(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0006447
(43) 공개일자 2011년01월20일

(51) Int. Cl.
G06F 17/28 (2006.01) *H04L 12/16* (2006.01)
(21) 출원번호 10-2009-0064082
(22) 출원일자 2009년07월14일
심사청구일자 2009년07월14일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 궁동 220번지 충남대학교
(72) 발명자
이영석
대전 유성구 관평동 대덕테크노밸리 8단지 금성백조아파트 809-602
김기수
대전 동구 가양2동 149-9 삼덕빌라 가동 101호
(뒷면에 계속)
(74) 대리인
김원준

전체 청구항 수 : 총 9 항

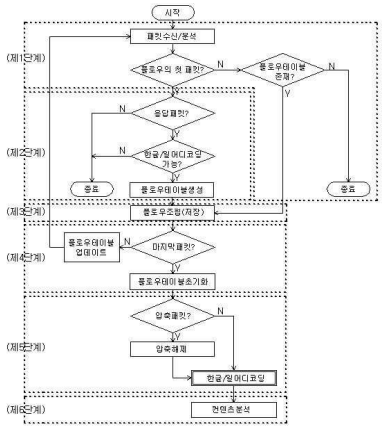
(54) HTTP 패킷에서 한글 또는 일본어 디코딩 방법과 장치 및 이를 이용한 한글 또는 일본어 웹 컨텐츠 분석방법

(57) 요 약

본 발명은 트래픽 모니터링 장비에서 HTTP 트래픽 분석을 수행할 때 수신된 HTTP 패킷 페이로드 내에 한글 또는 일본어가 인코딩되어 있는지를 탐지하고 이를 해독하는 방법/장치 및 이 방법에 의해 해독된 웹 컨텐츠의 내용을 분석하는 방법에 관한 것으로, 트래픽 모니터링 장비에서 HTTP 트래픽 분석을 수행할 때 HTTP 패킷 페이로드 내에 인코딩되어 있는 한글 또는 일본어 문자열을 탐지하여 해독하는 방법/장치와, 이를 이용하여 찾아낸 HTML 혹은 XML과 같은 웹 문서의 내용을 분석하여 사용자가 어떤 종류의 컨텐츠에 관심이 있는지를 분석하는 방법에 관한 것이다.

본 발명에 의해, HTTP 패킷에 인코딩되어 있는 한글 또는 일본어 문자열을 탐지하여 해독할 때 한글 완성형 코드 값과 비교하여 그 값에 해당하는 한글 또는 일본어 문자를 출력하는 것이 가능하게 된다. 또한 한글 또는 일본어 문자열이 여러 개의 패킷에 걸쳐 있을 경우 동일한 출발지와 목적지 IP 주소/포트번호를 갖는 연속적인 HTTP 패킷에 대해서도 인코딩 정보가 없더라도 한글 또는 일본어 문자열을 탐지하고 해독할 수 있도록 한다.

대 표 도 - 도1



(72) 발명자

권택근

대전 유성구 신성동 대림두레아파트 106-902

손현구

서울 동대문구 장안1동 391-21 성수연립 A동 305호

강원철

부산 동래구 온천2동 럭키아파트 5-1001

이 발명을 지원한 국가연구개발사업

과제고유번호 ITAC1090090200160001000200100 / ITAA1100090200630

부처명 정보통신진흥연구원 / 지식경제부

연구관리전문기관

연구사업명 대학IT연구센터 육성지원사업 / IT원천기술개발사업

연구과제명 해킹·바이러스 대응기술연구 / 초정밀 측정 및 분석기술연구

기여율

주관기관 충남대학교 산학협력단

연구기간 2003.08.01~2011.12.31 / 2008.03.01~2013.02.28

특허청구의 범위

청구항 1

플로우정보가 저장되는 플로우테이블이 개재된 한글 또는 일본어 디코딩 방법에 있어서,

(A) 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 제2단계(적합성분석단계)로 이동하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고, 존재하면 하기 제3단계(페이로드조립단계)로 이동하는 제1단계(플로우분석단계);

(B) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 제3단계(페이로드조립단계)로 이동하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 제2단계(적합성분석단계);

(C) 상기 단계(B)에서 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 것으로 분석된 패킷의 페이로드를 저장하고, 상기 단계(A)에서 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하는 것으로 확인된 패킷의 페이로드를 기존에 저장되어 있는 해당 플로우의 페이로드에 연결하여 플로우를 조립하는 제3단계(페이로드조립단계);

(D) 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하고 상기 제1단계로 이동하는 제4단계(플로우테이블관리단계);

(E) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 ① 상기 패킷이 압축된 것이라면 플로우의 마지막 패킷까지 조립된 플로우의 페이로드의 압축을 해제한 후, ② 압축된 것이 아니라면 그대로, 상기 플로우의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 제5단계(압축해제/디코딩단계);를 포함하여 이루어지는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 디코딩 방법.

청구항 2

제 1 항에 있어서,

상기 플로우테이블은 플로우를 식별할 수 있는 정보 필드, 삭제예약플래그 필드, 플로우유지시간 필드, 한글 또는 일본어 인코딩 타입 필드, HTTP 패킷의 페이로드 길이 필드 및 압축유무플래그 정보 필드를 포함하는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 디코딩 방법.

청구항 3

제 1 항에 있어서,

상기 제2단계(적합성분석단계)에서,

HTTP 헤더에 시그니쳐1("Content-Type: XXX" 및 {"charset=YYY" 또는 "encoding=YYY"}, 여기서 XXX가 "text/html" 또는 "text/xml")이 존재하고, YYY가 ① 한글 인코딩 타입인 UTF-8, utf-8, EUC-KR, euc-kr, KS_C_5601 또는 ks_c_5601이거나 ② 일본어 인코딩 타입인 UTF-8, utf-8, EUC-JP, euc-jp, SHIFT-JIS, Shift-JIS 또는 shift-jis인 경우에 플로우테이블을 생성하고 제3단계(페이로드조립단계)로 이동하는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 디코딩 방법.

청구항 4

제 1 항 내지 제 3 항 중 어느 한 항에 의한 방법을 수행하기 위한 장치로서,

(A) 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 적합성분석부로 전달하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고,

존재하면 하기 페이로드조립저장부로 전달하는 헤더분석부;

(B) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 페이로드조립저장부로 전달하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 적합성분석부;

(C) 상기 적합성분석부에서 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 것으로 분석된 패킷의 페이로드를 저장하고, 상기 헤더분석부에서 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하는 것으로 확인된 패킷의 페이로드를 기존에 저장되어 있는 해당 플로우의 페이로드에 연결하여 플로우를 조립하는 페이로드조립저장부;

(D) 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하는 플로우테이블관리부;

(E) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 압축된 것이라면 플로우의 마지막 패킷까지 조립된 플로우의 페이로드의 압축을 해제한 후 상기 플로우의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 압축해제/디코딩부;를 포함하여 이루어지는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 디코딩 장치.

청구항 5

제 1 항 내지 제 3 항 중 어느 한 항에 의한 방법에 따라 디코딩된 한글 또는 일본어 웹 컨텐츠를 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 것을 특징으로 하는 한글 또는 일본어 웹 컨텐츠의 분석방법.

청구항 6

플로우정보가 저장되는 플로우테이블이 개재된 한글 또는 일본어 웹 컨텐츠의 분석방법에 있어서,

(A) 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 제2단계(적합성분석단계)로 이동하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고, 존재하면 하기 제3단계(압축해제/디코딩단계)로 이동하는 제1단계(플로우분석단계);

(B) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 제3단계(압축해제/디코딩단계)로 이동하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 제2단계(적합성분석단계);

(C) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 ① 상기 패킷이 압축된 것이라면 패킷 페이로드의 압축을 해제한 후, ② 압축된 것이 아니라면 그대로, 상기 패킷의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 제3단계(압축해제/디코딩단계);

(D) 디코딩된 한글 또는 일본어 웹 컨텐츠를 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 제4-1단계(패턴매칭단계);

(E) 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하고 상기 제1단계로 이동하는 제4-2단계(플로우테이블관리단계);를 포함하여 이루어지는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 웹 컨텐츠의 분석방법.

청구항 7

제 6 항에 있어서,

상기 플로우테이블은 플로우를 식별할 수 있는 정보 필드, 삭제예약플래그 필드, 플로우유지시간 필드, 1바이트 저장공간 필드, 한글 또는 일본어 인코딩 타입 필드, HTTP 패킷의 페이로드 길이 필드및 압축유무플래그 정보

필드를 포함하는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 웹 컨텐츠의 분석방법.

청구항 8

제 6 항에 있어서,

상기 제2단계(적합성분석단계)에서,

HTTP 헤더에 시그니쳐1("Content-Type: XXX" 및 {"charset=YYY" 또는 "encoding=YYY"}, 여기서 XXX가 "text/html" 또는 "text/xml")이 존재하고, YYY가 ① 한글 인코딩 타입인 UTF-8, utf-8, EUC-KR, euc-kr, KS_C_5601 또는 ks_c_5601이거나 ② 일본어 인코딩 타입인 UTF-8, utf-8, EUC-JP, euc-jp, SHIFT-JIS, shift-JIS 또는 shift-jis인 경우에 플로우테이블을 생성하고 제3단계(압축해제/디코딩단계)로 이동하는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 웹 컨텐츠의 분석방법.

