

위성통신을 위한 웹 프록시 개발

전성윤[†], 김근형^{**}

요 약

육지에서 멀리 떨어진 해상의 선박 또는 항공기는 위성통신을 이용하여 웹 서비스를 이용해야 한다. 그러나 위성통신은 많은 비용이 들며 사용자가 만족할 만한 수준의 응답 속도를 제공하지 못하고 있다. 선박의 웹 서비스 사용자가 인터넷 뉴스의 기사 한편을 볼 때 불필요한 이미지와 광고들로 인하여 많은 양의 데이터를 전송 받게 된다. 따라서 사용자가 불필요한 데이터 사용 요금을 많이 지불하게 되는 것이다. 본 논문에서는 이러한 위성통신을 통한 웹 서비스 사용의 문제점인 비용, 속도의 문제를 해결하기 위해 설계된 프록시 모델을 제시한다. 본 논문에서 제안한 프록시는 이미지 차단, 광고 차단, 캐싱, 이미지 재요청 기능으로 위성망을 통한 데이터 전송량을 최대한 줄인다. 개발한 프록시의 성능을 위성통신을 이용한 데이터량 측정을 통해 증명하였다.

A Development of Web Proxy for the Satellite Communication

Sung-Yoon Jeon[†], Geun-Hyung Kim^{**}

ABSTRACT

In the maritime ships or airplanes, users should utilize the satellite channel in order to use web service. However, the satellite channel is costly and does not give users satisfied response time. In the ship, the users may receive plenty of extra data when they obtain internet news. The extra data may be unnecessary image and advertise. Therefore, they should pay unnecessary data usage charges as well. In this paper, we suggest a proxy model that solves the problem of cost and speed. The proposed proxy reduces the of data through the satellite link by image and advertising blocking, caching, image re-requesting functions. It's performance was tested by a real satellite communication.

Key words: Satellite Network(위성통신망), Web Browsing System(웹 브라우징 서비스), contents blocking(콘텐츠블록킹), contents re-transmission(콘텐츠 재전송), contents caching(콘텐츠 캐싱)

1. 서 론

최근 우리는 집이나 거리에서 초고속 인터넷 서비스를 아무 불편 없이 즐길 수 있으며 대부분의 정보를 인터넷을 통하여 접하고 있다. 이처럼 인터넷은 사람들에게 없어서는 안 될 존재가 되었다. 하지만 아직 해상의 선박에서는 인터넷 사용에 어려움이 존

재한다. 해상 선박에서는 인말새트를 이용한 웹 서비스 이용이 가능하다. 인말새트란 국제해사위성기구(INMARSAT)에서 지구 정지 궤도상의 4개의 통신 위성을 이용하여 이동체(선박, 차량, 항공기, 휴대용 단말기 등)와 전 세계 지상 간 전화, 팩스, 데이터 및 텔렉스 등의 통신을 제공하는 국제 위성 이동통신 서비스이다[1]. 그러나 인말새트는 데이터 사용량에

※ 교신저자(Corresponding Author): 김근형, 주소:부산광역시 부산진구 엄광로 995, 전화: 051) 890-2271, FAX: 051) 890-2265, E-mail: geunkim@deu.ac.kr
접수일: 2013년 7월 26일, 수정일: 2013년 10월 18일
완료일: 2013년 11월 7일

[†] 동의대학교 영상정보공학과
(E-mail: smy1598@naver.com)

^{**} 동의대학교 영상정보공학과

※ 본 연구는 중소기업청 서비스연구개발사업(S2061459)의 지원으로 수행되었음.

다른 비용을 지불해야 한다. 최근 무제한 요금제가 나왔지만 사용자에게는 여전히 부담스러운 비용이다.

전송 데이터 사용량에서 많은 양을 차지하는 콘텐츠는 이미지(image) 형태이다. 웹 서비스를 이용할 때, 대부분의 포털 사이트를 들어가 보면 불필요한 광고들과 이미지들이 보이도록 되어있다. 그러면 선박의 인터넷 사용자들을 원치 않은 정보를 제공받고 데이터 사용 요금을 지불하게 된다. 본 논문에서 제시하는 프록시[2] 모델에서는 이러한 불필요한 콘텐츠들이 요청되지 않도록 막아주는 기능이 추가되어 있다. 프록시를 선박 내에 두어 콘텐츠들의 요청이 서버로 가지 않게 한다. 결과적으로 차단된 요청에 대해서는 위성 링크를 사용하지 않게 된다. 제안하는 방법을 사용하여 광고, 이미지가 없고 텍스트와 프록시가 전송하여주는 대체 이미지만 보이는 웹 페이지가 제공된다. 이렇게 광고와 이미지를 차단해 주는 기술들이 이미 존재하나, 본 논문에서는 제시한 기술에는 차단된 이미지를 중에서도 사용자가 원하는 이미지에 대한 재요청을 가능하게 하여 이미지를 선택적으로 제공한다.

추가적으로 프록시에 캐시 기능을 추가하여, 사용자가 이전에 응답받았던 콘텐츠 이거나 선박 내의 다른 사용자가 응답을 받았던 콘텐츠에 대한 요청이 들어오면 캐시에 저장되어 있는 콘텐츠로 응답을 함으로써 위성 링크의 사용을 줄였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 기술인 Privoxy[3], 웹 캐시[4]에 대하여 설명한다. 3장에서는 전송 데이터 량 감소를 위해 구현된 프록시 모델의 이미지 차단, 광고 차단, 캐시 기능들에 대하여 설명한다. 4장에서는 실제 위성망을 통한 성능 평가 결과를 제시하고 5장에서 결론을 맺는다.

