하드웨어 기반 Anti-DDoS 대응 장비 고속 패킷 필터링을 위한 Hi-DPI 알고리즘 연구★

김점구*

요 약

인터넷 활용 범위의 폭발적인 증가는 점차적으로 네트워크 속도와 용량을 초고속화 하고 대용량화로 빠르게 진화해 가고 있다. 이에 따라 스위치 라우터 등 네트워크 장비들은 하드웨어에 기반 한 빠른 기술 진화로 대처를 하고 있으나 초연결사회에 가장 기본적이고 필수적인 네트워크 보안시스템의 기술 진화는 수만 가지의 보안 이슈와 시그니처(signature)에 대해서 수시 변경과 갱신을 필요로 하기 때문에 소프트웨어에 기반 한 기술적인 한계를 극복하기가 쉽지 않다. 본 논문은 이와 같은 DDoS 대응 장비를 설치 운영할 때의 패킷 필터링 속도 저하 문제점을 개선하고자 FPGA(Field Programmable Gate Array)의 하드웨어적인 특성과 병렬처리 특성을 최대한 반영한 DPI 알고리즘인 Hi-DPI를 제안하고 실용성을 검증하고자 한다.

Development Hi-DPI Algorithm for High Speed Packet Filtering of Anti-DDoS based on HW

Kim Jeom Goo*

ABSTRACT

The explosive increase in the range of Internet usage gradually makes the speed and capacity of network high-speed, rapidly evolving it into mass storage. Accordingly, network equipment such as switch and router are coping with it through hardware-based rapid technological evolution, but as the technological development of the most basic and essential network security system in the hyper-connected society requires frequent alterations and updates about the security issues and signatures of tens of thousands, so it is not easy to overcome the technical limitations based on the software. In this paper, to improve problems in installing and operating such anti-DDoS devices, we propose a Hi-DPI algorithm best reflecting the hardware characteristics and parallel processing characteristics of FPGA (Field Programmable Gate Array), and would verify the practicality.

Key-words: DPI, DDoS, Signature, FPGA, Packet filtering

접수일(2017년 6월 6일), 게재확정일(2017년 6월 26일)

* 남서울대학교 컴퓨터학과

★ 본 논문은 남서울대학교 2016년도 교내학술연구조성비 지원에 의하여 연구되었음.

1. 서 론

인터넷 활용 범위의 폭발적인 증가는 점차적으로 네트워크 속도와 용량을 초고속화 하고 대용량화로 빠르게 진화해 가고 있다. 이에 따라 스위치 라우터 등 네트워크 장비들은 하드웨어에 기반 한 빠른 기술 진화로 대처를 하고 있으나 초연결사회에 가장 기본적이고 필수적인 네트워크 보안시스템의 기술 진화는 수만 가지의 보안 이슈와 시그니처(signature)에 대해서 수시 변경과 갱신을 필요로 하기 때문에 소프트웨어에 기반 한 기술적인 한계를 극복하기가 쉽지 않다.

개발자가 말하는 기존 소프트웨어 기반 네트워크 보안시스템의 한계는 수년 전부터 문제가 제기되고 있는 TCP_PUSH_ACK 공격에 대해 명확한 해결책을 제시하지 못하고 있다는 것이다. 기존 네트워크 보안시스템에 TCP_PUSH_ACK 공격/방어정책을 추가하기 위해서는 장비 전체 설계를 수정해야 하며, 설계 수정을 통한 TCP_PUSH_ACK 공격/방어정책이 추가된 장비의 경우는 설치 이전의 성능보다 처리속도가 크게 떨어져 실질적인 문제 해결이 되었다고 보기가 어렵다[1][2].

또 보안전문가가 말하는 소프트웨어 기반 네트워크 보안시스템의 한계는 기본적으로 네트워크 장비인 스위치 및 라우터는 50μsec 범위의 낮은 지연시간(latency)을 갖는데 반해서 네트워크 보안시스템은 250μsec 이상 매우 높은 지연시간으로 네트워크 성능 저하에 영향을 끼치고 있다는 것이다. 일반적으로 네트워크 보안시스템의 지연시간은 스위치나 라우터와 수준을 유지하게 해야만 병목현상을 유발시키지 않을 것이며, 이 지연시간은 DDoS 같은 공격 시에도 유지 되어야 한다는 것이다[2][3].

그리고 사용자(고객)가 말하는 기존 소프트웨어 기반 네트워크 보안시스템의 한계는 CC 인증을 받은 장비라 믿고 도입해 사용하고 있으나 평시에는 문제가 없다가도 DDoS 공격 등과 같은 해킹공격이 발생되었을 때는 전체 패킷의 20~40%만 유입되므로 스위치 및 라우터가 유입을 허용한 100% 중 60~80%는 스위치 및 라우터에 머물게 되어 외부망에 대한 부하 증가와 정상 패킷의 내부 유입이 원활하지 못해 서비스의 불신을 가져오게 된다는 것이다. 이러한 현상을 해소하고 원활한 서비스 제공을 위해 추가적인 시스템 도입이 필요하며, 사람의 손을 빌어 수동으로 런아웃(run-out) 시키거나 다른 서비스에 피해가 없도록 하기 위해 서비스차단이라는 극단적인 선택을 하는 경우도 있다.

본 논문은 이와 같은 DDoS 대응 장비를 설치 운영할 때의 문제점을 개선하고자 FPGA(Field Programmable Gate Array)의 하드웨어적인 특성과 병렬처리 특성을 최대한 반영한 DPI 알고리즘인 Hi-DPI를 제안하고 실용성을 검증하고자 한다.