청구항 9

제 6 항에 있어서,

상기 제3단계(압축해제/디코딩단계) 및 제4-2단계(플로우테이블관리단계)에서,

(a) 상기 패킷(n)이 플로우의 마지막 패킷이 아닌 경우, 한글 또는 일본어 디코딩을 수행하고 마지막 1바이트가 잔류할 때 이를 플로우테이블의 1바이트저장공간에 임시로 저장하고,

(b) 다음 패킷(n+1) 처리시 상기 임시 저장된 1바이트 정보를 다음 패킷(n+1)의 페이로드 전단에 첨부한 후 디코딩하는 것을 특징으로 하는 HTTP 패킷에서 한글 또는 일본어 웹 컨텐츠의 분석방법.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 트래픽 모니터링 장비에서 HTTP 트래픽 분석을 수행할 때 수신된 HTTP 패킷 페이로드 내에 한글 또는 일본어가 인코딩되어 있는지를 탐지하고 이를 해독하는 방법/장치 및 이 방법에 의해 해독된 웹 컨텐츠의 내용을 분석하는 방법에 관한 것이다. 즉, 본 발명은, 트래픽 모니터링 장비에서 HTTP 트래픽 분석을 수행할 때 HTTP 패킷 페이로드 내에 인코딩되어 있는 한글 또는 일본어 문자열을 탐지하여 해독하는 방법/장치와, 이를 이용하여 찾아낸 HTML 혹은 XML과 같은 웹 문서의 내용을 분석하여 사용자가 어떤 종류의 컨텐츠에 관심이 있는지를 분석하는 방법에 관한 것이다.

배 경 기 술

[0002] 인터넷으로 대표되는 네트워크의 활성화에 따라 네트워크의 속성과 특징을 정확히 이해하고, 각종 네트워크상에서 발생하는 문제(트래픽 문제, 보안문제 등)의 원인을 명확히 밝히고 해결하기 위하여 트래픽분석 또는 패킷분석이 이루어지고 있다.

[0003] port번호를 이용한 트래픽분석 방법은 간단하기 때문에 많이 활용되고 있지만 분석의 정확성이 낮다는 문제가 있다. 보다 정확성이 있는 분석방법으로, 패킷 페이로드에서 어플리케이션의 특정 시그니쳐의 존재 유무를 판단하여 분석하는 방법이 있다. 그러나 특정 시그니쳐를 찾는 것이 쉽지 않으며, 시그니쳐가 바뀔 때마다 업데이트를 해줘야 한다는 단점이 있다. 그 밖에도 아이피 주소 혹은 tcp 포트번호와 같은 필드와 패킷 사이즈 같은 특징들을 SVM(Support Vector Machine)같은 기계학습(Machine Learning)에 적용시켜 트래픽을 분석하는 방법들도 있다. 그렇지만 이와 같은 방법론들은 인터넷 어플리케이션을 분류하는 것이 주된 목적이다.

[0004] 한편, 이러한 기존의 연구 및 분석관점에서 벗어나 네트워크에 흐르는 HTTP 프로토콜을 이용하는 웹 어플리케이

션 패킷들을 대상으로 한글 또는 일본어 HTML 혹은 XML 문서들을 추출하여 컨텐츠별로 분류함으로써 사용자들이 인터넷에서 어떤 종류의 컨텐츠에 관심이 많으며 어떠한 행동패턴을 보이는지 등을 파악할 필요성이 증대되고 있다.

[0005] 현재 인터넷 상에서는 패킷분석을 쉽고도 정확하게 해주는 다양한 툴(Ethereal, Wireshark, Sniffer 등)들이 존재한다. 그러나 이러한 툴들은 패킷의 페이로드의 내용을 ASCII코드 문자열로 보여주는 것에 그치고 있어 한글로 된 문자열이 HTTP 패킷에 포함되어 있을 경우 그 내용을 알 수 없다. 한국이나 일본의 경우 인터넷상의 정보는 대부분 한글이나 일본어로 인코딩된 웹 컨텐츠이기 때문에, 인터넷 트래픽의 내용을 정밀하게 분석하고, 사용자들의 행위를 파악하기 위한 컨텐츠 분석을 위해서는 ASCII 코드가 아닌 한글 또는 일본어 문자열로 자동으로 인식하고 해독하는 것의 필요성이 증대되고 있다.

[0006] 현재 HTTP에서 쓰이는 **한글 인코딩 방법**은 한글완성형코드(KS완성형 표준한글코드, KS C 5601)를 이용한 인코딩, EUC-KR, UTF-8, UTF-16으로 구분할 수 있다. 하지만 현재 거의 모든 웹서버가 한글완성형코드(KS C 5601), 특히 EUC-KR과 UTF-8을 이용하여 웹페이지를 전송하고 있는 상황이다.

[0007] 한글완성형코드는 2바이트 완성형 코드이며 2350자의 한글을 지원한다(KS C 5601). EUC-KR은 Bell Lab.에서 유닉스상에서 영문자 이외의 문자를 지원하기 위해 제안한 확장 유닉스 코드(Extened UNIX code)중 한글 인코딩 방식으로서 영문은 KS C 5636(새 이름은 KS × 1003)으로 처리하고 한글은 KS C 5601(새 이름은 KS × 1001)로 처리한다. 여기서 KSC5636은 영문자에 대한 표준으로서 한국공업표준 정보처리분야(C)의 5636번 표준안을 말하며 기존 ASCII Code에서 역슬래쉬(\)를 원(₩) 표시로 대체한 코드이다. 즉, EUC-KR은 KS C 5601과 KS C 5636을 합친 코드를 사용하는 8비트 문자 인코딩이라고 생각하면 된다.

[0008] UTF-8과 UTF-16은 유니코드를 위한 가변길이 문자 인코딩 방식의 하나이다(ISO/IEC 10646). UTF-8인코딩은 유니코드 한 문자를 나타내기 위해 1바이트에서 4바이트까지를 사용한다. 예를 들면 U+0000부터 U+007F 범위에 있는 ASCII 문자들은 UTF-8에서 1바이트만으로 표시된다. 마찬가지로 U+0080부터 U+07FF까지는 2바이트이며, U+0800부터 U+FFFF사이에 들어가는 한글은 3바이트로 인코딩된다. UTF-16은 기본 다국어 평면(BMP, Basic Multilingual Plane)에 속하는 문자들은 그대로 16비트 값으로 인코딩하고 그 이상의 문자는 특별히 정해진 방식으로 32비트로 인코딩한다.

[0009] 현재 HTTP에서 쓰이는 **일본어 인코딩 방법**은 SHIFT-JIS, EUC-JP, UTF-8, UTF-16으로 구분할 수 있다.

[0010] SHIFT-JIS(JIS × 0208:1997 Appendix 1)는 JIS × 0201과 JIS × 0208등을 사용하는 일본어 문자 인코딩이며, SJIS로 줄여 부르기도 한다. 1982년에 개발되었으나, 일본 내에서 널리 쓰이게 되자 JIS×0208:1997의 부속서 1로 표준화 되었다. 바이트 코드로 이루어진 SHIFT-JIS는 여러 확장들이 만들어 졌으나, 이중 JIS×0208의 확장으로 만들어진 마이크로소프트의 코드 페이지932가 가장 많이 쓰인다. 이중 히라가나(Hiragana)는 0x829F ~ 0x82F1, 가타카나(Katakana)는 0x8340 ~ 0x8396이며, 마지막으로 한자(kanji)는 몇 개를 제외하고 0x889F ~ 0xEEEC, 0xFA5C ~ 0xFC4B의 범위에 속한다.

[0011] EUC-JP는 Bell Lab.에서 유닉스 상에서 영문자 이외의 문자를 지원하기 위해 제안한 확장 유닉스 코드(Extened UNIX code)중 일본어 인코딩 방식으로서 EUC의 인코딩 방식위에 ASCII와 JIS×0208문자 집합을 배치한 것으로, 반각 가나(JIS×0201)와 JIS보조 한자(JIS×0212)도 포함할 수 있다. 이중 히라가나(Hiragana)는 0xA4A1 ~ 0xA4F3, 가타카나(Katakana)는 0xA5A1 ~ 0xA5F6이며, 마지막으로 한자(kanji)는 2바이트의 0xB0A1 ~ 0xFCED와 0x8FA2A0 ~ 0x8FFEFE 범위의 3바이트의 보조한자로 구성되어 있다.

[0012] UTF-8과 UTF-16은 유니코드를 위한 가변길이 문자 인코딩 방식의 하나이다(ISO/IEC 10646). UTF-8인코딩은 유니코드 한 문자를 나타내기 위해 1바이트에서 4바이트까지를 사용한다. 예를 들면 U+0000부터 U+007F 범위에 있는 ASCII 문자들은 UTF-8에서 1바이트만으로 표시된다. 마찬가지로 U+0080부터 U+07FF까지는 2바이트이며, U+0800부터 U+FFFF사이에 들어가는 일본어는 3바이트로 인코딩된다. UTF-16은 기본 다국어 평면(BMP, Basic Multilingual Plane)에 속하는 문자들은 그대로 16비트 값으로 인코딩하고 그 이상의 문자는 특별히 정해진 방식으로 32비트로 인코딩한다.

발명의 내용

해결 하고자하는 과제

[0013] 본 발명은 HTTP 패킷에서 한글 또는 일본어가 인코딩되어 있는지를 확인하고, 인코딩되어 있는 경우 이를 한글 또는 일본어 문자열로 디코딩하여 표현(출력)할 수 있는 방법 및 장치를 제공하는 것을 목적으로 한다.

[0014] 또한 본 발명은 상기 한글 또는 일본어 디코딩 방법에 따라 디코딩된 HTML/XML 문서 단위로 해독된 한글 또는 일본어 컨텐츠를 카테고리 별로 저장된 한글 또는 일본어 키워드들과의 매칭을 통해 해당 문서를 특정 컨텐츠 카테고리로 분류하여, 웹 컨텐츠 분석뿐만 아니라, 해당 망에서의 사용자의 행위까지도 파악할 수 있도록 하는 것을 목적으로 한다.

