이학석사 학위논문

자바 로봇과 네트워크 카드를 이용한 유해사이트 차단 시스템

지도교수 이 경 현

이 논문을 이학석사 학위논문으로 제출함.

2004年 2月

부경대학교 산업대학원

전 산 정 보 학 과

임 범 춘

임범춘의 공학석사 학위논문을 인준함

2003年 12月 13日

주 심 공학박사 김 창 수 

위 원 공학박사 박 만 곤 

위 원 이학박사 이 경 현 

<목 차>

< 표 목 차 >

< 그 림 목 차 >

Noxious site Blocking System using Java robot and Network Card

Bum-Chun Lim

Department of Computer Scinence Graduate School of Industry

Pukyong National University February 1997

Abstract

It's supposed that the hardware is equipped with network card and specified flash memory using URL contents which are collected and classified through JAVA robot.

The purpose is to block the noxious site using java robot.

The sampling is extracted by java robot application.

We examine the frequency of extracted words and give it to the weight and measure the site similarity. Then the databased information makes a consistent

update through the network card driver of client and communication .

The user's popular site also is processed to be automatically updated flash memory in order to reduce the server's load ,so noxious site blocking system is designed and implmented using java robot and network card.

1. 서론

컴퓨터단말기증후군이라는 병명이 생길 정도로 우리는 인터넷 환경 속에 많이 노출되어 있다. 인터넷상에는 수많은 정보들이 있으며, 이들 중엔 사회적 문제를 야기하는 음란, 폭력, 자살, 도박, 엽기 등의 정보는 이미 유해한 정보로 인식되고 있다. 이와 더불어 업무의 효율성과 교육의 효율성을 위해 수많은 자금을 들인 기업체나 기관에서는 인터넷 활용이 업무 향상과 교육 향상보다는 취미생활이나 유행에 따른 관심사를 검색하는데 인터넷의 활용도가 높다고 한다. 기업체의 경우 증권, 채팅, 레저, 스포츠 등과 같은 순으로 활용도가 높으며, 교육기관의 경우 게임, 채팅, 카페 등과 같은 순으로 인터넷 활용도가 높다고 한다. 유해 정보의 범위와 그 기준은 각 환경마다 차이가 있고, 이러한 유해 정보 차단을 위해 현재 개발되어 있는 차단 프로그램은 어플리케이션 계층에서 구현되어 동작되는 방식이 주류를 이루고 있으며, 차단 방법으로 도메인을 이용하는 방식과 현재 시행중인 내용등급제[2]를 이용하는 방식이 있다.

- 도메인을 이용하는 경우

유해 사이트의 도메인을 DB로 구축하여 목록을 만들어 이용하며, 유해 사이트에 차단 방법은 차단 목록만을 이용하거나 허용 목록만을 이용하는 방법이 있다.

차단 목록만을 이용한 방식의 경우 인터넷의 접근은 자유로우나 유해 사이트에 쉽게 노출되며 차단 목록의 주기적인 갱신 문제가 우선적으로 해결되어야 한다.

허용목록만을 이용하는 방식의 경우 유해 사이트에 노출되지는 않지만 인터넷의 접근이 제한적이다.

• 내용등급제를 이용하는 경우

사이트의 특정부분에 컨텐츠의 등급과 관련된 표기양식을 기술하여 브라우저에서 정의된 등급의 분류 정책에 따라 필터링 하는 방식이다. 다음 표 1은 등급의 분류 정책을 나타낸 것이다.

구 분	사이트 목록 차단	인터넷 내용 등급 시스템
기능	내용 차단	내용 차별
방법	특정목록에 포함된 정보내용 차단	정보내용에 등급을 부여
규제주체	목록을 배포하는 회사, 기관	등급서비스기관 및 이용자
정보선택권	타율/자율	자율
표현의 자유	제한	부분적으로 제한
사업친화도	낮음	상대적으로 높음
학부모의 개입	제한적/불필요	지속적인 감시 필요

표 1 사이트 목록 차단과 내용등급표시제의 비교

본 논문에서 위와 같은 어플리케이션 방식을 좀더 보안하여, 자바로봇을 통한 데이터베이스 확보와 하드웨어에서 차단하는 방식을 제안한다.

데이터베이스 확보에서 차단 목록만을 수집하는 것이 아니라, 허용목록 또한 데이터베이스화하여 데이터베이스 양분화로 서비스를 고려한 사항이 아래와 같다.

첫째, 자바 로봇을 이용하여 DB를 확보하고, 도메인을 이용하여 사이트를 차단하는 방식으로 차단 목록과 허용 목록을 동시에 이용한다.

둘째, 사이트 차단 모듈을 관리하는 별도의 서버와 지속적인 통신을 통하여 목록의 갱신이 용이하다.

셋째, 정책적인 부분이 사이트 차단모듈을 관리하는 서버에 적용되어 내용등급제와 같은 방법도 지원할 수 있는 유연성을 가지고 있다. 사이트 차단 모듈은 드라이버 수준에서 구현되어 사용자에게는 투명한 동작 환경을 제공하며 목록을 별도로 관리하지 않아도 되는 이점을 가진다.

2. 관련연구

본 장에서는 유해사이트의 현황과 차단 시스템의 개발 동향 그리고 기존의 차단 시스템의 구현 방법에 관하여 설명한다.

유해사이트 차단 방법은 응용 프로그램들이 인터넷 접속 프로그램과 운영체제 사이에서 동작하며 사용자의 인터넷 접속을 감시한다. 이러한 응용프로그램들은 유해사이트에 대한 기술적인 차단방법을 차단유형에 따라 분류하며 차단 방법에 의한 구현방법과 차단 위치에 의한 구현방법 두 가지로 분류할 수 있다.

2.1 기존의 사이트 차단 시스템 현황

단일 PC에서 수행되는 개인용 유해정보차단 제품은 1995년 7월

에 사이버 패트롤을 시작으로 일반에게 판매되기 시작하였으며, 현재는 미국 마이크로 시스템즈 소프트웨어사의 사이버 패트롤, Surfwatch, CYBERsitter, Net Nanny, 한국전산원이 개발한 NC Apatrol 1.5가 있으며, PLUS TECH에서 개발한 수호천사 3.0, 수호천사 배포판, 한국정보공학에서 개발한 안티엑스, 아이탑에서 개발한 파로스 등이 있다.

이러한 소프트웨어들은 유해정보 주소목록에 포함된 유해한 사이트를 차단하고, PC 관리자가 사용 중에 추가로 찾은 사이트를 주소목록에 입력하여 차단시킬 수 있다. 또한, 차단하고 싶은 단어를 입력하여 단어가 포함된 유해정보를 차단할 수도 있다.

사이버 패트롤은 요일별로 인터넷 사용 가능 시간대를 설정하거나 최대 인터넷 사용시간을 설정하여 인터넷 사용시간을 제한할 수 있으며, 인터넷 사용내역을 기록하여 PC관리자가 볼 수 있도록 해 줌으로써 PC 사용자들의 건전한 인터넷 활용을 유도할 수 있다.

네트워크용 정보차단 S/W는 인터넷 접속 관문에서 주소목록에 포함되어 있는 사이트를 차단한다.

기존의 클라이언트용 소프트웨어를 서버용으로 확장한 제품과 인터넷 프락시 소프트웨어에 정보 차단 기능을 추가한 제품으로 분류된다. 미국 마이크로시스템즈 소프트웨어사의 Cyber Patrol Proxy와 Secure Computing 사의 Web Track 등이 대표적인 네트워크용 정보차단 소프트웨어이다.

표 2는 기존의 차단 시스템을 정리 하였다.

표 2. 기존의 국내외 차단 시스템

국가	개발사	제품명	특징
미국	마이크로시스템즈	사이버 패트롤	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	Surfwarch	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	CYBERsitter	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	Net Nanny	· 개인용 · 차단목록 · 단어선별 차단기능
미국	마이크로시스템즈	사이버 패트롤 프록시	· 네트워크용
미국	Secure Computing	Web Track	· 네트워크용
한국	한국전산원	NC Apatrol	· 개인용 · 차단목록 · 이용시간 통제
한국	플러스기술	수호천사	· 개인용 · 차단목록 · 이용시간 통제
한국	인터정보	컴지기	· 개인용 · 차단목록
한국	이엠테크놀러지	웹그린	· 개인용 · 차단목록
한국	에이엘테크	아이키퍼	· 개인용 · 차단목록
한국	스마트시스템	SecureDesk	· 개인용 · 차단목록 · IC카드를 이용하여 연령별 통제
한국	아이탑	파로스	· 개인용 · 차단목록
한국	인터피아월드	지키미	· 개인용 · 차단목록

2.2 기존의 유해사이트 차단 시스템의 구현방법

기존의 유해사이트 시스템은 차단방법과 차단 위치에 따라 분류할 수 있다.

2.2.1 차단 방법에 따른 분류

<그림 1>은 차단 소프트웨어가 차단 목록이나 허용 목록을 이용하여 클라이언트의 접근을 제어하는 방법의 개요이다.

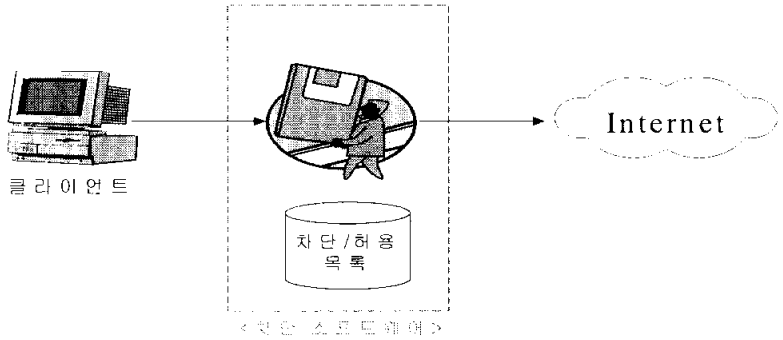

그림 1 차단/허용 목록에 의한 차단 방법

(1) 차단목록에 따른 차단

유해한 정보를 제공한다고 판단되는 사이트만을 차단하는 방법으로 관리자의 눈을 피해 PC 사용자는 인터넷에서 제공되는 대부분의 사이트 접근이 가능하다. 이를 보완하기 위해 수신 정보의 키워드 및 구문 검색을 통해 유해정보를 포함하고 있는 사이트를 차단하는 방법을 함께 사용한다.