2. 고속 DPI 방식

2.1 멀티 코어 방식

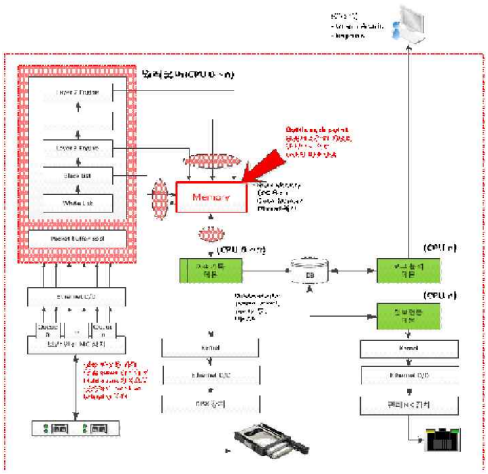
멀티코어 방식은 Intel/AMD 기반의 서버 제품들로 이 방식을 바탕으로 플랫폼이 양산 공급되었고, S/W 개발사들은 여기에 자사의 제품을 탑재하여 소비자에게 IDS, IPS, Firewall, VPN 등의 서비스를 제공하였다. 또한 대부분의 오픈소스 그룹에서 공개된 소프트웨어가 이 기반에서 쉽게 컴파일 할 수 있도록 되어 있어 쉽게 보안 서비스를 운영할 수 있도록 소스를 배포하였다. 이로 인해 1990년대 국내외 보안 기업들이 쉽게 자사의 핵심 기술과 오픈 소스를 바탕으로 쉽게 제품화하여 빠른 시장 진입을 할 수 있었다.

2000년대에 들어서면서 네트워크의 트래픽이 기하급수적으로 증가하면서 보안제품들도 성능에 민감하게 대처하지 않으면 안되는 상황이 발생되었다. 이 때 보안장비는 멀티코어 기반의 방식으로 1 GIGA 망에서 약 200 ~ 300Mbps의 성능을 가지고 시장점유율 30%를 점유하였고, 이로 인해 보안 장비를 설치한 소비자는 패킷의

유실로 인한 재전송 패킷의 양이 늘어나고 네트워크가 느려지는 현상이 발생되어 불만이 고조되었다[3].

최근 사이버테러(7.7 DDoS 대란, 3.20 대란, 6.25)에서도 볼 수 있듯이 공격의 형태가 좀 더 정밀해지고 복잡해짐에 따라 이를 탐지/차단하기 위해서는 Header lookup, L7 Pattern matching, Session status 등의 정보를 바탕으로 공격자의 행위를 분석하여 정책을 수립하여야 한다. 이로 인해 보안 기능은 좀 더 복잡해지고, 많은 기능이 필요해짐에 따라 프로세스의 처리량은 갈수록 증가된다. 보안업체들은 멀티코어 방식의 Intel/AMD의 클럭 속도를 높여 이를 해결하려고 노력하였으나 인텔은 이 시점에 클럭 속도 경쟁을 포기하여 3.6GHz이상을 개발하지 않고 매니코어 방식으로 전환하였다. 이로 인해 보안업체는 다른 대안을 모색하기 시작했다[4].

(그림 1)의 화살표부분에서 보는 바와 같이 프로세스 간에 메모리를 중첩으로 사용하므로 병목현상이 발생하여 성능 저하에 결정적인 원인이 된다.

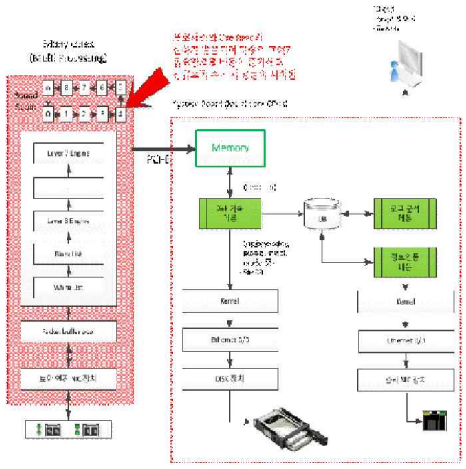


(그림 1) 멀티코어 방식

2.2 매니코어 방식

매니코어 방식은 병렬프로그램으로 재설계 없이 기존 코드를 사용하면 다수의 코어가 Round Robin 형태로 돌아가면서 처리 블럭을 처리함으로써 기존 단일 프로세서에서 순차적으로 처리하는 구조보다는 높은 성능 향상을 가져온다. 매니코어 방식도 멀티코어 방식과 마찬가지로 기존 코드(보안 개발 소스)로 이식하여 바로 운영하는 데에는 어려움 없이 가능하다. 그러나 단일코어나 멀티코어 방식의 코드를 재설계 없이 이식한다고 하면 멀티코어 보다는 성능은 좋아지겠지만 매니코어 장점을 최대한 활용하기에는 무리가 따른다. 따라서 기존 코드를 재사용하지 않고 매니코어 방식으로 재설계가 되어야 성능과 기능부분을 처리할 수 있을 것이다.

프로세스 간 공유메모리를 통해 정보 공유를 수행할 때, 메모리 접근에 따른 Lock의 필요성으로 인해 성능 저하가 발생되기 때문에 설계 시 메모리 중첩을 최소화할 수 있는 설계가 필요하다. 패턴 매칭과 같은 패턴 검색 또는 정렬 등은 소프트웨어 방식에서 CPU자원이 많이 소모되는 작업이다. CPU자원을 덜 소모하고 성능을 보장받기 위해서는 특수 용도의 고속 메모리를 탑재하여 활용할 수 있어야 하고, 다양한 하드웨어 자원과 매니코어를 결합시키기 위해서는 하드웨어 설계/개발과 양산할 수 있는 인력 구성과 기반이 있어야 한다. 하지만 대부분 멀티코어와 매니코어 방식을 도입하는 회사의 특성상 하드웨어 인력을 확보하고 있지 않아 어려움이 따른다[7].