과제 해결수단

[0015] 이하에서 설명의 편의를 위해 몇가지 용어를 정의한다.

[0016] "플로우(flow)"란 공통의 주소쌍(송신자 주소, 송신자 포트 번호, 수신자 주소, 수신자 포트), 호스트 쌍(송신자 호스트 주소, 수신자 호스트 주소), 네트워크 주소 쌍(송신자 네트워크 주소, 수신자 네트워크 주소), AS 번호 쌍(송신자 AS 번호, 수신자 AS 번호) 등으로 주어지는 조건을 만족시키는 제한된 시간 내에 도착하는 IP 패킷들의 흐름으로 정의된다. 따라서 수신된 패킷의 헤더를 분석하면 그 패킷이 특정 플로우에 속하는지, 속한다면 플로우의 몇 번째 패킷인지 또는 마지막 패킷인지 등을 확인할 수 있다. IP 패킷 헤더분석은 종래 널리 알려진 사항이므로 상세한 설명을 생략한다.

[0017] "페이로드(payload=message body)"는 패킷의 헤더에 대응되는 개념으로, 사용자 정보(컨텐츠)를 보유하는 패킷의 부분을 의미한다. 페이로드는 압축되어 있을 수도 있다.

[0018] 전술한 과제를 해결하기 위한 본 발명은, 플로우정보 등이 저장되는 플로우테이블이 개재된 한글 또는 일본어 디코딩 방법/장치, 이에 의해 디코딩된 한글 또는 일본어 웹 컨텐츠의 분석방법에 관한 것이다.

[0019] **(1)** 플로우에 해당하는 패킷의 페이로드를 조립한 후 디코딩하는 방법

[0020] 본 발명에 의한 한글 또는 일본어 디코딩 방법은, 플로우정보 등이 저장되는 플로우테이블이 개재된 한글 또는 일본어 디코딩 방법에 있어서, **(A)** 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 제2단계(적합성분석단계)로 이동하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고, 존재하면 하기 제3단계(페이로드조립단계)로 이동하는 제1단계(*플로우분석단계*); **(B)** 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 제3단계(페이로드조립단계)로 이동하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 제2단계(*적합성분석단계*); **(C)** 상기 단계(B)에서 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 것으로 분석된 패킷의 페이로드부분을 저장하고, 상기 단계(A)에서 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하는 것으로 확인된 패킷의 페이로드를 기존에 저장되어 있는 해당 플로우의 페이로드에 연결하여 플로우를 조립하는 제3단계(*페이로드조립단계*); **(D)** 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하고 상기 제1단계로 이동하는 제4단계(*플로우테이블관리단계*); **(E)** 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 ① 상기 패킷이 압축된 것이라면 플로우의 마지막 패킷까지 조립된 플로우의 페이로드의 압축을 해제한 후, ② 압축된 것이 아니라면 그대로, 상기 플로우의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 제5단계(*압축해제/디코딩단계*);를 포함하여 이루어지는 HTTP 패킷에서 한글 또는 일본어 디코딩 방법이다.

[0021] 별도의 상세한 부가설명이 없더라도, 본 발명의 특성상 통신상의 하자 등에 의해 더 이상 수신 또는 분석할 패킷이 없는 경우 상기 디코딩 방법의 수행이 중단됨은 당연하다(이하 동일함).

[0022] 본 발명에서 상기 플로우테이블은 플로우를 식별할 수 있는 정보 필드와, 삭제예약플래그 필드, 플로우유지시간 필드, 한글 또는 일본어 인코딩 타입 필드, HTTP 패킷의 페이로드 길이 필드 및 압축유무플래그 정보 필드를 포함하는 것이 바람직하다. 만일 플로우유지시간 필드가 있는 경우, 후속 패킷이 소정의 플로우유지시간(예컨대 30초) 내에 수신되지 않으면 디코딩을 종료하도록 할 수 있을 것이다. 삭제예약 플래그 필드가 있는 경우, 초기 플로우분석단계에서 패킷이 마지막 패킷인 경우(즉 패킷의 TCP 헤더에 FIN 플래그가 설정된 경우) 이를 삭제예약 플래그에 기록해 두고, 이 기록의 유무를 확인함으로써 패킷이 마지막 패킷인지를 확인하도록 할 수도 있다. 즉, 플로우테이블에 설정된 삭제예약 플래그가 있는지, 또는 소정의 플로우유지시간을 넘긴 플로우가 있는지를 체크하여 있다면 플로우테이블을 삭제(초기화)한다.

[0023] 본 발명의 상기 제2단계(적합성분석단계)에서, HTTP 헤더에 시그니쳐1("Content-Type: XXX" 및 {"charset=YYY" 또는 "encoding=YYY"}, 여기서 XXX가 "text/html" 또는 "text/xml")이 존재하고, YYY가 ① 한글 인코딩 타입인 UTF-8, utf-8, EUC-KR, euc-kr, KS_C_5601 또는 ks_c_5601이거나 ② 일본어 인코딩 타입인 UTF-8, utf-8, EUC-JP, euc-jp, SHIFT-JIS 또는 shift-jis인 경우에 플로우테이블을 생성하고 제3단계(페이로드조립단계)로 이동하는 것이 바람직하다.

[0024] (2) 또한 본 발명에 의한 한글 및 일본어 디코딩 장치는, 전술한 방법을 수행하기 위한 장치로서, (A) 수신된 패킷의 IP헤더를 분석하여 첫 패킷인지를 확인하는 헤더분석부; (B) 상기 HTTP 패킷헤더와 페이로드의 일부정보를 참조하여 HTTP 응답패킷인지, 한글 또는 일본어 디코딩이 가능한 패킷인지를 판단하는 적합성분석부; (C) 적합성이 확인된 패킷의 페이로드를 조립하는 페이로드조립저장부; (D) 플로우정보를 생성, 관리하는 플로우테이블관리부; 및 (E) 조립이 완료된 플로우의 페이로드를 파싱을 통하여 디코딩하는 압축해제/디코딩부;를 포함하는 HTTP 패킷에서 한글 또는 일본어 디코딩 장치이다.

[0025] (3) 본 발명에 의한 한글 또는 일본어 웹 컨텐츠의 분석방법 1은, 전술한 방법에 따라 디코딩된 한글 또는 일본어 웹 컨텐츠를 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 것을 특징으로 하는 한글 또는 일본어 웹 컨텐츠의 분석방법이다.

[0026] (4) 패킷마다 페이로드를 순차적으로 디코딩한 후 컨텐츠 분석하는 방법(페이로드조립단계가 없는 방법)

[0027] 본 발명에 의한 한글 또는 일본어 웹 컨텐츠의 분석방법 2는, 플로우정보 등이 저장되는 플로우테이블이 개재된 한글 또는 일본어 웹 컨텐츠의 분석방법에 있어서, (A) 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 제2단계(적합성분석단계)로 이동하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고, 존재하면 하기 제3단계(압축해제/디코딩단계)로 이동하는 제1단계(*플로우분석단계*); (B) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 제3단계(압축해제/디코딩단계)로 이동하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 제2단계(*적합성분석단계*); (C) 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 ① 상기 패킷이 압축된 것이라면 패킷 페이로드의 압축을 해제한 후, ② 압축된 것이 아니라면 그대로, 상기 패킷의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 제3단계(*압축해제/디코딩단계*); (D) 디코딩된 한글 또는 일본어 웹 컨텐츠를 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 제4-1단계(*패턴매칭단계*); (E) 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하고 상기 제1단계로 이동하는 제4-2단계(*플로우테이블관리단계*);를 포함하는 HTTP 패킷에서 한글 또는 일본어 웹 컨텐츠의 분석방법이다.

[0028] 이 경우 상기 플로우테이블에는, 전술한 (1)에서의 필드에 1바이트저장공간 필드가 추가되는 것이 바람직하다. 1바이트저장공간은 한글과 일본어의 2바이트 인코딩방식을 반영한 것으로서 다음과 같이 활용된다.

[0029] 즉, 상기 제3단계(압축해제/디코딩단계) 및 제4-2단계 (플로우테이블관리단계)에서, (a) 상기 패킷(n)이 플로우의 마지막 패킷이 아닌 경우, 한글 또는 일본어 디코딩을 수행하고 마지막 1바이트가 잔류할 때 이를 플로우테이블의 1바이트저장공간에 임시로 저장하고, (b) 다음 패킷(n+1) 처리시 상기 임시 저장된 1바이트 정보를 다음 패킷(n+1)의 페이로드 전단에 첨부한 후 디코딩하게 되는 것이다.

[0030] 상기 제2단계(적합성분석단계)는 전술한 (1)과 동일할 수 있다.

효 과

[0031] 본 발명에 의해, HTTP 패킷에 인코딩되어 있는 한글 또는 일본어 문자열을 탐지하여 해독할 때 한글 완성형 코드 값과 비교하여 그 값에 해당하는 한글 또는 일본어 문자를 출력하는 것이 가능하게 된다.

[0032] 또한 본 발명에 의해, 한글 또는 일본어 문자열이 여러 개의 패킷에 걸쳐 있을 경우 동일한 출발지와 목적지 IP 주소/포트번호를 갖는 연속적인 HTTP 패킷에 대해서도 인코딩 정보가 없더라도 한글 또는 일본어 문자열을 탐지하고 해독할 수 있도록 한다.

