한국ITS학회논문지
제11권, 제4호 (2012년 8월)
pp.121~130

40G급 Aladdin 시스템의 모델링 및 입력 트래픽에 따른 시스템 구조 효율성 연구

The Study of Efficiency of System Architecture According to the Modeling of 40G Aladdin System and Input Traffic

황 유 동* (Yu-Dong Hwang) 박 동 규** (Dong-Gue Park) 장 종 수*** (Jong-Soo Jang)

요 약

본 논문에서는 고속의 DDoS 방어도구로 한국 전자통신 연구원에서 개발 중인 40G bps급 Aladdin 시스템의 성능 평가를 위하여 패트리네트로 Aladdin 시스템을 모델링하였고, 이를 기반으로 Aladdin 시스템의 입력 트래픽에 따른 시스템 구조 효율성 분석을 수행하였다.

Abstract

In this paper, the structure of the Aladdin system was modeled by Petri nets for performance evaluation of 40G bps class Aladdin system which was developed in ETRI as the high-speed DDoS defensive tool. The efficiency analysis of the system architecture according to the input traffic of the Aladdin system was performed based on the modeling.

Key words : DDoS, Aladdin System, Petri nets

Ⅰ. 서 론

최근 7.7 DDoS와 3.4 DDoS 대란과 같이 대규모 봇 넷을 이용한 대규모 DDoS의 위협이 점차 증가하고 있다. 따라서 대규모 DDoS 공격에 빠르게 대처할 수 있는 고속의 DDoS 방어 도구에 대한 연구가 절실히 필요한 상황이며, 이에 따라 최근 한국전자통신연구원에서는 고속의 DDoS 방어 도구로 40G bps급 Aladdin 시스템을 개발하고 있다.

본 연구에서는 Aladdin 시스템 중에서 핵심요소인 loadbalancer와 Anti-DDos엔진 사이의 동작을 패트리네트로 모델링하여 입력 트래픽에 대한 시스템의 성능을 분석하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 Aladdin 시스템의 성능 평가 연구와 시스템의 구조 효율성 분석에 대한 연구를 위하여 Aladdin 시스템을 모델

* 주저자 : 순천향대학교 전기전자공학과대학원 정보보호전공 박사과정
** 공저자 및 교신저자 : 순천향대학교 정보통신공학과 교수
*** 공저자 : 한국전자통신연구원 책임연구원
† 논문접수일 : 2012년 5월 25일
† 논문심사일 : 2012년 6월 29일
† 게재확정일 : 2012년 8월 13일

링 할 수 있는 패트리네트에 대해 살펴보고, 3장에서는 Aladdin 시스템을 패트리네트를 이용하여 모델링하여 시뮬레이션하고, 4장에서는 시뮬레이션 결과 및 성능 분석을 한 후, 5장에서 결론을 내린다.

Ⅱ. 관련 연구

1. 패트리 네트〔9〕

패트리 네트[9] 이론은 1962년 Carl. Adam. Petri가 통신 시스템을 모델링하고 해석하기 위해서 네트이론 방법을 개발한 것으로, 동시성, 비동기적인 요소들 간의 상호 교류에 대한 특징을 갖는 시스템에 대해서 아주 유용하다.

패트리 네트는 현재 많은 이산적 사건 시스템 분야에 매우 적극적으로 사용되고 있으며, 패트리 네트는 시스템에서의 비동기적이고 불확실한 이산적인 사건을 모델링하고, 모니터하고 분석하는데 매우 유용하다. 따라서, 병렬 시스템이나 통신 프로토콜, 유연 생산 시스템과 같이 병행으로 일어나는 시스템을 모델링하고 분석하는데 사용될 수 있다.

패트리 네트는 많은 시스템의 모델링에 사용되는 그래픽 형이면서 수학적인 모델링 도구로 수학적인 면으로는, 상태방정식이나, 산술적 방정식, 또는 기타 다른 수학적인 형태를 이용하여 시스템의 상태를 검증하거나 분석 할 수가 있다. 그래픽 형으로 패트리 네트는 플로우 차트나, 블록 다이어 그램이나, 네트워크 형태로 표현되어 사용되고 있다. 무엇보다도, 이러한 모델링에서 토큰의 흐름을 통하여 생동적이고 병행적인 시스템의 흐름을 시뮬레이션 할 수 있는 장점을 갖는다.

패트리 네트의 장점 중의 하나는 모델을 통하여 시스템의 성질을 분석할 수 있다는 것이다. 이러한 성질 중에는 패트리 네트가 가지는 독특한 성질, 즉 도달성(Reachability), 안전성(Boundedness)과 생존성(Liveness)등을 이용하여 시스템의 분석 및 검증이 가능하다. 이러한 패트리 네트에 트랜지션과 플레이스, 점화시간이라는 개념을 부가하여서 처리 시간을 분석 하는 형태를 타임 패트리 네트(Time Petri nets)이라고 부르며, 이는 스케쥴링 분석을 시작으로 각종 병행 처리나, 동시 처리성이 있는 모델링 분석 및 검증에 적극적으로 활용 되고 있다.