(그림 2) 매니코어 방식

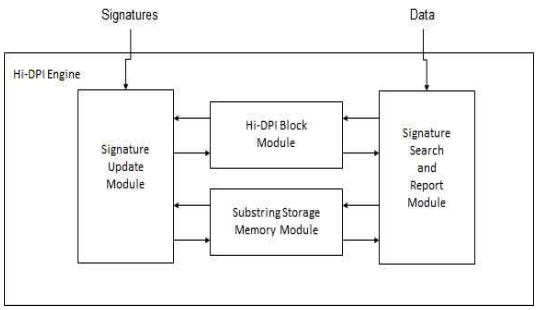
(그림 2)의 화살표 부분에서처럼 프로세서가 연관되어 실행되므로 새로운 로직의 추가 시에는 전체적인 프로그램 수정이 필요하고 프로세서의 오버헤드가 자주 발생되므로 비용 및 성능 저하의 원인이 된다[8][9].

3. Hi-DPI 알고리즘

3.1 Hi-DPI 개요

Hi-DPI 엔진을 이용한 시그니처 탐지 방식에 대은 (그림 3)은 Hi-DPI의 최상위 다이어그램을 보이고 있다. Hi-DPI는 시그니처 갱신 모듈, 시그니처 검색 보고서 모듈, Hi-DPI 블럭 모듈과 서브스트링 저장 모듈로 구성이 되어 있다.

Hi-DPI는 연속적으로 흘러가는 데이터로부터 시그니처 서브스트링을 탐지하는 목적으로 사용된다. 시그니처들은 시그니처 갱신 모듈에서 처리(compile) 되어 그 결과 값들이 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈에 저장된다. 시그니처 갱신 모듈에서는 여러 개의 서브스트링으로 분리되는데 각 길이들은 다를 수 있다.



(그림 3) Hi-DPI 최상위 다이어그램

각 서브스트링은 또한 여러 개의 키 값으로 변환된다. 이 키 값들은 Hi-DPI 모듈에 있는 Hi-DPI들의 주소로 사용이 된다. 키 값들은 특별한 해싱처리를 하지 않고 서브스트링의 특정 위치 비트 정보 값을 이용하여 구성이 된다. 서브스트링의 일부분으로 부터 선택되는 비트들은 서브스트링을 구성하는 문자들의 상관관계를 나타내며 시그니처 자체의 정보를 가지게 되는데 각 서브스트링의 길이에 따라 각 키의 길이가 다를 수 있다.

Hi-DPI 모듈은 소량의 메모리 그룹으로 구성이 되어 있는데 이것은 Hi-DPI이고, 키 값들이 메모리주소로 사용이 되며, 메모리 공간에는 시그니처 갱신 모듈에서 처리되는 서브스트링의 특정 정보가 저장된다. 키 값들은 시그니처 갱신 모듈에서 만들어지는데 Hi-DPI의 메모리주소로 사용이 된다. 키들이 지정하는 Hi-DPI 메모리 공간은 최대 적용횟수가 있다. 특정 사용횟수에 이르면 해당 키는 선행 필터링을 위해 다른 서브스트링을 위해 적용되지 못한다. 이 기능은 시그니처를 하나씩 등록하고 삭제하는 과정에서 다른 시그니처 들에게 영향을 주지 않게 하는데 필요하다.

Hi-DPI 블럭 모듈에 속한 Hi-DPI 블럭의 수에 따라 서브스트링 저장 모듈이 다수의 영역으로 분할된다. 서브스트링 저장 모듈의 전체 메모리 용량은 가능하면 큰 용량으로 적용을 하면 좋은데 예를들면, Hi-DPI 블럭 모듈의 Hi-DPI 블럭 수와 같게 나누어진다. 해싱 구조를 이용하여 Hi-DPI 블럭 모듈의 Hi-DPI 블럭들과 서브스트링 저장 모듈에서 분할된 메모리영역들과 연관을 시킬 수 있다.

동작속도에 따라 서브스트링 저장 모듈의 분할된 메모리영역을 표시하는 해싱 주소는 Hi-DPI 블럭 모듈로 부터 부분적으로 저장된 매치 정보를 위해 다수의 시그니처 서브스트링이 사용될 수 있다.

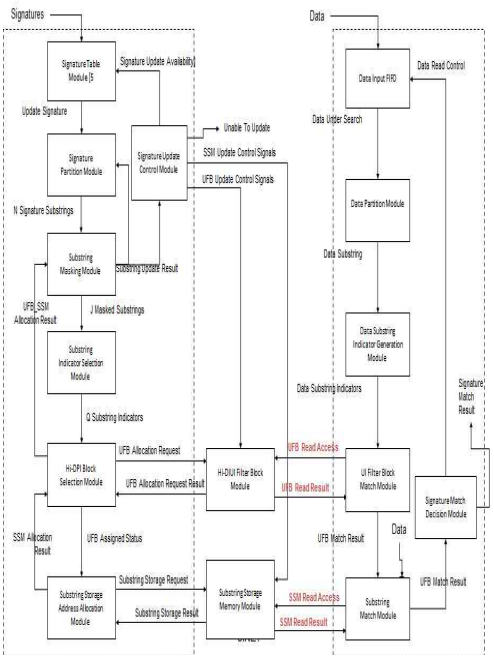
시그니처 갱신 모듈에 의해 시그니처가 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈로 등록이 된 후, 시그니처 검색과 보고서 모듈에 실제 데이터가 입력이 되면 시그니처 검색과 보고서 모듈은 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈을 액세스하여 시그니처 매칭 절차가 실시간으로 수행된다.