[0033] 또한 본 발명에 의해, 탐지된 한글 또는 일본어 문자열 정보를 이용할 수 있게 되어 HTTP 트래픽의 상세한 분석을 가능하게 한다.

[0034] 본 발명에 의하면, 탐지된 한글 또는 일본어 문자열 정보를 이용하여 트래픽 모니터링 시스템이 설치된 네트워크 내의 사용자들이 어떠한 웹 컨텐츠를 즐기는지 파악할 수 있다. 이러한 정보를 바탕으로 웹 컨텐츠 제작자는 특정 웹 포탈에서의 인기 컨텐츠 결과가 아닌 다양한 웹 포털을 이용하는 종합적인 결과를 얻을 수 있을 것이며, 네트워크 관리자나 새로 네트워크를 설계하는 엔지니어는 네트워크 우선순위 경로의 설정 등의 네트워크 관리를 최적화 하는 데 활용할 수 있을 것이다.

발명의 실시를 위한 구체적인 내용

[0035] 이하 첨부된 도면을 참조하여 본 발명을 보다 상세히 설명한다. 그러나 이러한 도면은 본 발명의 기술적 사상의 내용과 범위를 쉽게 설명하기 위한 예시일 뿐, 이에 의해 본 발명의 기술적 범위가 한정되거나 변경되는 것은 아니다. 또한 이러한 예시에 기초하여 본 발명의 기술적 사상의 범위 안에서 다양한 변형과 변경이 가능함은 당업자에게는 당연할 것이다.

[0036] 도 1은 본 발명에 따른 한글 또는 일본어 디코딩 방법 및 그 결과 얻어진 웹 컨텐츠를 분석하는 방법의 일예를 보여주는 전체 흐름도이고, 도 2 본 발명에 따른 한글 또는 일본어 디코딩 방법의 다른 예를 보여주는 흐름도이다. 도 1 및 도 2의 예는, 플로우에 해당하는 모든 패킷의 페이로드를 조립한 후 디코딩하는 방식이다.

[0037] 먼저 도 1에 관해 설명한다. 도 1에 의한 방법은 크게 [플로우분석단계→적합성분석단계→페이로드조립단계→플로우테이블관리단계→압축해제/디코딩단계]를 포함하여 구성된다. 이때 발명의 전 취지 상, 플로우테이블관리단계는 흐름상 적합성분석단계 이후라면 어느 때 수행되더라도 동일한 결과를 나타냄은 당업자에게 있어 당연한 것이다. 따라서 본 발명 및 그 설명/도면에서 플로우테이블관리단계를 비록 제4단계로 표현하였으나, 이는 순서를 나타내는 것이 아니다(이하 동일하다).

[0038] (A) 제1단계(플로우분석단계)에서는 수신한 TCP/IP 패킷이 여러 개의 HTTP 패킷들 중에서 이미 받은 HTTP 응답패킷들과 관련된 나머지 패킷인지(즉, 플로우에 해당하는지)를 판단한다. [물론 플로우의 첫 패킷이면서 적합성(한글/일본어디코딩가능성)이 확인된 경우에는 플로우테이블이 생성된다.]

[0039] 본 단계에서는 수신된 패킷이 플로우의 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 HTTP 응답패킷이 아닌 플로우거나 한글/일본어디코딩이 불가능한 플로우임이 첫 패킷에 의해 이미 확인된 것이므로 추가조치 없이 종료한다. 물론 헤더 정보가 없는 패킷이면서 플로우테이블에도 없는 패킷이라면 어떠한

처리 없이 종료한 후 바로 다음 패킷이 있는지를 조사하는 식의 순서를 따른다.

[0040] 첫 패킷이 아니면서 해당 플로우테이블이 존재하면, HTTP 응답패킷이면서 한글/일본어 디코딩이 가능한 플로우임이 첫 패킷에 의해 이미 확인된 것이므로, 상기 플로우에 해당하는 연속된 패킷으로 인식하고 제2단계(적합성분석단계)를 거치지 않고 바로 제3단계(페이로드조립단계)로 이동한다. 여기서 "패킷의 해당 플로우테이블"이란 패킷과 같은 출발/목적 IP주소, 출발/목적 포트를 갖는 선행 패킷의 정보가 저장된 플로우테이블을 의미한다.

[0041] 수신된 패킷이 플로우의 첫 패킷이면 제2단계(적합성분석단계)로 이동한다.

[0042] (B) 제2단계(적합성분석단계)에서는 플로우의 첫 패킷인 경우, 수신 패킷이 응답패킷인지와, 한글 또는 일본어 디코딩이 가능한지를 확인하여 그에 대한 플로우테이블을 생성하는 단계이다.

[0043] 즉, 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 2바이트 한글 또는 일본어 디코딩이 가능한 패킷이면 플로우테이블을 생성한 후 제3단계(페이로드조립단계)로 이동한다. 한편 HTTP 응답패킷이 아니거나 2바이트 한글 또는 일본어 디코딩이 불가능한 패킷이면 추가적 조치없이 종료한다.

[0044] 상기 플로우테이블에는 플로우를 식별할 수 있는 정보(출발 IP 주소, 목적 IP 주소, 출발 port, 목적 port, 수신한 시간, 식별자, 플래그, 단편화옵셋 등의 전부 또는 일부) 및 삭제예약플래그 및 플로우유지시간 등의 필드로 구성되도록 할 수 있다.

[0045] 이렇게 생성된 플로우테이블은 제1단계(플로우분석단계)에서 플로우의 첫 번째가 아닌 패킷의 플로우테이블인지를 확인하는 용도로 활용되며, 하기 제4단계(플로우테이블관리단계)에서 관리된다.

[0046] 한편, 본 발명은, 상기 플로우테이블을 제2단계 전에 생성하였다가 적합성이 없는 것으로 판단되면 초기화하는 것을 배제하지 않는다.

[0047] 제2단계에 대해서는 뒤에서 다시 상세히 설명한다.

[0048] (C) 제3단계(페이로드조립단계)에서는 한글 또는 일본어 디코딩 적합성이 확인된 플로우의 패킷 HTTP 페이로드의 스트링을 조립/저장하는 단계이다.

[0049] 즉, HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 것으로 분석된 패킷의 페이로드를 저장한다. 이때 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하는 것으로 확인된 패킷(n번째 패킷)의 페이로드(p_n)을 기존에 저장되어 있는 해당 플로우의 페이로드가 조립된 스트링($p_1+p_2+\cdots p_{n-1}$)에 연결하여 추가된 페이로드가 조립된 스트링($p_1+p_2+\cdots p_{n-1}+p_n$)을 생성하는 것이다.

[0050] (D) 제4단계(플로우테이블관리단계)는 한글 또는 일본어 문자열이 여러 개의 HTTP 패킷에 걸쳐있을 때(즉, 패킷이 플로우로 전달될 때), 첫 HTTP 패킷이 응답패킷이고 한글 또는 일본어 변환이 가능한 경우 뒤에 계속적으로 오게 될 동일 플로우의 패킷들을 곧바로 페이로드조립하기 위해 플로우테이블을 삽입, 유지, 삭제하는 단계이다. 즉, 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고 종료하며, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하고 상기 1단계(플로우분석단계)로 이동하는 단계이다.

[0051] 수신된 패킷이 마지막 패킷인 경우(즉 패킷의 TCP 헤더에 FIN 플래그가 설정된 경우) 이를 플로우테이블의 삭제예약 플래그에 기록해 두고, 이 기록의 유무를 확인함으로써 패킷이 마지막 패킷인 경우 플로우테이블을 삭제(초기화)한다. 또한 만일 플로우유지시간 필드가 있는 경우, 후속 패킷이 소정의 플로우유지시간(예컨데, 30초) 내에 수신되지 않으면 디코딩을 종료하고 플로우테이블을 삭제(초기화)할 수 있을 것이다.

[0052] 이상의 과정을 통하여 수신된 패킷들의 페이로드를 조립하는 것이 가능하게 되고, 플로우의 마지막 패킷의 페이로드까지 조립된 후에는 플로우테이블의 모든 정보가 삭제(초기화)되는 것이다.

[0053] (E) 제5단계(압축해제/디코딩단계)는 소정 플로우의 페이로드조립이 완료된 후에 이를 일괄해서 디코딩하는 단

계이다.

[0054] 즉, 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 압축된 것이라면 플로우의 마지막 패킷까지 조립된 플로우의 페이로드의 압축을 해제한 다음 한글 또는 일본어로 디코딩하고, 압축된 것이 아니라면 바디를 그대로 한글 또는 일본어를 디코딩한다. 구체적 디코딩 방법에 대하여는 뒤에서 다시 상세히 설명한다.

[0055] 본 발명의 제1항은 제5단계까지에 관한 것이고, 제5항은 제6단계(컨텐츠분석단계)가 추가된 것이다.

[0056] **(F)** 제6단계(컨텐츠분석단계)는 전술한 과정에 따라 디코딩된 한글 또는 일본어 웹 컨텐츠를 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 단계이다. 상세한 것은 뒤에 설명한다.