2. 관련 기술

2.1 Privoxy

Privoxy는 웹 프록시 프로그램이다. 웹 프록시는 웹 브라우저와 웹 서버의 사이에서 웹 브라우저가 웹서버에 어떠한 문서를 요구했을 경우 프록시가 그것을 가로채어 해당 역할을 수행한 뒤, 다시 웹서버에 요청을 하고, 그 요청된 결과를 웹브라우저에 다시 보내주는 역할을 한다.

보통 웹 프록시는 웹 브라우저의 성능 향상을 위

해 사용된다. 예를 들어, 한 사무실 내에서 수백 대의 컴퓨터가 있고, 이 컴퓨터를 통해 매일 아침 신문 사이트에서 신문을 본다고 하면, 각 컴퓨터가 매번 신문 사이트에 연결되어 웹페이지를 가져오는 것보다, 회사 내 프록시에서 신문사의 웹 페이지를 저장하고 있고, 회사 내의 모든 브라우저가 그 프록시에서 내용을 가져오면, 훨씬 빠른 웹브라우저가 가능할 것이다. 마치 브라우저 내의 캐시 역할을 네트워크 내의 프록시 서버가 담당하는 것이다.

이렇게 캐시 역할을 통한 속도 향상 외에 프록시 서버를 통한 내용 변경이나 사이트 검열도 할 수 있다. 프록시 서버를 통해 특정 사이트의 내용을 볼 수 없게 하거나, 웹 페이지 내의 특정 내용을 변경해서 웹 브라우저로 보내는 것도 가능하다.

Privoxy의 주요 기능은 광고 이미지 제거와 광고 배너 차단, 웹 페이지 내 원치 않는 내용을 차단하는 것이다. 웹 브라우저는 Privoxy에게 웹서버를 통해 해당 웹페이지를 가져오도록 요청하고, Privoxy는 해당 웹서버에 접속하여 해당 웹페이지의 콘텐츠를 가져와 적절한 규칙을 통해 광고라고 생각되는 이미지를 삭제해서 웹브라우저로 넘겨준다. 그러면, 웹브라우저에서는 광고 이미지가 삭제되어 보이지 않는다.

아래 그림 1은 프록시를 사용하지 않을 때의 일본 기상청 사이트의 기상도이다. 그리고 그림 2는 Privoxy를 사용하여서 같은 웹 페이지의 이미지가 차단되고 특정 패턴의 이미지로 대체된 모습을 보여준다. 이 경우 필요한 기상도임에도 불구하고 웹 브라우저에 전달되지 않는 문제점을 가지고 있다.

하지만, 이러한 광고 차단이나 내용 변경의 주된 내용 외에도, Privoxy가 필요한 다른 기능이 있다.

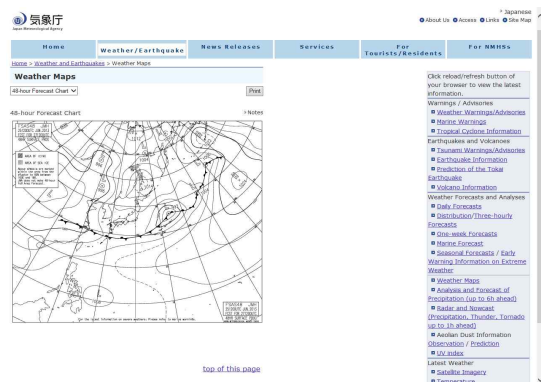


그림 1. 일본 기상청 사이트

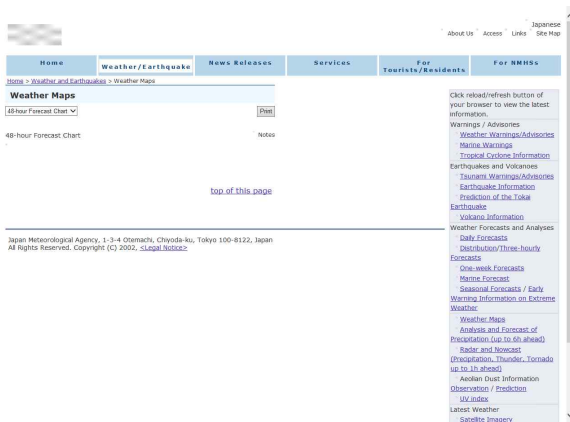


그림 2. Privoxy를 이용한 일본 기상청

바로, 웹 내용 변경이다. Filter 기능을 통해서 웹 페이지 내의 특정한 내용 혹은 태그를 변경할 수 있다. 국내 웹사이트의 경우 윈도우즈 환경의 Internet Explorer에 주로 맞추어져 있기 때문에, 매킨토시 상의 Mozilla, Chimera, Internet Explorer와 같은 브라우저로 보다 보면, 제대로 화면에 표시되지 않는 경우가 많다. 이 경우 Privoxy의 Filter 기능을 이용하여 보정하여 볼 수 있다.

2.2 웹 캐시

웹 캐시는 동일한 서버에 다시 접근할 때에는 근처에 있는 프록시 서버의 웹 캐시에 저장된 정보를 불러오므로 더 빠른 열람이 가능하다. 웹 캐시의 기본 개념은 다음 그림 3과 같다. 웹 캐싱은 주로 요청에 대한 지연을 최소화하고 네트워크 대역폭을 줄이기 위해 주로 사용한다. 웹 캐시 서버의 성능이 뛰어나수록 사용자들에게 보다 빠른 서비스를 제공할 수 있고, 데이터 전송을 최소화할 수 있다.