검색되고 있는 실제 데이터로부터 키들이 발생이 되어 Hi-DPI 블럭 모듈에 있는 Hi-DPI 블럭들을 동시에 액세스한다. Hi-DPI 블럭의

Hi-DPI들로 부터 부분 매치가 발생이 되면 해당 데이터 영역이 표시되어 저장이 되고 해싱의 결과 값이 주소로 사용되어 서브스트링 저장 모듈을 액세스한다. 부분 매치가 되어 표시된 데이터 부분과 서브스트링 정보를 비교하여 시그니처 서브스트링 비교결과가 나오게 된다. 하나의 시그니처에 속하는 연속적인 서브스트링 비교 결과가 나타나면 완전한 매칭이 일어난 것으로 판단한다.

3.2 Hi-DPI를 이용한 시그니처 검색 방법

(그림 4)에는 시그니처 갱신 모듈과 시그니처 검색과 보고서 모듈의 세부 구성과 두 모듈들 간의 신호를 보이고 있다. (그림 4)에서는 시그니처가 어떻게 처리되어 갱신되는 지에 대한 일반적인 방식을 보이고 있다. 그리고 시그니처 검색이 어떻게 진행이 되는지도 예시하고 있다.

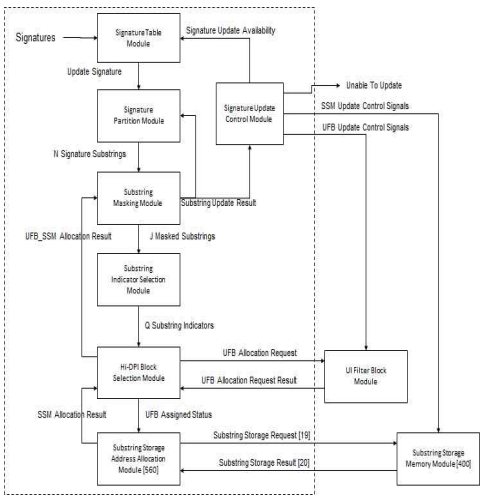


(그림 4) 시그니처 검색 방법

3.2.1 시그니처 갱신 모듈

시그니처 갱신 모듈은 시그니처 테이블 모듈, 시그니처 파티션 모듈, 서브스트링 마스킹 모듈, 서브스트링 식별 선정 모듈, Hi-DPI 블럭 선정 모듈, 서브스트링 기억 주소 할당 모듈과 시그니처 갱신 제거 제어 모듈로 구성이 된다.

시그니처 테이블 모듈은 시그니처 갱신 모듈에 입력되는 시그니처들을 처리되기 전에 일시적으로 저장하는 모듈이다. 시그니처 갱신 제어 모듈로 부터의 시그니처 갱신 가능 신호가 시그니처에 등록을 할 수 있음을 알려주면 시그니처 테이블 모듈에서 갱신 시그니처가 처리되어 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈에 등록될 시그니처 파티션 모듈로 전달된다. 각 시그니처의 길이는 최소 4 바이트 이다.



(그림 5) 시그니처 갱신 모듈

(그림 5)에서 시그니처 파티션 모듈은 K바이트 길이의 시그니처 가 N 개의 시그니처 서브스트링으로 나누어진다. 처음 (N-1) 개의 서브스트링의 서브스트링 길이(SBL)은 길이가 S 바이트이고 마지막 N 번째의 SBL은 S 바이트 이하일 수 있다. 서브스트링 마스킹 모듈로 부터의 서브스트링 갱신 결과 신호는 시그니처 파티션이 제대로 진행이 되어 그 다음 단에서 시그니처 갱신이 바로 되었는지를 알려준다. 서브스트링 갱신

결과는 다음과 같은 4개 모듈로(서브스트링 마스킹 모듈, 서브스트링 식별 선정 모듈, Hi-DPI 블럭 선정 모듈, 그리고 서브스트링 기억 주소 할당 모듈) 부터의 결과를 나타내 준다. 여기에 대한 자세한 설명은 다음 섹션에서 설명이 된다.

서브스트링 마스킹 모듈은 인접한 서브스트링 바이트들의 그룹에 마스크를 할당하여 각 시그니처 서브스트링 키 바이트로 정의한다. 이렇게 구성된 마스크 된 서브스트링은 서브스트링 식별 선정 모듈로 보내진다.

J 개의 마스크 된 서브스트링은 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈에서 메모리영역을 찾게 된다. 이렇게 진행되는 메모리 영역 할당 시도 결과는 UFB_SSM 할당 결과 신호로서 리포트 된다.

각 서브스트링 갱신 절차의 결과는 Hi-DPI 블럭 선정 모듈로 부터의 UFB_SSM 할당 결과 신호에 나타난다. N 개의 서브스트링 갱신 결과인 UFB_SSM 할당 경과 신호들을 기본으로 하여 서브스트링 갱신 결과 신호들이 생성되어 시그니처 패턴 모듈과 시그니처 갱신 제어 모듈로 전달된다.

시그니처 서브스트링에 대해 서브스트링 마스킹 모듈에서 처리한 결과는 서브스트링 갱신 결과는 모두 positive일 때 시그니처 갱신 가능이 positive로 되어 다른 시그니처가 갱신될 수 있음을 알리게 된다.

갱신 시그니처는 시그니처 갱신 제어 모듈에서 모든 절차가 완료된 후에 시그니처 갱신 가능이 positive 일 때 시그니처 테이블 모듈로부터 시그니처 패턴 모듈로 보내지게 된다.

이 경우에서만 시그니처 갱신 제어 모듈은 N 개의 시그니처 서브스트링들을 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈에 UFB 갱신 제어 신호와 SSM 갱신 제어 신호를 이용하여 갱신 한다.