이 방법의 단점은 차단 목록의 주기적인 갱신이 어렵고, 포괄적 기준 설정으로 인한 무차별 차단의 가능성이 있다. 또한, 차단 목록 제공자에 의한 실질적인 검열권 행사의 우려가 있다. 그러나, 현재 상용 차단 소프트웨어가 대부분 이 방법에 근간을 두고 있다.

(2) 허용목록에 의한 차단

정보의 내용이 건전하다고 검증되어 허용목록에 등록된 사이트에 대해서만 접근을 허용하고 이외의 사이트는 모두 차단하는 방식으로 유해정보에 노출될 위험은 거의 없으나 접근할 수 있는 사이트가 극히 제한적인 것이 단점이다. 일반적으로 허용목록을 관리하는 관리자는 지정된 패스워드를 입력함으로써 등록되지 않은 새로운 사이트로의 접근이 가능하도록 하고 있다. 이는 학교 등의 특수 환경에서 사용하기에 적합하다.

(3) 내용등급에 따른 차단

일정 기준에 의해 정의된 내용등급에 따라 차단하는 방식으로 인터넷 정보에 유해정도를 나타내는 등급 값을 삽입하여, 인터넷 사용자가 인터넷 정보에 접근할 때 웹브라우저나 차단 소프트웨어에서 인터넷 정보에 삽입되어 있는 등급 값이 관리자가 미리 해당 사용자에게 설정해 놓은 허용 등급 값보다 크지 않을 때만 접근하도록 하는 방법이다.

웹브라우저나 차단 소프트웨어에서 등급에 대한 정보를 가지고 있으며 허용목록에 의한 방법과 마찬가지로 관리자의 관리를 필요로 한다. 그러나, 차단·허용 목록을 보유할 필요가 없으며 연령에 따라 차별적인 등급값을 적용할 수 있으나, 웹서비스를 제공하는 해당 사이트에 의해 등급이 표기되므로 특정 사이트에서 등급값을 표기하지 않았을 경우 효력이 없다.

<그림 2>는 내용등급에 의한 차단 방법의 개요를 보여준다.[3]

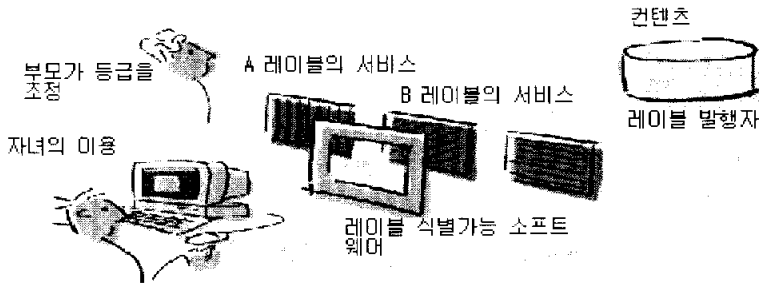

그림 2. 내용등급에 의한 차단 방법

A 레이블과 B 레이블은 내용 등급 정책 표기 부분을 나타낸 것이며, 이 레이블을 통하여 내용등급 정책을 기준으로 동작되는 소프트웨어가 레이블 식별가능 소프트웨어다.

2.2.2 차단위치에 따른 분류

(1) 네트워크 상에서의 차단

라우터나 방화벽이 설치되어 있는 학교나 회사 등에서 사용될 수 있는 방법으로 <그림 3>은 라우터 또는 프락시 서버를 통하여 운영되는 네트워크의 동작방식을 나타내고 있다.

방화벽이나 별도의 서버에만 차단소프트웨어를 설치·운영하므로 관리가 용이하며, 차단 소프트웨어가 클라이언트 시스템과는 별도로 존재하므로 소프트웨어 삭제 등의 위험으로부터 안전하다.

방화벽을 이용할 경우 로컬 네트워크의 모든 호스트들이 방화벽이 설치된 서버를 통하여 인터넷에 접근하는 구조를 가지며 유해사이트 차단 소프트웨어는 방화벽과 일체형이거나 방화벽과 다른 소프트웨어로 존재하게 된다. 따라서, 유지 및 관리는 쉬우나 서버 시스템에 부하가 생기면 성능의 저하가 발생할 수 있다.

그리고, 라우터를 이용하는 경우 라우터가 이해하는 것은 패킷의 IP(Inter- net Protocol)가 존재하는 네트워크 계층이다. 그러므로, 라우터에서는 목적지 주소를 사용하여 차단하고, 클라이언트용 웹브라우저에서 라우터 외부의 프락시(Proxy)를 설정해 놓고 사용하면, 라우터 측면에서는 목적지 주소가 프락시 주소로 되어 있으므로 연결사이트가 유해사이트일지라도 차단할 수 없다.

또 다른 방법으로 네트워크상에 유해정보 차단을 위한 전용서버를 연결하고 네트워크의 패킷을 모니터링하여 차단하거나 프락시

서버와 같이 특정서비스(예 : HTTP)에 대하여 서비스 중계를 하는 과정에서 유해정보를 차단할 수 있다. 이러한 방법도 단지 사이트 주소 목록에 의해서만 선별하기 때문에 내용등급 등을 이용한 탄력적인 선별은 불가능하다.

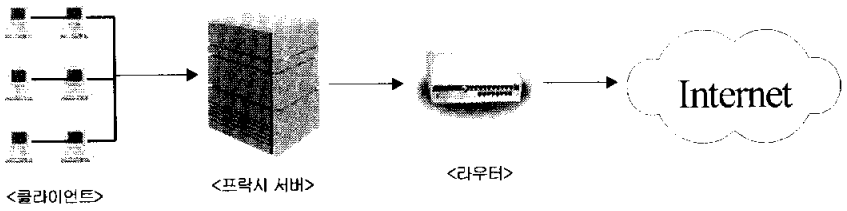

그림 3 네트워크 상의 차단 방법

(2) 클라이언트상에서의 차단

성능 면에서 부하가 거의 없으나 모든 개인용 PC마다 차단 소프트웨어를 설치해야 하며, 초보자일지라도 PC를 관리할 수 있어야 하고, 컴퓨터를 잘 다루는 사용자인 경우 관리자 몰래 차단 소프트웨어에 설치된 안전장치를 무효화시키는 방법을 찾아낼 수 있으므로 안전성이 떨어지는 단점을 가진다.[1]

본 논문에서는 허용목록에 따른 선별 방식과 차단목록에 따른 선별 방식을 적용시켜 유해사이트 차단 기법을 설계 및 구현하고자 한다.

3. 자바 로봇과 네트워크 카드를 이용한 차단 시스템의 설계

본 장에서는 차단 시스템의 개요와 설계에 대하여 살펴본다. 차단 시스템의 설계는 2장에서 기술한 바와 같이 자바 로봇을 이용한 정책 결정에서 URL을 수집 분류를 하여, 여러 가지 방법 중 어느 한가지 방법만으로 관리자가 지정하는 사이트를 차단하는 것은 어려우므로, 착안하여 허용목록과 차단목록을 같이 사용하도록 설계하였다.

클라이언트측 차단 시스템은 네트워크 카드 드라이버의 모듈로 구현되어 드라이버의 일부처럼 동작하므로, 사용자 브라우저의 형태 및 구성에 영향을 받지 않는다.

그리고, 드라이버는 사용자 트래픽에 대하여 최소한의 간섭을 하도록 설계되어 있다. 즉, 사용자가 웹에 접근하는 트래픽만을 실시간 감시의 대상으로 한다. 따라서 드라이버에서는 패킷을 분석할 수 있는 능력이 요구되며 송수신시의 패킷을 분석하여 원하는 패킷의 "ACCEPT" 또는 "DROP" 여부를 결정한다.

3.1 개요

본 논문에서 구현한 모듈은 선마이크로시스템즈사 "java"로 로봇을 구현하였으며, 네트워크 카드 부분은 마이크로소프트사의 윈도우즈 운영체제를 기반으로 ANSI. C로 구현하였다.

드라이버는 마이크로소프트사의 윈도우즈 운영체제상에서 동작되는 커널 모드 드라이버인 미니포트(miniport) 드라이버를 사용하

었다.[3]

미니포트 드라이버의 특징은 하드웨어를 드라이버 제작자가 직접 제어할 수 있으므로, 네트워크 카드와 직접적으로 연결되어 네트워크 카드를 제어할 수 있으며, 표준 하드웨어 이외에도 개발자가 정의하는 하드웨어를 사용할 수 있다.

본 논문에서는 자바 로봇을 통하여 수집·분류된 URL 목록들을 이용하여 별도의 플래쉬 메모리를 네트워크 카드에 장착한 하드웨어를 사용한다고 가정하였다.

플래쉬 메모리는 사이트의 목록을 저장하는 용도로 사용된다. <그림 4>는 전체 시스템의 동작방식을 설명하고 있다.