패트리 네트는 다양한 시스템에 적용할 수 있는 수학적이며 그래프 개념을 이용하여 그 흐름 활동을 표현하고 분석 할 수 있는 하나의 그림형태로 표현한 모델링 도구이다. 패트리 네트는 병행적, 비동기적, 분산적, 병렬적 또는 확률적 성질을 지닌 정보처리 시스템의 성능 분석, 통신 프로토콜의 설계 검증 및 FMS등 다양한 분야에 이용되고 있다.

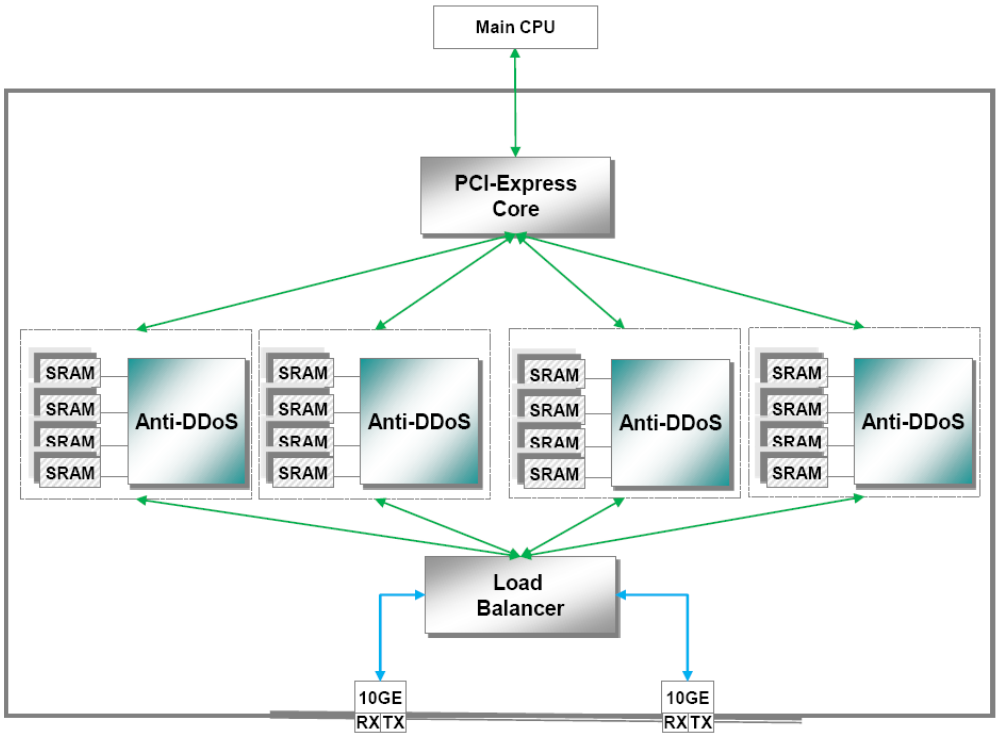
그림 형태로 표현한 모델링 도구 측면에서 볼 때 패트리 네트는 플로 차트나 블록 다이아그램과 유사하다. 패트리 네트는 시스템의 상태(state) 혹은 조건(condition)을 나타내는 플레이스(place), 행동(event)을 나타내는 트랜지션(transition), 흐름을 나타내는 아크(arc) 및 플레이스 조건의 진위 또는 시스템의 가용 자원을 나타내는 토큰(token)으로 구성되어 있으며, 토큰은 시스템의 동적이며 병행적 동작 특성을 나타내기 위해 사용된다. 패트리 네트의 분석은 패트리 네트의 특성인 생존성(liveness), 유한성(boundedness), 보존성(persistence) 및 발달가능성(reachability)과 행렬방정식(matrix equation)등으로 시스템을 분석 할 수 있다.

Ⅲ. Aladdin 시스템 모델링

1. 패트리 네트를 사용한 Aladdin 시스템 모델링

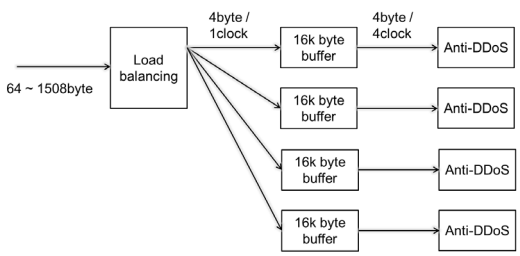
Aladdin 시스템[1,2]은 한국 전자통신연구소에서 실시간 트래픽 분석 및 처리를 통해 DDoS 공격을 탐지 및 대응하는 시스템으로 개발되었다. 개발된 Aladdin 시스템의 내부 구조는 다음 <그림 1>과 같다.

<그림 1>과 같은 Aladdin 시스템의 구조 효율성 분석을 위하여 Aladdin 시스템의 구조에 대한 성능 평가 연구가 반드시 수반되어야 한다. 본 연구에서는 Aladdin 시스템 중에서 핵심요소인 loadbalancer와 Anti-DDos엔진 사이의 동작을 패트리 네트로 모델링하여 입력 트래픽에 대한 시스템의 성능을 분석하고자 한다.