동작 원리는 다음과 같다. 먼저 해당 객체의

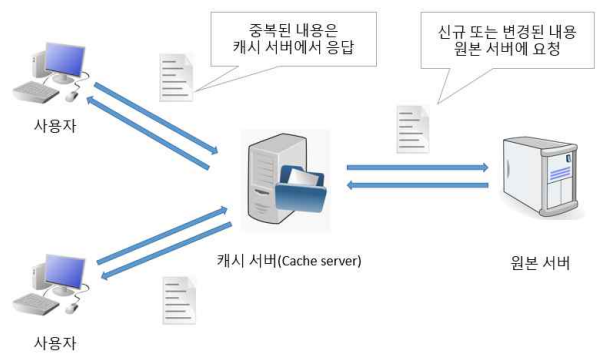


그림 3. 기본적인 캐시 서버 동작 원리

HTTP[5] 헤더 정보를 참조하여 캐싱 유무를 판단한다. 캐싱 된 객체는 재요청 시 웹 서버로 요청하지 않고 바로 웹 캐시 서버를 이용하여 사용자에게 전달하게 된다. 이때 아래 표 1에 나타난 HTTP 캐시 헤더 항목들을 이용하여 'fresh/stale'에 대한 검증 후 원본 서버 참조 여부를 결정한다. 해당 객체의 유효기간이 만료된 경우에는 다시 원본 웹 서버에 요청하여 해당 객체를 갱신한다.

웹 캐싱 기능은 서버뿐만 아니라 각 사용자들이 사용하는 웹 브라우저에도 기본적으로 탑재되어 사용된다. 웹 브라우저의 캐싱은 사용자가 접속한 사이트에 대한 콘텐츠를 저장하여 재방문 시 효과를 거두고, 옵션에서 캐싱 방법에 대한 설정을 할 수 있지만, 한 사용자에 대하여 캐싱 효과를 볼 수가 있다. 그리고 복잡하고 다양한 인터넷 사이트가 존재하므로 성능 면에서 한계가 있다. 일반적으로 사용하는 웹 캐싱은 이보다 더 확장된 형태의 고성능 캐시 서버이다. 대표적인 웹 캐시 서버의 종류로는 Nginx[6], Squid[7], Apache Traffic Server[8], Varnish[9] 등이 있다.

본 논문에서 제시하는 프록시의 캐싱 기능은

표 1. 캐시 유효성 관련 HTTP 헤더

| 항 목 | 내 용 | 비 고 |
|---------------|---|----------|
| Last-Modified | 해당 콘텐츠의 마지막 변경 시간을 기록 | GMT 기준 |
| E-Tag | 서버에서 생성되는 유일한 인자로 오브젝트의 중복성과 캐싱 유효성 여부를 확인할 수 있는 항목 | HTTP 1.1 |
| Cache-Control | 해당 오브젝트에 대한 캐싱 관련 정책을 제어하는 항목 | HTTP 1.1 |
| Age | 해당 오브젝트가 갱신된 시점을 기준으로 초기화되며 새롭게 갱신되는 시점까지 1초 단위로 증가함 | |
| Expires | 해당 오브젝트가 얼마 동안 유효한지를 표시 해주는 항목으로 명시된 시간 이후 캐싱 유효성 재확인 | GMT 기준 |

Squid의 소스를 참조하여 개발하였다. Squid는 공개 웹 프록시 캐시 서버이며, 웹 프록시 캐싱 서비스를 제공하고 접근 제어 규칙을 만들 수 있다. 그리고 캐시 DNS(Domain Name System) 참조를 설정할 수 있는 기능이 내장되어 있다. 현재 대역폭의 절감과 보안 강화 그리고 웹서핑 속도를 높이기 위한 노력이 끊임없이 이루어지고 있으며, 이런 기능이 구현된 프록시 서버 프로그램들이 개발되었는데, 이 프로그램들은 몇 가지 문제점을 가지고 있었다. 첫 번째는 오픈 소스가 아니라는 점이고 두 번째는 ICP(Internet Cache Protocol)[10]를 지원하지 않는다는 것이다. ICP는 이웃하는 캐시에 특정 요청 정보가 존재하는지 정보를 교환하는데 사용되며 인접해 있는 다른 캐시 서버에 먼저 저장돼 있는 캐시 정보를 전송받아 사용할 수 있다.

그러나 Squid는 이 두 가지 모두를 지원한다. 또한 프록시의 부가적인 것으로 대역폭을 경제적으로 사용할 수 있게 해 준다는 것이다. 프록시는 최근 사람들이 공통적으로 방문한 사이트를 디렉터리에 저장해 두었다가 같은 URL에 대한 요청이 들어오면 서버에 연결하지 않고도 디렉터리에서 읽어 즉시 브라우저에게 보내준다. 웹서버 성능의 최적화를 위해 두 가지 다른 방법으로 Squid 서버를 설치하고 설정하는 방법이 있다.

첫 번째는 웹서버 성능을 극대화하고 httpd 가속 서버로 구성한다. 두 번째는 일반적인 프록시 캐싱 서버로 구성하는 것이다. 이렇게 구성함으로써 회사 또는 조직 내의 모든 사용자들로 하여금 Squid를 통해서만 인터넷에 접근하도록 할 수 있다. 이것은 상당한 보안 효과와 시스템 자체의 속도를 향상시킬 수 있다. 또한 Squid 자체 설정을 통해 인터넷의 접속 제한 여부를 설정할 수도 있다. 이렇게 구성함으로써 유해 사이트도 간단히 접근을 제어할 수 있다.

