Journal of Korea Multimedia Society Vol. 9, No. 3, March 2006(pp.333-341)

멱함수 네트워크 특성을 이용한 랜덤확산형 웜의 동적 제어

노병규[†], 박두순[††]

요 약

최근의 웜은 CPU 자원, 네트워크 대역폭등 주어진 자원을 최대한 소모하여 네트워크 전체 가용성을 심각히 저해하는 랜덤확산형(Random Constant Spreading) 웜이 점차 늘어나고 추세이다. 본 논문에서는 이러한 웜의 확산을 동적으로 억제하기 위하여 선호적 성장 특성을 가지는 멱함수 네트워크를 분석한다. 그리고 이러한 네트워크에서 공통적으로 나타나는 전달노드의 깊이분포 특성을 이용하여 랜덤확산형 웜을 동적으로 제어하는 모델을 제안하고 시뮬레이션을 통하여 각 노드의 부하가 최소화되면서 웜 확산이 효과적으로 제어됨을 검증한다.

Dynamic Control of Random Constant Spreading Worm Using the Power-Law Network Characteristic

Byung-Gyu No[†], Doo-soon Park[††]

ABSTRACT

Recently, Random Constant Spreading worm is increasing. The worm retards the availability of the overall network by exhausting resources such as CPU resource and network bandwidth, and damages to an uninfected system as well as an infected system. This paper analyzes the Power-Law network which possesses the preferential characteristics to restrain the worm from spreading. Moreover, this paper suggests the model which dynamically controls the spread of the worm using information about depth distribution of the delivery node which can be seen commonly in such network. It has also verified that the load for each node was minimized at the optimal depth to effectively restrain the spread of the worm by a simulation.

Key words: Worm(웜), Random Constant Spreading(랜덤확산), Dynamic Network(동적 네트워크), Power-Law(멱함수), Bandwidth Control(대역폭 제어)

1. 서 론

급변하는 정보화 사회에서 정보전달에 대한 신뢰성이 더욱 더 중요해지고 있는 반면 정보유출, 정보의 위·변조, 웜·바이러스 및 해킹과 같은 정보화 역기능도 동시에 늘어나고 있다.

1982년 제록스연구소의 허프에 의해 "웜"이라는 단어가 최초로 사용된 이후[1] 다양한 웜들이 출현하여 인터넷의 발전에 심각한 위협이 되고 있다. 코드레드나 님다의 경우는 수백만 대나 되는 시스템을 감염시켰으며[2], 슬래머 웜의 경우 10분이라는 짧은 시간에 전 세계 감염 가능한 호스트의 90%가 감염되었고, 국내의 경우 약 12시간 이상 인터넷이 마비되는 현상까지 발생하였다[3].

※ 교신저자(Corresponding Author): 노병규, 주소 : 서울시 송파구 가락동 78 IT 벤처타워 한국정보보호진흥원(138-803), 전화 : 02)405-5500, FAX : 02)405-5519, E-mail : nono@kisa.or.kr

접수일 : 2005년 10월 6일, 완료일 : 2005년 10월 24일
[†] 정회원, 한국정보보호진흥원 보안성평가단 단장
[††] 종신회원, 순천향대학교 정보기술공학부 교수
(E-mail: parkds@sch.ac.kr)

최근 웜 추세의 첫 번째 동향은 시스템 취약점이 발표된 후 이를 이용한 해킹이 이루어지는 시간이 점차 짧아지는 제로-데이(Zero-Day) 위협이다[4]. 2002년 슬래머 웜의 경우 6개월에서 2005년 조톱의 경우 4일로 줄어들어 대처시간이 점차 부족해지고 있다.

두 번째는 기존에는 사용자 부주의로 인해 전염하는 형태에서 최근에는 시스템 취약성만 존재하면 자기증식코드(Self Propagation Code)를 이용하여 스스로 복제가 되는 형태로 발전하고 있다[5].

자기증식코드 웜은 대부분 랜덤확산 특성(Random Constant Spreading)[4]을 가지고 있으며, 이는 랜덤 스캔・확산 과정에서, 과도한 트래픽에 의해 네트워크 자원이 소모되어 감염 호스트만 피해를 입는 기존 웜에 비해 그 피해가 더욱 광범위하다. 제로-데이의 현실화와 랜덤확산형 웜들의 빠른 전파속도는 기존 대응방법에 문제점을 노출하여 새로운 대응 방법이 필요하게 되었다[6].

본 논문에서는 네트워크 관련자원을 급속히 소모시키는 랜덤확산형 웜을 제어하는 모델을 제안하기 위하여 단순 성장 네트워크와 선호적 성장 네트워크를 시뮬레이션을 통해 분석하였다.