〈그림 1〉 Aladdin 시스템의 내부구조
〈Fig. 1〉 The internal fabric of the Aladdin system

패트리 네트를 사용하여 Aladdin 시스템의 성능을 분석하기 위하여 먼저 Aladdin 시스템의 loadbalancer 와 Anti-DDos 엔진의 동작을 시간 패트리 네트의 플레이스와 트랜지션으로 모델링한다.



〈그림 2〉 Aladdin 시스템의 load balancing 블록 다이어그램
〈Fig. 2〉 The load balancing block diagram of the Aladdin system

본 논문에서는 패트리 네트로 모델링하기 위하여 위 <그림 1>의 Aladdin 시스템의 내부 구조를 다음 <그림 2>와 같이 블록 다이어그램으로 표현하였다.

입력 트래픽은 64byte ~ 1,508byte 사이의 크기로 랜덤하게 수신되고, 입력되는 트래픽은 4byte 패킷으로 16개 ~ 377개의 패킷단위로 수신된다.

Load balancer 는 입력 트래픽을 1클럭당 1개의 패킷(4byte)을 버퍼로 전송하고 버퍼에 저장된 패킷은 4클럭 당 1개의 패킷을 Anti-DDoS로 전송한다.

2. 패트리 네트를 사용한 Aladdin 시스템 시뮬레이션

본 논문에서는 한국 전자통신 연구소에서 개발하고 DDoS 대응 및 탐지 시스템인 40G bps급 Aladdin 시스템의 모델링 및 성능 분석을 위하여 Aladdin 시스템의 핵심 구조인 loadbalancer 와 Anti-DDos 엔진의 동작을 패트리 네트로 모델링하고 시뮬레이션을 수행한다. Aladdin 시스템의 내부 요소들을 패트리 네트의 플레이스와 트랜지션으로

모델링하고 입력 트래픽을 토큰으로 모델링하여 시뮬레이션을 하였다.

위 <그림 1>의 Aladdin 시스템의 내부구조를 위 <그림 2>와 같이 블록 다이어그램으로 표현 할 수 있고, 이 블록 다이어그램을 이용하여 다음 <그림 3>과 같은 패트리 네트를 모델링 할 수 있다.

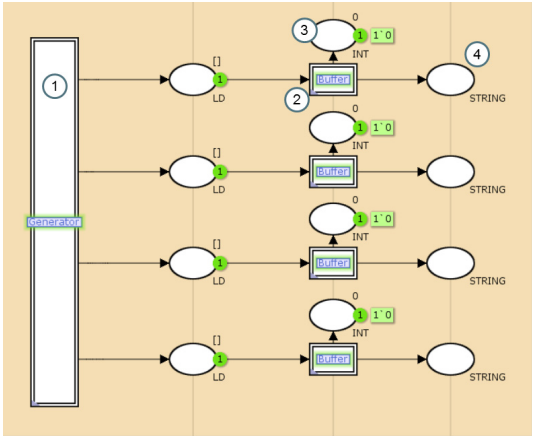
다음 <그림 3>의 각 부분에 대한 설명은 다음과 같다.

① 입력 트래픽을 생성하고 버퍼로 분배하는 generator 서브 네트 : Aladdin 시스템의 loadbalancer에 해당 한다.

② 입력된 패킷이 저장되는 버퍼 서브 네트 : Aladdin 시스템 내부의 4개 Anti-DDoS 엔진에 연결된 각 버퍼에 해당 한다.

③ 버퍼에 저장된 패킷의 개수를 확인하는 플레이스

④ Anti-DDoS 시스템을 의미하는 플레이스



〈그림 3〉 Aladdin 시스템의 패트리 네트 모델링
〈Fig. 3〉 The petri nets modeling of the Aladdin system

위 <그림 3>의 패트리 네트는 다음과 같이 동작한다.

1) 위 <그림 3>에서 ①의 generator 서브 네트에서 랜덤한 크기(64byte ~ 1508byte : 16packet ~ 377packet)의 패킷을 생성한다.
2) 생성된 트래픽은 1클럭당 1패킷(4Byte)의 단위로 플레이스로 전송된다.
3) 각 플레이스로 전송된 입력 트래픽은 ②의 버퍼 서브네트에 저장된다.
4) ③의 플레이스에서 ②의 버퍼 서브네트에 저장된 패킷의 개수를 확인한다.
5) 버퍼에 저장된 패킷은 4클럭 당 1 패킷의 단위로 ④의 플레이스에 해당하는 Anti-DDoS 시스템으로 전송된다.

본 논문에서는 다양한 환경에서 동작해야할 Aladdin 시스템 성능을 분석하기 위하여 가상 보안존을 적용하여 load balancer가 입력 트래픽을 버퍼로 전송할 때 전송될 버퍼를 선정하는 방법에 따라 다음의 세 가지 방법을 패트리 네트로 모델링하여 시험하였다.

1) 트래픽이 수신되면 버퍼를 랜덤하게 선택하여 전송.
2) 트래픽이 수신되면 각 버퍼에 저장되어 있는 데이터의 양을 비교하여 가장 작은 데이터가 저장되어 있는 버퍼로 전송.
3) 트래픽이 수신되면 그 시점까지 각 버퍼에 전송된 패킷의 개수가 가장 작은 버퍼로 전송.