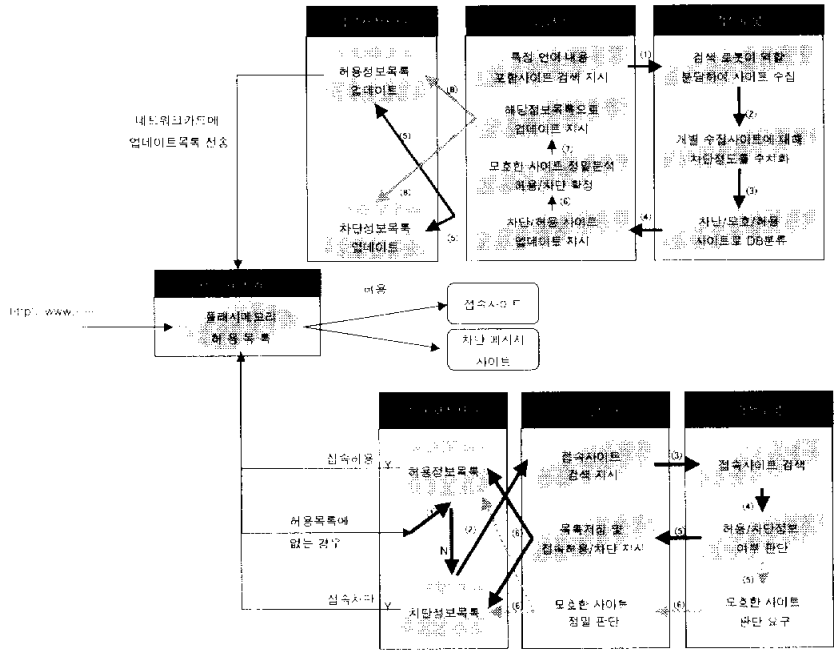

그림 4. 전체 시스템의 동작방식

3.2 자바 로봇의 설계 및 구현

3.2.1 설계

(1) 개요

본 논문에서는 표본 추출 과정은 유해 사이트 100개, 유사 사이트 100개를 사람이 선별하였고, 자바 로봇을 응용하여, 유해 사이트와 유사 사이트의 단어들만을 추출하여 빈도수를 조사한 결과로 단어의 가중치는 표 3과 같은 결과이다.

표 3. 가중치 측정 결과

유해 단어	가중치	유사단어	가중치
sex	1.0000	information	1.0000
porn	1.0000	health	0.9260
pussy	1.0000	therapy	0.7410
xxx	1.0000	sexuality	0.3952
nude	0.9750	marriage	0.1710
sexy	0.4613	abuse	0.1192
suck	0.0371	research	0.1598
lust	0.0046	success	0.0444
:	:	:	:

실제 구현에서는 정확한 사이트의 값을 측정하기 위해 유해 사이트에서 추출한 단어 50개, 유사 사이트에서 추출한 단어 50개로 사이트의 유사도 측정하였다.

(2) 구성도

자바 로봇에서 수집한 문서를 분석하여 자동 분류하는 과정과 분류 엔진을 결과에 따라 DB에 저장된다. 분류 엔진은 형태소 분석과 유사도 측정을 담당한다.

자바 로봇에 구성은 <그림 5>와 같다.

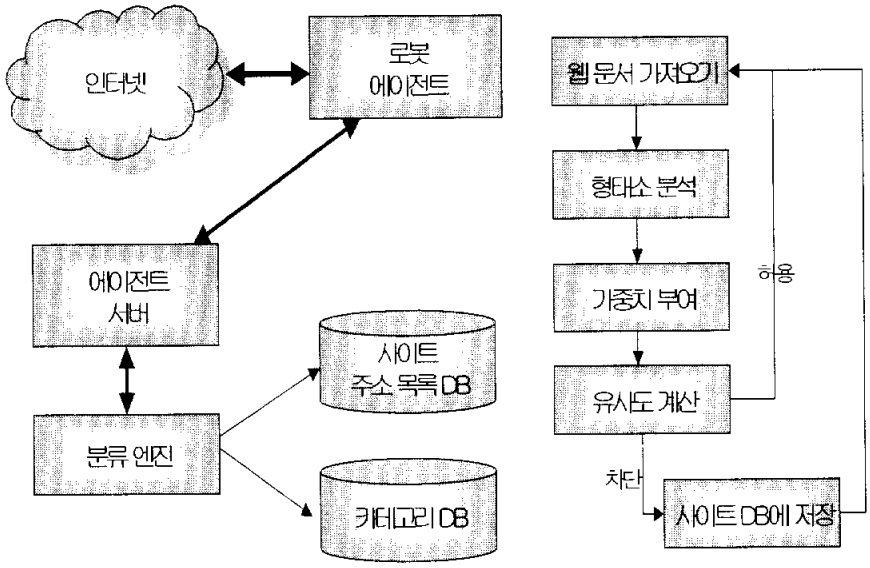

그림 5. 자바 로봇의 구성도

3.2.2 기능 및 구현

(1) 기능

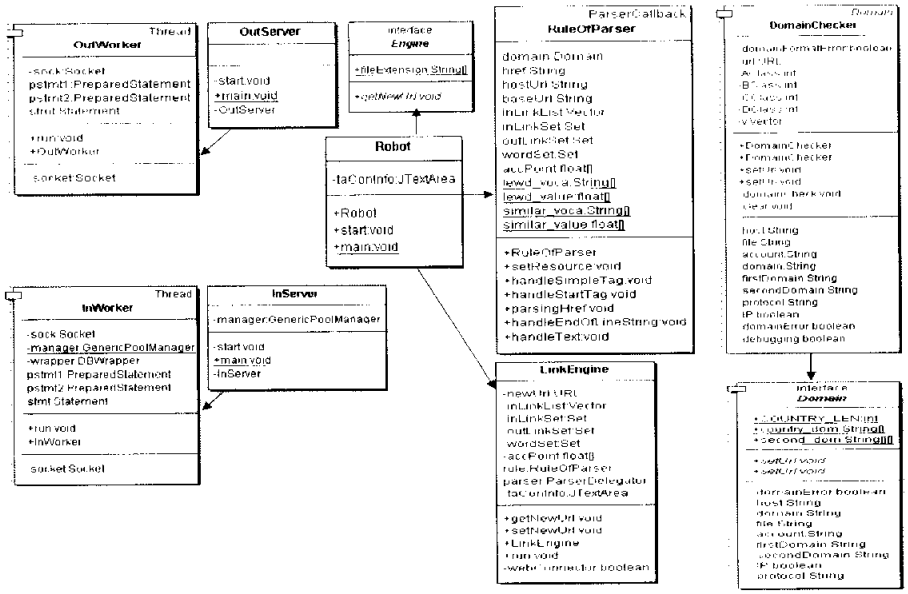

그림 6. 자바 로봇의 개념도

<그림 6>은 자바 로봇의 내부 흐름을 알 수 있는 UML의 클래스 다이그램이다. 각 클래스들과 인터페이스의 역할이다.

- Robot : OutServer에서 출발정보를 받아, 출발지점을 넘기며, 수집된 URL을 InServer로 전달하는 기능.
- RuleOfParser : 사이트의 정보를 규정에 맞게 파싱하는 기능.
- LinkEngine : 새로운 URL과 검색 중인 사이트 하부 URL로 구분하여 문서 검색을 담당하는 기능.
- Domain : 각각 도메인 정책을 기준을 제공하는 기능.

• DomainChecker : 국가별 도메인을 선별하는 도메인을 분리하는 기능.

• OutServer : 로봇과 통신을 할 수 있는 연결고리 기능.

• Outworker : DB에 새로운 출발정보를 요구하여, OutServer을 통해 로봇에게 전달하는 기능.

• InServer : 로봇과 통신을 할 수 있는 연결고리 기능.

• InWorker : 로봇이 InServer 통하여 사이트 관련정보 (URL, 유사도) 저장을 요구하는 기능.

(2) 구현

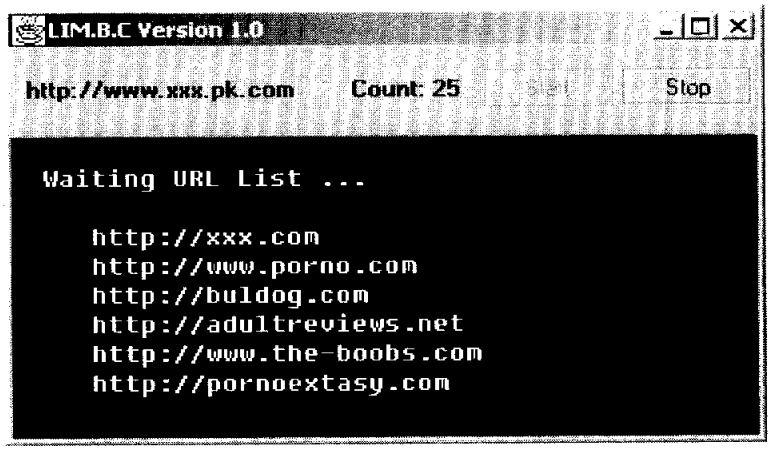

그림 7 자바 로봇 클라이언트 구동 화면

3.3 차단 모듈의 구성 요소

본 논문에서 제안하고자 하는 사이트 차단시스템은 클라이언트에서 구동되는 드라이버와 드라이버의 모듈, 드라이버의 모듈에서 질의시 적절한 응답을 되돌려주는 판별서버로 구성되어 있다.

3.3.1 허용목록

네트워크 카드에 장착된 플래쉬 메모리에 저장되는 목록으로 도메인이 저장되어야 하지만 길이가 도메인마다 다르므로 해쉬함수를 이용하여 128bit의 고정된 길이로 만들어 저장하였다. 표 4에서 플래쉬 메모리에 구성되는 허용 목록의 구성이다.

표 4. 플래쉬 메모리내의 허용목록 구조

index	Hashed 도메인
0	AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
10	55 0A 4F 44 0C 3B 36 66 80 38 06 04 86 6A 1A 7B
20	05 2B 40 53 17 79 7A 59 3B 69 35 3F 0C 0A 05 28
:	:
:	:

플래쉬 메모리의 크기는 512KB의 크기를 가지며, 드라이버에서 플래쉬 메모리를 제어한다. 허용목록을 위해서 다음과 같은 사항들을 고려하여 설계하였다.

(1) 허용목록의 갱신 방법

판별 서버에서는 허용목록과 차단목록을 동시에 유지하고 있다. 차단목록은 자바 로봇을 통하여 목록을 업데이트 할 수 있으며, 허용목록의 갱신은 사용자가 접근하려는 사이트를 분석하여 갱신이 이루어지도록 한다.