시뮬레이션 결과 선호적 성장 네트워크가 랜덤확산형 웜의 동적 제어에 더 적합함을 나타내어 선호적 성장 네트워크의 멱함수(Power-Law) 특성[7]과 노드의 분포 특성을 이용하는 확산 제어모델을 제안하고, 제안된 모델이 웜의 확산을 효과적으로 제어 할 수 있음을 검증하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 랜덤확산 특성을 가지는 웜 모델, 실제 확산 사례 및 기존 대응사이클의 문제점을 분석하고 제 3장에서는 멱함수 특성을 가지는 동적 네트워크의 특성 및 구성, 랜덤 에러와 의도된 공격의 경우 랜덤확산 모델에 대한 견고성에 대하여 시뮬레이션 해본다. 제 4장에서는 동적 네트워크의 멱함수 특성을 이용한 웜 확산제어 모델을 제안한 후 시뮬레이션을 통해 이의 분석결과를 보이고 마지막으로 제 5장에서는 결론과 향후 계획에 대해 기술한다.

2. 랜덤확산형 웜 모델과 관련연구

2.1 랜덤확산형 웜의 출현 및 사례분석

2001년 7월 자기증식코드 능력을 가진 코드레드가 취약 호스트를 매우 빠르게 감염시켰다. 또한 2003년 1월에는 슬래머 웜, 2003년 8월에는 마이크로소프트 블라스터 웜등이 출현하여 이전의 코드레드나 님다보다 더 빠르게 확산하여 국내・외적으로 커다란 피해가 발생하였다. 슬래머 웜은 8.5초당 2배씩 증가할 뿐 아니라, 약 30M~150M의 대역폭을 소모하면서 약 30여분 만에 전 세계 감염 가능한 호스트의 90%인 약 75,000대를 감염시켰다.

(그림 1)의 슬래머 웜 확산 모델에서 각 노드에서 발생한 패킷들은 임의의 UDP 패킷이므로 약 90% 이상의 확률로 최상위 국외접속 노드로 전송된다[3].

전송패킷들은 국외접속 노드의 대역폭을 초과하여 국제 연동이 필요한 루트 네임서버에 연결 이상이 발생하고, 이에 따라 국내 도메인 네임서버 연결 불능으로 국내 인터넷 전체가 사용불가능한 상태가 된다.

최근 웜의 특징인 랜덤확산형 웜은 웜 그 자체보다 수분 내로 지역망을 과 부하시키고 단 시간에 확산되는 자원 소모과정에서 다른 서비스를 불가능하게 만드는 이차적인 문제가 더 심각하다. 따라서 인터넷의 생존성을 보장하기 위해서 감염 직후 동적으로 확산을 억제하는 모델이 필요하다.

2.2 랜덤확산형 웜 모델 분석

전염 및 확산 사이클에 대한 연구로는 (그림 2)와 같은 3단계 확산 모델이 있다[8].

감염가능단계에서는 취약점을 사전에 제거하는 예방활동을 통해, 감염단계에서는 감염율 등을 줄이는

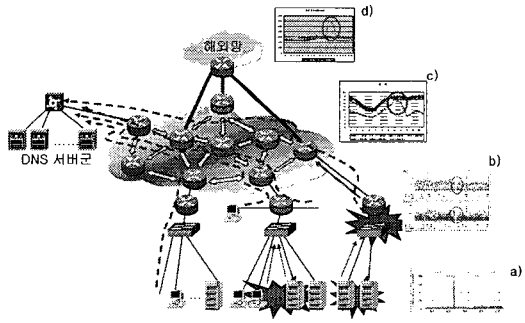

그림 1. 국내 슬래머 웜 확산 모델

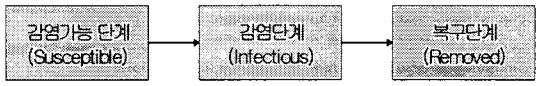

그림 2. 3단계 확산 모델

방법을 통해, 복구단계에서는 감염노드의 치료율을 증가시키는 방법 등을 통해 각 단계에서의 확산을 줄일 수 있다.

3단계 확산 모델 중 감염단계 t 시점에서의 감염된 호스트 수는 (식 1)과 같다[5].

- 가정 : 감염 호스트는 다른 호스트에 대하여 랜덤성을 가지고 공격하고 한번 감염된 호스트는 다시 감염되지 않음
- T : 처음 감염이 발생한 시간
- t : 측정을 위한 임의의 시간
- N : 감염 가능한 총 호스트 수
- K : 확산상수(K는 CPU, 네트워크 연결 속도, 감염 위치에 영향을 받지 않는 전역 변수)
- S : t 시점에서 감염 가능한 호스트
- A : t 시점에서 감염된 호스트
- $s = S/N$: 비감염 호스트 비율
- $a = A/N$: 감염된 호스트 비율
- U : 치료 활동을 통해 감염 상태에서 정상 상태로 돌아온 호스트
- $u = U/N$: 정상적으로 돌아온 호스트 비율
- r : 복구율이라 할 때

$\frac{dA}{dt} = A \times K \times s$ 이고, $s = 1 - a$이므로

$$\frac{da}{dt} = a \times K \times (1-a) \tag{1}$$

따라서 임의의 시간 t 에서 감염된 호스트의 비율을 a라 하면 dt 시간동안 감염될 호스트 수는 (식 2), (식 3)과 같이 구해진다.

$$Nda = (Na)K(1-a)dt \tag{2}$$

$$\Rightarrow a(t) = \frac{e^{K(t-T)}}{1+e^{K(t-T)}} \tag{3}$$

2.3 네트워크 모델과 에러와 공격에 대한 견고성

네트워크 모델에는 정적 네트워크와 동적 네트워크 모델이 있으며 정적 네트워크 모델은 에르되스-레니 모델[9]과 와츠-스트로가츠 모델[10]로 구분된다.