UFB 갱신 제어 Signals[25] 은 Hi-DPI 블럭 모듈에 있는 관련된Hi-DPI를 접속하기 위한주소, 데이터 그리고 관련 쓰기 가능 신호로 구성되어 있다. SSM 갱신 제어 신호는 N 개의 시그니처 서브스트링들을 갱신하기 위한 주소, 데이터 그리고 쓰기 가능 신호로 구성이 된다.

만약 시그니처 서브스트링들 중 한 개라도 갱신되지 못한다면 시그니처 파티션 모듈이 S 의 길이를 증가하여 시그니처 서브스트링을 갱신하는 절차를 거치게 된다.

서브스트링 마스크 모듈에서는 M 바이트들이 시그니처 서브스트링을 나타내는 데 사용될 키 바이트들로 선택이 된다. 마스킹에 사용되는 바이트의 수는 M 개 이다. J 개의 다른 마스크가 구성될 수 있는데 J 값은 (S + 1 - M) 와 같다. 따라서 마스크의 길이인 M 값이 서브스트링 의 길이인 S 과 같으면 J 값은 1이 되게 된다.

Hi-DPI 블럭 선정 모듈 의결과 값인 UFB-SSM 할당 결과 신호가 서브스트링 값이 갱신되지 못한다고 알리면 서브스트링 마스킹 모듈은 다른 마스크 값을 선택하게 된다. UFB_SSM 할당 결과 값이 서브스트링 갱신 시도가 J 번 불가능한 것으로 나타나면 서브스트링 갱신 결과 시그니처 파티션 모듈이 시그니처를 다른 길이의 S 로 나누게 하도록 알린다.

서브스트링의 마스크된 키 바이트들로 부터 서브스트링 식별 선정 모듈은 Q 개의 서브스트링 식별들을 선택한다. 서브스트링 식별의 길이는 F 비트이다. 서브스트링 식별은 마스크 서브스트링 키 비이트 영역에 속하는 비트들의 조합으로 구성된다. 따라서 Q 개의 서브스트링 식별들로 구성된 조합이 J 개 있을 수 있다.

Hi-DPI 블록 선정 모듈은 Q 개의 서브스트링 식별을 Hi-DPI 블럭 모듈의 메모리의 주소로 사용하여 Hi-DPI 블럭 모듈을 접속하게 된다. 이 과정은 UFB 할당 요청 신호에 의해서 시작된다. Hi-DPI 블록 이 J 번째의 서브스트링 식별들로부터 Q 개의 메모리 영역의 주소에 액세스하여 공간이 할당이 될 수 있으면 Hi-DPI 블럭은 UFB 할당 요청 결과 신호를 positive 로 만든다. 그렇지 않은 경우에는 UFB 할당 요청 결과 상태가 negative 로 된다.

UFB 할당 요청 결과가 J 번째의 UFB 할당 요청에 대해 positive 한 결과를 나타내면 Hi-DPI 블럭 선정 모듈은 UFB 지원 상태를 positive로 만들어서 서브스트링 저장 주소 할당 모듈로 전달해 준다. 그런 후에 해싱 방식을 이

용하여 주소가 생성되고 적절한 서브스트링 저장 결과가 서브스트링 저장 모듈에 전달해 준다.

서브스트링 저장 모듈에서는 서브스트링을 저장하기에 사용가능한 메모리 저장 공간이 확인되어 positive 서브스트링 저장 결과 신호가 발생이 된다. 사용가능한 서브스트링 저장 공간이 확인이 되면 SSM 할당 결과 신호가 생성되어 Hi-DPI 블럭 선정 모듈로 전달이 되게 된다.

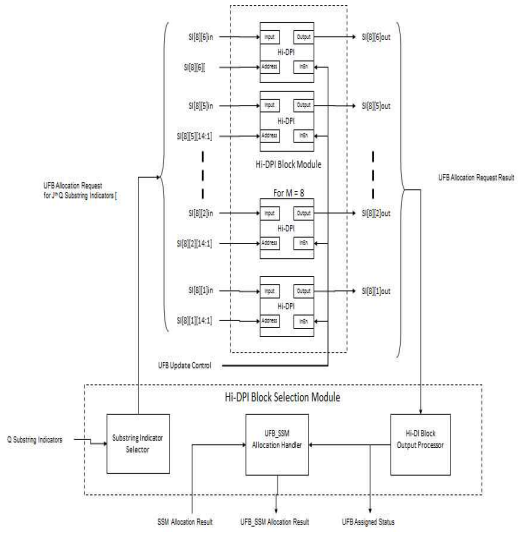
UFB 할당 요청 결과와 SSM 할당 결과의 값에 따라 Hi-DPI 블럭 선정 모듈은 UFB_SSM 할당 결과를 생성하여 서브스트링 마스킹 모듈로 보내준다. 만약 SSM 할당 결과의 값이 positive 이라면 서브스트링 마스크 모듈은 그 다음의 마스크 된 서브스트링을 서브스트링 인식선정 모듈로 보내어 서브스트링 갱신 절차를 시작하게 한다. 만약 UFB 할당 요청 모듈의 결과가 negative 라면 서브스트링 마스크 모듈은 현재의 시그니처 서브스트링을 위해 시그니처 파티션 모듈로 부터 제공된 마스크들 중 다음으로 사용가능한 마스크를 이용하여 서브스트링 갱신 절차를 시작한다.

갱신 시그니처를 위한 처리가 진행이 되는 동안 각 처리 결과는 시그니처 갱신 제어 모듈에 전달이 된다. 시그니처 갱신 제어 모듈이 N 개의 positive SSM 할당 결과가 있었다는 것을 확인하면 UFB 갱신 제어 신호와 SSM 갱신 제어 신호들을 이용하여 Hi-DPI 블럭 모듈과 서브스트링 저장 모듈을 갱신 하게 된다. 만약 시그니처 갱신 제어 모듈이 갱신 시그니처가 Hi-DPI 블럭 모듈이나 서브스트링 저장 모듈에 갱신 될 수 없음을 알게 되면 Unable To 갱신 신호를 발생하게 된다.