사용자가 접근하려는 사이트에 대한 분석은 사용자가 사이트에 접근시 플래쉬 메모리에 존재하지 않는 사이트일 경우 서버에 질의할 때, 질의 정보에 사용자가 접근하고자 하는 사이트의 도메인을 정보로 하여 서버에 전송하여 준다. 이것은 클라이언트 시스템이 판별 서버로의 질의시 반드시 필요한 정보이기 때문에 가능하다. 플래쉬 메모리에 들어가는 내용은 허용목록이며, 이러한 허용목록은 차단목록과 마찬 가지로 수시로 사라지거나 새롭게 생긴다. 게다가 사용자별로 방문하는 사이트의 종류도 다양할 것이다.

따라서, 이러한 허용목록은 사용자가 요구하는 사이트를 플래쉬 메모리에 기록하는 것이 사용자를 위해서는 가장 바람직한 방법일 것이다. 그러나, 모든 사용자의 플래쉬 메모리를 다르게 설정할 수 없기 때문에 사용자가 자주 접속하는 사이트를 플래쉬 메모리에 기록하도록 한다.

드라이버에서 판별 서버로 질의한 사이트 중에서 허용 사이트로 분류되어 있는 사이트에 대해 질의를 하였을 경우 응답 패킷에서 해당 사이트는 허용목록임을 플래쉬 메모리에 기록하도록 응답한다.

만약 사이트가 분류되어 있지 않다면 서버에서는 질의 정보 중

에 포함된 도메인 스트링을 DB에 저장한 다음 전문 검색 요원의 판단아래 판단 결과를 DB에 저장 할 수 있다. 그리고, 이후 같은 사이트에 대한 질의가 오면 서버에서는 드라이버에 해당 사이트의 유해 여부에 따라 드라이버에 기록 여부를 알려준다.

따라서, 플래쉬 메모리에 기록을 하게 되는 시점은 서버에서 유지하는 허용 사이트의 목록에 기록되어 있으며, 사용자가 접근을 시도하는 사이트가 이러한 허용 목록에 존재하는 경우일 것이다.

(2) 클라이언트에서의 허용목록 갱신 및 유지

클라이언트에서 허용목록은 네트워트 카드에 장착된 플래쉬 메모리에 기록되어 유지된다. 따라서 허용목록의 유지는 플래쉬 메모리를 어떻게 관리하는가의 문제로 볼 수 있다.

플래쉬 메모리의 관리 기법은 단순하다. 실시간으로 플래쉬 메모리에 접근하여 쓰고 지우는 것은 상당한 지연시간을 요구하며 보다 빠른 응답을 요구하는 사용자에 대하여 비효율적인 방법이다.

따라서, 특정한 시간에 플래쉬 메모리를 쓰거나 지워야만 한다. 이 시점은 사용자가 컴퓨터를 사용하는데 불편을 느끼지 않는 시간이어야 하며 차단 시스템에서도 부담이 없는 시점을 이용하여야 한다. 이러한 시점을 분류하면 표 5와 같다.

표 5. 플래쉬 메모리의 갱신 시점

시점	장 점	단 점
컴퓨터의 부팅시	쓰기 동작의 오동작 문제가 없다.	플래쉬 메모리에 기록할 내용을 하드디스크에 기록하여 보관하여야 한다.
컴퓨터의 종료시	플래쉬 메모리에 기록할 내용을 하드디스크에 보관할 필요가 없다.	쓰기 동작의 오동작이 일어날 수 있다.

이외에도 사용자가 컴퓨터를 사용중인 상황에서 플래쉬 메모리를 업데이트 할 수 있으나 실시간으로 플래쉬 메모리를 기록할 경우 지연시간이 길어진다면 시스템 전체가 정지상태가 되어 사용자는 OS의 오동작으로 시스템이 비정상인 것으로 오인할 수 있다. 따라서, 컴퓨터의 종료시에 플래쉬 메모리의 기록과 삭제를 수행하는 것이 가장 효과적이므로, 본 논문에서는 컴퓨터 종료시에 플래쉬 메모리의 기록과 삭제를 수행한다.

(3) 사용자의 조작 가능성 방지

허용목록은 사용자가 조작을 할 수 없도록 하여야 한다. 사용자가 허용목록을 조작할 수 있다면 차단 효과를 무력화 시킬수 있다.

3.3.2 WEB DB Access

질의를 할 수 있는 서버의 정보를 가지고 있는 DB로 사용자가 접근하려는 웹사이트가 클라이언트 차단 시스템의 허용목록에 존재하지 않을 경우 구축된 판별 서버로 질의하여 질의 결과에 따라 접근을 제어하도록 하는 것이다. 이때 구축된 판별 서버의 DB는 허용목록과 차단목록을 함께 유지한다. 서버측의 시스템을 구축하기 위해 다음과 같은 사항들을 고려하여 설계하였다.

(1) 차단목록의 갱신 방법

차단목록의 갱신은 주기적으로 웹을 자바 로봇을 통하여 차단정보를 포함하는 사이트의 도메인을 DB에 갱신하는 방법을 사용한다.

(2) 클라이언트 시스템의 관리 및 제어

클라이언트 시스템은 사용자가 사용하고 있는 드라이버와 네트워크 카드상에 장착되어 있는 플래쉬 메모리를 지칭한다. 상술한 바와 같이 플래쉬 메모리의 제어는 드라이버에서 하지만 내용의 삽입, 삭제는 판별 서버에서 전적으로 책임을 진다. 드라이버는 판별 서버의 결정에 따라서 플래쉬의 내용을 삽입 또는 삭제한다.

3.3.3 도메인 추출기

HTTP 헤더[8]에서 도메인 부분을 추출하는 모듈로 패킷 필터링이라 불리는 기법을 사용한다. 패킷 필터링은 어디서, 무엇을 필터링 하는지에 따라 수행하는 기능이 다르다. TCP/IP헤더의 내용을 분석하여 이미 정해진 룰에 따라 처리하는 방식이 있으며, 또 다른 방식으로는 데이터 부분의 내용을 필터링하여 내용에 따라 정해진 룰을 적용시키는 방식이 있다.

내용을 기반으로 한 필터링 방법은 정보의 흐름을 제한하기 위해 패킷의 데이터를 필터링 하는 방법으로 본 논문에서 제안하는 사이트 차단 시스템의 사이트 차단 모듈도 이 방법을 사용한다.

내용을 기반으로 한 필터링 기술은 응용계층에서 생성되는 프로토콜을 이해하고 있어야 하며 낮은 계층의 프로토콜을 이해하여 필터링하는 방법보다는 상대적으로 구현이 어려우며 성능도 떨어지는 단점을 가진다.

차단 시스템의 필터링은 이런 내용을 기반으로 한 필터링을 실시하되 모든 컨텐츠를 검사하는 방식이 아니라 "HTTP request" 페이지만을 검사하여 사용자의 목적지를 알아낸다.

3.3.4 해쉬함수

도메인은 MD5 해쉬함수[10]를 적용시킨 128bit 해쉬값으로 만든다.

드라이버에서 추출하는 도메인 정보는 스트링으로 되어있으며, 이는 플래쉬 메모리에 저장되는 허용목록의 리스트도 스트링으로 저장될 것을 요구한다. 그러나, 도메인 데이터는 가변길이로 최대 255자까지의(255byte) 길이를 가질수 있다. 이러한 데이터를 그대로 사용하는 것은 제한된 크기의 플래쉬 메모리를 비효율적으로 사용하는 방법이다. 게다가 스트링 데이터의 탐색과 유지 관리 또한 어렵다.

따라서, 일정한 길이의 데이터로 만들 필요가 있으며 스트링보다는 수치화 시키는 것이 탐색과 유지 관리등의 작업을 용이하게 만들 수 있다.

이러한 요구 조건에 따라 도메인을 MD5라는 해쉬 함수를 이용하여 128bit 길이의 정량화된 수치로 대신한다.

그러므로 플래쉬 메모리에 저장되는 사이트의 도메인은 MD5로 해쉬화 되어 고정된 길이를 갖는 수치값으로 저장된다. 다음은 도메인을 MD5함수를 이용하여 수치화 시킨 예를 보여 주고 있다.

예) http://kr.yahoo.com -> MD5() ->

55 0A 4F 44 0C 3B 36 66 80 38 06 04 86 6A 1A 7B

위의 예와 같이 유해사이트 차단 시스템에서 사용되는 모든 도메인들을 MD5()를 이용 수치화 하여 플래쉬 메모리 또는 판별 서

버의 DB에 저장을 하며 드라이버에서 HTTP request 패킷의 목적지 도메인을 추출하면 허용목록과 비교하기 위하여 해쉬화 한다.

3.3.5 프로토콜 분석기

송·수신되는 패킷들의 프로토콜[4-8]을 분석한다.

프로토콜의 분석은 클라이언트 차단 시스템의 기능 확장을 고려하여 설계된 부분이다.

프로토콜을 분석할 수 있다는 것은 네트워크 카드가 이더넷에 사용되는 카드를 사용한다라는 전제하에 데이터링크 계층보다 상위 계층에서 사용되는 프로토콜들을 이해하고 있다는 것을 의미한다. 즉, 데이터링크 계층에서 이더넷 헤더를 제외한 데이터를 분석하여 어떠한 프로토콜이 사용되었는지를 분석할 수 있다.

본 논문에서 제안하는 클라이언트 유해 차단 시스템은 데이터링크 계층의 이더넷 프로토콜 외에 국내에서 널리 사용되고 있는 ADSL(Asymmetric Digital Subscriber Line)의 PPPoverEthernet 프로토콜, IP, TCP, UDP 그리고 HTTP를 지원하도록 설계되었다.

3.3.6 패킷 변조기

판단 모듈에서 모듈 매니저 서버로의 질의를 요구한다면 HTTP request 패킷을 모듈 매니저 서버로 전송하도록 변조하여 질의에 필요한 데이터를 패킷에 삽입한다. 패킷의 변조는 "IP spoofing" 기법을 참조하여 본 논문에서 제안한 시스템에 맞게 수정하여 사용 하였다.

