

무선 인터넷 프록시 서버

곽후근^o 정규식

송실대학교 정보통신전자공학부

{gobarian^o, kchung}@q.ssu.ac.kr

A Wireless Internet Proxy Server

Hukeun Kwak^o Kyusik Chung

School of Electronics Engineering, Soongsil University

요약

사회적으로 큰 관심의 대상이 되고 있는 무선 인터넷은 유선 인터넷과 달리 기술 환경과 그 특성상 여러 가지 제약점들을 가지고 있다. 대역폭이 낮고, 접속이 빈번하게 끊기며, 단말기내의 컴퓨팅 파워가 낮고 화면이 작다. 또한 사용자의 이동성 문제와 네트워크 프로토콜, 보안 등에서 아직 기술적으로 부족한 부분을 보이고 있다. 그리고 급속도로 증가하는 수요에 따라 무선 인터넷 서버는 대용량 트래픽을 처리할 수 있는 확장성이 요구되어지고 있다. 이에 본 논문에서는 무선 인터넷 프록시 서버 클러스터를 사용하여 앞에서 언급된 무선 인터넷의 문제와 요구들을 캐싱(Caching), 압축(Distillation) 및 클러스터(Clustering)를 통하여 해결하려고 한다.

TranSend는 클러스터링 기반의 무선 인터넷 프록시 서버로 제안된 것이나 시스템적인(Systematic) 방법으로 확장성을 보장하지 못하고 불필요한 모듈간의 통신구조로 인해 복잡하다는 단점을 가진다. 기존 연구에서 시스템적인 방법으로 확장성을 보장하는 CD-A라는 구조를 제안하였으나 이 역시 모듈간의 불필요한 통신 구조를 가진다는 단점을 있다. 이에 본 논문에서는 시스템적인 확장성과 단순한 구조를 가지는 새로운 클러스터링 기반의 무선 인터넷 프록시 서버를 제안한다. 16대의 컴퓨터를 사용하여 실험을 수행하였고 실험 결과 TranSend 시스템과 CD-A 시스템에 비해 각각 216%, 40%의 성능 향상을 보였다.

1. 서론

현재 무선 인터넷의 사용이 증가하고 있지만 무선 인터넷의 본질적인 문제 역시 무시할 수 없는 요소로 부각되고 있다. 현재까지 나와 있는 무선 인터넷의 근본적인 문제점은 낮은 대역폭, 빈번하게 연결이 끊김, 단말기내의 낮은 컴퓨팅 파워 및 작은 화면, 단말기 사용자의 이동성, 네트워크 프로토콜, 보안 등이 있다. 이러한 나열된 문제들을 캐싱(Caching)과 압축(Distillation)을 통하여 해결하는 방법으로 무선 프록시 서버를 사용하며, 이의 기능으로는 캐싱과 압축을 비롯해 보안(Security), 멀티미디어 전송, FEC(Forward Error Correction), 핸드오프(Handoff), 프로토콜 변환(Protocol Translation), 오류 복구(Failure Recovery) 등이 있다. 그림 1은 무선 인터넷에서 사용되고 있는 무선 프록시 서버를 나타내고 있다.

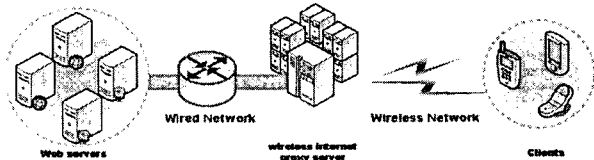


그림 1 무선 인터넷 프록시 서버

본 연구자들은 기존 논문[1]에서 기존 클러스터링의 기반의 무선 인터넷 프록시 서버인 TranSend[2]를 개선한 CD-A라는 구조를 제안하였다. 본 논문에서는 TranSend와 CD-A의 구조상의 문제점을 지적하고 이를 해결하는 새로운 구조를 제안하고 실험을 통해 이들을 비교하는데 초점을 맞춘다.