3.2.2 Hi-DPI 블록 모듈

(그림 6)은 Hi-DPI 블럭 선정 모듈과 Hi-DPI 블럭 모듈간의 상관관계를 보이고 있다. Hi-DPI 블럭 선정 모듈은 서브스트링 인식 선택자, UFB_SSM 할당 핸들러, Hi-DPI 블록 출력 처리기로 구성이 된다. 서브스트링 인식 선택자는 서브스트링 식별 선정 모듈로 부터 Q 서브스트링 인식를 Hi-DPI 블럭 모듈에 있는 Hi-DPI 블럭에 연결을 시킨다. 그림 8에서의 예는 Q 서브스트링 인식들이 8 바이트 마스크 된 서브스트링이므로 8 개의 서브스트링 인식들이 사용되어 관련된 Hi-DPI 메모리 영역을 UFB 할당 요청 신호의 일부로 액세스하게 된다. 이 메모리들이 액세스되어 READ 되어 나오는 데이터들이 UFB 할당 요청 결과 신호로 사용된다.

Hi-DPI의 메모리 영역에 논리적인 HIGH 정보가 제한된 횟수를 정해두고 저장할 수 있게 하면 액세스된 후 읽어지는 값이 모두 LOW 이거나 모두 HIGH인 것을 보고 사용가능한 메모리 영역인지를 확인할 수 있게 된다. UFB 할당 요청 결과 신호는 Hi-DPI 블록 출력 처리기로 전달이 될 것이다. 그러면 Hi-DPI 블록 출력 처리기는 이 정보를 서브스트링 기억 주소 할당 모듈에 UFB 할당 상태의 형태로 전달하게 된다. UFB 할당 상태 값이 positive 이면 서브스트링 기억 주소 할당 모듈은 서브스트링 저장 기억 모듈을 액세스 하기 위한 메모리 주소들을 생성하여 시그니처 서브스트링을 저장할 수 있는 메모리 공간이 있는지 확인하게 된다. 이 절차의 결과는 SSM 할당 결과로서 Hi-DPI 블럭 선정 모듈로 전달이 된다.

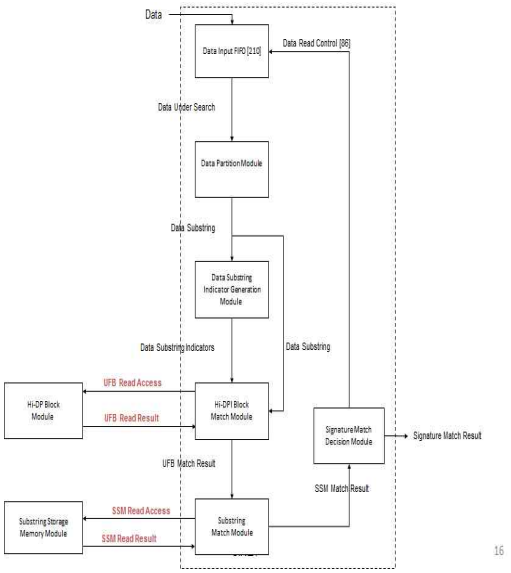


(그림6) Hi-DPI 블록 모듈

SSM 할당 결과가 positive 이면UFB_SSM 할당 조작자는 UFB_SSM 할당 결과 신호를 만들어서 서브스트링 마스킹 모듈로 전달한다. 이와 같은 과정이 N 번 수행되고 SSM 할당 결과 값이 메모리 영역의 가용사항을 N 번 나타내게 되면 시그니처 갱신 제어 모듈은 Hi-DPI 블럭 모듈 과 서브스트링 저장 기억 모듈을 갱신하게 된다. (그림 6)에서는 UFB 갱신 제어 신호가 Hi-DPI 메모리에 메모리 주소 (Q 서브스트링 인식), 입력 데이터, 그리고 enable 신호들로 갱신 하도록 알린다.

3.2.3 시그니처 검색 및 보고 모듈

(그림 7)은 시그니처 검색과 보고서 모듈을 구성하는 내부 모듈들 간의 관계를 그리고 있다. 시그니처 검색과 보고서 모듈은 데이터를 연속적으로 수신하여 Hi-DPI 블럭 모듈과 서브스트링 저장 기억 모듈 안에 저장되어 있는 시그니처와 비교한다. 시그니처 검색과 보고서 모듈은 수신되는 데이터를 시그니처 갱신 모듈에서 시그니처를 갱신하는 방식과 같이 처리하여 비교한다.

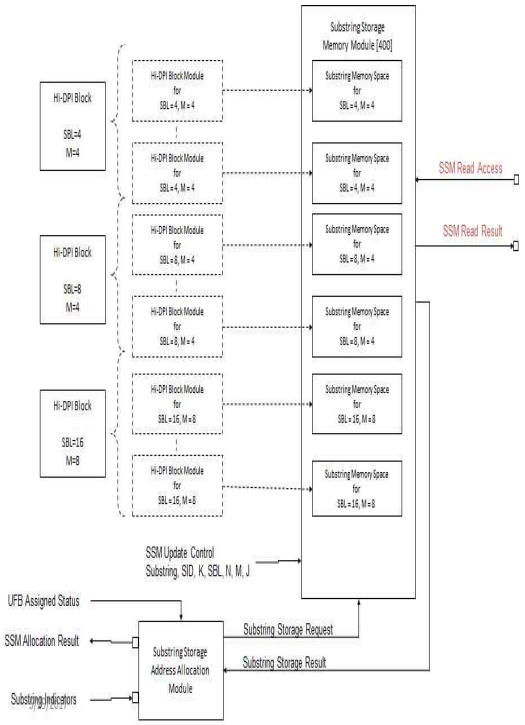


(그림 7) 시그니처 검색 및 보고 모듈

시그니처 검색과 보고서 모듈은 데이터 입력 FIFO 모듈, 데이터 파티션 모듈, 데이터 서브스트링 인식 일반 모듈, Hi-DPI 블럭 매치 모듈, 서브스트링 매치 모듈, 그리고 시그니처 매치 결정 모듈로 구성된다.

