Usage

Haesun Byun*, Meejeong Lee
**Jonghwa Lee, **Singak Kang,
*Dept of Computer Science & Engineering, Ewha Womans University
Electronics and Telecommunications Research Institute

요 약

P4P는 P2P 통신을 함에 있어서 가장 적절한 피어를 선택하여 P2P 어플리케이션의 성능을 향상 시키면서 네트워크 자원의 효율적인 사용을 극대화하기 위해 제안된 프레임워크이다. 본 논문에서는 P4P 프레임워크에 기존 CoolStreaming 시스템을 적용하는 방안을 설명하고, ISP 개체와 P2P 시스템 개체의 협력 동작과정 및 그 개체들이 유지/관리하는 정보 등을 설명한다.

1. 서론

P2P 기술은 대규모의 네트워크 어플리케이션을 지원하기 위한 주요 기술로서 많은 주목을 받고 있지만, 대부분의 P2P 어플리케이션들은 효율적인 네트워크 자원 사용을 고려하지 않고 있기 때문에 P2P 통신에 있어서 효율적이고 공정한 네트워크 자원 활용은 해결해야 할 새로운 문제로 대두되고 있다. 이를 위해, DCIA(Distributed Computing Industry Association)의 P4P(Provider Portal for Applications) 워킹 그룹에서는 P2P 어플리케이션 측면에서 가능한 한 어플리케이션 성능을 최대화 시키고, 네트워크 서비스 제공자 측면에서 어플리케이션 요구사항을 만족시키면서 네트워크 투자 및 운용비용을 감소하기 위해 네트워크 자원을 효율적이고 공정하게 사용하기 위한 협력 프레임워크인 P4P 방안을 제시하였다[1].

본 논문에서는 P4P 프레임워크에 대한 간단한 설명과 함께, 현재 상용화와 아카데믹 버전 모두에서 알려져 있는 P2P 시스템인 CoolStreaming[2]의 동작과정을 설명한다. 또한 CoolStreaming에 P4P 프레임워크를 적용한 구조와 각 개체에서의 협력 동작과정을 살펴본다.

2. 관련 연구

P4P[1]의 목적은 가장 적절한 피어를 선택하여 통신함으로써 P2P 어플리케이션의 성능을 향상 시키고, 효율적인 네트워크 자원의 사용을 이끌어내기 위한 것이다. P4P 구조에서, 각 네트워크 제공자는 그 네트워크와 관련한 모든 정보, 즉, 정책, 성능(Capability), 피어간 비용(cost) 또는 거리(distance) 등의 정보를 관리하는 포털서버와 같은 역할을 담당하는 iTracker를 유지한다. 이 iTracker는 P2P 어플리케이션 시스템의 서버인 appTracker(또는 피어)와 서로 통신하는데, 이때 자신이 관리하는 정보를 전달한다. appTracker는 서로 협력할 가장 적당한 피어를 선택하는데 있어서 iTracker(또는 피어)로부터 받은 정보를 이용한다. appTracker가 가장 적당한 피어를 선택하기 위해서는 iTracker로부터 받은 정보가 네트워크 상황을 적절하게 반영해야 하며, 이를 위해 iTracker는 추구하고자 하는 목적에 따라 피어와 피어간 거리 및 비용을 최소화해야 한다. 이를 위해, P4P[2]에서는 최대 링크 활용율(Maximum Link Utlization:MLU)을 최소화하기 위한 목적으로 최적화된 P_{ij}(피어 i로부터 j까지의 distance)를 계산하기 위해 $p_e \geq 0$, $e \in E$의 이중 변수(dual variable)을 도입하여 라그랑쥐 듀얼 함수를 정의하였다. 그밖에 대역폭-거리(Bandwidth-Distance Produce:BDP)의 비용을 최소화 하거나, underlying 트래픽이 피크 대역폭(Peak Bandwidth)에 도달했을 때 MLU나 BDP를 최소화 하기 위한 함수, 인터 도메인간 멀티호밍 비용을 제어하기 위한 사용-기반 부가 스킴(Usage-based charging scheme)을 정의하였다.

"이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. 2009-0060034)."

CoolStreaming[2]은 메시 기반의 스트리밍 전달 구조의 대표적인 시스템으로 2004년 처음 소개된 이후, 메시와 트리를 혼합한 오버레이 구조로 발전되었다. CoolStreaming 시스템은 오버레이 구조의 부분 뷰(partial view)를 유지하고 있는 멤버십 매니저, 피어들과 파트너쉽을 설립 유지하면서 버퍼맵을 주기적으로 교환하는 파트너십 매니저, 스트리밍 데이터의 전달을 책임지는 스트림 매니저로 구성되어 있다.

3. P4P-enabled CoolStreaming 시스템

본 연구에서는 CoolStreaming 시스템에 P4P를 적용한 구조와 CoolStreaming의 적용을 위해 iTracker 및 appTracker에서 유지해야 하는 정보 및 동작과정을 살펴본다. 그림 1은 P4P-enabled CoolStreaming 시스템의 피어링 동작과정을 보이고 있다.