3. 프록시 구현

3.1 프록시 전체 동작 흐름

위성 링크 사용량을 줄이기 위해 개발된 프록시는 C언어와 윈도우즈 소켓 프로그래밍을 이용하여 HTTP 요청 메시지를 송/수신할 수 있도록 하였다. 프록시는 클라이언트의 요청마다 하나의 스레드를 생성하여 처리한다. 스레드가 생성되면 소켓에 있는

요청 메시지를 읽어 들인다. 그 후 캐시를 탐색하여 객체가 캐시에 존재하면 캐시 내용에 대한 유효성 검사를 한다. 캐시 내용이 유효하다면 클라이언트로 전송하고 그렇지 않으면 요청 차단 기능으로 넘어간다. 이미지나 광고에 대한 요청으로 판단되면 서버로 요청이 차단되고 대체 이미지 또는 차단 알림 HTML 문서로 클라이언트로 응답한다. 요청이 차단되지 않았다면, 서버로 요청을 전송하고 응답 메시지를 받아 캐싱한 후에 클라이언트로 전달한다. 이와 같은 요청 및 응답 동작이 모두 완료되면 클라이언트와 연결이 종료되고 대기 상태가 된다.

3.2 이미지 차단

사용자가 웹 서비스 이용할 시, 데이터 전송량의 대부분을 차지하는 객체는 이미지 형태이다. 그러므로 이미지를 차단할 경우 많은 양의 데이터 사용을 줄일 수 있다. 이미지 차단 기능은 클라이언트의 요청 메시지의 요청 URL과 설정 파일 image_blocking_list에서 읽어 들인 이미지 관련 차단 키워드를 비교하여 요청 URL에 차단 키워드가 존재하면 서버로 요청을 보내지 않고 차단할 수 있다. 이미지와 관련된 URL에는 이미지 확장자나 이미지와 관련된 단어인 image, img 등이 대부분 들어있다. 그러므로 설정 파일에 내용을 .jpg, .gif, img 등으로 설정하면 관련 이미지가 차단된다. 설정 파일의 키워드가 들어 있지 않은 URL을 가진 이미지가 존재하는 경우에는 이미지에 대한 URL을 확인하여 적절한 키워드를 설정 파일에 추가하면 이미지 차단이 가능하다. 아래의 그림 4는 그림 1과 같은 사이트를 본 논문의 구현한 프록시로 이미지를 차단한 모습이다. 차단된 이미지에 대하여 'S'라는 대체 이미지를 제공하였다.

그림 4의 모습을 보면 그림 2에서 보았던 Privoxy의 이미지 차단 모습과 유사하다. 그러나 기상도가 있던 곳에서 파란색의 대체 이미지를 볼 수 있다. 파란색의 대체 이미지는 본 논문에서 설명하고 있는 프록시와 함께 개발된 육상 프록시와 함께 연동하여 제공된다. 서버 응답의 이미지에 링크가 걸려 있지 않으면 육상 프록시에서 HTML 문서의 이미지 URL을 수정하여 클라이언트에 전달한다. 본 논문에서 개발한 프록시 서버는 수정 URL에 대한 요청을 수신하면 재요청 이미지라고 판단하여 파란 이미지를 제공한다. 만약 사용자가 재요청을 하여 이미지 하나만



그림 4. 개발 프록시로 일본 기상청 이미지 차단 모습

을 전송 받는다고 하여도 이미지 하나에 대한 데이터량을 무시할 수가 없다. 그래서 몇 가지 옵션을 두었다. 아래의 그림 5는 재요청이 가능한 이미지, 즉 파란색 'S' 이미지를 선택하였을 때 나타나는 페이지이다.

그림 5에서 나타난 옵션 'original', 'high', 'low', 'gray'들은 각각 원본 영상, 압축 레벨이 낮은 이미지, 압축 레벨이 높은 이미지, 흑백 이미지를 제공한다. 'original'에서 'gray'로 갈수록 압축률이 높아지고 위성을 통해 전달되는 데이터량은 줄어든다. 압축에 대한 처리는 따로 분리되어 개발된 선박 외부 프록시에서 제공하고 있으므로, 본 논문에서는 자세하게 다루지 않는다.

이미지 차단 기능에는 화이트 리스트 기능을 갖는다. 지도나 메뉴 등과 같이 사용자가 꼭 봐야 할 이미지인 경우 차단하지 않기 위해 image_blocking_whitelist라는 설정 파일에 통과시킬 URL 키워드를 설정하면, 차단 키워드가 URL에 있다고 하여도 차단



그림 5. 재요청 옵션 선택 페이지

하지 않는다. 그림 6은 이미지가 아닌 요청 메시지가 왔을 때 위성망을 사용하는 경우의 흐름을 나타내고 그림 7은 이미지에 대한 요청 메시지를 수신하였을 때 위성망을 사용하지 않고 중간에서 차단되는 흐름을 나타낸다. 개발 프록시의 이름을 선박(Ship)의 첫 글자를 이용해 Proxy_S로 표기하였다.