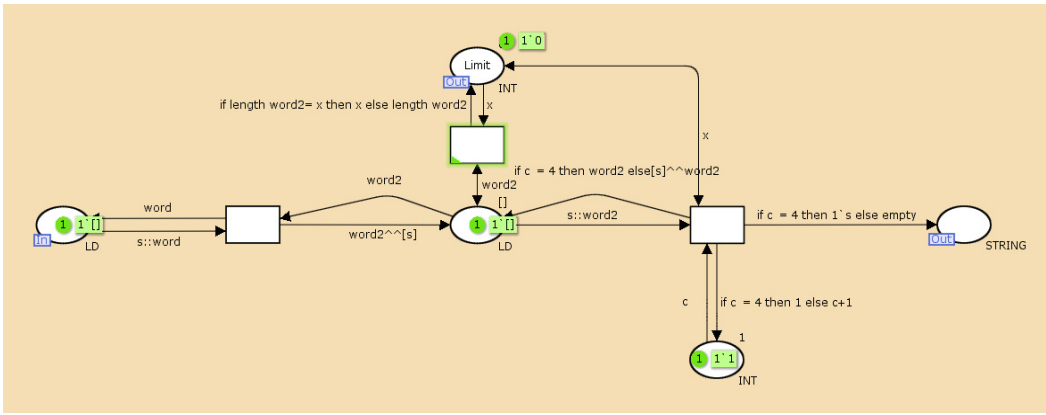
위 세 가지 방법을 적용한 패트리 네트 모델을 비교하면, 입력 트래픽을 어떤 버퍼로 전송할 것인지 버퍼를 선택하는 부분만이 다르고 패킷 생성, 버퍼 등은 모두 동일하다.

따라서, 본 연구에서는 서브 네트를 모델링하여 패트리 네트를 구성하였고, 입력 트래픽의 생성, 버퍼로의 분배를 하는 generator 서브 네트와 수신된 패킷을 저장하는 버퍼 서브 네트를 모델링하였다.

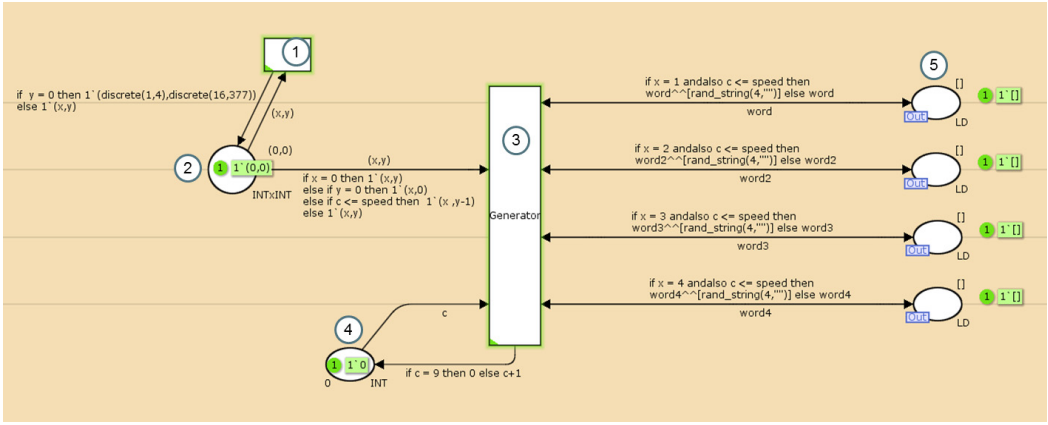
위 세 가지 시험 방법에 모두 적용이 되는 메인 패트리 네트 모델은 위 <그림 3>과 같다.

다음 <그림 4>는 세가지 시험 방법에 따른 패트리 네트 모델에 모두 사용되는 버퍼 서브 네트이다.

버퍼는 FIFO (First In First Out) 구조로 이루어져 있으며, 현재 버퍼에 저장된 패킷의 개수를 count 한다. Aladdin 시스템에서 버퍼는 16k byte 크기를 가지므로, 버퍼에 저장된 패킷(4byte)의 갯수가



〈그림 4〉 버퍼 서브 네트
〈Fig. 4〉 The buffer sub nets



〈그림 5〉 generator 서브 네트 - 버퍼 랜덤 선택
〈Fig. 5〉 The generator sub nets - buffer random selection

4,000개가 넘어가면 버퍼 오버플로우가 발생한 것으로 간주한다.

서브 네트 중 generator 서브 네트에 따라 위 세 가지 시험 방법을 적용할 수 있다. 시험 방법에 따른 세가지 generator 서브 네트는 다음과 같다.