에르되스-레니 모델은 랜덤 네트워크 모델로서 각 노드의 연결 링크는 확률 pER을 가지고 랜덤하게 연결되므로 모든 노드가 임의의 링크로 연결될 확률은 동일하게 주어진다. N이 노드의 총 개수일 때 링크의 총 수 n은 (식 4)와 같다[9].

$$n = pER\frac{N(N-1)}{2} \tag{4}$$

에르되스-레니 모델에서 $pER=0$일 경우 모든 노드는 고립되며 $pER=1$일 경우 완전연결 그래프가 되며 각 노드가 k개의 링크를 가질 확률인 연결선 분포 $P(k)$는 (식 5)와 같다.

$$\lambda = \binom{N-1}{k} pER^k (1-pER)^{N-1-K} \text{일 때}$$

$$P(k) = e^{-\lambda}\lambda^k / k! \tag{5}$$

에르되스-레니 모델에서는 모든 노드-링크간의 연결 확률이 같기 때문에 노드의 클러스터링 현상을 해석할 수 없어 와츠-스트로가츠는 정규 격자 그래프에 약간의 랜덤 링크를 추가함으로써 클러스터링 문제를 해결하였다[10].

$pWS=0$일 경우 네트워크는 높은 밀도로 클러스터링되고 두 노드간의 평균 거리 $<l>$은 노드의 총 개수 N에 대하여 선형적으로 증가하나 $pWS=1$일 경우 네트워크의 형태는 완전한 랜덤 그래프가 되어 클러스터링은 최소화되고 두 노드간의 평균거리 $<l>$은 노드의 총 개수 N에 대하여 지수 함수적으로 증가한다. $0 < pWS < 0.01$일 경우 높은 클러스트링이 유지되면서 전체 노드간의 거리가 급격하게 짧아지는 특성이 나타난다.

정적 네트워크 모델이 노드 수가 정해져 있고 링크만으로 네트워크를 설명하므로서 실 네트워크의 성장, 소멸을 반영하지 못해 바라바시는 동적 네트워크 모델링을 위해 척도없는(Scale Free) 네트워크 모델을 제안하였다[7].

척도없는 네트워크 모델은 두 단계의 알고리즘에 의해 노드가 추가되고 성장한다.

1) 노드의 성장(g-네트워크) : 적당한 초기의 노드(m_0)로부터 네트워크 성장이 시작하고 각 시간단위마다 m_0 보다 작은 m개의 링크를 가지는 새로운 노드를 시스템에 추가한다. 이때 생성 링크는 추가된 노드를 m개의 서로 다른 기존 노드에 연결한다.

2) 선호적 연결(p-네트워크) : 새로운 노드와 연결할 m개의 서로 다른 노드를 선택할 확률 Π는 (식

6)에 의해 선택한다.

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \quad (6)$$

즉, 선택 과정에서 링크를 많이 가진 노드가 덜 가진 노드에 비해 링크 수 만큼 더 높게 주어진다. 척도없는 네트워크 모델에서 시간 t가 지난 후 전체 네트워크의 노드 총 수 N는 $N=t+m_0$, 링크의 총 수는 mt이며 연결선 분포 $P(k)$는 (식 7)과 같다.

$$P(k) = \frac{2m^2t}{m_0+t}k^{-3} \quad (7)$$

(식 7)에서 $t \to \infty$ 일 경우 이 함수는 지수 $r=3$인 멱함수 즉, $P(k) \sim 2m^2k^{-3}$로 수렴하며 r과 m은 네트워크 크기 $N=t+m_0$와는 독립적인 변수이다.

네트워크 구조는 네트워크 성능뿐 아니라, 전체 노드에 대해 랜덤한 분포를 따르는 고장이나 특정 지역을 선별하여 진행할 수 있는 공격에 대해 서로 다른 견고성을 보인다[11].

p-네트워크의 인터넷은 전체 노드 중 80%이상의 고장에 견딜 만큼 견고한 반면, 단지 18%의 의도된 공격만으로 붕괴할 정도로 취약한 것으로 밝혀졌으나[12] 2장에서 랜덤확산형 웜일 경우에는 랜덤공격임에도 네트워크의 자원소모에 의해 쉽게 붕괴됨을 볼 수 있다.