본 논문의 구조는 다음과 같다. 2장에서는 기존 무선 프록시와 이들이 가지는 문제점을 소개한다. 3장에서는 기존 무선 프록시가 가지는 문제점들을 해결하는 제안된 구조를 설명하고, 4장에서는 실험 및 토론을, 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 연구 배경

2.1 기존 무선 인터넷 프록시 서버

무선 프록시는 캐싱(Caching), 압축(Distillation), 대용량 트래픽에 대한 확장성(Scalability)을 고려하여야 한다. 기존 무선 프록시들 [2-12]은 기본적으로 캐싱(Caching)과 압축(Distillation) 기능을 제공하고 있다. 그러나 확장성의 관점과 구조적인 관점에서는 무선 프록시들 간의 개별적 특성에 따라 차이점을 가지고 있다. 먼저, 확장성 관점에서는 일부 [2, 7, 8]는 고려하고 있는 반면에 대부분은 고려하고 있지 않다. 구조적 관점에서도 캐싱과 압축 기능이 대부분 하나의 호스트에 통합(Untity)되어 있지만, 일부 [2, 3]는 다른 호스트로 분리(Separation)되

어 있다.

기존 무선 프록시로는 TranSend[2], Mowgli(Mobile Office Workstations using GSM Links)[3], WebExpress[4], Class-based Proxy[5], KWU Proxy[6], Distributed Proxy[7, 8], TranSquid[9, 10], Grid Proxy[11], SHAKE[12] 등이 있다.

2.2 TranSend

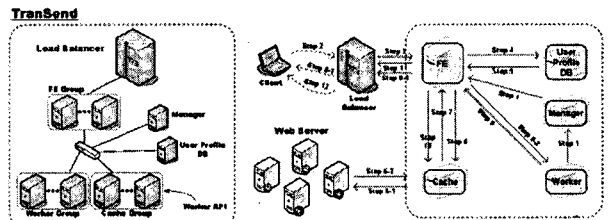


그림 2 TranSend 무선 인터넷 프록시 서버

그림 2는 TranSend 프록시 시스템의 전체적인 구조 및 동작 과정을 나타낸다. TranSend는 각 모듈로써 Front End (FE), User Profile .DB, Cache(\$), Worker, Manager로 구성된다. 모듈 별 구체적인 기능에 대한 자세한 내용은 참고 문헌 [2]을 참조하십시오.

FE는 부하 분산기(Load Balancer)를 통해 사용자 요청을 받고 이를 캐시로 보낸다. 요청 데이터가 캐시에 없다면 캐시는 이 데이터를 웹 서버에 요청한다. 캐시가 데이터를 받고 이는 다시 FE로 보내진다. 압축이 필요하다면, FE는 캐시에서 받은 데이터를 Distiller로 보낸다. FE는 Distiller로부터 받은 압축된 데이터를 캐시에 저장하고 사용자에게 응답한다.

2.3 CD-A (CD & All-in-one)

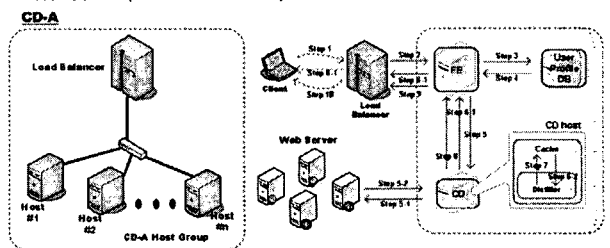


그림 3 CD-A 무선 인터넷 프록시 서버

그림 3은 CD-A 프록시 시스템의 전체적인 구조 및 동작 과정을 나타낸다. CD-A는 TranSend 모듈(FE, Cache, Distiller)에서 Distiller를 없애고 Cache에 압축(Distillation) 기능을 추가한 것이다(이하 CD 모듈: Cache & Distiller). 그리고 이 모듈(FE, CD)을 하나의 호스트에 넣고(이하 CD-A 구조: CD-All-in-one) LVS(Linux Virtual Server)를 사용하여 부하 분산을 하는 것이다. TranSend에서는 각각의 모듈들(FE, Cache, Distiller) 각각이 클러스터링 되어 있는 반면에 CD-A에서는 각 모듈들(FE, CD)을 하나의 호스트들에 포함하고 이러한 호스트를 클러스터링하는 구조로 되어 있다.

Distiller를 없애고 Cache에 압축 기능을 추가한 것은 복잡한 구조를 단순화하고 불필요한 통신을 줄이기 위해서이다. 그리고 각 모듈(FE, CD)을 하나의 호스트에 넣은 이유는 시스템적으로 확장하는 구조를 만들기 위해서이다. 즉, TranSend는 새로운 모듈을 추가 시에 동작과정중의 병목을 찾아 그 모듈을 추가해야 하는(No Systematic) 반면에, CD-A는 병목에 상관없이 새로운 호스트를 추가하면(Systematic) 그 호스트 내에 모듈(FE, CD) 중에 필요한 모듈이 상대적으로 많이 사용된다.