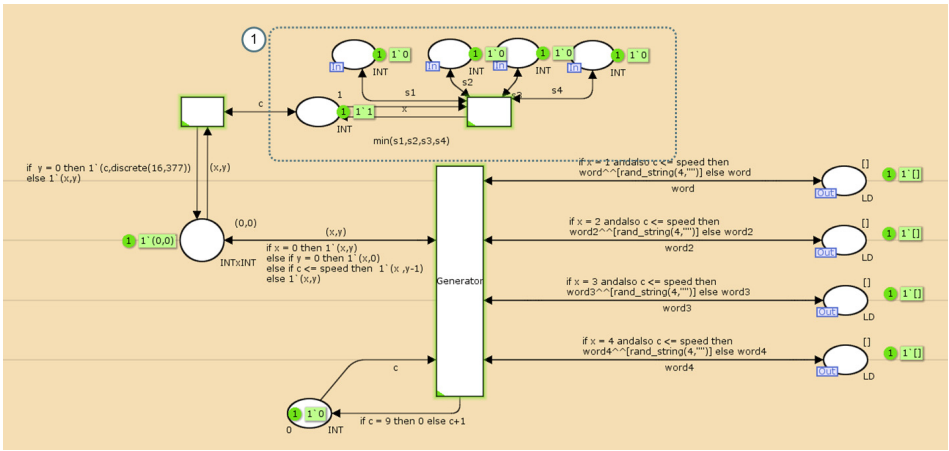
1) 버퍼를 랜덤하게 선택

Aladdin 시스템의 loadbalancer 가 입력된 트래픽을 전송할 버퍼를 랜덤하게 선택하여 전송하는 generator 서브 네트는 <그림 5>와 같다.

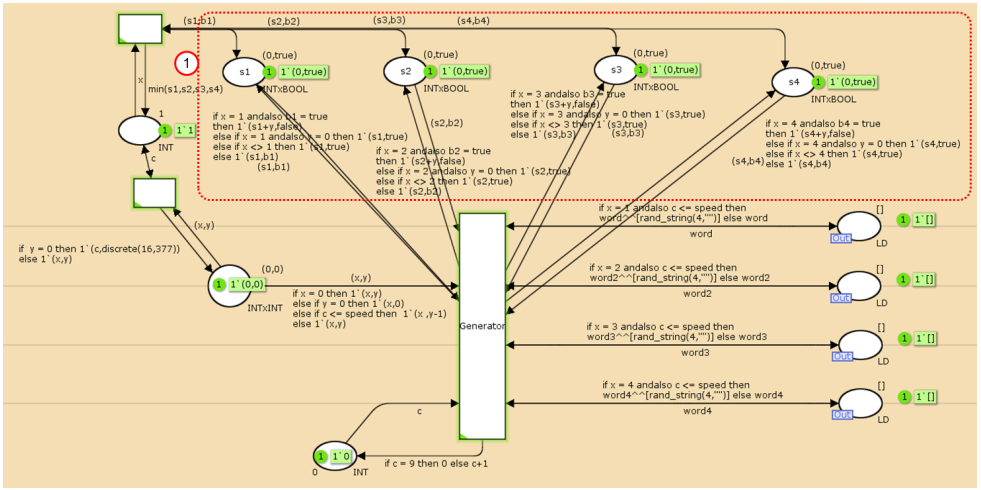
<그림 5>의 generator 서브 네트 동작 과정은 다음과 같다.

1) 입력 트래픽이 전송될 버퍼를 1부터 4중 랜덤하게 생성하여 선택하고, 버퍼로 전송될 입력 트래픽을 16 ~ 377 패킷의 크기로 랜덤하게 생성
2) 1)에서 생성된 입력 트래픽 데이터를 1클럭 당 1패킷의 단위로 전송
3) 1)에서 선택된 버퍼로 수신된 패킷을 전송
4) 패킷을 전송하기 전 패킷 전송 속도 제어.
5) 패킷을 버퍼로 전송.

본 연구에서는 패트리 네트 모델을 다양한 환경에서 시험하기 위하여 입력되는 패킷의 속도를 조절한다. 속도는 패킷이 실시간으로 연속적으로 입



〈그림 6〉 generator 서브 네트 - 버퍼에 저장된 패킷 개수가 가장 작은 버퍼 선택
〈Fig. 6〉 The generator sub nets - buffer select in which the packet count saved in the buffer is the smallest



〈그림 7〉 generator 서브 네트 - 버퍼로 전송된 패킷 개수가 가장 작은 버퍼 선택
〈Fig. 7〉 The generator sub nets - buffer select in which the transmitted packet count is the smallest

력되는 경우를 100%로 설정하고 10 ~ 100%까지 10% 단위로 시험하였다.

2) 버퍼에 저장된 패킷의 개수가 가장 작은 버퍼를 선택

Aladdin 시스템의 loadbalancer 가 버퍼에 저장된 패킷의 개수를 비교하여 가장 작은 개수의 패킷이 저장된 버퍼를 선택하여 입력 트래픽을 전송하는 generator 서브 네트는 <그림 6>과 같다.

<그림 6>의 저장된 패킷 개수가 가장 작은 버퍼를 선택하는generator 서브 네트와 <그림 5>의 버퍼를 랜덤하게 선택하는 generator 서브 네트의 차이는 <그림 6>의 ①부분이다.

①에서 네개의 플레이스를 이용하여 각 버퍼로부터 저장된 패킷의 수를 가져오고 비교하여 저장된 패킷의 수가 가장 작은 버퍼를 선택한 다음, 입

력되는 트래픽을 생성하여 버퍼로 전송한다.