[0057] 도시하지는 않았지만, 전술한 한글 또는 일본어 디코딩 방법(제1단계~제5단계)은, **(A)** 수신된 패킷의 헤더를 분석하여 상기 패킷이 플로우의 첫 패킷인지를 확인하여 ① 첫 패킷인 경우 하기 적합성분석부로 전달하고, ② 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하지 않으면 종료하고, 존재하면 하기 페이로드조립저장부로 전달하는 *헤더분석부*; **(B)** 상기 패킷의 HTTP 헤더 또는 바디의 정보를 참조하여 상기 패킷이 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 패킷이면 상기 패킷에 대한 플로우테이블을 생성한 후 하기 페이로드조립저장부로 전달하고, HTTP 응답패킷이 아니거나 한글 또는 일본어 디코딩이 불가능한 패킷이면 종료하는 *적합성분석부*; **(C)** 상기 적합성분석부에서 HTTP 응답패킷이면서 동시에 한글 또는 일본어 디코딩이 가능한 것으로 분석된 패킷의 페이로드를 저장하고, 상기 헤더분석부에서 첫 패킷이 아니면서 상기 패킷에 대한 플로우테이블이 존재하는 것으로 확인된 패킷의 페이로드를 기존에 저장되어 있는 해당 플로우의 조립된 페이로드에 연결하여 플로우를 조립하는 *페이로드조립저장부*; **(D)** 상기 패킷이 플로우의 마지막 패킷인 경우 플로우테이블을 초기화하고, 마지막 패킷이 아닌 경우 플로우테이블을 업데이트하는 *플로우테이블관리부*; **(E)** 상기 패킷의 HTTP 헤더 또는 페이로드의 정보를 참조하여 상기 패킷이 압축된 것이라면 플로우의 마지막 패킷까지 조립된 플로우의 페이로드의 압축을 해제한 후 상기 플로우의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 *압축해제/디코딩부*;를 포함하는 HTTP 패킷의 한글 또는 일본어 디코딩 장치에 의해 수행될 수 있다.

[0058] 한편, 본 발명에서는 ① 전술한 바와 같이 일단 소정의 플로우에 해당하는 패킷의 모든 페이로드를 조립한 후에 압축여부를 확인하여 압축을 해제하는 방식(도 1)도 가능하지만 ② 소정의 플로우가 압축패킷인지를 먼저 확인하여 두고 패킷의 페이로드를 조립한 후에 압축을 해제하는 방식도 가능할 것이다. 후자의 방식에 관한 흐름도를 도 2에 도시하였다. 도 2에 예시된 방식은 도 1의 방식에 비해 다소 복잡한 구성을 보이지만, 궁극적으로 동일한 개념에 기반한 것이고, 동일한 결과를 얻을 수 있다. 도 1에 대한 설명부분을 참조하면 도 2를 용이하게 이해할 수 있으므로 추가적인 설명을 생략한다.

[0059] 도 3은 본 발명에 따른 한글 또는 일본어 디코딩 방법 및 그 결과 얻어진 웹 컨텐츠를 분석하는 방법의 또 다른 예를 보여주는 전체 흐름도이다. 이것은 패킷마다 페이로드를 순차적으로 디코딩하고 즉시 패턴매칭 등의 방법으로 컨텐츠를 분석하는 방식이다.

[0060] 도 3의 예에서 플로우분석단계, 적합성분석단계 및 플로우테이블관리단계는 도 1의 예에서와 동일하므로 그에 대한 설명을 생략하고 압축해제/디코딩단계(제3단계) 및 패턴매칭단계(제4-1단계)에 대해서만 설명한다.

[0061] **(A)** 제1단계(플로우분석단계)는 도 1에서와 동일하다.

[0062] **(B)** 제2단계(적합성분석단계)는 도 1에서와 동일하다.

[0063] **(C)** 제3단계(압축해제/디코딩단계)는 패킷이 압축된 것이라면 패킷 페이로드의 압축을 해제한 후, 압축된 것이 아니라면 그대로, 상기 패킷의 HTTP 페이로드의 스트링으로부터 한글 또는 일본어를 디코딩하는 단계이다. 즉, 플로우를 조립하지 않고 패킷별로 즉시 디코딩하는 것이다. 따라서 이 단계에서 디코딩된 컨텐츠는 플로우 전체가 보유한 컨텐츠의 일부(부분)가 된다.

[0064] **(D)** 제4-1단계(패턴매칭단계)는 패킷별 페이로드가 디코딩된 한글 또는 일본어 웹 컨텐츠의 일부에 대해 소정의 키워드 세트와 패턴매칭을 수행하여 상기 웹 컨텐츠의 내용을 분석하는 것이다.

[0065] (E) 제4-2단계(플로우테이블관리단계)는 도 1에서와 동일하다.

[0066] 이 예에 따르면, 특정 시점에서는 컨텐츠의 일부에 대한 내용이 분석되는 것이지만, 소정의 플로우에 해당하는 모든 패킷이 제3단계와 제4-1단계를 거치게 되면 결국은 플로우 전체가 가진 웹 컨텐츠에 대한 내용이 분석되게 된다. 이에 따라 결과는 도 1에 예시된 방식에 의한 결과와 동일하게 된다.

[0067] 한편, 2바이트로 인코딩된 한글 또는 일본어 문자열의 상위 1바이트와 하위 1바이트가 나뉘어 서로 다른 패킷의 페이로드에 실려 전달될 경우가 생긴다. 이러한 경우를 감안하여 플로우테이블의 필드에는 1바이트저장공간이 추가되는 것이 바람직하다. 이 경우, 전 패킷(n-1번째)에서 디코딩 후 마지막 1 바이트가 남게 되면 이를 플로우테이블의 1바이트저장공간에 저장해 둔다. 이어서 다음 패킷(n번째)을 디코딩할 때 플로우테이블의 1바이트저장공간에 상위 1바이트가 저장되어 있는지를 확인하고, 상위 1 바이트가 존재하면 현재 패킷(n번째)의 맨 처음에 추가한 후 디코딩을 수행한다.

[0068] 본 발명의 **제2단계(적합성분석단계)**에서는 플로우의 첫 패킷이면서 상기 패킷에 대한 플로우테이블이 존재하지 않는 수신 패킷이 응답패킷인지와, 한글 또는 일본어 디코딩이 가능한지를 확인하고, 둘 다 만족하는 경우 그 패킷에 대한 플로우테이블을 생성한다. 물론 헤더 정보도 없고 플로우테이블도 없는 패킷이라면 어떠한 조치 없이 흐름을 종료한다.

[0069] 먼저 플로우의 첫 패킷이 HTTP 응답패킷인지를 확인해야 한다.

[0070] 즉, HTTP 헤더에 "HTTP/1.1 200 OK"라는 스트링을 찾아 HTTP 응답 패킷 여부를 조사한다. 참고로 TCP 연결을 위한 3 단계 핸드쉐이킹(3-way handshaking) 방법에 따라 HTTP 요구 패킷, 그에 대한 서버측의 ACK가 온 뒤에 사용자가 요청한 HTTP 응답 패킷이 오게 된다.

[0071] 이어서 수신된 HTTP 응답 패킷에 대해서 HTTP의 헤더와 페이로드의 정보 중 일부를 추출하여 한글 또는 일본어 디코딩이 가능한지와, 가능하다면 어떤 인코딩 타입인지를 조사한다. 확인된 인코딩 타입은 플로우테이블의 해당 필드에 기록된다.

[0072] 예컨데, HTTP 헤더에 시그니쳐1("Content-Type: XXX" 및 {"charset=YYY" 또는 "encoding=YYY"}, 여기서 XXX가 "text/html" 또는 "text/xml")이 존재하고, YYY가 ① 한글 인코딩 타입인 UTF-8, utf-8, EUC-KR, euc-kr, KS_C_5601 또는 ks_c_5601이거나 ② 일본어 인코딩 타입인 UTF-8, utf-8, EUC-JP, euc-jp, SHIFT-JIS, Shift-JIS 또는 shift-jis인 경우에 각각 한글 또는 일본어 인코딩이 가능한 패킷으로 해석한다. 한글 또는 일본어 인코딩이 가능한 패킷으로 인정된 경우 상기 패킷의 정보를 저장하는 플로우테이블을 생성하고 제3단계(페이로드조립단계)로 이동한다. 그 이외의 경우 아무런 조치 없이 종료한다.

[0073] 한편, 디코딩되는 컨텐츠의 크기(길이)를 알아야 할 경우가 있다. 컨텐츠의 크기는 "Content-Length:"와 "Transfer-Encoding: chunked"를 통해서 구할 수 있다. 전자로부터는 컨텐츠의 길이를 바로 구할 수 있는 반면, 후자의 경우는 컨텐츠와 관련된 패킷(즉 플로우)을 다 수신한 뒤에 구할 수 있다. 예를 들면, Content-Length는 HTTP 헤더필드 중 컨텐츠의 길이를 직접적으로 나타내는 필드로써 "Content-Length: ZZZ"와 같이 표현되며 컨텐츠의 길이가 "ZZZ" 바이트임을 의미한다. Transfer-Encoding에 대해서는 압축해제와 관련한 부분에서 설명하도록 한다.

[0074] 정리하면, 본 발명에서 디코딩 적합성분석단계는, 수신된 패킷이 응답패킷인지, 한글 또는 일본어 디코딩이 가능한지 여부를 조사하여 이에 해당하는 패킷들만을 걸러내는 과정이다. 한글 또는 일본어를 어떤 인코딩방식을 써서 인코딩했는지를 알아내고, 소정의 조건을 만족하는 패킷에 대해서는 출발지 IP 주소, 목적지 IP 주소, 출발지 포트번호, 목적지 포트번호, 삭제 예약 플래그, 플로우 유지시간, 사용된 인코딩 타입, HTTP 패킷의 페이로드 길이, 압축 유무 플래그 정보 등이 저장된 플로우테이블을 생성한다.

[0075] 본 발명에서 **페이로드조립단계**는 전술한 적합성분석단계를 통과한 패킷의 플로우에 해당하는 모든 패킷의 페이로드를 패킷 순서에 따라 연결하여 저장하는 단계이다. 구체적인 내용은 위에서 설명한 바 있다.