3.4 모듈 매니저 서버

서버측의 구성은 차단 모듈을 관리하고 차단 목록의 지속적인 업데이트가 가능하도록 설계한다.

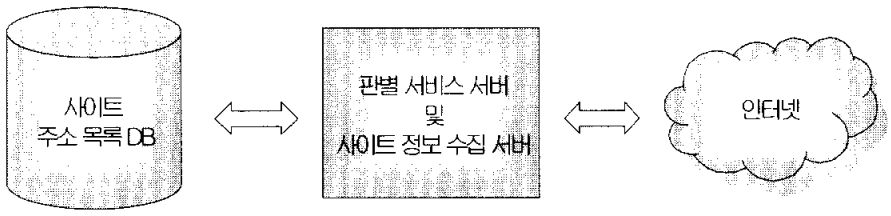

그림 8. 클라이언트 관리 및 사이트 정보 수집 서버의 개요

차단 시스템의 차단 모듈에서의 질의를 처리하거나 차단 모듈의 허용목록 업데이트와 서버측 차단 목록의 업데이트를 수행한다.

차단 목록의 수집은 검색 로봇을 이용하여 수집할 수 있으며, 모듈 매니저의 역할은 차단 모듈을 관리하는 정책을 결정한다. 정책의 내용은 차단 모듈의 허용 목록 업데이트 또는 제거 방법, 차단 목록의 업데이트 또는 제거 방법, 차단 모듈에 적용할 카테고리(사이트의 분류, 예를 들어 폭력, 도박, 음란, 증권, 게임, 채팅 등)의 범위 결정 등의 정책을 수립하여 모듈 매니저에 적용하여야 한다. 모듈 매니저의 차단 목록 수집에 의해 주기적으로 업데이트 된다.

4. 로봇 에이전트와 네트워크 카드를 이용한 사이트 차단 시스템의 구현

본 논문에서 사용되는 차단 방법은 허용목록을 사용하여 차단하는 방법을 적용하였으며 모듈관리자에 의해 허용목록이 업데이트되도록 설계되었다.

송·수신되는 패킷들은 차단모듈에서 감지해야 하는 HTTP패킷이거나 모듈 매니저의 응답이거나 차단 모듈에서 처리가 필요하지 않은 패킷일수도 있으므로, 패킷의 프로토콜을 분석하여 적절한 처리를 하여야 한다.

차단 모듈은 클라이언트에서 동작하도록 구현하되 어플리케이션이 아닌 드라이버 수준에서 구현하여 차단 시스템이 무력화되는 것을 보완하으며, 본 논문에서 구현한 도메인 추출기는 현재 널리 사용되고 있는 웹 프락시(Proxy) 서버를 통한 트래픽 제어도 가능하도록 설계하였다.

웹 프락시를 통한 트래픽은 HTTP request부터 시작하는 것이 아니라 웹 프락시에 목적지 사이트의 도메인을 질의어나 데이터로 전송하는 방식을 취한다. 따라서, 첫 번째 HTTP request 이후의 HTTP 패킷의 헤더를 지속적으로 분석하여야 하는 단점이 있다.

4.1 차단 모듈의 송·수신 동작

<그림 9>에서 차단 모듈의 판단 모듈에서의 판별결과가 모듈 매니저 서버로의 질의를 요구하고 있으며 사용자의 HTTP request 패킷을 UDP(User Datagram Protocol)패킷으로 변조하여 판단에 필요한 데이터를 삽입하여 서버로 보내게 된다.

UDP를 사용할 경우 패킷이 유실 될 수 도 있으나, HTTP request 패킷을 보내는 것을 의미는 사용자의 PC와 사용자가 접근하려는 웹 서버 사이에 TCP의 three-way hand shake가 성공적으로 수행되었다는 의미이기도 하다.[7]

그러므로, 운영체제의 TCP(Transmission Control Protocol) /IP(Internet Protocol) 스택에서는 드라이버에 정상적인 HTTP request 패킷이 전달된 것으로 판단된다.

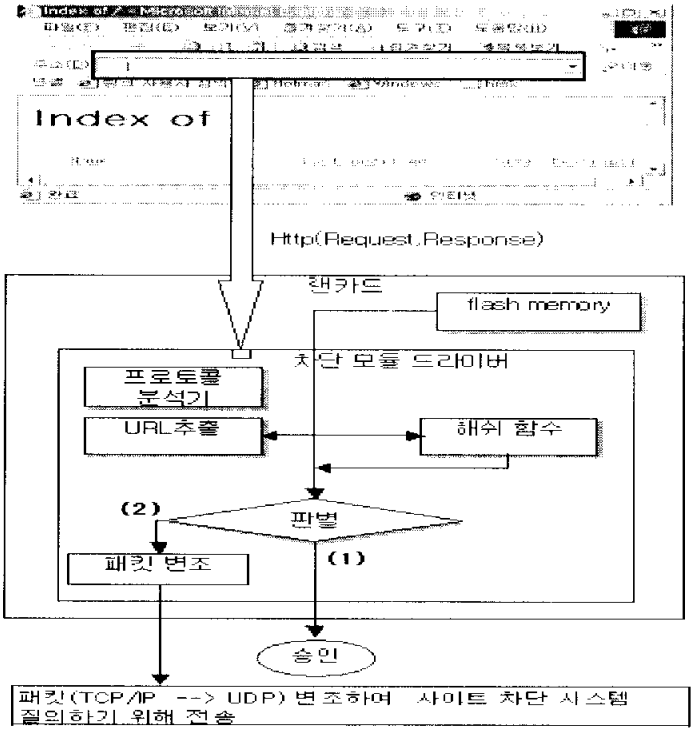

그림 9. 차단 모듈의 송신시 동작과정

모듈 매니저 서버에 보낸 UDP 패킷이 유실되어도 운영체제의 TCP/IP 스택의 재전송 메커니즘을 이용할 수 있게 된다. 그리고, UDP[9]의 속도는 TCP에 비해 빠르다. <그림 9>의 (1)은 판단 모듈의 판별 결과가 허용사이트로 판단된 결과이며 HTTP request 패킷을 정상적으로 처리하도록 한다.

본 논문에서 제안한 차단 시스템의 차단 모듈은 HTTP request 패킷을 대상으로 하여 차단하고 있어 차단 사이트일 경우 HTTP request 패킷은 차단 사이트에 도달하지 못한다.

따라서, HTTP request 패킷을 변조하여 모듈 매니저로의 질의

를 수행하게 되며 질의에 대한 모듈 매니저의 응답 UDP 패킷을 수신 과정에서 처리한다.

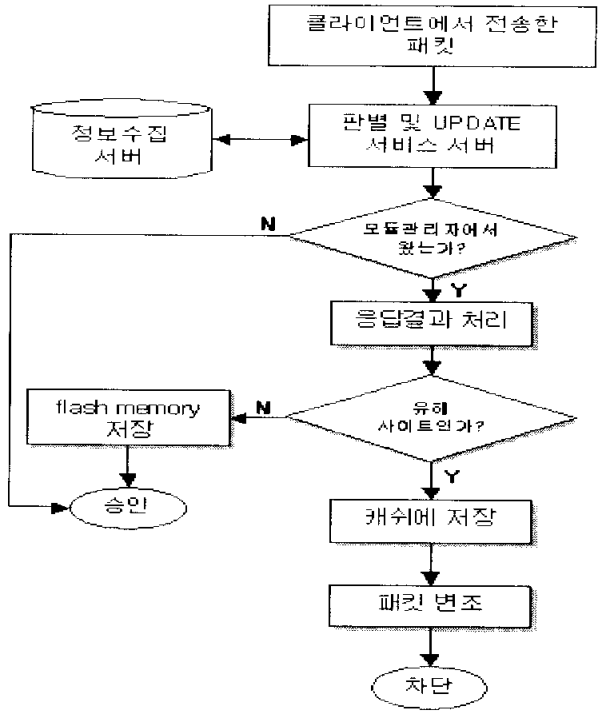

그림 10. 차단 모듈의 수신시 동작과정

즉 수신되는 HTTP 패킷이나 다른 용도의 패킷은 처리를 하지 않는다.

모듈 매니저가 보낸 응답 패킷을 받았을 경우 <그림 10>과 같은 처리를 하게 되며 응답 패킷의 판단결과 데이터가 유해 사이트인 경우 응답 UDP패킷을 사용자가 접근하려던 웹사이트의 응답 HTTP redirection 패킷으로 변조하여 TCP/IP 스택에게 세션의 응답으로 보내주고 임시 메모리에 도메인의 해쉬값을 기록한다. 응답 패킷의 판단결과 데이터가 허용 사이트인 경우 해당 사이트의

도메인의 해쉬값을 플래쉬 메모리에 기록하게 된다.

4.2 패킷 필터링

차단 모듈의 필터링 기능은 내용에 기반한 필터링 방식을 사용하고 있다. 다음 <그림 11>은 필터링 절차를 보여준다.

<그림 11>에서와 같이 핵심적인 내용은 HTTP 헤더의 내용을 검사하는 부분이다.

HTTP request 헤더의 내용 중 필요한 부분은 도메인이 존재하는 필드로, 도메인을 필터링하여 네트워크 카드의 플래쉬 메모리에 저장되어 있는 허용목록 사이트와 비교하여 허용 여부를 결정한다.