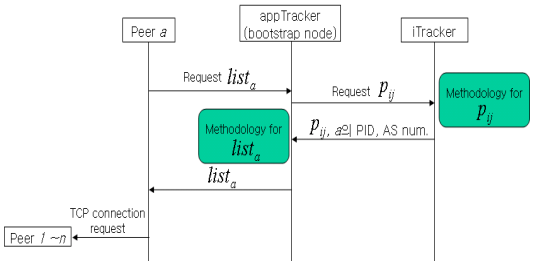

그림 1. 피어링 동작과정

피어 a는 CoolStreaming의 부트스트랩 서버인 appTracker에게 등록 요청을 한다. appTracker는 iTracker에게 P_{ij}를 요청한다. P_{ij}는 PID-i로부터 PID-j의 P4P-distance를 의미한다. P4P에서 PID(Peer ID)는 다양한 타입으로 정의될 수 있는데, 가장 일반적인 타입으로 여러 피어들의 집합을 대표하는 군집 노드(aggregation node)의 ID, 예를 들어, POP(Point of presence)를 가정할 수 있다. iTracker는 자신의 ISP 내에 모든 P_{ij}를 계산하여 appTracker에게 전달한다. P_{ij}의 계산은 네트워크 상태에 따라 주기적으로 동적 정보를 적용하여 계산하거나, 또는 OSPF weights이나 BGP preferences와 같은 정적 정보를 이용하여 계산할 수 있다. appTracker는 iTracker로부터 받은 P_{ij}값을 이용하여 등록 요청한 피어와 파트너십 또는 멤버십을 맺을 피어들을 선택하여 등록 요청한 피어 a에게 그 리스트를 전달한다.

- iTracker에서의 P_{ij}값을 계산하기 위한 Methodology

P4P-enabled CoolStreaming에서는 정적정보와 동적정보를 동시에 이용하여 P_{ij}를 계산한다. 먼저, 정적정보인 OSPF Weights과 BGP Preferences를 수집하여 P_{ij}를 계산하기 위한 P_e(링크 e에 대한 비용)로 사용한다. 이때 주기적으로 링크 e에 대한 활용률(utilization)인 u_e를 계산하여 u_e가 링크 e에 대한 threshold인 T_e를 초과하면 그 P_{ij}는 무한대로 설정한다.

P2P 통신에 참가하는 피어들은 서로 다른 ISP에 속할 수 있다. 따라서 임의의 ISP에 속하는 iTracker는 다른 ISP의 링크 비용이나 선호도 또는 링크 활용률 등을 알 수 없기 때문에 모든 피어들 간 P_{ij}를 계산하기 어렵다. 그림 2는 임의의 AS내의 iTracker가 P_{ij}를 계산할 수 있는 범위를 표현하고 있다. Intra AS에 있는 PID를 위한 P_{ij}는 앞에서 설명한 바와 같이, OSPF weights과 주기적으로 수집한 u_e를 기반으로 계산하고, Inter AS간 PID를 위한 P_{ij}는 BGP 선호도(preference) 및 비용에 따라 타 AS로부터 다운로딩 받기 원하는 선호도 정도를 제시해주는 것으로 가정하였다.

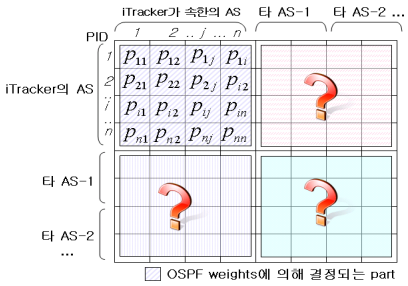

그림 2. P_{ij}의 계산범위

- appTracker에서의 P_{ij}값을 기반으로 피어링 할 피어를 선택하기 위한 Methodology

appTracker는 3단계를 통해 m(시스템 파라미터)개의 피어를 선택한다. 1단계로, Intra-PID 피어 선택은 등록 요청한 피어 a의 Intra-PID 내에서 동일 스왐(채널)에 속하는 피어들 중 랜덤으로 최대 m의 70%까지 선택한다. 2단계로, Inter-PID 피어 선택은 1단계를 포함하여 m의 80%까지 선택하며, 이를 위해 수식(1)과 (2)에서와 같이 PID-i로부터 PID-j까지의 weight인 w_{ij}를 계산한다. 3단계에서는 Inter-AS 피어 선택으로 1,2단계에서 선택된 피어를 포함하여 m에 도달할 때까지 해당 AS내에서 등록 요청하는 피어 a가 참여하기 원하는 AS의 선호도를 따라 그 AS내에서 피어를 랜덤하게 선택한다.

$$w_{ij} = \frac{1}{p_{ij}}, \text{ for } p_{ij} \neq 0, \quad w_{ij} = 1, \text{ for } p_{ij} = 0 \quad ---(1)$$

$$w_{ij} = \frac{w_{ij}}{\sum_{i \neq j} w_{ij}} \quad ---(2)$$

4. 결론

본 논문에서는 ISP와 P2P 어플리케이션의 협력 방안으로 CoolStreaming에 P4P를 적용하는 방안을 간단하게 설명했다. 추후, 순수 CoolStreaming과 P4P를 적용한 CoolStreaming을 구현하여 성능을 평가할 것이다.

참고문헌

[1] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Liu, Avi Silberschatz, "P4P: Provider Portal for Applications" ACM SIGCOMM'08

[2] S. Xie, B. Li, G. Y. Keung, and X. Zhang, "CoolStreaming: Design, Theory, and Practice," IEEE Transactions on Multimedia, Vol.9, No.8, December 2007.