[0076] 도 4는 본 발명의 **압축해제/디코딩단계**의 일예를 보여주는 세부 흐름도이다.

[0077] 도 1 또는 도 2의 경우는 해당 플로우의 모든 정보가 조립된 후에 압축해제/디코딩되고, 도 3의 경우는 해당 플로우의 패킷별로 순차적으로 압축해제/디코딩되는 점에서 차이가 있으나 실제 "압축해제/디코딩"은 동일하게 이루어지므로 묶어서 설명한다.

[0078] ① (압축해제과정) 압축해제/디코딩단계에서는, 먼저 현재 패킷이 압축된 패킷인지를 검사한다. 이는 시그니쳐3("Content-Encoding: YYY", YYY = "gzip" or "deflate")의 존재 유무로써 판단한다. "gzip" 이라는 문자열 대신에 "x-gzip", "deflate" 대신에 "x-deflate"이 라는 문자열이 왔을 때도 각각 gzip, deflate와 같은 압축 알고리즘으로 이해한다. 시그니쳐3이 존재할 경우 플로우테이블에 압축플래그를 설정하고, 이용된 압축 알고리즘을 기록한다.

[0079] 압축되어 있는 데이터를 해제하는데 있어서 가장 중요한 것은 데이터의 무결성과 그 사이즈이다. 또한 압축이 되어있지 않은 HTTP 플로우라 할지라도 온전한 HTML/XML 문서단위로 분석을 하기 위해서는 데이터의 사이즈를 알아내는 것은 필요하다.

[0080] HTTP에서 전달하려는 데이터의 사이즈는 위에서도 잠깐 언급했지만, "Content-Length"(도 4의 시그니쳐1) 필드와 "Transfer-Encoding"(도 4의 시그니쳐2) 필드를 이용하여 웹브라우저에게 알린다. 만약 HTTP 헤더에 시그니쳐 1이 설정되어 있다면, 바로 그 사이즈를 알 수 있기에 다음 단계로 진행할 수 있다(400). 하지만 시그니쳐 2가 설정되어 있다면, 전체적으로 데이터를 검사해 가면서 chunk 데이터의 사이즈와 실제 데이터, chunk 구분자를 구분하여 저장해야 한다(410). 즉, 시그니쳐 2가 설정되어 있는 HTTP 패킷의 경우에는 헤더와 페이로드를 구분하는 구분자인 "₩r₩n₩r₩n"이후에 "SIZE₩r₩n" + "데이터" + "₩r₩n"의 형태로 전달된다. 따라서 하나의 플로우 내에서 각각의 chunk데이터마다 16진수형태의 사이즈 값들을 추출하여 더한 값이 실제 전송하고자 하는 총 데이터의 사이즈가 된다(420). 그리고 보내는 데이터가 마지막이라는 것을 알리기 위해 해당 패킷의 마지막 부분에 시그니쳐3을 같이 실어 보내게 된다. 만약 시그니쳐 3이 없다면 해당 플로우가 패킷을 완전하게 받지 않은 것으로 판단하고 해당 플로우는 폐기한다(430). 위 과정을 거치게 되면 웹서버에서 보내고자 하는 데이터의 사이즈를 알아 낼 수 있게 된다.

[0081] 데이터의 사이즈를 알아낸 다음, 압축알고리즘이 적용된 데이터에 대해 압축 해제 과정을 거치게 된다. HTTP에서 가장 많이 사용되는 압축 알고리즘인 gzip은 10바이트의 gzip 헤더필드와 압축된 데이터, 4바이트의 CRC32 필드, 4바이트의 ISIZE 필드로 구성되어 있다. 10바이트의 헤더필드 중 플래그의 설정 여부에 따라 헤더와 압축 데이터 사이에 추가적으로 옵션필드가 있을 수 있지만, HTML/XML을 전송하는 HTTP에서는 위에 설명한 기본적인 부분만 고려하면 된다. 또한 gzip은 내부의 압축 데이터를 생성하기 위해서 deflate 알고리즘을 사용한다. 즉 거기에 gzip헤더와 기타 필드들이 더해져서 gzip 포맷의 압축 데이터가 생성되는 것이다. 대부분의 gzip과 deflate은 Zlib library (http://www.zlib.net, RFC 1950)를 이용하여 압축을 수행하고 해제한다. 본 발명에서도 Zlib library를 적용하여 압축을 해제할 수 있다.

[0082] HTTP 패킷에 적용된 압축 알고리즘이 deflate인 경우, 압축된 데이터와 그 데이터 사이즈가 정확하다면 Zlib library에서 지원하는 함수를 이용하여 압축을 해제할 수 있다(450). 다만, 몇몇 zlib 헤더를 포함하지 않고 데이터를 전송하는 웹서버들이 있어 2바이트의 Zlib 헤더를 압축 데이터 맨 앞에 삽입하여 압축해제를 하는 경우도 있다.

[0083] 이와 같은 과정으로 압축되어 있던 페이로드가 압축해제된다. 물론 수신된 패킷이 압축된 것이 아니라면 압축해제과정은 생략된다.

[0084] ② (디코딩과정) 압축해제되거나, 처음부터 압축되지 않은 페이로드를 한글 또는 일본어로 디코딩하게 된다. 도 1 또는 도 2의 경우 조립이 완료된 플로우 전체를 디코딩한 후에 패턴매칭하는 것도 가능하지만, 플로우를 디코딩 하는 것과 동시에 패턴매칭을 수행하여 컨텐츠분석을 하는 것도 가능할 것이다. 물론 도 3의 경우에는 당연히 패킷별로 순차적으로 디코딩된 부분을 즉시 패턴매칭하게 된다.

[0085] 물론, 패킷이 UTF-8과 같은 다국어를 지원하는 인코딩 타입이면 인코딩된 3바이트를 2바이트의 유니코드 값으로 변환한다.

[0086] 한글 또는 일본어 유무는 변환된 2바이트 값으로 판단한다. ⓐ 유니코드 문자코드 차트

(http://www.unicode.org/charts/)를 참조하면, 0x1100~0x11FF(Hangul Jamo), 0xFFA1~0xFFDC(Halfwidth Jamo), 0x3130~0x318F(Hangul Compatibility Jamo), 0xAC00~0xD7AF(Hangul Syllables)이 한글을 나타내는 코드 범위이고, ⓑ 0x3040~0x309F(Hiragana), 0x30A0~0x30FF(Katakana), 0xFF00~0xFFEF(Full-width roman characters and half-width Katakana), 0x4E00~0x9FAF(CJK unified ideographs - Common and uncommon kanji), 0x3400~0x4DBF(CJK unified ideographs Extentsion A - Rare Kanji)이 일본어를 나타내는 코드 범위이다.

[0087] 디코딩된 2바이트가 이 범위내의 코드 값을 가지고 있다면, 혹은 유니코드 테이블을 참조 할 수 있다면 UTF-8로 인코딩된 문서가 한글 또는 일본어를 포함하고 있는지를 알 수 있다. 즉, 아래 표 1과 같은 방식으로 인코딩이 이루어지게 된다.

표 1

코드 범위(십육진법)	UTF-8(이진법)	설명
0000-007F	0xxxxxxx	ASCII와 동일한 범위
0080-07FF	110xxxxx 10xxxxxx	
0800-FFFF	1110xxxx 10xxxxxx 10xxxxxx	한글 또는 일본어 코드 범위

[0088]

[0089] 예를 들면, 패킷 내에서 "위"라는 글자는 EC 9C 84의 3바이트로 표현되며 이진수로는 1110**1100** 10**011100** 10**000100**로 표현된다. 굵게 표시된 부분만을 조합하면 1100-0111-0000-0100과 같으며 16진수로 표현하게 되면 C704의 "위"(U+C704)라는 유니코드 값을 갖는 것을 알 수 있다. 이러한 방식으로 UTF-8에서 한글로 디코딩할 수 있다.

[0090] 압축해제/디코딩을 거친 데이터에 대해 패턴매칭을 거치는 **컨텐츠분석단계**가 이루어진다.

[0091] 패턴매칭은 전술한 바와 같이, 조립이 완료된 플로우 전체를 디코딩한 후에 수행하는 것도 가능하지만, 플로우를 디코딩 하는 것과 동시에 수행하는 것도 가능하다. 이 과정은 크게 키워드들을 저장, 관리하는 컨텐츠분류사전과, 컨텐츠 분류 함수 혹은 알고리즘을 활용하여 이루어진다.

[0092] ① (컨텐츠분류사전) 컨텐츠 분류를 위한 카테고리의 구분방법, 구분수, 각 카테고리에 대응하는 키워드의 종류 및 수 등은 사용자가 임의로 설정할 수 있다.

[0093] 예를 들면, 대형 포털사이트들의 분류를 참조하여 쇼핑/택배, 성인, 주식/금융, 인터넷커뮤니티, 게임, 음악, 영화, 메일/메신저, 교육, 뉴스/웹서비스 등으로 카테고리를 분류할 수 있을 것이다. 이어서 각 카테고리별로 키워드를 상호배타적으로(즉, 하나의 키워드는 하나의 카테고리에 배당되도록) 선정하고 저장한다.

[0094] 이렇게 선정/저장된 데이터베이스를 "컨텐츠분류사전"이라 칭한다.

[0095] 여기서는, 예시적으로 간단한 카테고리 분류 방법을 제시(표 2)하고 예를 참조하여 설명한다. 표 2에서는 한글로만 표현하였지만, 일본어로도 동일한 컨텐츠분류사전을 작성할 수 있음은 당연하다. 필요하다면 영어 등 다른 언어로된 키워드도 추가될 수 있을 것이다. 키워드는 카테고리 당 15~20개 정도로 선정하는 것이 좋다.