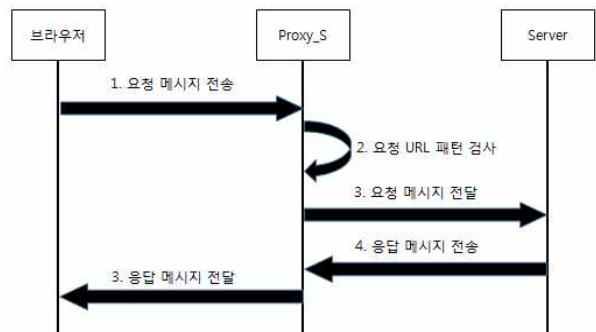


그림 6. 이미지가 아닌 HTTP 메시지 요청 응답 시 흐름

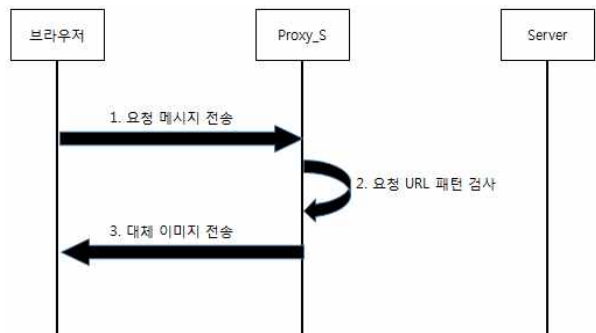


그림 7. 이미지 요청 차단 흐름

3.2 광고 차단

최근 인터넷 사용량이 급속히 증가하면서 인터넷을 이용한 광고의 효과가 높아 웹 사이트에서 많은 인터넷 광고를 볼 수 있다. 하지만 웹 사이트를 이용하는 사용자 입장에서는 인터넷을 이용하는 목적과 맞지 않는 광고를 접하게 되면서, 산만함과 불편함을 느낀다. 또한 전체 웹 화면이 나타나는 시간이 늘어나고, 비싼 요금을 내면서 위성을 통하여 웹 서비스를 이용하는 선박들의 경우에는 광고를 보게 되어 원치 않는 데이터 요금을 지불하게 된다. 이런 불편함을 해결하기 위해 광고를 차단하는 프로그램이 필요하게 되었다. 그중 앞서 말했던 Privoxy가 있는데, Privoxy는 사용자가 객체를 요청하는 URL에 광고에 관한 특정한 문자열 패턴을 적용하여 광고를 차단한다. 본 논문에서는 사용자의 요청 URL이 프록시 서버에 설정되어 있는 광고 사이트이거나 관련된 특정 키워드를 포함하고 있으면 차단을 시키는 기능을 개발 프록시에 추가하였다.

프록시에 추가된 광고 차단 기능의 세부 구현 내용은 다음과 같다. 우선 ad_blocking 설정 파일에 차단할 광고와 관련된 키워드를 입력한다. 그 후에 클라이언트에서 요청 메시지가 오면 메시지의 요청 URL과 설정 파일의 키워드와 비교한다. 키워드가 포함된 URL이면 서버로 요청을 보내지 않고 프록시에서 광고 차단 HTML 문서로 클라이언트에게 응답한다. 아래 표 2는 몇몇 유명 사이트들의 광고 URL을 조사한 표이다.

조사한 URL의 공통적인 키워드를 찾아보면 광고를 뜻하는 'advertising'의 'ad'가 대부분 들어가 있는

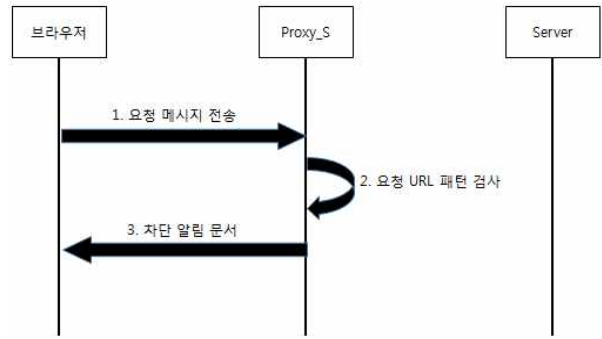


그림 8. 광고 차단 시퀀스

것을 발견할 수 있었다. 표 2에 있는 사이트들 외에 50여 개 이상의 사이트들을 조사해 보았다. 마찬가지로 광고 창의 주소에 모두 'ad'가 포함되어 있다. 이러한 조사를 바탕으로 몇 가지의 키워드를 찾아낼 수 있었다. 'ad.', '/ad', '/AD/', '/banner' 이 네 가지의 키워드를 개발 프록시에 적용시키고 웹 서비스를 이용하였을 때, 거의 모든 광고들을 차단할 수 있었다. 조사를 통해 찾아내지 못한 광고들을 감안해도 90% 이상의 광고를 차단한다고 볼 수 있다. 만약 앞서 말한 네 가지의 키워드들로 광고가 차단되지 않았을 경우 차단되지 않은 URL을 참조하여 적절한 키워드를 설정하면 그 광고 역시 차단이 가능하다. 그림 8의 광고 차단 시퀀스를 통해 선박 외부로의 네트워크 트래픽이 발생하지 않음을 보여준다.

그림 9는 이미지 차단과 광고 차단 기능이 함께 적용된 네이버 홈페이지이다. S라고 된 이미지는 기존 요청이 차단되어 프록시 내에서 브라우저로 제공한 이미지이고, blocked advertising 문구가 있는 부분은 광고가 차단되고 프록시가 가지고 있던 HTML 문서가 제공됨을 나타낸다.