2.4 랜덤확산형 웜의 확산 제어

랜덤확산 특성을 가진 웜에 대한 연구는 TCP 연결을 가로채 이를 강제로 유지시켜 TCP 기반의 웜 확산을 지연시키는 LaBrea 프로젝트가 있으나[13] 비 TCP-기반 연결에는 대처하지 못하고, Williamson은 새로운 호스트가 추가될 때마다 이를 주어진 비율 이하로 제한하는 스로틀(Throttle) 기능 도입을 제안하였는데[14] 인터넷에 연결된 모든 호스트가 동시에 스로틀 기능을 채택하여야 하므로 실현성이 어렵다.

또, Weaver등은 네트워크 대역폭 조절을 통한 확산제어 방법을 제안하였는데[15] 일괄적인 대역폭 조절을 통해 랜덤확산형 웜의 확산속도를 조절할 수 있으나 최적의 제어지점 및 동적으로 반응하는 시스템을 제시하지는 못하였다.

3. 네트워크 모델 시뮬레이션

랜덤확산 모델의 특징은 최초 감염 직후 감염 정도가 기하급수적으로 늘어나며 임계시간이 지난 후에는 더 이상 감염시킬 대상이 없어져 커브의 기울기가 완만하게 변하게 된다.

(그림 3)은 랜덤확산 모델 함수에 의해 생성된 확산 그래프와 코드레드의 실제 확산 경우이다. 여기서 N은 취약성이 내재된 약 360,000개의 호스트이고, K는 10호스트/sec이다.

감염단계를 최소화하기 위해서는 웜의 빠른 탐지와 탐지 후 웜의 확산과정에서 트래픽 제어를 통한 확산제어 방법이 필요하다. 이 과정은 랜덤확산형 웜의 특성상 빨리 진행되므로 탐지와 확산 억제가 동적으로 진행되어야 한다. 이를 위해 랜덤확산형 웜에 적합한 네트워크를 선택하고 노드분포 등을 이용하여 웜 확산을 최소화하는 동적 웜 확산제어모델을 제안한다.

시뮬레이션은 100개에서 10,000개의 노드를 가지는 g, p-네트워크를 설계하여 p-네트워크의 경우 모든 경우에 있어 멱함수 특성이 나타나는지 확인하였고 평균 분석이 필요한 경우에는 각각 10,000개의 노드를 가지는 독립 네트워크 100개를 구성하여 3% 오차 범위를 넘어설 경우 무의미한 가정으로 간주하였다.

시뮬레이션을 통해 20개 노드를 생성한 g-네트워크와 p-네트워크는 (그림 4-a), (그림 5-a)와 같으며 $t \to \infty$에 따라 노드의 총 수 N이 충분히 크다면 (그림 4-b), (그림 5-b)와 같은 형태로 성장한다.

(그림 6)의 20개의 노드로 이루어진 g-네트워크에서 모든 노드가 분절되도록 하기 위해서는 총 6개의 노드 훼손이 필요하다.(루트노드인 0번 노드 제외)

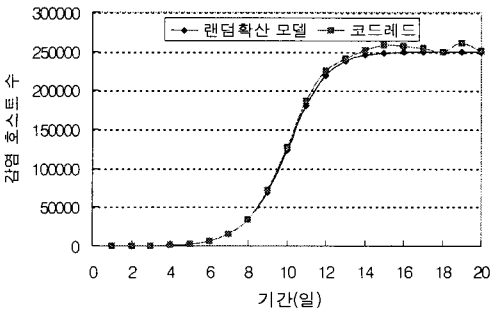

그림 3. 랜덤확산 모델과 코드레드 확산 그래프

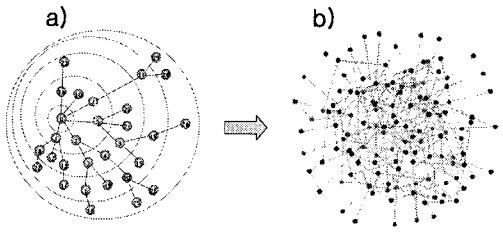

그림 4. g-네트워크의 성장

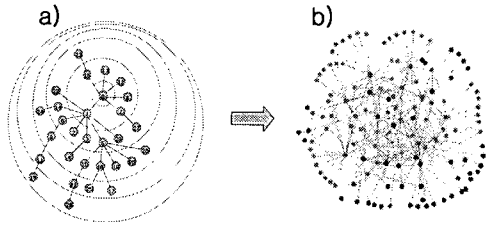

그림 5. p-네트워크의 성장

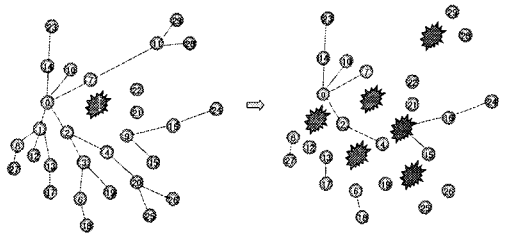

그림 6. g-네트워크에서의 분절

(그림 7)에서 20개의 노드로 이루어진 p-네트워크에서 1번 전달노드가 기능을 잃어버릴 경우 약 40%의 네트워크 분절이 발생할 뿐 아니라 단 3개의 노드 훼손으로도 네트워크 전체가 붕괴된다. 즉, g-네트워크에 비해 약 1/2의 특정노드에 대한 공격만으로 p-네트워크 피해가 더 심각하다.