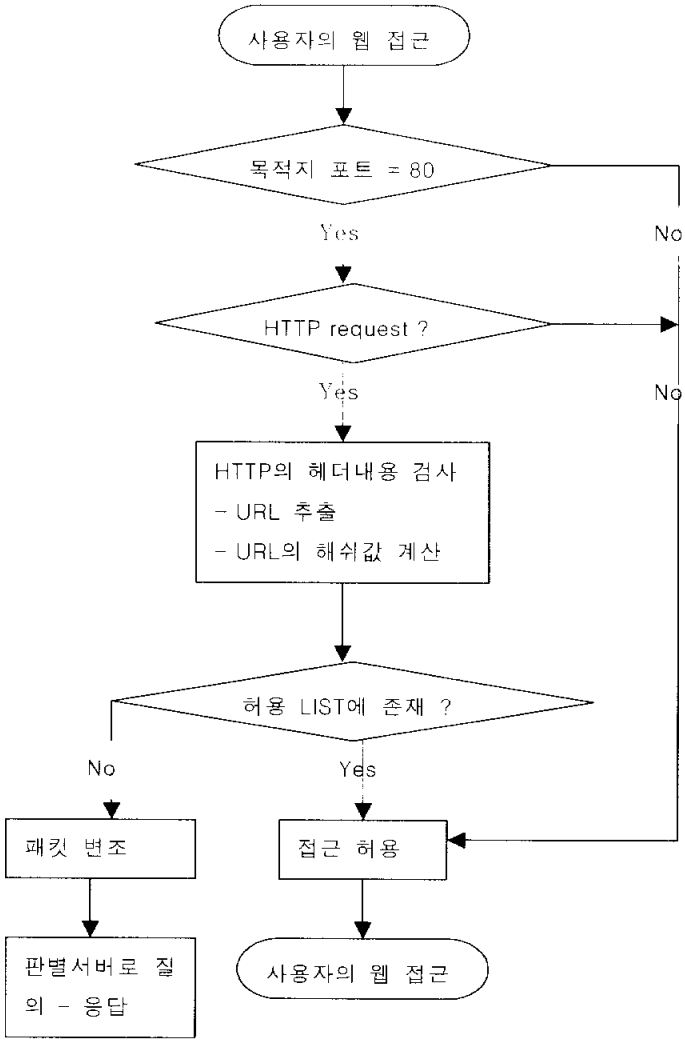

그림 11. 차단 모듈에서의 패킷 필터링

이와 같이 정보의 필터링을 위해서는 HTTP를 이해하고 있어야 한다. 그리고, 헤더 내용을 검색하기 위해서는 응용계층처럼 문자열 기반으로 검색을 하여야 하므로 특정 주소나 포트만을 필터링하는 방식 보다 상당히 효율이 떨어지는 단점을 가진다.

커널 기반의 소프트웨어인 드라이버에서는 응용계층에서 지원하

는 함수들을 이용할 수 없으므로 문자열을 검색하는 기능을 구현하여야 한다.

드라이버에서는 이러한 일련의 패킷들을 다시 역캡슐화를 이용하여 사용자의 메시지를 알아내어 적절한 처리를 수행하게 되는 것이다.

4.3 패킷 변조

패킷 변조는 의미 그대로 패킷의 내용을 수정하는 것으로, 사용자가 접속하려는 사이트가 사용자 컴퓨터에 보관되어 있는 허용목록의 리스트(네트워크 카드의 플래쉬 메모리에 저장되어 있다.)에 존재하지 않으면 세션의 방향을 바꾸어 본 논문에서 구현한 모듈 매니저 서버로 질의를 할 필요가 있다.

이런 사용자의 질의에 모듈 매니저 서버에서 사용자가 접근하려는 사이트의 차단 여부를 판별하여 응답을 보낸다. 응답의 결과가 차단 사이트라면 사용자의 연결을 차단 페이지로 리다이렉션을 하게 되고 허용 사이트라면 접근을 허용하는 것이다.

이렇게 사용자의 세션을 이용하여 패킷을 변조하여 내보냄으로 사용자와 OS는 정상적인 접근이 이루어지고 있다고 생각하게 된다. 그러므로, TCP의 흐름제어를 방해하지 않으면서 목적을 달성할 수 있는 것이다.

한가지 유의점은 서버에서는 패킷의 리다이렉션에 관련되는 정보를 알 수가 없다. 리다이렉션에 관련된 정보를 사용자의 시스템

메모리에 보관할 수 도 있으나 세션의 관리와 성능에 따르는 문제가 생길 수 있으므로, 사용자의 패킷에서 추출된 정보를 질의 패킷에 같이 보내어 서버에서의 응답으로 리다이렉션에 관련된 정보를 취득하는 것이 효율적이다. 이런 방식으로 리다이렉션에 관한 정보를 드라이버에서 사용할 수 있다.

차단 모듈에서의 패킷 변조는 송신시에 한번 그리고 수신시에 두번에 걸쳐 일어난다.

송신시의 패킷 변조는 모듈 매니저 서버로 질의 데이터를 보내기 위해 사용되고 수신시의 패킷 변조는 사용자의 접속을 리다이렉션 시키기 위해 사용된다.

패킷 변조시에는 TCP의 흐름을 방해하지 않기 위해 해당 세션의 흐름을 파악하고 있어야 하며, HTTP request 패킷이 나가는 상태는 3 way hand- shake가 이루어진 상태이므로 이후의 상태를 예측하여 동작하도록 하면 된다.

표 6. 송신시의 패킷 변조

내용 / 위치	추출	삽입	변조 여부
IP 헤더	· Destination Address · Identification	· 모듈 매니저 서버의 주소 · 변조된 패킷길이 · 변조된 CheckSum	변조
TCP 헤더	· Sequence Number · Acknowledge Number · Source Port	· UDP로 대체 · 변조된 CheckSum	변조
HTTP 헤더 or 데이터	· 도메인	· 도메인의 해쉬값 · Destination Address · Identification · Sequence Number · Acknowledge Number · Source Port	변조

송신시의 패킷 변조는 사용자의 HTTP request 패킷을 서버로 질의하도록 하기 위해 UDP로 변조하는 것으로 표 6과 같은 조건과 과정으로 이루어진다.

다음의 응답형식에는 질의시에 없는 데이터가 추가되어 있다. 그것은 리다이렉션 페이지의 정보와 플래쉬 메모리의 관리에 사용되는 정보들이다.

수신시의 패킷의 변조는 사용자에게 적절한 응답을 주기 위해 사용되며, 사용자에게 적절한 응답을 주기 위해서는 모듈 매니저 서버로의 질의시에 이용한 사용자 세션의 복구가 필요하다.

즉, 사용자의 HTTP request에 대한 응답을 해 주어야 한다. 수

신시에 패킷의 변조가 필요하다는 것은 사용자가 접근을 시도 했던 사이트가 차단 사이트라는 판정을 받은 경우이므로, 모듈 매니저 서버에서의 응답 패킷을 HTTP response 패킷으로 변조하여 사용자에게 되돌려 주어야 한다.

표 7. 수신시의 패킷 변조

내용 / 위치	추출	삽입	변조 여부
IP 헤더		· Source Address · Identification · 변조된 패킷길이 · 변조된 CheckSum	변조
UDP 헤더		· TCP로 대체 · Sequence Number · Acknowledge Number · Destination Port · 변조된 CheckSum	변조
HTTP 헤더 or 데이터	· 판단결과 · 서비스 코드 · HTTP 리다이렉션 헤더	· HTTP 리다이렉션 헤더	변조

수신시의 패킷 변조는 송신시의 패킷 변조와 반대 과정을 거쳐 패킷을 재조립하여, 데이터의 내용은 드라이버에서 응답 데이터의 해석을 끝내고 리다이렉션에 관련된 데이터를 채운다.

리다이렉션에 관련된 데이터는 HTTP 헤더로 이 헤더는 서버에서 결정하여 보내준다. 즉 서버측에서 리다이렉션 방향을 결정할 수 있는 것이다.

1.4 통신 방법 및 데이터 형식

서버로의 질의와 응답 절차는 사용자의 세션이 이루어진 이후에 일어나는 상황이므로, 사용자에게 빠르고 적절한 응답을 해주어야 할 책임이 있다.

다음 <그림 12>은 사용자의 세션과 세션 도중의 질의와 응답이 이루어지는 과정을 보여준다.

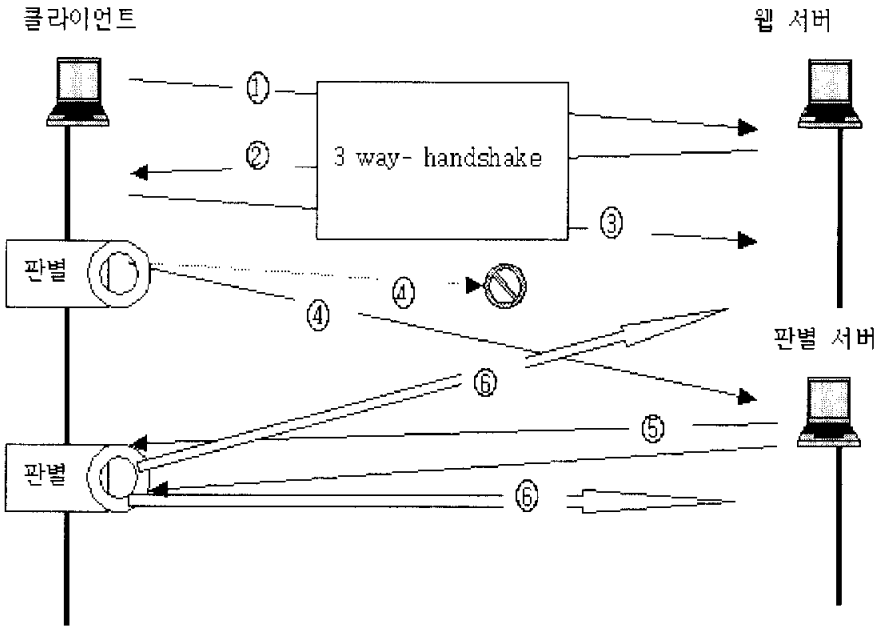

그림 12. 차단 모듈의 Session hijacking

<그림 12>에서 나타나는 통신과정은 차단 모듈의 유해 사이트 차단 과정 중 드라이버의 동작 부분을 제외하고 서버와의 통신에 관련되는 부분을 설명하고 있다.