2.4 접근 방식

본 절에서는 TranSend 무선 프록시 서버 및 이의 개선 구조인 CD-A의 문제점을 정리하고, 3장에서는 이를 해결할 새로운 무선 프록시 구조를 제안한다.

(1) TranSend 구조의 문제점

· 확장성(Scalability) : 그림 2 구조에서 보면 FE, Cache, Distiller는 각각 여러 개의 노드(Node)들로 구성가능하다. 프록시 서버를 확장성 있게 만들려면 노드들을 추가해야 하지만 어느 분류(FE 그룹, Cache 그룹, Distiller 그룹)의 노드들을 언제 추가해야 하는지 시스템적인(Systematic) 방법이 없다. 즉, 실험 결과에 의존하여 특정 모듈 그룹이 병목 현상이 발생하면 그룹 모듈을 추가하는 방식으로 하게 된다.

· 복잡성(Complexity) : TranSend는 FE, Cache, Distiller로 구성되어 FE를 중심으로 서로 간에 통신(Communication)을 하도록 구성되어 있다. 이러한 구조는 FE로 모든 통신이 편중되어 있고 Cache와 Distiller가 분리되어 있어 복잡한 통신을 하는 단점을 가진다.

(2) CD-A 구조의 문제점

· 복잡성(Complexity) : CD-A 구조는 TranSend의 복잡한 구조를 단순화하기 위해 Distiller를 Cache 모듈에 포함시켰다. 그러나 이 구조도 TranSend와 마찬가지로 FE로 모든 통신이 편중되어 있다는 단점을 가진다.

(3) 본 연구의 접근 방식

본 논문에서는 TranSend 및 CD-A 구조의 단점인 복잡성(Complexity)을 단순화(Simplification)하고 시스템적인 확장성을 보장하는 새로운 구조를 제안한다.

3. 제안된 무선 인터넷 프록시 서버

제안된 구조

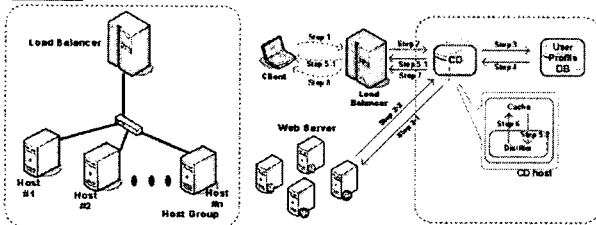


그림 4 제안된 무선 인터넷 프록시 서버

(1) 구조

그림 4는 2.4절에서 분석된 TranSend 및 CD-A 무선 인터넷 프록시 서버의 문제점을 기반으로 이를 해결할 수 있도록 제안된 새로운 구조 및 전체적인 동작 과정을 나타낸다. 제안된 새로운 구조에서는 CD-A에서 사용된 기본 모듈에서 FE를 없애고 FE의 기능을 CD에 추가한 것이다. TranSend에서는 모듈들(FE, Cache, Distiller) 각각이 클러스터링 되어 있고, CD-A에서는 CD-A 호스트가 클러스터링 되어 있는 반면에

제안된 구조에서는 FE와 Distiller를 제외한 Cache가 독립 호스트로서 클러스터링 되어 있다.

제안된 구조에서 FE를 없앤 이유는 다음과 같다. TranSend의 FE의 주요 기능은 여러 대의 캐시 중에 하나를 선택하는 것이었으나 시스템적인 확장성을 보장하는 CD-A 구조에서는 불필요한 모듈이다. 왜냐하면 CD-A 구조에서는 TranSend를 구성하는 모듈들을 하나의 호스트에 집어넣고 이를 클러스터링한 구조이고, FE가 선택할 수 있는 캐시는 자신의 호스트에 있는 것뿐이기 때문이다. 즉, CD-A 구조에서 FE의 기능은 데이터가 지나가는 단순한 통로 역할에 지나지 않는다.

(2) 기존 구조와의 비교 (TranSend, CD-A vs. 제안된 구조)

표 1은 TranSend와 CD-A를 제안된 시스템과 구조적으로 비교한 표이다.