랜덤확산형 웜의 동적 제어를 위한 적절한 네트워크를 선택하기 위하여 아래의 제한사항을 두고 g, p-네트워크를 시뮬레이션 하였다.

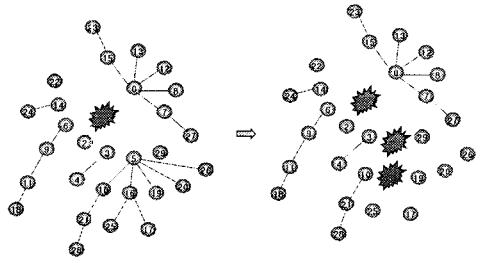

그림 7. p-네트워크에서의 분절

1) 제안모델은 거시모델이므로 네트워크 프로토콜 및 서비스 등의 특성은 고려하지 않는다.
2) 전송역할을 하는 비 터미널 노드는 감염되지 않고 최종 터미널 노드만 감염된다.
3) 최초 노드 m_0 및 링크 수 m의 값은 성장 네트워크의 특성에 아무런 영향을 미치지 않으므로[7] $m_0 = m$은 1로 한다.

선호적 성장모델 p-네트워크가 단순 성장모델인 g-네트워크와 가장 큰 차이는 노드-링크 연결 분포에서 특정노드의 링크수가 지수분포보다 크게 나타나는 두터운 꼬리(Heavy-Tail) 분포의 유무이다.

(그림 8)은 멱함수 $P(k) \sim k^{-3}$의 노드-링크 분포를 로그변환한 값이며, (그림 9), (그림 10)은 g, p-네트워크에서 10,000개의 노드를 10번씩 수행한 노드-링크 분포의 평균값과 이를 로그변환 한 값이다.

(그림 8)과 (그림 10)의 로그변환값이 동일한 단순 감소 형태임을 보여주고 있어 동일한 네트워크임을 알 수 있다. 노드 생성 후 g-네트워크에서는 최대 링크를 보유한 노드가 기껏해야 15개 정도인데 반해 p-네트워크는 약 90~400의 링크를 보유한 두터운 꼬리 노드들이 반드시 생성되었다.

약 10,000개의 노드를 가진 네트워크로 100번의 실험을 통해 나타난 g, p-네트워크에서의 단말 · 비단말 평균 분포는 <표 1>과 같다.

표 1. g, p-네트워크의 단말 · 비단말 분포

샘플	g-단말	g-비단말	p-단말	p-비단말
평균	5000.8	4999.2	6697.8	3302.2
편차	28.6	28.6	40.8	40.8
편차 비율	0.6%	0.6%	0.6%	1.2%