다음은 통신 과정에 대한 설명이다.

① 사용자의 호스트는 접속하려는 웹서버로 연결을 요구하는

"TCP Syn" 패킷을 전송한다.

② 웹서버가 존재하면 사용자의 호스트로 "TCP Syn" 패킷에 대한 응답과 자신의 "TCP Syn" 패킷을 전송한다.

③ 사용자의 호스트는 웹서버의 응답과 연결을 허락하는 패킷을 받음과 동시에 응답을 보낸다. 이로써 웹서버와 사용자의 호스트는 세션을 시작할 수 있게 된다.

④ 사용자의 호스트는 요청할 HTML문서의 위치와 요청형식이 담긴 HTTP 헤더를 웹서버로 전송한다..

이때, 차단 모듈의 사이트 차단 기능이 동작을 하게 되어 드라이버에서 자체 판단을 할 수 없는 경우 판별 서버로 질의문을 보내게 된다. (본 논문에서는 질의문을 보내는 것으로 가정한다.)

여기서 사용자 호스트의 HTTP request 헤더는 판별 서버로의 질의문으로 변경되고, 질의 패킷은 UDP 패킷이다.

⑤ 질의문에 대한 응답으로 판별 서버에서는 두개의 UDP 패킷을 전송한다. 이것은 사용자가 접근하려는 웹서버가 차단 사이트일 경우 리다이렉션을 염두에 둔 것이다.

⑥ 응답 패킷의 판별 여부에 따라 사용자의 접근을 허락 또는 거부하게 된다. 사용자의 접근을 거부하게 되면 응답이 왔던 것 처럼 보이기 위해 리다이렉션(판별 서버에 있는 접근 거부 페이지로)을 하게 된다.

현재 질의와 응답에 사용되는 통신 형태는 신뢰성이 없는 UDP

를 사용하고 있으며, UDP를 사용하는 이유는 다음과 같다.

(i) TCP와 같이 연결을 유지하기 위한 작업이 없으므로 속도가 TCP보다 빠르다.

(ii) UDP를 사용하더라도 신뢰성을 보장할 수 있다.

이 과정에서 유해차단 모듈의 질의 또는 응답 패킷이 소실된다면 사용자의 브라우저는 자신의 TCP세션에 문제가 있는 것으로 생각을 하게 된다. 따라서, 차단 모듈이 가로채었던 HTTP request 패킷이 분실되었거나 웹서버의 응답이 분실 된 것으로 착각을 하게 되는 것이다.

그러므로, 사용자의 브라우저는 다시 HTTP request 패킷을 내보내게 되고, 차단 모듈은 UDP를 사용하지만 TCP의 신뢰성을 이용 할 수 있게 된다.

리다이렉션은 사용자가 접근을 시도한 사이트가 차단정보를 포함하고 있을 때 일어나게 된다.

사용자가 차단 사이트에 접속시 리다이렉션을 시키지 않는다면 사용자는 TCP의 응답 대기 시간동안 아무런 응답이 없는 상태로 기다려야 하며 결과로 서버를 찾을 수 없다는 브라우저의 메시지를 받게 된다.

이런 대기 상태와 네트워크의 상태에 대한 오해를 피하고 사용자에 즉각적인 응답을 주기 위해 리다이렉션을 사용한다. <그림 13>은 리다이렉션의 절차를 나타낸다.

<그림 13>에서처럼 리다이렉션 과정은 두개의 UDP 즉 응답 패

킷을 사용자가 접속을 시도한 웹서버의 응답 패킷으로 속여 목적을 달성한다.

데이터들은 드라이버에서 서버로의 질의시 사용자의 HTTP request 패킷을 변조하여 판별 서버로의 질의 패킷으로 재조립을 하는 과정에서 리다이렉션에 필요한 데이터를 따로 분류하여 질의 패킷에 데이터로 저장 한다.

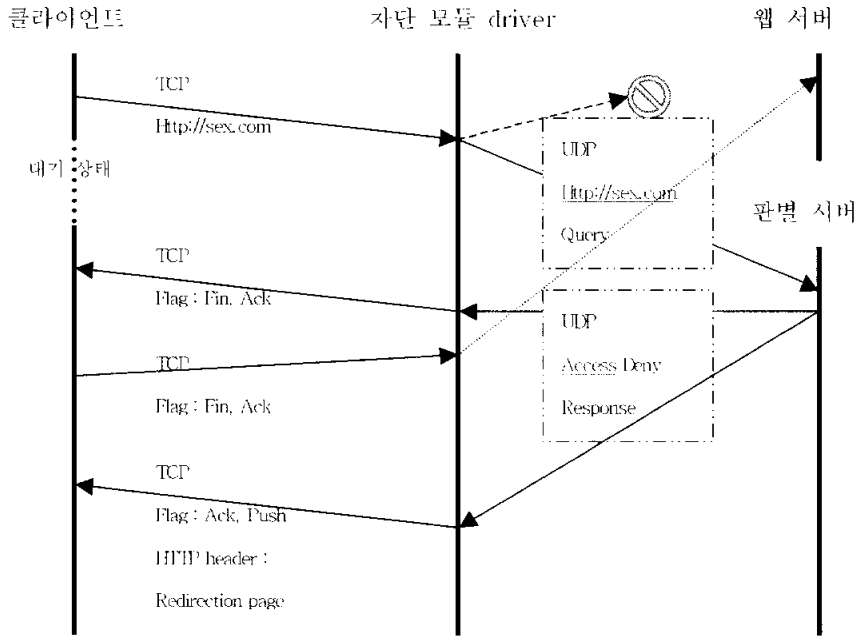

그림 13. 접근 거부에 의한 리다이렉션의 절차

그리고, 서버에서 응답 패킷에 이 데이터들을 저장하여 드라이버에게 보내면 드라이버에서 수신받은 패킷에 포함되어 있는 데이터들을 이용하는 구조이다. 이 경우 드라이버에서는 세션을 감시하거나 데이터를 저장하는 등의 번거로운 작업을 피할 수 있다.

이러한 방법이 가능한 것은 TCP의 흐름이 예측 가능하기 때문이며 언제나 같은 흐름을 유지하기 때문이다.

<그림 14>는 모듈 매니저와 드라이버의 질의 응답에 대하여 설명하고 있다. 차단 모듈의 질의 데이터 형식과 모듈 매니저의 응답 데이터 형식은 다음 표 8과 같다.

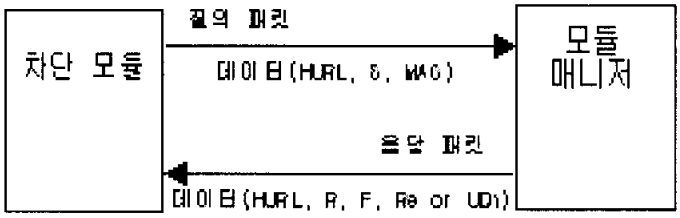

그림 14 차단 모듈과 모듈관리자와의 통신

표 8. 차단 모듈과 모듈관리자와의 통신 데이터 형식

데이터 형식	내 용
H	· 도메인 해쉬를 취한 도메인 데이터
S	· 서비스 코드
MAC	· 사용자 랜카드의 이더넷 어드레스
R	· 모듈 메니저의 판단결과
F	· Flag 데이터의 존재 유·무와 종류 길이를 표기
Re	· Redirection에 사용될 HTTP 헤더의 내용
UD	· 허용 목록에 업데이트될 데이터 · 데이터의 크기는 16바이트 이며 해쉬 함수를 이용하여 얻어지는 도메인의 해쉬값

서비스 코드는 현재의 차단 모듈을 관리하기 위한 코드로 차단되는 차단정보의 카테고리를 식별하도록 도와주는 역할을 한다. 즉, 사용자의 차단 모듈은 증권정보를 가진 사이트만을 차단하고 싶을 경우 서비스 코드의 값은 증권 정보에 대한 값을 가지게 되는 것이다.

모듈 매니저는 서비스와 차단 모듈의 관리를 위한 체계적인 정책을 필요로 하다.

허용 목록의 업데이트는 허용 목록의 버전정보를 모듈 매니저가 DB로 구축하여 관리하며 차단 모듈과 DB의 허용 목록 버전 정보와의 연관은 MAC으로 해결할 수 있다.

그리고 이러한 버전 정보를 통하여 차단 모듈의 질의시 마다 응답 패킷에 업데이트 데이터의 삽입여부를 결정할 수 있다.

4.5 플래쉬 메모리 관리방법

전용 네트워크 카드에는 플래쉬 메모리가 탑재되어 있으며 플래쉬 메모리의 용도는 차단 모듈에서 이용하는 허용목록이 유지된다.

허용목록은 플래쉬 메모리의 용량이 한계가 있음으로 공장에서 초기값으로 일부분만을 채워 넣는다. 이후의 목록은 사용자가 자주 접근하는 허용 사이트의 도메인을 서버에서 분석하여 허용 사이트로 판별된 사이트의 도메인 해쉬값이 플래쉬 메모리에 업데이트 된다.