입력 트래픽이 전송되어야 하는 버퍼를 선택하는 과정을 제외한 나머지 과정은 위의 버퍼를 랜덤하게 선택하는 패트리 네트와 동일하다.

3) 버퍼로 전송된 패킷의 개수가 가장 작은 버퍼를 선택

Aladdin 시스템의 loadbalancer 가 각 버퍼로 전송한 패킷의 개수를 비교하여 가장 작은 개수의 패킷이 전송된 버퍼를 선택하여 입력 트래픽을 전송하는 generator 서브 네트는 <그림 7>과 같다.

<그림 7>의 버퍼로 전송된 패킷 개수가 가장 작은 버퍼를 선택하는generator 서브 네트와 <그림 5>의 버퍼를 랜덤하게 선택하는 generator 서브 네트의 차이는 <그림 7>의 ①부분이다. ①에서 네개의 플레이스를 이용하여 각 버퍼로 전송된 패킷의 수를 가져오고 비교하여 전송된 패킷의 수가 가장 작은 버퍼를 선택한 다음, 입력되는 트래픽을 생성하여 버퍼로 전송한다.

입력 트래픽이 전송되어야 하는 버퍼를 선택하는 과정을 제외한 나머지 과정은 위의 버퍼를 랜덤하게 선택하는 패트리 네트와 동일하다.

Ⅳ. 시뮬레이션 결과 및 성능 분석

본 논문에서는 한국 전자통신 연구소에서 개발한 DDoS 대응 및 탐지 시스템인 40G bps급 Aladdin 시스템의 모델링 및 성능 분석을 위하여 Aladdin 시스템의 핵심 구조인 loadbalancer 와 Anti-DDos 엔진의 동작을 패트리 네트로 모델링하고 시뮬레이션을 수행하였다. Aladdin 시스템의 내부 요소들을 패트리 네트의 플레이스와 트랜지션으로 모델링하고 입력 트래픽을 토큰으로 모델링하여 시뮬레이션을 수행하여 전체 시스템을 모델링하였으며, 성능을 분석하였다. 그리고 이를 바탕으로 40G bps급 Aladdin 시스템의 성능을 개선하기 위하여 다양한 환경에서 시스템의 성능을 분석하였다.

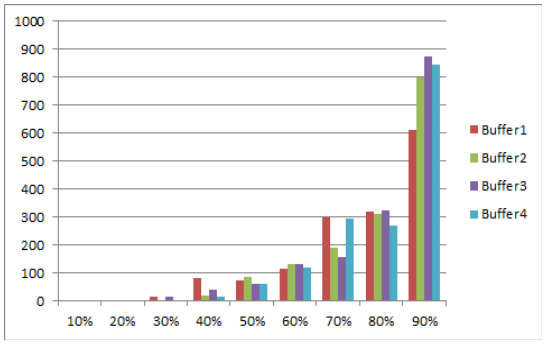
Load balancer의 성능을 분석하기 위하여 입력 트래픽이 실시간으로 발생하고 전송되는 경우를 100% 의 속도로 가정하고 이 속도를 점차 낮추어 10% ~ 100% 까지(10% 단위로 증가)의 속도로 시험을 수행하였다.

패트리 네트에서 입력 트래픽을 발생시킬 때 정확한 속도로 발생 시킬 수가 없으므로, 입력 트래픽 생성에 아무런 조건이 없는 경우를 100%의 속도로 하였고, 입력 트래픽 생성에 조건을 부가하여 10개 생성하여 1개 사용 시 10%의 속도(10% 단위로 증가)로 하였다.

〈표 1〉 버퍼의 랜덤 선택 - 100% 속도
〈Table 1〉 The random selection of the buffer -100% speed

회차	저장된 패킷의 수				버퍼 오버플로우 도달시간 (msec)
	Buffer1	Buffer2	Buffer3	Buffer4	
1	4176	690	1571	0	220,000
2	3037	2182	4162	0	340,000
3	1907	1432	4206	0	260,000
4	0	4009	633	115	627,000
5	145	1379	1118	4091	1,010,000
6	137	4042	804	1883	1,200,000
7	4698	3130	285	0	300,000
8	4575	1994	876	32	750,000
9	4557	1497	3289	0	300,000
10	4445	411	2693	477	2,800,000

다음 <그림 8>은 버퍼의 랜덤 선택 패트리 네트의 속도별 시험 결과이다.



〈그림 8〉 버퍼의 랜덤 선택 패트리 네트의 속도별 시험 결과
〈Fig. 8〉 The test result by speed of petri nets having the random selection of the buffer

위 <그림 8>의 세로축은 버퍼에 저장된 패킷의 개수이고, 가로축은 입력 트래픽이 전송되는 속도이고, 시험은 20,000,000 msec 동안 수행하였다.

다음 <그림 8>을 보면 속도가 증가 할수록 버퍼에 쌓이는 패킷의 개수가 증가함을 알 수 있다.

위 <표 1>에서와 같이 속도 100%(입력 트래픽이 실시간으로 발생하는 상태)에서는 버퍼 오버플로우를 확인할 수 있었으나, 10 ~ 90%의 속도에서는 버퍼 오버플로우를 쉽게 확인할 수가 없었다.