표 2. 유명 사이트들의 광고 URL

| 사 이 트 | URL |
|---------------------|---|
| SINA.com(중국 포털 사이트) | http://www.skoovyads.com/english/2011/ad49.html |
| 163.com(중국 포털 사이트) | http://www.gongheshengshi.com/adiframe/163/... |
| 네이버 | http://nv1.ad.naver.com/adshow?unit=002AG&nr... |
| 다음 | http://amsv2.daum.net/ad/adview?secid=0K904... |
| 조선일보 | http://cad.chosun.com/RealMedia/ads/adstream... |
| G마켓 | http://adscriptgmarket.co.kr/js/adscript/ad_center... |
| 잡코리아 | http://ads.jobkorea.co.kr/RealMedia/ads/adstream |
| 한게임 | http://ad.hangame.com/adshow?unit=335C... |

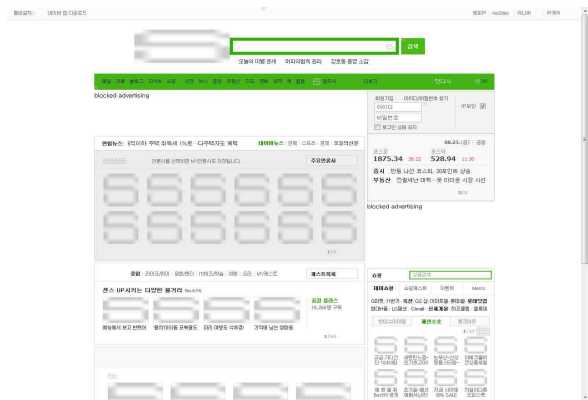


그림 9. 네이버 홈페이지 이미지 및 광고를 차단하지 않은 모습(좌)과 차단한 모습(우)

3.3 캐싱

개발한 프록시에 캐싱 기능을 추가함으로써 해상 선박 내의 한 사용자가 방문하였던 사이트들의 객체들에 대하여 위성통신 링크를 사용하지 않고 제공받을 수 있다. 최초에 서버로의 요청에 대한 응답이 오면 프록시의 캐시에 저장된다. 이때 저장하는 방식은 해싱을 사용하였다. 해시 테이블에 키와 함께 저장되는 캐시 파일의 경로가 들어가고, 캐시 파일에는 응답 메시지가 저장된다. 그런데 다른 객체들 사이에서 같은 키가 생성될 경우가 있다. 그러면 사용자의 요청에 잘못된 응답을 할 수가 있다. 이 경우를 대비하여 캐시 파일에 URL을 함께 저장하여 키를 확인한 다음 URL도 확인을 한다.

사용자의 요청에 대한 동작 원리는 다음과 같다. 처음 사용자가 서버로 객체를 요청할 때 HTTP 메시지를 통해 URL과 Method를 참조하여 해시 키를 생성한다. 그에 대한 응답이 왔을 때 객체를 저장한 후 생성하였던 키를 해시 테이블에 저장한다. 이때 파일의 위치도 함께 테이블에 저장한다. 이후에 다른 사용자로부터 같은 객체에 대한 요청이 오면 URL과

Method를 이용하여 키를 생성한 후에 키를 이용하여 해시 테이블에서 해당 객체를 찾게 된다. 캐시 되어 있음을 확인하면 저장된 데이터가 최신의 것인지 아닌지를 확인한다. 이를 확인하는 방법은 크게 'Expiration' 방식과 'Validation' 방식 두 가지로 나뉜다.

Expiration 방식은 roundtrip 과정을 최소화하기 위해 User-Agent가 웹 문서를 요구할 때 캐시에 저장되어있는 만기일을 프록시에서 확인하여 만기일이 지나지 않았을 경우에 캐시 된 문서를 브라우저로 응답하는 매커니즘을 사용한다. 이 과정에서는 HTTP 헤더의 Expires와 Cache-Control이 사용된다. 그림 10에서 Expiration 방식의 처리 과정을 나타냈다.

다음 Validation 방식은 full response의 발송을 줄여 네트워크 대역폭을 절약하는 방식이다. 사용되는 HTTP 헤더로는 E-tag와 Last-modified 가 사용된다. Expiration 방식과 다르게 서버로 검증 요청을 보내게 된다. 서버에서 검증을 마치고 변화가 없는 경우는 304(Not Modified)를 응답하여 주고, 변화가

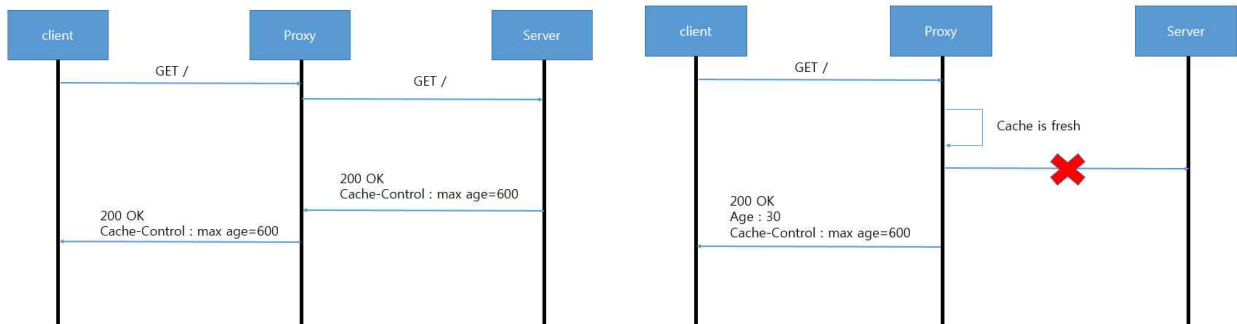


그림 10. Expiration 방식의 처리 과정. 최초 요청 시(왼쪽), 두 번째 요청 시(오른쪽).