표 1 기존 구조 vs. 제안된 구조(LOC)

	TranSend	CD-A	제안된 구조
확장성(Scalability)	No Systematic	Systematic	Systematic
구조(Complexity)	LVS-FE-Cache-Distiller	LVS-FE-Cache	LVS-Cache

4. 실험 및 토론

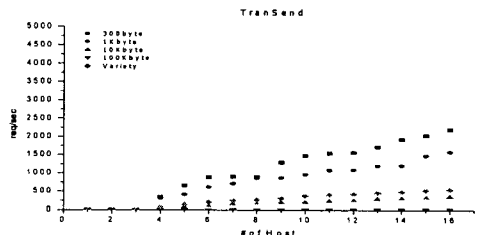
4.1 실험 환경

프록시 서버는 PC 16대로 구성되어 있고 TranSend 및 제안된 시스템에서 FE의 부하 분산과 All-in-one 시스템의 부하 분산을 위하여 LVS라는 Load Balancer를 사용하였다. Apache Bench라는 프로그램을 Client에서 수행하여 프록시 서버에 영상(이미지)을 요청하는 방식으로 실험하였다. 표에서 Client와 LVS가 Host보다 하드웨어 성능이 좋은 이유는 확장성 실험을 할 때 Client와 LVS에서는 병목이 발생하지 않는 상황에서 프록시 서버내 호스트들 사이의 확장성을 확인하고자 했기 때문이다.

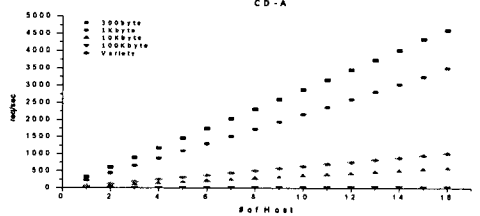
4.2 실험 결과

(1) TranSend & CD-A

그림 5(a)는 TranSend 구조에서 이미지 크기를 다르게 요청했을 경우 호스트 개수에 따른 초당 요청수를 나타내고 그림 5(b)는 CD-A 구조에서 이미지 크기를 다르게 요청 했을 경우 호스트 개수에 따른 초당 요청수를 나타낸다.



(a) TranSend



(b) CD-A

그림 6 호스트 개수에 따른 초당 요청수 (TranSend, CD-A)

(2) 제안된 구조

표 2와 그림 7은 이미지 크기를 다르게 요청했을 경우 호스트 개수

에 따른 초당 요청수를 나타낸다.

표 2 호스트 개수에 따른 초당 요청수 (CD)

# of Hosts	1	5	10	15
300 bytes	620	3035	5973	6178
1 Kbytes	363	1805	3605	5401
10 Kbytes	39	198	396	595
100 Kbytes	2.4	12.02	24.14	36.35

그림 6을 보면 300 bytes에서 호스트가 10에서 15대로 갈 때 초당 요청수가 선형적으로(Linear) 증가하지 않는 것을 볼 수 있다. 이는 사용된 부하 분산기(LVS)가 많은 수의 요청을 처리하지 못해 발생한 병목이다. 즉, LVS가 사용한 구조는 NAT(Network Address Translation)으로 모든 패킷이 부하 분산기를 통과하는 구조로서 많은 수의 요청을 처리하지 못하는 단점을 가진다. 이러한 병목을 해결하기 위해서는 다른 구조(Direct Routing or IP Tunneling)로의 변환이 필요하다.

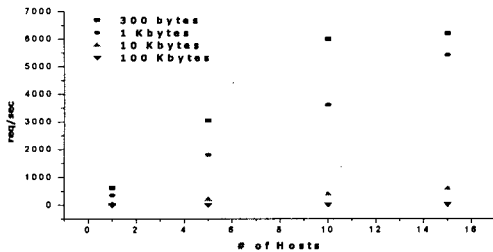


그림 6 호스트 개수에 따른 초당 요청수 (제안된 구조)

(3) TranSend, CD-A vs. 제안된 구조

표 3은 TranSend와 CD-A에 대한 제안된 구조의 평균 성능 향상률을 나타낸 것이다. 제안된 시스템은 TranSend와 CD-A에 비해 각각 평균 216%와 40%의 성능 향상을 가진다. 작은 크기의 이미지는 큰 크기의 이미지보다 상대적으로 모듈간의 통신에서 많은 시간이 소요되므로(FE를 통과하는 회수가 많음으로) 제안된 시스템에서 높은 성능 향상률을 가짐을 알 수 있다.