다음 <표 2>는 시험 결과를 이용하여 버퍼 오버플로우에 도달할 step을 예상하였다.

〈표 2〉 버퍼의 랜덤 선택-속도별 버퍼 오버플로우 도달 예상 시간

〈Table 2〉 The number of packet stored in the buffer -expectation time of buffer-overflow by speed

속도	10%	20%	30%	40%	50%
버퍼오버플로우 도달 예상 시간 (msec)	2700억	890억	51억	9억 6천만	9억 3천만
속도	60%	70%	80%	90%	100%
버퍼오버플로우 도달 예상 시간 (msec)	6억	2억 7천만	2억 5천만	4천 6백만	7십8만

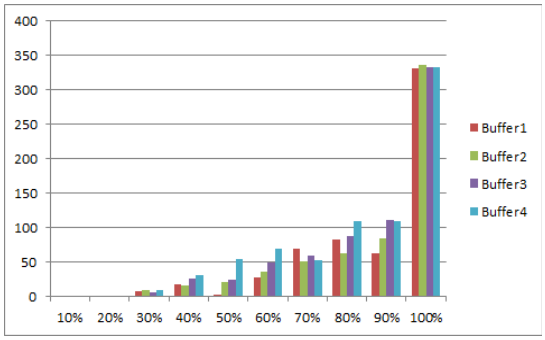
다음 <그림 9>는 버퍼에 저장된 패킷의 수를 비교하여 가장 작은 수의 패킷이 저장된 버퍼로 패킷을 분배하는 모델의 시험 결과이다.

〈표 3〉 버퍼에 저장된 패킷의 수 - 속도별 버퍼 오버플로우 도달 예상 시간

〈Table 3〉 The number of packet stored in the buffer -buffer-overflow reach expectation time by speed

속도	10%	20%	30%	40%	50%
버퍼오버플로우 도달 예상 시간 (msec)	2000억	1000억	84억 2천만	26억 4천만	14억 9천만
속도	60%	70%	80%	90%	100%
버퍼오버플로우 도달 예상 시간 (msec)	11억 6천만	11억 7천만	7억 3천만	7억 2천 5백만	1억 1천만

다음 <표 3>은 위 <그림 9>의 시험 결과를 이용하여 버퍼 오버플로우에 도달할 시간을 예상하였다. 위 표 2와 마찬가지로 속도가 증가할수록 버퍼에 저장되는 패킷의 수가 증가함을 알 수 있다.



〈그림 9〉 버퍼에 저장된 패킷의 수 비교 선택 패트리 네트의 속도별 시험 결과

〈Fig. 9〉 The test result by speed of petri nets having selection of comparing number of the packet saved in the buffer

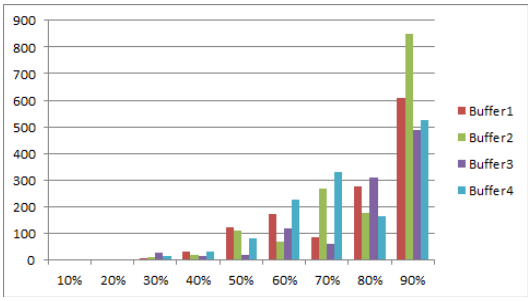
다음 <그림 10>은 버퍼로 전송된 패킷의 수를 비교하여 가장 작은 수의 패킷이 전송된 버퍼로 패킷을 분배하는 모델의 시험 결과이다.

다음 <표 4>는 시험 결과를 이용하여 버퍼 오버플로우에 도달할 step을 예상하였다. 위 <표 2, 3>과 마찬가지로 속도가 증가할수록 버퍼에 저장되는 패킷의 수가 증가함을 알 수 있다.

〈표 4〉 버퍼로 전송된 패킷의 수 - 속도별 버퍼 오버플로우 도달 예상 시간

〈Table 4〉 The number of transmitted packet to the buffer - expectation time of buffer-overflow by speed

속도	10%	20%	30%	40%	50%
버퍼오버플로우 도달 예상 시간 (msec)	1600억	1142억 9천만	27억 6천만	24억 4천만	6억 5천만
속도	60%	70%	80%	90%	100%
버퍼오버플로우 도달 예상 시간 (msec)	3억 5천만	2억 4천만	1억 3천만	9천 4백만	150만



〈그림 10〉 버퍼로 전송된 패킷의 수 비교 선택 패트리 네트의 속도별 시험 결과

〈Fig.10〉 The test result by speed of petri nets having selection of comparing number of the packet transmitted to the buffer

위 세 가지 패트리 네트 모델의 시험 결과로, 패킷이 전송될 버퍼를 랜덤으로 선택하여 패킷을 분배하는 시험 결과가 제일 좋지 않음을 알 수 있었고, 세 가지 패트리 네트 모델 중 가장 좋은 성능을 보인 모델은 버퍼에 저장되어 있는 패킷의 수를 비교하여 가장 작은 수가 저장되어있는 버퍼로 패킷을 우선 분배하는 모델이다.