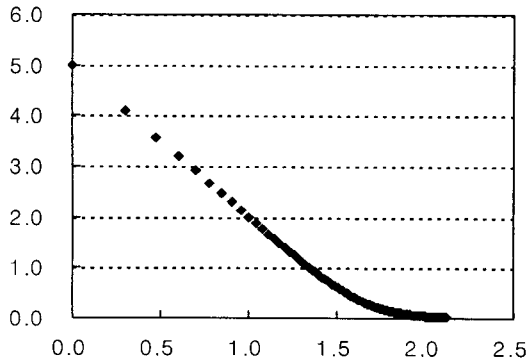

그림 8. $r=3$인 멱함수의 로그 변환

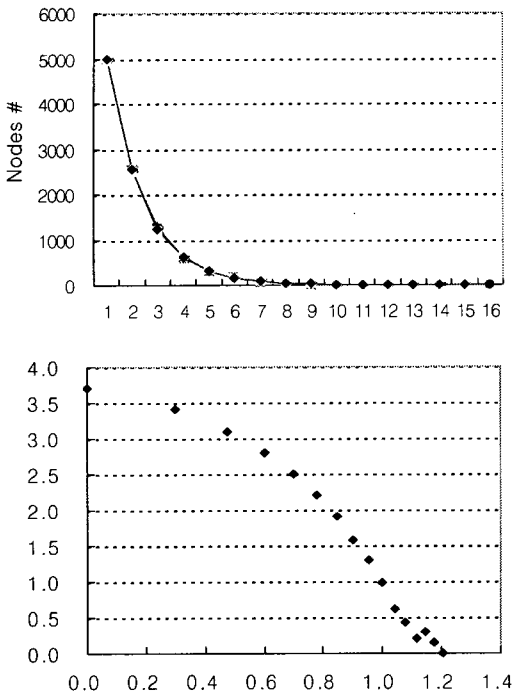

그림 9. g-네트워크 노드-링크/로그변환

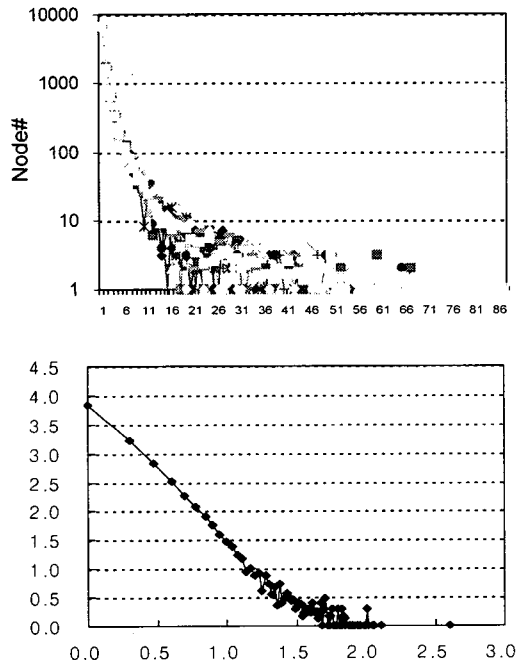

그림 10. p-네트워크 노드-링크/로그변환

p-네트워크가 단말 · 비단말 노드분포가 50:50으로 비슷한 반면 g-네트워크는 단말 · 비단말 노드 분포가 약 67:33의 비율로 나타난다. 각 경우에 있어 평균 편차는 최대 1.2%로서 동일한 결과이며 이는 단말노드 수가 많은 p-네트워크가 악성 웜 등에 감염될 비율이 약 33.9% 높고 소수의 전달노드에 공격을 가하는 분산서비스거부공격이 효율적임을 의미한다. 하지만 비 단말노드가 적으므로 적은 노력으로 웜 확산을 억제할 수 있는 있는 p-네트워크를 대상 모델로 결정하였다.

4. 랜덤확산형 웜의 확산 제어 모델 및 성능 평가

4.1 랜덤확산형 웜의 확산 제어 모델

p-네트워크가 적절히 성장 할 경우 멱함수의 연결선 분포 특성에 따라 두터운 꼬리노드가 나타나게 되는데 시뮬레이션 결과 노드가 약 20개 이상만 되어도 멱함수의 특징이 나타난다. p-네트워크는 선호적 성장모델이기 때문에 링크를 많이 소유한 노드는 m_0와 같은 초기 생성 노드 부근에 많이 분포가 될 것이다. 이를 위해 p-네트워크의 깊이 $Depth(n_i)$를 (식 8)과 같이 정의한다.

$Depth(m_0) = 0$ 이라 할때

$$Depth(n_i) = Min(Length(n_i) \rightarrow Length(m_0)) \qquad (8)$$

(그림 11)과 같이 m_0 노드 근처 즉, $Depth(n_i)$가 0에 가까울수록 과 부하없이 네트워크 전체 대역폭을 효율적으로 조절 할 수 있는 동적 제어모델을 제안한다. 제어 모델의 비교기는 m_0에서 발생되는 패킷이 임계값이 넘을 경우 깊이 $Depth(n_i)$의 노드에서 발생되는 패킷과 비교하여 결정기 부분에서 확산제어가 필요할지를 결정한다. 만일 임계값이 일정 시간 이상 또는 랜덤확산형 모델을 따를 경우 깊이 $Depth(n_i)$의 대역폭 조절 값을 피드백한다.

4.2 성능 평가

(그림 12)는 확산상수 K 및 대역폭의 변화에 따라 p-네트워크에서의 확산속도를 정규화 한 결과로서.

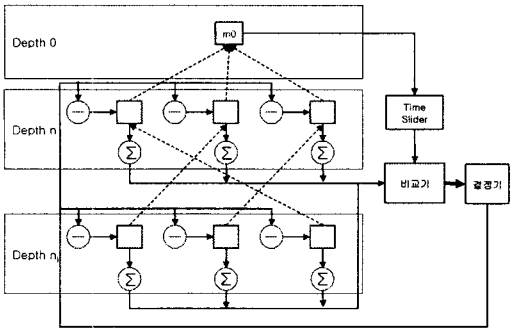

그림 11. p-네트워크의 확산제어 모델

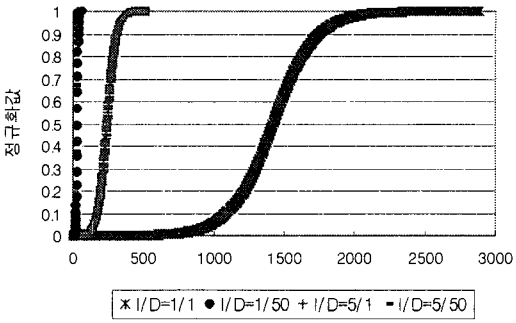

그림 12. K/대역폭 변화에 따른 웜 확산

같은 대역폭 B_i에 대하여 1/5의 K와 비교하여 약 20%의 확산 억제효과가 있었으며 같은 비율 K_i에 대하여 네트워크 대역폭을 1/2 줄일 경우 약 43%의 확산 억제효과가 있었다.