3.2.4 시그니처 저장 모듈

(그림 8)은 서브스트링 기억 주소 할당 모듈과 서브스트링 저장 기억 모듈 사이의 관계를 보여주고 있다. Hi-DPI 블럭 선정 모듈은 서브스트링 저장 기억 모듈에서의 사용가능한 메모리 서치 값이 positive 로 나오는 경우 서브스트링 저장 주소 할당 모듈은 서브스트링 저장 결과를 만드는데 이 값은 서브스트링 인식의 해싱 값이다.



(그림 8) 시그니처 저장 모듈

이 주소로 액세스 된 결과 값은 서브스트링 저장 결과인데 서브스트링 정보, SID, K, SBL, N,M 그리고 J 값들이 들어 있다. 메모리 공간이 있는 것으로 판단이 되면 서브스트링 저장 주소 할당 모듈은 positive SSM 할당 결과를 발생하게 된다. 이러한 결과는 시그니처 갱신 제어 모듈에게 시그니처 서브스트링이 서브스트링 저장 기억 모듈에 저장이 될 수 있는 것을 알려준다.

서브스트링 저장 기억 모듈 에서 사용되는 메모리영역은 시그니처의 갯수와 문자를 여러 방식으로 나눌 수 있다. (그림 8)에서는 각 Hi-DPI Block이 할당된 메모리 영역을 사용하고 있는 예를 보이고 있다.

4. 시험 평가

국제공통평가 기준에 따라서 목표 평가 값을 성정하여 <표 1>과 같은 장비를 이용하여 평가하였다.

<표 1> 시험장비 목록

구 분	장 비	수 량	비 고
Hi-DPI	H-DPI적용 DDoS 대응 장비	2 EA	대상 장비
계측기	IXIA XM2	1 EA	IXAutomate license 포함
GiGA Switch	L2/L3 Switch	2 EA	Fiber 2 ports/UTP 2 ports이상
	L2/L3 Switch	1 EA	DDoS 장비 MGT Port 용
PC + 모니터	DDoS 장비 모니터링	1 EA	Windows 7 기반 Laptop PC
	계측기 운영	1 EA	Windows 7 기반 (프로그램 포함)

<표 2>는 평가결과 그리고 평가 방법에 대해서 기술하였다.

<표 2> 평가지표와 평가결과

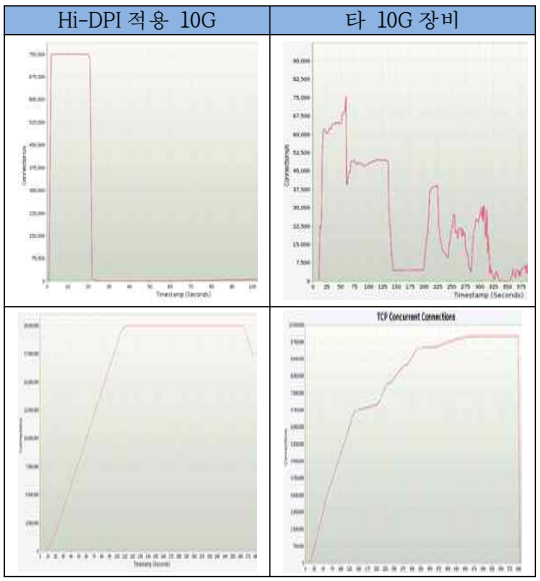
평가항목	세계 수준	국내 수준	목표치	평가결과
1. 처리량	10 Gbps		10 Gbps	10 Gbps
2. 지연시간	80μs 이하	250μs 이상	80μs 이하	20μs
3. 초당연결 생성율	613,000 cps	500,000 cps	1,000,000 cps	1,500,000 cps
4. 초당 최대 패킷 처리량	9,000,000 pps	7,500,000 pps	14,480,000 pps	14,480,950 pps
5. 미탐지율	3% 미만	5% 이상	3% 미만	0.00%
6. 오탐지율	1% 미만	5% 이상	1% 미만	0.00%

① 처리량(Throughput)의 경우 최대치인 초당 10기가비트까지 전송하는 것을 목표로 하며, 측정은 계측장비(Ixia, BP)를 이용하여 CC 기준인 10분간 시험하여 측정하였다. 계측기 IXIA와 시료를 10G portX2(Tx, Rx)로 연결한 뒤, 각 패킷 길이(64, 128, 256, 1024, 1280, 1518 Byte)로 나눠서 측정한 결과 10Gbps의 결과를 얻었다.

② 지연시간(Latency)의 경우 패킷 처리 시 걸리는 시간으로 80μs 미만을 목표로 하며, 측정은 계측장비(Ixia, BP)를 이용하여 CC 기준인 10분간 테스트하여 지연시간을 측정하였다. [그림 2-30]은 계측기 IXIA와 시료를 10G portX2(Tx, Rx)로 연결한 뒤, 각 패킷 길이(64, 128, 256, 1024, 1280, 1518 Byte)로 나눠서 지연시간을 측정한 결과 평균 20μs가 나왔다.