표 2

카테고리	키워드
쇼핑/택배	쇼핑, 옥션, 택배, 배송,...
성인	성인, 19금, 에로, 섹스,...
주식/금융	주식, 증권, 은행, 주가, 코스피, 코스닥,...
인터넷커뮤니티	블로그, 클럽, 홈페이지, 포럼, 방명록, 게시판, 카페,....
게임	게임, 롤플레잉, 전략, 닌텐도, 플레이스테이션(플스1,2,3),...
음악	음악, 가수, 신곡, 인기곡, 댄스, 발라드, 힙합, 앨범,...
영화	영화, 극장, CGV, 시네마,...
메일/메신저	메일, 메신저, 네이트온,...
교육	교육, 영어, 수학, 어학, 과외,...
뉴스/웹서비스	뉴스, 기사, 기자, 신문,...
....	....

[0096]

[0097] ② (분류함수/알고리즘) 이어서 키워드 패턴매칭을 통한 컨텐츠 분류가 이루어진다.

[0098] 패킷 페이로드의 데이터를 바탕으로 컨텐츠를 분류하는 이 방법은 기본적으로 텍스트 데이터의 문서 분류와 일맥상통한다고 할 수 있다. 즉, 각각의 카테고리에 속한 키워드들이 텍스트 문서에 얼마나 많이 속해있는지의 정보를 바탕으로 베이시안 학습(Baysian Learning) 혹은 SVM(Support Vector Machine)과 같은 기계학습을 이용하여 컨텐츠를 분석한다. 즉 HTML/XML 문서 단위로 합쳐진 패킷들의 페이로드에서 키워드 매칭을 통해 카테고리별 키워드 매칭 빈도수를 계산한다. 이 정보를 바탕으로 텍스트 데이터들을 분류하는데 가장 좋은 성능을 발휘한다고 알려진 알고리즘 중 하나인 베이시안 학습(Baysian Learning)과 SVM(Support Vector Machine)방법을 이용하여 해당 HTML/XML 문서단위의 패킷들이 어떠한 컨텐츠를 저장하고 있는지 기계적으로 판단하게 된다.

[0099] 일반적으로, 텍스트 데이터 마이닝에서는 머신러닝 알고리즘에 넣을 데이터를 bags-of-words와 같은 방법을 이용한다. 이런 방법은 높은 정확도를 제공하지만 상당한 연산 시간을 요구한다.

[0100] 카테고리 당 키워드 개수 차이가 심할 경우, 정규화(normalization)의 개념을 적용시켜 키워드 hit수(매칭된 카운트)를 해당 카테고리의 총 키워드 수로 나누어 다음 표 3과 같은 데이터를 얻을 수 있다.

표 3

문서번호	카테고리 1	카테고리 2	....	카테고리 K	패킷 수	바이트 수	F_{simple}
1	2	5		0	10	12057	1
2	7	0		1	8	9934	1
....							
N	3	5		10	5	7385	2

[0101]

[0102] - N: 한글 디코딩을 통한 총 문서 수(마지막 문서 번호)

[0103] - K: 컨텐츠 카테고리 총 수(마지막 카테고리 번호)

[0104] - F_{simple}: 해당 문서 내에서 가장 먼저 발견된 키워드가 속한 카테고리

[0105] - 문서의 실제 카테고리 번호: Optional. 검증을 위해 매뉴얼하게 구한 해당 문서의 실제 카테고리.

[0106] 위와 같은 데이터를 얻은 경우, 다음과 같은 컨텐츠 분류 함수를 이용할 수 있다.

[0107] F_{simple}: 해당 문서 내에서 가장 먼저 발견된 키워드가 속한 카테고리

[0108] F_{max}: 해당 문서 내에서 가장 많은 키워드 hit를 기록한 카테고리

[0109] F_{svm}: 위의 카테고리별 hit수 정보를 바탕으로 SVM 머신러닝 알고리즘을 통해 분류된 결과

[0110] 실제 필요한 정보(정확도 혹은 속도 등)에 맞춰 분류 함수를 이용할 수 있다. 이는 전수 조사 혹은 샘플링을 통해서 검증된 데이터와의 비교를 통해 결정하면 된다.

도면의 간단한 설명

[0111] 도 1은 본 발명에 따른 한글 또는 일본어 디코딩 방법 및 그 결과 얻어진 웹 컨텐츠를 분석하는 방법의 일예를 보여주는 전체 흐름도이다.

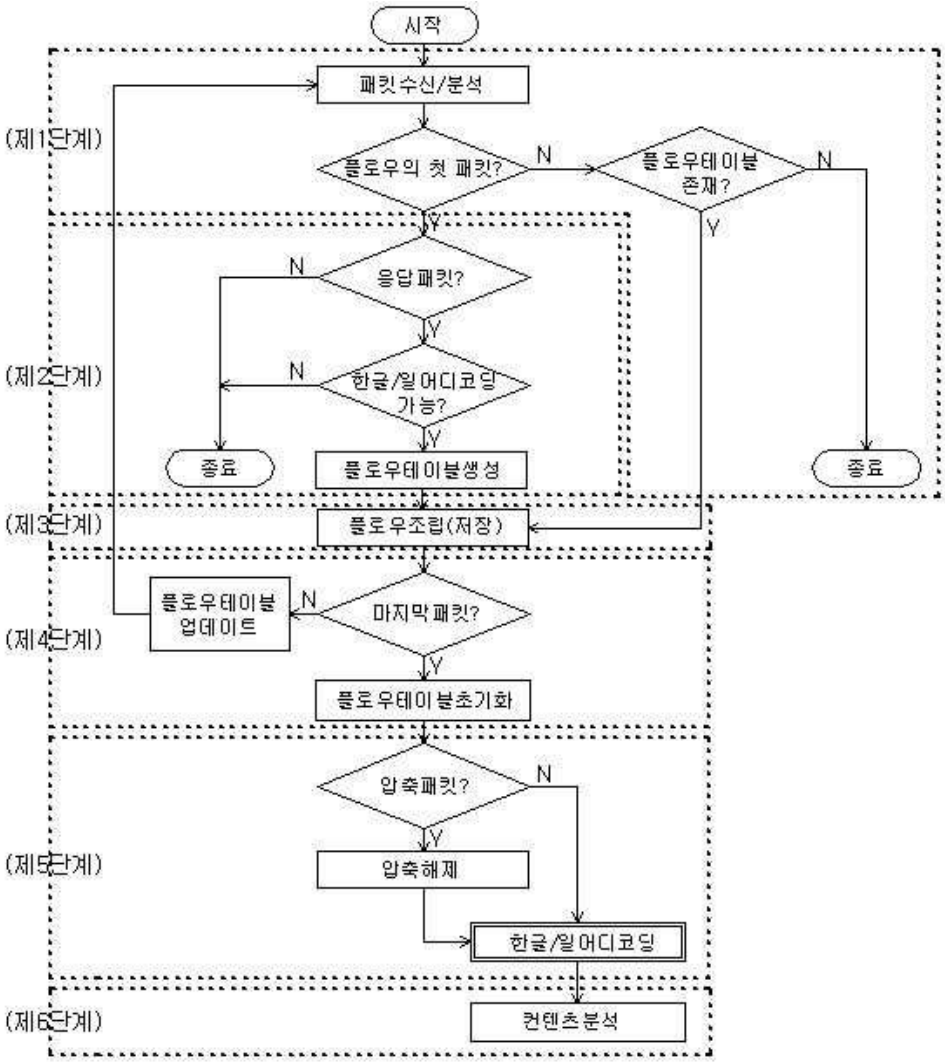
[0112] 도 2 본 발명에 따른 한글 또는 일본어 디코딩 방법의 다른 예를 보여주는 흐름도이다.

[0113] 도 3은 본 발명에 따른 한글 또는 일본어 디코딩 방법 및 그 결과 얻어진 웹 컨텐츠를 분석하는 방법의 또 다른 예를 보여주는 전체 흐름도이다.