표 3 평균 성능 향상률 (제안된 구조)

%	300 bytes	1 Kbytes	10 Kbytes	100 Kbytes	Average
TranSend vs. 제안된 구조	292.14	293.74	190.32	88.23	216.11
CD-A vs. 제안된 구조	86.73	63.01	8.39	0.13	39.57

4.4 토론

제안된 시스템이 기존 시스템(TranSend, CD-A)에 비해 성능이 좋아진 이유는 구조적 관점에서 복잡한 구조를 단순한 구조로 바꾼 결과로 설명할 수 있다. 제안된 단순화된 구조는 모듈(FE, Cache, Distiller)간의 통신이 상대적으로 빈번한 작은 크기의 이미지 요청 실험에서 더 좋은 성능 향상률을 보임을 알 수 있다.

5. 결론

본 논문에서는 무선 인터넷의 근본적인 문제점 중 일부를 해결할 수 있도록 제안된 TranSend 및 이의 개선 구조인 CD-A 무선 인터넷 프록시 서버의 문제점을 지적하고, 구조 및 성능 개선을 제안하였다. TranSend 및 CD-A 무선 인터넷 프록시 서버의 문제점을 확장성, 복잡성 관점에서 분석하였고, 이를 해결하기 위해 새로운 구조를 제안하였다. 실험을 통해 제안된 구조가 성능 향상에 기여했음을 확인하였다.

향후 연구 방향을 요약하면 다음과 같다.

LVS의 NAT(Network Address Translation) 방식을 DR(Direct Routing) 방식으로 변환 : NAT 방식은 외부와 내부 패킷이 모두 LVS를 통과함으로 호스트를 증가하는데 한계를 가진다.

참고문헌

[1] 박후근, 정규식, "무선 인터넷 프록시 서버 클러스터 성능 개선", 한국정보과학회논문지 : 정보통신, 게재 예정, 2005. 6.
 [2] A. Fox, "A Framework For Separating Server Scalability and Availability From Internet Application Functionality", Ph. D. dissertation, U. C. Berkeley, 1998.
 [3] M. Liljeberg, H. Helin, M. Kojo and K. Raatikainen, "Enhanced Services for World Wide Web in Mobile WAN Environment", Department of Computer Science, University of Helsinki, Report C-1996-28, 1996.
 [4] B. Housel, G. Samaras and D. Lindquist, "WebExpress: A client/intercept based system for optimizing web browsing in a wireless environment", Mobile Networks and Applications, ACM, pp. 419-431, 1998.
 [5] J. Lee, M. Kim, H. Youn, Y. Hahm and D. Lee, "Class-based proxy server for mobile computers", Proceedings of International Workshops on Parallel Processing, IEEE, pp. 559-566, 2000
 [6] K. Ham, S. Jung, S. Yang, H. Lee and K. Chung, "An Enhanced Proxy Architecture for Efficient Web Browsing over Cellular Networks", Proceedings of the 14th International Conference on Information Networking, pp. 5A 4.1-4.5, 2000
 [7] K. Kim, H. Lee and K. Chung, "A Distributed Proxy Server System for Mobile Web Service", Proceedings of the 15th International Conference on Information Networking, IEEE, pp. 8A 749-754, 2001
 [8] Anindya Datta and et al., "Proxy-based acceleration of dynamically generated content on the World Wide Web: an approach and implementation", Proceedings of the 2002 ACM SIGMOD international conference on management data, 2002.
 [9] A. Maheshwari, A. Sharma, K. Ramamritham and P. Shenoy, "TranSquid: transcoding and caching proxy for heterogeneous e-commerce environments", Proceedings of 12th International Workshop on RIDE-2EC, IEEE, pp. 50-59, 2002.
 [10] B. Knutsson, H. Lu, J. Mogul, "Architecture and pragmatics of server-directed transcoding", Proceedings of 7th International Workshop on Web Content Caching and Distribution, 2002.
 [11] T. Phan, L. Huang, and C. Dulac, "Integrating mobile wireless devices into the computation grid", Proceedings of the 8th annual international conference on mobile computing and networking", 2002.
 [12] Y. Konishi and et al., "Web SHAKE: A fast WWW access method for mobile terminals on temporary cluster networks", IEEE International Conference on Communications, 2002.