억제효과를 정규화하면 확산상수 K가 대역폭 제어보다 억제효과가 약 7% 더 높으나 K가 웜 자체 상수이므로 대역폭 억제에 의한 효과도 확산상수 K와 거의 같은 수준이다.

p-네트워크의 두터운 꼬리 노드의 분포는 낮은 $Depth(n_i)$ 값을 가지는 위치에서 높게 나타난다. 10,000개의 노드로 구성된 p-네트워크에서 평균깊이는 약 9.9이고 평균 링크 수는 약 486개이다.

링크 집중도가 커지면 커질수록 노드 당 평균 패킷처리를 위한 부하율이 커지지만 상대적으로 높은 집중도를 갖는 노드들이 낮은 깊이에 집중되어 있어 트래픽 분산의 효과가 크게 나타난다.

다양한 노드 수를 가지고 실험해 본 결과 (그림 13)에서 노드 수에 상관없이 전 구간에서 비슷한 패킷처리율을 보인다. 이는 제안된 모델이 깊이에 따른 확산효과와 깊이에 따른 노드당 부하율에 의해 최적의 깊이를 선택할 수 있음을 의미한다. 노드가 같은 깊이일 경우, p-네트워크에 비해 g-네트워크가 약 9배의 패킷처리율을 나타내어 p-네트워크가 웜 확산 제어에 훨씬 효과적임을 보여준다.

(그림 14)는 각 깊이에 분포된 노드와 누적 노드 수이다. 전체 노드의 분포는 n_i가 5인 깊이, 약 41% 깊이 지점에서 최대를 이루고 최대 깊이지점에서 깊이가 깊어질수록 즉, $\lim_{i\to\infty} Depth(n_i)$ 일수록 조밀하게 분포됨을 알 수 있다.

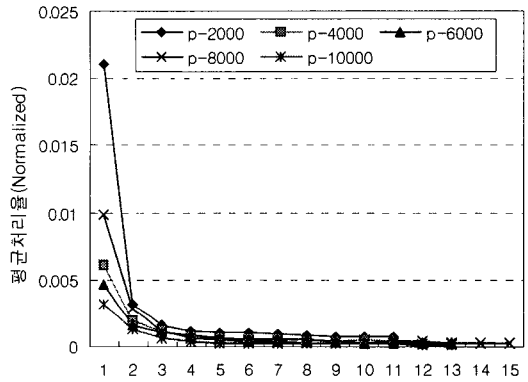

그림 13. p-네트워크의 노드깊이당 패킷처리율

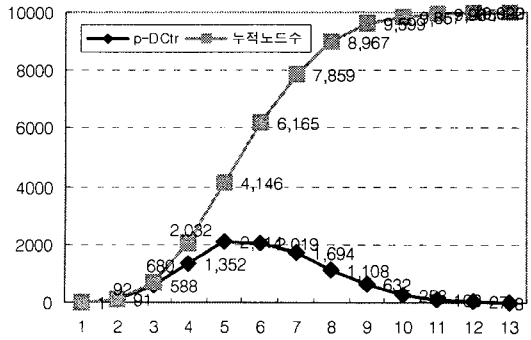

그림 14. 깊이에 따른 노드 수

좋은 효율을 보이는 $Depth(n_i)$의 범위는 (그림 15)의 빗금 친 부분으로서 n_i가 $1 \le n_i \le 4$인 깊이, 즉, $0 \le Depth(n_i) \le (Max(Depth(n_i)))_{33\%}$로 나타났으며 $Depth(n_i)$를 $(Max(Depth(N_i)))_{33\%}$ 이상으로 선택할 경우 확산제어 효과가 급격히 저하되었다. 이 깊이 범위에서의 대역폭 제어를 담당하는 노드 수는 2,030여 개로 약 20%이다.