③ 초당 연결 생성율(CPS, Connections Per Second)의 경우 초당 신규 연결 수로 평균 1,000,000 cps를 목표로 하며, 측정은 계측장비(Ixia, BP)를 이용하여 CC 기준인 10분간 테스트하여 평균 신규 세션 연결 수를 측정하였다. HTTP 세션을 BP 장비에서 제공되는 최대값 20,000,000 세션까지 연결 후 400초간 유지하여 초당 연결 생성율을 측정하였다. 테스트 결과 L4의 CPS는 1,500,000 CPS로 목표 보다 좋은 결과를 나타냈다.

BP와 같은 계측기는 TCP 연결율을 측정할 수 있는데 이는 IXIA XM2 또는 N2X 계측기로 단순 패킷만 보내 계측하는 방식이 아닌 실망과 유사하게 세션을 맺고 공격 패킷을 넣어 시험해 볼 수 있다. BP는 양방향의 각기 다른 세션을 맺어 초당 MAX 750,000(2포트 기준 : 계측기마다 다름) 까지 맺고 20,000,000 까지 MAX CPS를 생성해서 시험할 수 있어 가장 실망과 유사하게 시험할 수 있는 장점을 지닌다.



(그림 9) CPS 측정 비교

성능의 중요한 결과 포인트가 TCP 연결율에 대한 그래프 진폭의 변화량이다. [그림 9]을 보면 타 장비의 경우 그래프 진폭 변화가 매우 크고, 7만 이상에서 꺾이는 그래프를 보인다. 이는 세션을 더 이상 맺지 못하고, 세션이 많이 생성되면서 네트워크의 불안정을 가져오는 것을 보여주는 것이다. (그림 9)와 같이 타 동급 장비와 CPS 측정 비교를 한 결과 당사 개발 제품은 CPS 측정치가 일정하게 나온 반면 타사 장비는 불규칙하게 나온 것을 확인하였다.

④ 초당 최대 패킷 처리량(PPS)의 경우 14,480,000을 목표로 하며, 측정은 계측장비(Ixia, BP)를 이용하여 CC 기준인 10분간 테스트하여 평균 신규 세션 연결 수를 측정하였다. 계측기 IXIA와 시료를 10G portX2(Tx, Rx)로 연결한 뒤 초당 최대 패킷 처리량을 측정하였다. 테스트 결과 64bytes 기준 단방향 14,880,950 pps, 양방향: 29,761,900pps로 목표치를 초과한 결과를 얻었다.

5. 결 론

본 논문은 HW기반 고속의 패킷 필터링을 위한 Hi-DPI 알고리즘을 제안하고, 이 기술을 DDoS 대응 장비에 장착하여 그 성능의 우수성을 검증하였다. 이 기술은 FPGA의 병렬성을 최대한 이용하였고, 시그니처의 갱신이 실시간으로 가능하므로 정보보안 제품에 적용하기에 적합함을 입증하였다.

실험을 통해서 DDoS 대응 장비를 설치 운영할 때의 속도 저하의 문제점이 획기적으로 개선되어짐을 검증하였고, FPGA의 하드웨어적인 특성과 병렬처리 특성을 확실하게 증명할 수 있는 Hi-DPI 알고리즘의 실용성을 검증하였다. 향후 Hi-DPI 알고리즘을 활용한 정보보호제품에 대한 연구에 진척이 있을 것으로 기대한다.

참 고 문 헌

[1] Internet Security - Mission Critical, Techpress Inc, 2015.
[2] W.Richard Stevens, "UNIX Networking Programming", Vol.1, 2nd Ed., 1998.
[3] Robert L. Ziegler, "Linux Firewalls", New Riders, 2014.

[4] Sean Walton, "Linux Socket Programming",SMS, 2011.
[5] AnalyZ Demonstration Copy User Guide, Zergo Limited, June 2015.
[6] Welcome to the World of BDSS, and OPA Inc. The Integrated Risk Management Group, OPA Inc., Janury 2016.
[7] ZUM Strarten hier kicken, "IP VPN Solution for Service Provider“ Cisco Systems, 2015.
[8] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, November 2015.
[9] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload(ESP)“ , RFC 2406,November 2015.
[10] Zenkins, Buddy, Security Analysis and Management Manual, Countermeasures Inc, 2014.
[11] Tom Olzak, Improve data protection processes with content discovery, monitoring and filtering, 2007.
[12] Jay Heiser, Understanding data leakage, Gartner, 2007.
[14] R. Erbacher, K. Christensen, and A. Sundberg, “Designing Visualization Capabilities for IDS Challenges,” IEEE Symp. on Information Visualization’s Workshop on Visualization for Computer Security (VizSEC), Oct. 2015.
[15] Kim Jeom Goo, “A Study of Connection Maintenance Techniques using TCP Hijacking”, 융합보안논문지, 2014.
[16] 김점구, 이도현, “리눅스 기반 침입 방지를 위한 로그 분석 방법 연구”, 융합보안논문지, 2015.
[17] 김점구, 노시춘, “네트워크 보안 환경에서의 현장적용 중심 암호품질 만족도 평가 메트릭스 설계 프로세스”, 융합보안논문지, 2015.
[18] 한국 전자통신 연구원, ESM 개발 동향, 2013.5.
[19] 한국정보보호진흥원, “국제공통평가기준(v.3.1)", 2015.
[20] 펜타 시큐리티 시스템(주), “자동화된 위험분석 툴의 구현”, 2015.

[저 자 소 개]

김 점 구 (Jeom Goo Kim)
1990년 2월 광운대학교 전자계산학과 이학사
1997년 8월 광운대학교 전자계산학과 석사
2000년 8월 한남대학교 컴퓨터공학 박사
1999년 3월~ 현재 남서울대학교 컴퓨터학과 교수
IT융합연구소장
email : jgoo@nsu.ac.kr