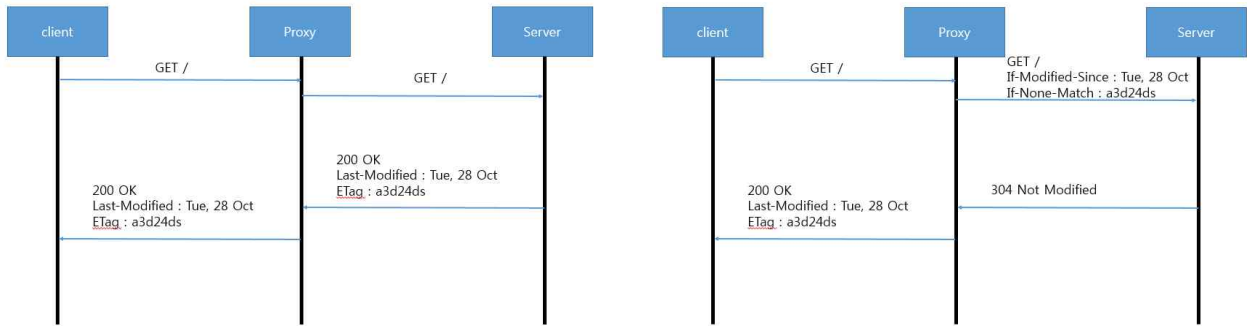


그림 11. Validation 방식의 처리 과정. 최초 요청 시(왼쪽), 두 번째 요청 시(오른쪽).

있을 경우 200(OK)를 리턴하고 변경된 객체를 다시 전송한다. 그림 11에서 이 과정의 예를 보여준다.

4. 실험

구현된 프록시를 사용하여 데이터 사용량을 얼마나 줄일 수 있는지 실험을 하였다. 프록시가 선박 내에 존재하여 사용자들이 브라우저를 통해 객체를 요청할 경우 프록시를 거쳐서 위성망으로 요청 메시지가 나가게 된다. 이때, 이미지와 광고가 차단되면

서 요청하는 데이터의 양이 감소하게 된다. 예로 그림 12는 프록시 서버가 선박 내에 구축되어 동작하는 모습이다.

구현된 프록시의 성능을 측정하기 위하여 프록시를 사용한 경우와 그렇지 않은 경우의 전송 데이터량을 비교하였다. 실제 해상에서 주로 접속하는 사이트를 대상으로 측정하였다. 아래의 표 3은 실험의 결과로, 이미지 차단과 광고 차단 기능을 사용하여 접속한 대상 사이트별 데이터 감소량을 나타낸다.

표 3에서 나온 결과를 보면 데이터 사용량 감소

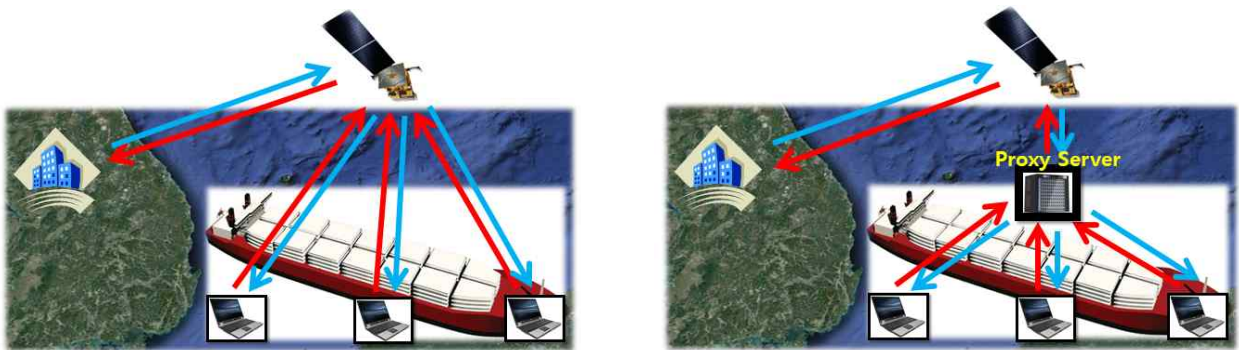


그림 12. 선박에서의 일반적 웹 서비스(왼쪽)와 프록시 서버를 구축한 웹서비스(오른쪽)

표 3. 이미지 차단, 광고 차단 기능을 이용한 전송 데이터 량 감소

| 사이트 | 프록시 미 사용 시 데이터 량(KB) | 프록시 사용 시 데이터 량(KB) | 데이터 감소 비율 (%) |
|-----------------|----------------------|--------------------|---------------|
| Noaa(미국 해양 기상청) | 47 | 40 | 15 |
| 한국 기상청 | 1126 | 204 | 82 |
| 일본 기상청 | 131 | 30 | 78 |
| 조선일보 | 3276 | 1843 | 44 |
| 한겨레 | 2457 | 542 | 78 |
| 뉴스시스 | 2662 | 153 | 95 |
| 다음 | 1228 | 204 | 84 |
| 네이버 | 1433 | 399 | 77 |

표 4. 캐시 기능을 이용한 전송 데이터량 감소

| 사이트 | 캐시 미 사용시 전송 패킷 수/데이터량(KB) | 캐시 사용시 전송 패킷 수/데이터량(KB) |
|-----------------|------------------------------|-------------------------|
| Noaa(미국 해양 기상청) | 20 / 49.9 | 0 / 0 |
| 일본 기상청 | 12 / 131 | 0 / 0 |

비율이 각각 다르게 나오는데 이미지나 광고의 수가 많고 크기가 클수록 데이터 감소 비율이 높게 나온 것이다. 추가적으로 캐시 기능까지 함께 동작 된다면 더욱 많은 양의 데이터가 줄어들게 된다. 표 4에서 캐시 기능만을 사용한 결과를 나타내고 있다.