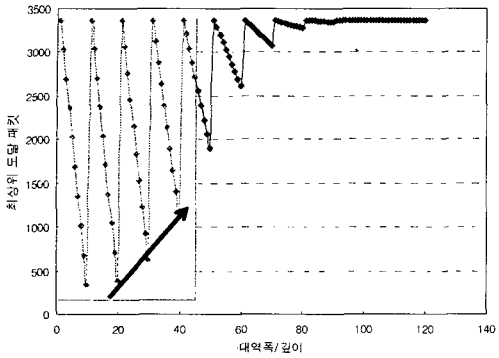

그림 15. 대역폭-깊이에 따른 확산제어 효과

5. 결 론

본 논문에서는 확산 과정에서 네트워크의 자원을 과도하게 소모하는 랜덤확산형 웜의 확산모델을 해석하고, 멱함수 특성을 가진 동적 네트워크의 노드깊이분포를 이용하여 확산을 동적으로 제어하는 모델을 제안하였다. 시뮬레이션 결과 실험결과 깊이별 패킷처리율에서 좋은 효율을 보이는 $Depth(n_i)$의 범위는 약 $0 \le Depth(n_i) \le (Max(Depth(n_i)))_{33\%}$로 나타났으며 $Depth(n_i)$를 $(Max(Depth(N_i)))_{33\%}$ 이상으로 선택할 경우 확산제어효과가 급격히 저하되었고 노드부하율은 $Depth(n_i)$ 범위가 약 $(Max(Depth(n_i)))_{16\%} \le Depth(n_i) \le (Max(Depth(n_i)))_{42\%}$ 일 경우 최하가 되는 것으로 나타났다.

깊이별 패킷처리율과 노드의 부하를 동시에 고려할 경우 최적의 $Depth(n_i)$는 $(Max(Depth(n_i)))_{16\%} < Depth(n_i) < (Max(Depth(n_i))_{33\%}$이며 이 깊이범위에서 제어기능을 담당하는 노드 수는 약 20%로 나타났다.

본 논문에서는 $Depth(n_i)$에 해당하는 모든 노드의 대역폭을 일률적으로 조절하였으나, 향후 필요 노드만 적절히 선택한다면 랜덤확산형 웜에 대응하는 최적의 시스템을 만들 수 있을 것이다.

참 고 문 헌

[1] J. F. Shoch and J. A. Hupp. "The Worm Programs - Early Experience with a Distributed Computation," *Communications of the ACM*, Vol. 25 No. 3, pp. 172-180, Mar. 1982.

[2] Eeye Digital Security, "Code Red Disassembly", http://www.eeye.com/html/advisories/codered.zip, 2001

[3] 정보통신부 정보통신망 침해사고 합동조사단, 정보통신망 침해사고 조사결과, pp. 2-5, Feb. 2003.

[4] S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," *Proc. of the 11th USENIX Security Symposium*, pp. 3-10, 2002.

[5] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code," *Proc. of the 2003 IEEE Infocom Conf.*, pp. 3-5, Apr. 2003.

[6] E. Rice, "The Effect of Infection Time on Internet Worm Propagation," *Math.* Vol. 164, Scientific Computing at Harvey Mudd College, pp. 3-4, May, 2004.

[7] R. Albert, H. Jeong, and A.-L. Barabasi, "Mean-Field Theory for Scale-Free Random Networks," *Physica A*, pp. 175-181, 1999.

[8] D. Daley and J. Gani, *Epidemic modeling*, Cambridge University Press, 1999.

[9] P. Erdos and Renyi, "On the evolution of random graphs," *Publ., Math.*, Ins., Hung., Acad., Sci., Vol. 5, pp. 17-60, 1960.

[10] D. J. Watts and S. H. Strogatz, "Collective Dynamics of small-world networks," *Nature* 393, pp. 440-441, 1998.

[11] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of Scale-Free Networks: Error and Attack Tolerance," *Physica A*, pp. 642-650, 2003.

[12] R. Albert, H. Jeong, and A.-L. Barabasi, "Error and Attack Tolerance of Complex Networks," *Nature* 406, pp. 379-381, 2000.

[13] http://labrea.sourceforge.net/labrea-info.html

[14] M. Williamson, "Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code," *18th Annual Computer Security*

Applications Conf., pp. 6-7, Dec. 2002.

[15] N. Weaver, I. Hamadeh, G. Kesidis, and V. Paxson, "Preliminary Results Using Scale Down to Explore Worm Dynamics," *Proc. of the 2004 ACM workshop on Rapid Malcode*, pp. 3-6, Oct. 2004.

노 병 규

1988년 2월 충남대학교 이과대학 전산학 학사
1995년 2월 충남대학교 이과대학 전산학 석사
2006년 2월 순천향대학교 공과대학 전산학 박사
1988년 1월~1997년 1월 한국전자통신연구원
1997년 1월~2005년 1월 한국정보보호진흥원 평가2팀장, 평가기준팀장, 기반보호기획팀장
2005년 1월~현재 한국정보보호진흥원 보안성평가단 단장
관심분야 : 정보보호시스템 평가, 시스템·네트워크 정보보호

박 두 순

1988년 고려대학교 전산학전공 (이학박사)
1985년~현재 순천향대학교 정보기술공학부 교수
2000년~현재 한국 멀티미디어학회 편집위원
2004년~2005년 미국 U. of Colorado 객원교수
2002년~2003년 순천향대학교 공과대학 학장
2000년~2005년 한국 멀티미디어학회 이사, 논문지 분과위원장
2001년~2004년 한국정보처리학회 편집위원
관심분야 : 병렬처리, 멀티미디어 컨텐츠, 데이터마이닝, 유비쿼터스 컴퓨팅, 컴퓨터 교육