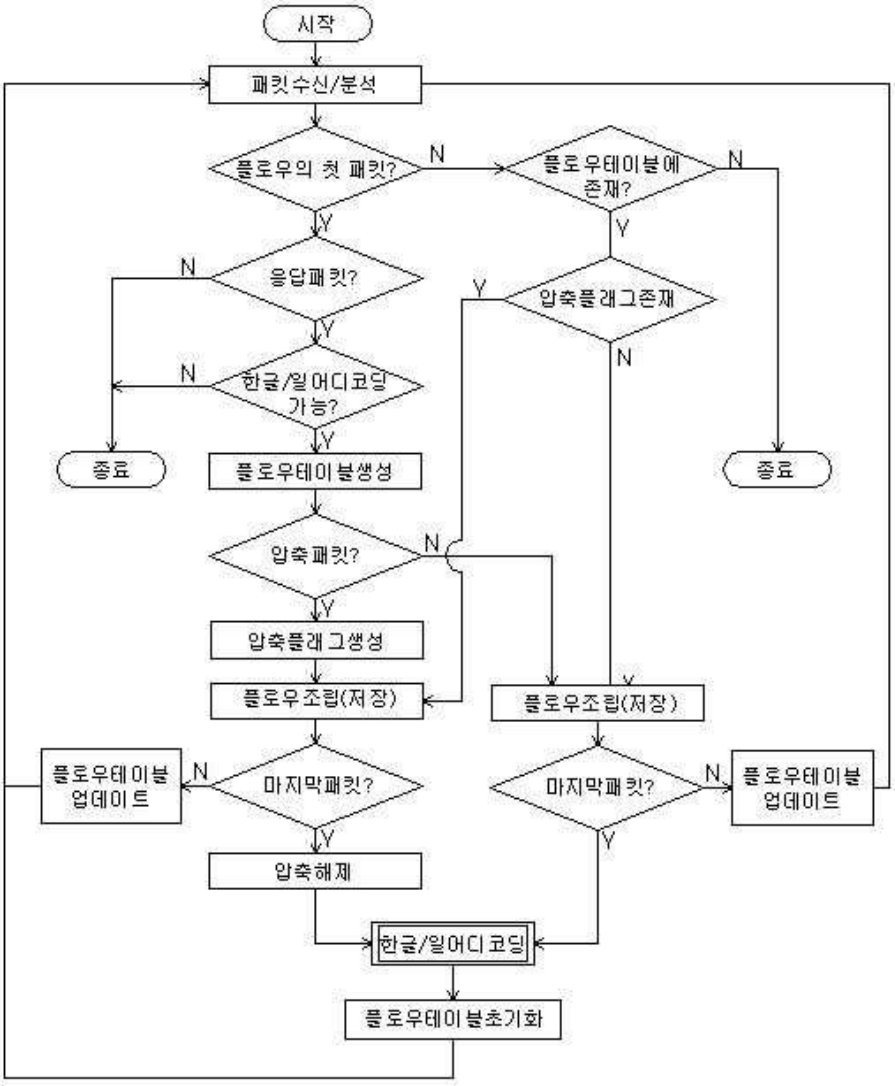
[0114] 도 4는 본 발명의 압축해제/디코딩단계의 일예를 보여주는 세부 흐름도이다.

도면

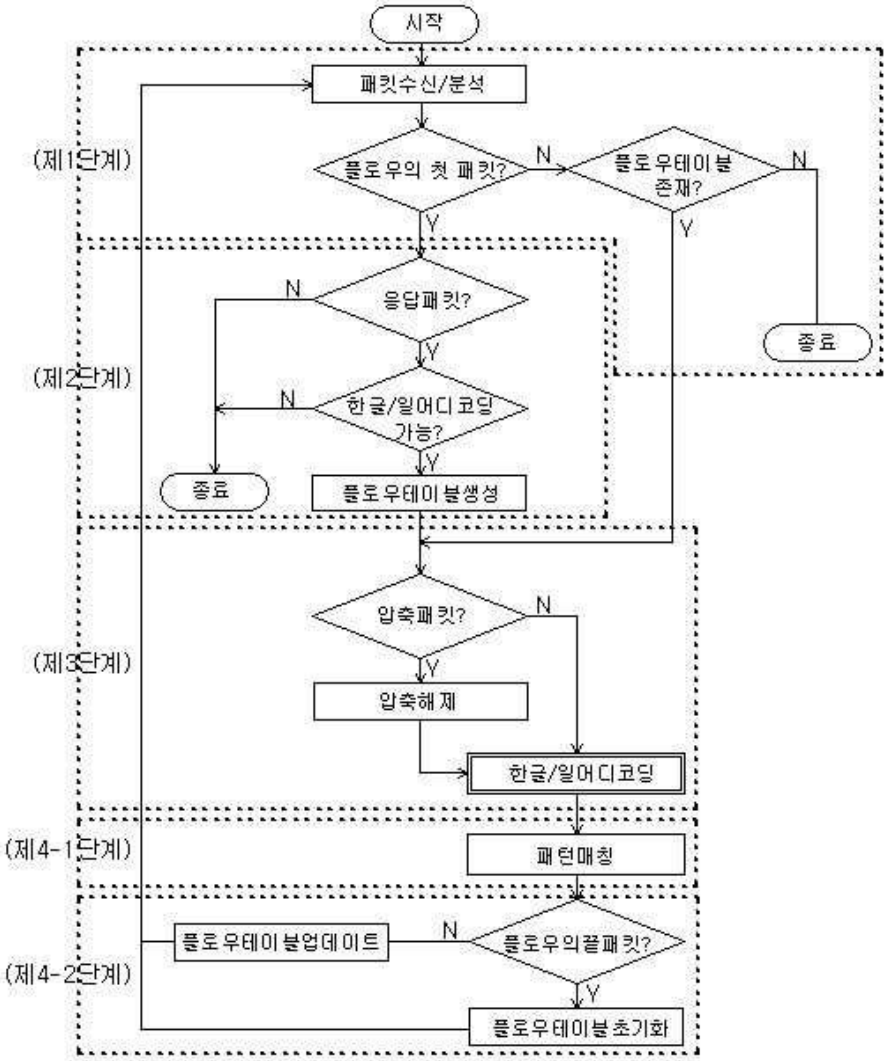
도면1



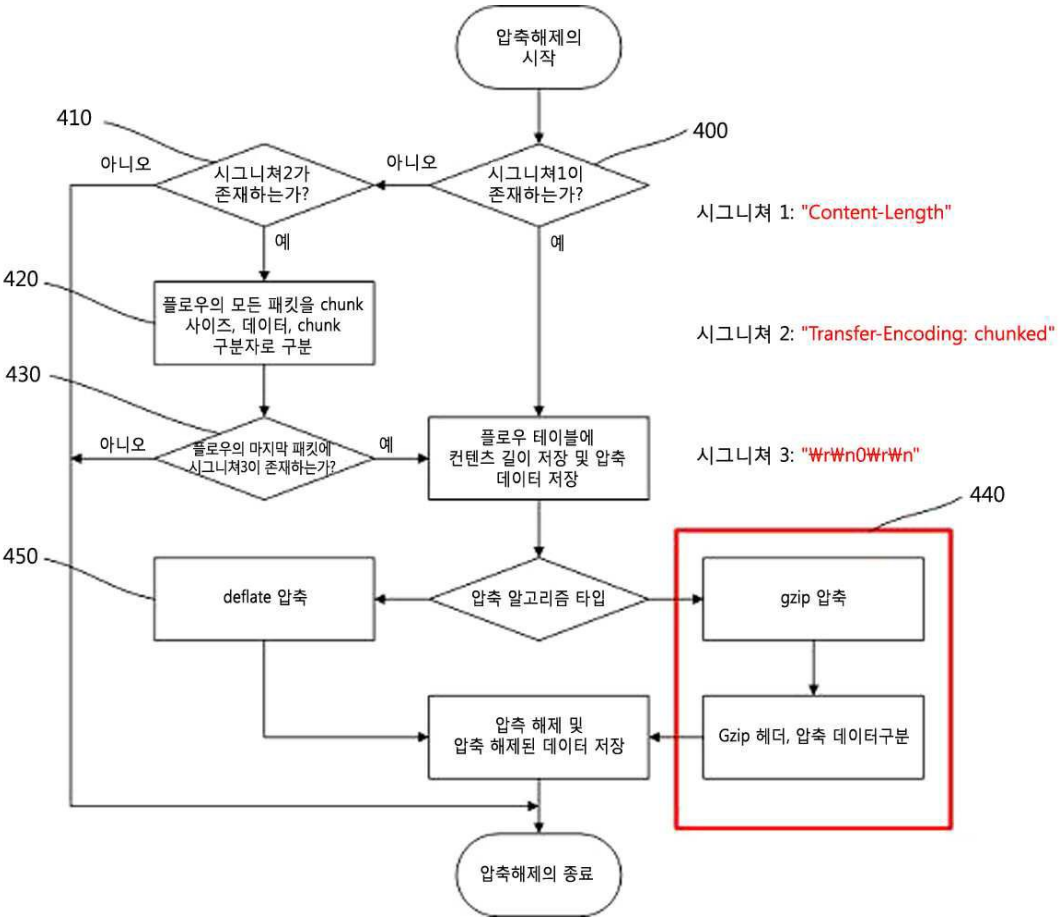
도면2


시작
패킷수신/분석
플로우의 첫 패킷?
N
플로우테이블에 존재?
N
Y
Y
응답패킷?
N
Y
압축플래그존재
Y
N
한글/일어디코딩 가능?
N
Y
종료
플로우테이블생성
종료
압축패킷?
N
Y
압축플래그생성
플로우조립(저장)
플로우조립(저장)
플로우테이블 업데이트
N
마지막패킷?
마지막패킷?
N
플로우테이블 업데이트
Y
Y
압축해제
한글/일어디코딩
플로우테이블초기화

도면3


시작
패킷수신/분석
(제1단계)
플로우의 첫 패킷?
N
플로우테이블 존재?
N
Y
Y
종료
(제2단계)
응답패킷?
N
Y
한글/일어디코딩 가능?
N
Y
종료
플로우테이블생성
압축패킷?
N
Y
(제3단계)
압축해제
한글/일어디코딩
(제4-1단계)
패턴매칭
(제4-2단계)
플로우테이블업데이트
N
플로우의끝패킷?
Y
플로우테이블초기화

도면4


압축해제의 시작
400
410
시그니쳐1이 존재하는가?
시그니쳐2가 존재하는가?
아니오
예
420
플로우의 모든 패킷을 chunk 사이즈, 데이터, chunk 구분자로 구분
430
플로우의 마지막 패킷에 시그니쳐3이 존재하는가?
플로우 테이블에 컨텐츠 길이 저장 및 압축 데이터 저장
440
450
압축 알고리즘 타입
deflate 압축
gzip 압축
Gzip 헤더, 압축 데이터구분
압축 해제 및 압축 해제된 데이터 저장
압축해제의 종료
시그니쳐 1: "Content-Length"
시그니쳐 2: "Transfer-Encoding: chunked"
시그니쳐 3: "₩r₩n0₩r₩n"