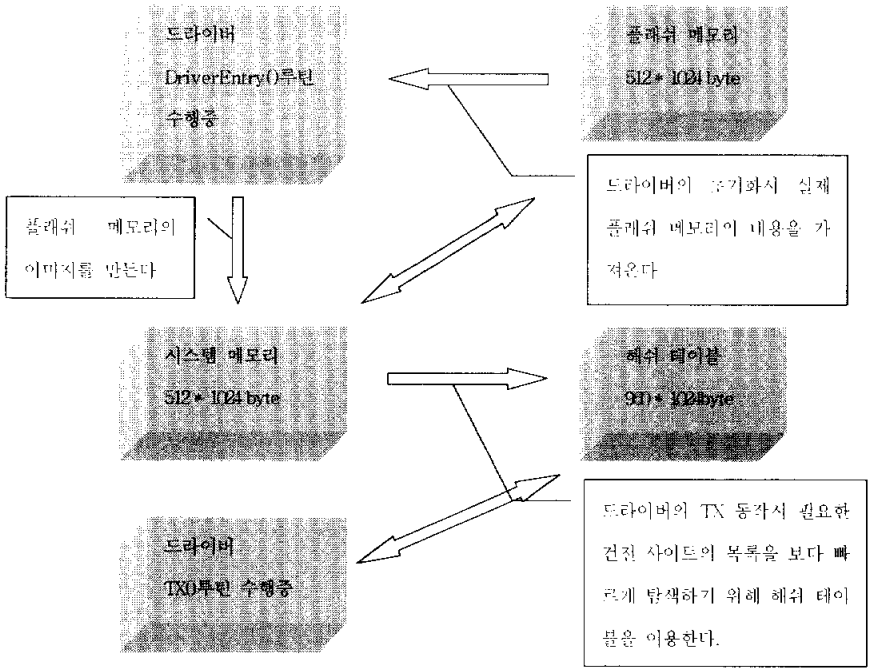

그림 15. 플래쉬 메모리의 관리와 탐색기법의 개요

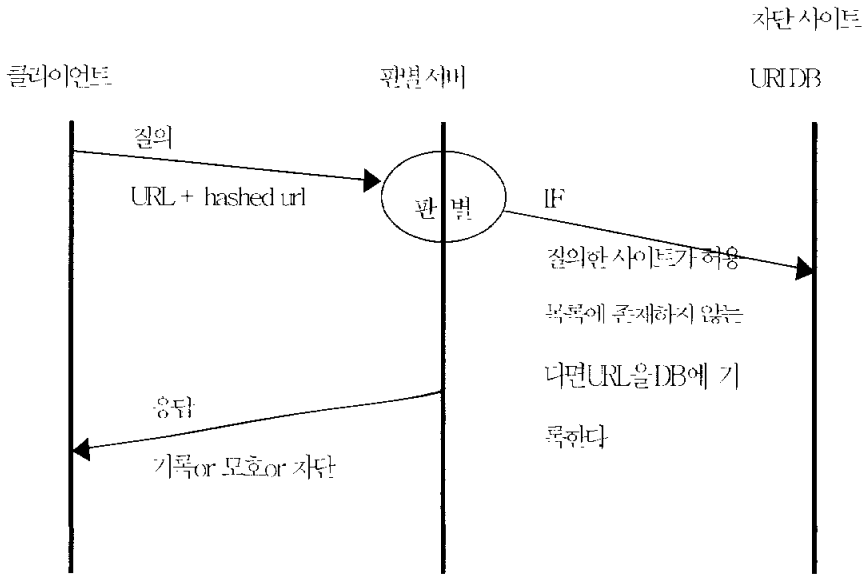

그림 16. 플래쉬 메모리의 업데이트

플래쉬 메모리에 실시간으로 기록되는 도메인들은 서버에서 건전 사이트라는 판별이 되어 있는 것들로 <그림 16>를 보면 응답 패킷에서 3가지의 상태 코드를 볼 수 있다.

이 상태 코드는 표 8에 정의된 데이터 형식의 F에 정의된다. 상태 코드의 의미는 다음의 표 9과 같다.

표 9. 모듈 매니저 서버의 응답 내용

상태코드	의 미
기록	· 플래쉬 메모리에 사이트를 기록 · 서버의 허용 사이트 목록에 존재
모호	· 서버의 DB에는 존재를 하지만 아직 판별이 되지 않은 사이트 · 차단 사이트의 목록에도 없으며 허용 사이트의 목록에도 존재하지 않는 사이트.
차단	· 서버의 차단 사이트 목록에서 발견된 사이트 · 드라이버에서 이런 응답을 받으면 리다이렉션을 함

이러한 상태 코드에 따라 드라이버에서는 플래쉬에 기록을 하게 된다.

해쉬 테이블은 실제의 플래쉬 메모리 내용을 포함하고 있지만 플래쉬 메모리의 업데이트에는 영향을 미치지 않는다.

5. 시뮬레이션 및 분석

본 논문에서 구현한 유해 사이트 차단 시스템은 2장에 기술된 방법들의 단점들을 상당 부분 보완하였으며 모듈 매니저의 구현 시 체계적인 정책을 수립하여 적용한다면 유연한 구조의 시스템을 구축할 수 있을 것이다.

<그림 17>은 구현된 유해사이트 차단 시스템의 결과를 보여 주고 있다. 사용자가 http://www.sex.com의 사이트로 접근을 시도하면 차단 페이지가 브라우저에 나타난다.

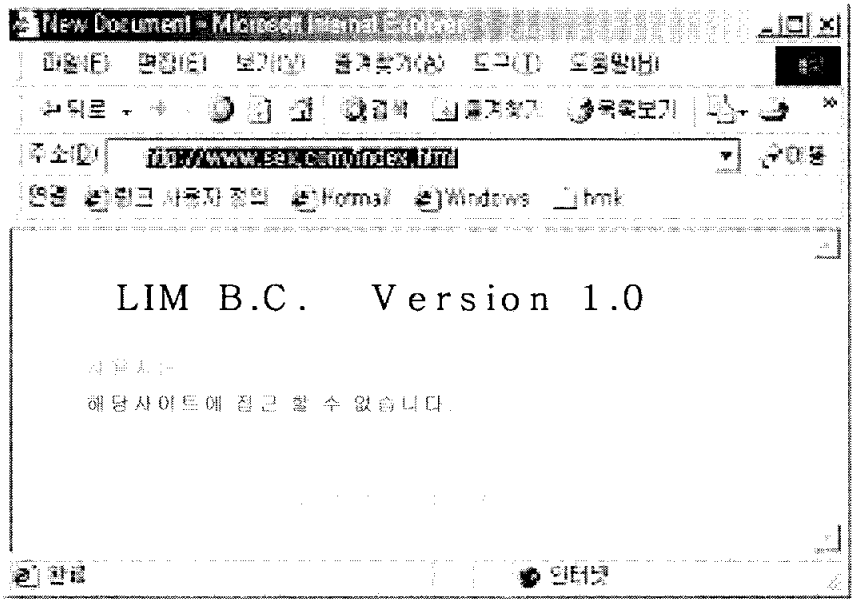

그림 17. 구현된 차단 시스템을 적용한 시스템에서의 테스트 결과

차단 모듈의 동작환경이 운영체제의 커널 부분 중에서도 하위레벨이므로 차단 사이트의 차단 이외에도 다양한 응용이 가능하다. 예를 들어 기초적인 침입차단 시스템을 구축한다면 드라이버는 모든 트래픽을 직접적으로 처리하므로 침입차단 모듈을 드라이버에 추가하여 침입차단 시스템을 구축할 수 있다.

표 10는 기존의 구현된 차단 시스템과 본 논문에서 구현한 시스템을 비교하였다.

표 10. 기존 시스템과 구현 시스템의 비교

	장점	단점
허용 목록	노출되지 않는다.	접근 가능한 사이트가 제한적이다.
차단 목록	인터넷에 제공되고 있는 대부분의 정보에 접근 가능	차단목록의 주기적 갱신이 어렵다.
내용 등급	다양한 환경에 적용 가능한 융통성과 보안성을 제공	차단 S/W와 웹브라우저에서 등급시스템을 지원하지 않거나 등급표시가 안된 사이트에는 내용등급에 의한 선별방법의 적용이 불가능
제안 방법	상술한 세가지 방법을 모두 적용 가능, 허용 목록의 갱신가능	차단 목록의 갱신이 어렵다.

6. 결론

본 논문에서는 응용 어플리케션을 주류로 이루고 있는 차단 시스템의 문제점과 차단 방법을 살펴보았다.

기존의 차단 소프트웨어와 본 논문에서 제안한 차단 시스템은 다음과 같은 차이점이 있다. 첫째, 관리의 용이성이다. 본 논문에서 구현한 차단 시스템은 허용 목록이 지속적으로 업데이트 되도록 설계되어 있으며 목록의 작성도 실시간으로 사용자가 자주 방문하는 사이트를 위주로 작성하도록 구현 되어있다. 둘째, 사용자의 시스템에서 투명하게 동작한다. 셋째, 호환드라이버가 아닌 전용 네트워크 카드를 가지는 드라이버 방식으로 제작할 경우 성능의 향상과 함께 클라이언트 방식의 유해 사이트 차단 소프트웨어의 단점인 차단 모듈의 무력화를 보완 할 수 있다.

현재 상용화되어 있는 유해 사이트 차단 소프트웨어들은 클라이언트용이며 일부 네트워크용 차단 소프트웨어가 존재하지만 성능의 개선이 시급한 상황이다. 네트워크와 PC의 성능 향상에 따라 본 논문에서 제안한 차단 모듈의 관리를 전담하는 서버도 성능의 향상이 이루어져야 한다. 현재 본 논문에서 차단 모듈의 체계적인 관리 정책이 추가로 필요하여 차단 모듈의 관리 체계에 따라 사용자에게 보다 질 좋은 서비스가 가능하리라 예상된다.

참 고 문 헌

[1] 선우종성, 이병만, 김남욱, 홍성명, 송 의, "NCApatrol 1.5 개발보고서", 한국전산원, pp. 5-10, 1998.

[2] Paul Resnick, Jim Miller, "PICS: Internet Access Controls Without Censorship", Communications of the ACM, 1996, vol. 39(10), pp. 87-93..

[3] "Driver Developent Toolkit", Microsoft coporation, http://msdn.microsoft.com, June 2000.

[4] Postel,J., "Internet Protocol", RFC 791, USC/Information Sciences Instutute, September 1981.

[5] Reynolds, J., Postel, J., "Assigned Numbers", RFC 1340, USC/Information Sciences Instutute, July 1992.

[6] Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Instutute, August 1980.

[7] Postel, J., "Transmission Control Protocol", RFC 761, USC/Information Sciences Instutute, January 1980.

[8] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

[9] Richard Stevens, W., "TCP/IP Illustrated: the protocol", Addison-Wesely, 1994.

[10] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.