뉴스, 포털 사이트에서는 많은 데이터들이 실시간으로 변화하기 때문에 정확한 성능 테스트가 어렵다. 그렇기 때문에 데이터들이 많이 바뀌지 않는 기상청 사이트들로 테스트를 하였다. 프록시를 이용하는 한 사용자가 기상청 사이트를 방문하면, 전송 받은 객체는 프록시에 저장된다. 여기서 다른 사용자가 동일한 사이트를 방문하였을 때 추가적인 데이터 전송 없이 기상청 사이트를 이용할 수 있는 결과를 얻었다. 실험한 Noaa와 일본 기상청의 경우를 보면 각각 20개, 12개의 패킷이 발생하였는데 해당 패킷의 요청에 대한 응답이 모두 캐싱 되어 있으므로 위성망을 사용하지 않고도 웹 페이지를 제공받을 수 있었다.

기존 프록시를 이용하여 본 논문에서 제안한 기능을 요청 차단과 캐시 기능을 하는 프록시 각각을 이용하여 구현할 수 있다. 그러나 요청 차단 기능의 서버와 캐시 서버 각각 필요하게 된다. 구현된 프록시는 서버 하나로 두 기능 구현하였고 차단된 이미지를 재요청한다는 점에서 성능을 비교하는 것이 어렵다. 또한 구현된 프록시는 통신 요금 절감에 필요한 기능과 캐시와 필터링 기능만을 구현하였기 때문에 프로그램 용량이 적어 낮은 컴퓨터 사양에서도 동작이 된다.

5. 결 론

현재 해상의 선박들은 위성망을 통하여 웹 서비스를 제공받고 있으며, 그에 따른 많은 요금에 부담을 느끼고 있다. 본 논문에서는 위성망을 사용한 웹 브라우징 서비스에서 문제가 되는 데이터 요금과 속도 문제를 해결하기 위하여 프록시 모델을 개발하였다. 데이터 사용량에 따른 많은 요금을 지불하는 위성망

에서 사용자들의 불필요한 콘텐츠들의 요청을 선박 내에서 차단함으로써 전송 데이터 량을 줄일 수가 있었다. 기존 이미지/광고 차단 프로그램에서는 캐시 기능이 함께 동작되는 사례가 없지만 차단 기능과 캐시 기능을 합쳐서 사용하였고, 또한 기존 이미지 차단 기술에는 차단된 모든 이미지를 사용자가 볼 수 없는 불편함이 있었지만, 본 논문에서 구현한 프록시 모델에서 사용자가 원하는 이미지에 대한 재요청을 가능하게 하여 더욱 실용적인 방안을 제시하였다. 구현된 프록시에서는 이미지/광고 차단 방식이 키워드를 통해 이루어지고 있는데, 기능을 확장시켜 정규표현식을 사용하면 더 정확한 필터링 효과를 나타낼 것이다. 이러한 광고 차단, 이미지 차단, 캐싱 기술을 프록시에 추가하여 사용함으로써 선박의 선원들이 해상에서의 웹 서비스를 저렴한 비용으로 이용할 수 있다.

선박에서는 주요 사이트들을 지정하여 접속 가능하도록 하고 그 외의 사이트들은 접속을 제한시켜 데이터 사용량을 줄이기를 원한다. 선박에 캐시 서버가 존재한다면 캐시 적중률이 높아 많은 데이터 사용량을 줄이는 효과가 나타난다. 그러나 구현된 프록시의 캐시 기능을 지상의 일반적인 유/무선 통신에서 사용한다면 클라이언트들이 더 다양한 웹 사이트를 사용하므로 캐시 적중률이 떨어지게 될 것이다.

본 논문에서는 위성통신을 사용하고 있는 예로 선박을 다루었다. 선박이 아닌 위성통신을 사용하는 다른 환경에서도 프록시 사용을 통해 통신비 절감 효과를 볼 수 있다. 모바일 사용 시에도 구현된 프록시와의 연결이 가능하기 때문에 더 나아가 모바일로 사용한다면 데이터 통신 요금 절감의 효과도 가져올 수 있을 것이라고 본다.

참 고 문 헌

[1] KT Inmarsat 홈페이지(Inmarsat FB 서비스), <http://www.ktinmarsat.com>, 2012.

- [2] Wikipedia, Proxy Server, http://en.wikipedia.org/wiki/Proxy_server, 2012.
- [3] Privoxy. A Web Proxy with Advanced Filtering Capabilities, <http://www.privoxy.org/>, 2001.
- [4] Wikipedia, Web Cache, http://en.wikipedia.org/wiki/Web_cache, 2012.
- [5] HTTP/1.1, RFC2616, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.
- [6] Nginx 홈페이지, <http://wiki.nginx.org/>, 2012
- [7] Caching Proxy, Squid 홈페이지, <http://www.squid-cache.org>, 2012.
- [8] Apache Traffic Server 홈페이지, <http://trafficserver.apache.org>, 2012.
- [9] Varnish Cache, Varnish 홈페이지, <http://www.varnish-cache.org>, 2012.
- [10] 최현희, 김근형, "HTML5 기반 HTTP 스트리밍 환경에서의 서비스 이동성 연구," 멀티미디어학회논문지, 제14권, 제7호, pp. 905-916, 2011.



전 성 운

2008년 3월~현재 동의대학교 영
상정보공학과 재학
관심분야 : 웹, 웹 기반 융합 서비
스 이동, 멀티미디어 스트
리밍



김 근 형

1986년 2월 서강대학교 전자공학
과 학사
1988년 2월 서강대학교 전자공학
과 석사
2005년 2월 포항공과대학교 컴퓨
터공학과 박사

1988년~1990년 LS 산전 연구소 연구원
1990년~1993년 삼성종합기술원 선임연구원
1993년~2007년 8월 (주) KT BcN 본부 수석연구원
2007년 9월~현재 동의대학교 영상정보공학과 부교수
관심분야 : 웹 기반 융합 서비스 이동, P4P 스트리밍,
Content Centric Network