버퍼로 전송된 패킷의 수를 비교하여 가장 작은 수가 전송된 버퍼로 패킷을 우선 분배하는 모델은 4개의 버퍼로 균등하게 패킷이 분배되지 못하는 단점이 있다.

V. 결 론

한국 전자통신연구소에서는 최근 빈번하게 발생하고 있는 각종 네트워크 피해 사고에 대응하기 위하여 실시간 트래픽 분석 및 처리를 통해 DDoS 공격을 탐지 및 대응하는 Aladdin을 개발하였고 현재 40G bps급 초고속 시스템을 개발하고 있는 중이다.

개발 중인 Aladdin은 성능 평가 연구를 통하여 시스템의 구조 효율성 분석에 대한 연구가 수반되어야 한다. 이를 위하여 본 연구에서는 loadbalancer와 Anti-DDos엔진 사이의 동작을 패트리 네트로 모델링하여 입력 트래픽에 대한 시스템의 성능을 연구하고 이를 기반으로 Aladdin 시스템의 구조에 대한 효율성 분석을 수행하였다.

4장에서 입력된 트래픽이 전송될 버퍼를 랜덤하게 선택하는 모델, 버퍼에 저장된 패킷의 수를 비교하여 가장 작은 수가 저장된 버퍼를 선택하는 모델과 버퍼로 전송된 패킷의 수를 비교하여 가장 작은 수가 전송된 버퍼를 선택하는 모델을 패트리 네트로 모델링하여 성능을 분석하였다. 또한 이 세 가지 모델 중 버퍼에 저장된 패킷의 수를 비교하여 가장 작은 수가 저장된 버퍼를 선택하는 모델이 가장 좋은 성능을 보임을 알 수 있었다.

이러한 연구 결과를 바탕으로 virtual security zone을 구성하여 load balancing에 적용하면 보다 높은 성능 향상이 있을 것으로 예상되며, 이에 따라 지속적인 연구가 필요할 것으로 사료된다.

참고문헌

[1] 유승엽, 박동규, 오진태, 전인오, “ALADDIN의 어플리케이션 계층 공격 탐지 블록 ALAB 알고리즘의 최적 임계값 도출 및 알고리즘 확장”, 정보처리학회지 제18-C권 제 3호, 2011

[2] 유승엽, 박동규, 장종수, “URI 및 브라우저 행동 패턴의 특성을 이용한 HTTP get flooding 공격 탐지 알고리즘”, 한국정보기술학회지 제9권 제1호 159p ~ 170p, 2011

[3] Markus J, Zulfikar R., “Crimeware: Understanding New Attacks and Defenses”, Addison Wesley Professional, ISBN 0-321-50195-0, April 2008.

[4] Tuncer, T. and Tatar, Y. 2008 Detection SYN Flooding Attacks Using Fuzzy Logic. Proc. Int. Conf. Information Security and Assurance ISA’08, Washington, DC, USA, April 24-26, p. 321-325. IEEE Computer Society, NewYork, NY, USA

[5] U.Payer, M.Lamberger, and P.Teufl. Hybrid engine for polymorphic code detection. In Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment, pages 19-31

[6] Takeshi Yatagai, et al., “A HTTP Flooding Detection Method Based on Browser Behavior”, 2007.

[7] Jinghe Jin, Nazarov Nodir, Chaetae Im, and Seung

Yeob Nam, “Mitigating HTTP GET Flooding Attacks through Modified NetFPGA Reference Router”, 1st Asia NetFPGA Developers Workshop, Daejeon, Korea, June 14, 2010.

[8] KINDER, J., KATZENBEISSER, S., SCHALLHART, C., and VEITH, H. “Detecting Malicious Code by Model Checking” In Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), 2005.

[9] TADAO MURATA, “Petri Nets:Properties, Analysis and Applications”, PROCEEDING OF THE IEEE, VOL. 77, NO. 4, APRIL 1989

[10] D.A.Zaitsev, T.R.Shmeleva, “Simulating of Telecommunication Systems with CPN Tools”, Nov. 2006

[11] http://www.cpntools.org/

저자소개

황 유 동 (Hwang, Yu-Dong)

2003년 : 순천향대학교 박사과정 수료(정보보호전공)
2001년 3월 ~ 현 재 : 순천향대학교 전기전자공학과 박사과정(정보보호전공)
1998년 ~ 2000년 : 순천향대학교 전기전자공학과 공학석사(정보통신전공)

박 동 규 (Park, Dong-Gue)

1992년 : 한양대학교 박사(전자공학전공)
2004년 ~ 현 재 : 순천향대학교 정보통신공학과 교수
1999년 ~ 2003년 : 순천향대학교 정보기술공학부 부교수

장 종 수 (Jang, Jong-Soo)

2000년 : 충북대학교 박사(컴퓨터공학전공)
1989년 7월 ~ 현 재 : 한국전자통신연구원 책임연구원
2004년 ~ 2008년 : 한국전자통신연구원 네트워크 보안그룹 그룹장
2000년 ~ 2003년 : 한국전자통신연구원 네트워크 보안구조팀 팀장
2004년 ~ 현 재 : 한국정보보호학회 이사(부회장)