DOI : 10.3745/JIPS.2009.5.1.033

Dynamic Control of Random Constant Spreading Worm using Depth Distribution Characteristics

Byung-Gyu No*, Doo-Soon Park, Min Hong**, HwaMin Lee** and Yoon Sok Park*****

Abstract: Ever since the network-based malicious code commonly known as a 'worm' surfaced in the early part of the 1980's, its prevalence has grown more and more. The RCS (Random Constant Spreading) worm has become a dominant, malicious virus in recent computer networking circles. The worm retards the availability of an overall network by exhausting resources such as CPU capacity, network peripherals and transfer bandwidth, causing damage to an uninfected system as well as an infected system. The generation and spreading cycle of these worms progress rapidly. The existing studies to counter malicious code have studied the Microscopic Model for detecting worm generation based on some specific pattern or sign of attack, thus preventing its spread by countering the worm directly on detection. However, due to zero-day threat actualization, rapid spreading of the RCS worm and reduction of survival time, securing a security model to ensure the survivability of the network became an urgent problem that the existing solution-oriented security measures did not address.
This paper analyzes the recently studied efficient dynamic network. Essentially, this paper suggests a model that dynamically controls the RCS worm using the characteristics of Power-Law and depth distribution of the delivery node, which is commonly seen in preferential growth networks. Moreover, we suggest a model that dynamically controls the spread of the worm using information about the depth distribution of delivery. We also verified via simulation that the load for each node was minimized at an optimal depth to effectively restrain the spread of the worm.

Keywords: *Worm, Random Constant Spreading, Dynamic Network, Depth Distribution Characteristic, Bandwidth Control*

1. Introduction

In knowledge-based information society, an authenticity plays a key role for the information communication and numerous reverse functions such as leak of information, forgery of information, worm or virus, access deny by hacking, and computer crimes are substantially increasing. Among these reverse functions, the appearance of worm that destroys the network system is one of the most serious problems.

After Hupp[1] defined worm in 1982, more worms have been developed as network evolved. This is perhaps the greatest threat for the Internet. Code Red or Nimda infected millions of computers worldwide and it serious problems for the international economy [2]. Slammer Worm infected 90% of hosts in the world in 10 minutes and it anesthetized the Korean internet system around 12 hours.

The attacking technique of worm is evolving from a guessing password to buffer overflow, attacking of format string, DDOS (Distributed Denial of Service), and so on. The common aspect of these trends is faster and wider spread. It's almost too fast to react to the attack. When the system is connected to a network, there are two possible attacks. First, it is an attack against individuals. Individuals are at risk of revealing their personal information such as credit card information, behavior pattern, email and etc.. The second type of attack is DDOS against routers, name server, internet main services, and main facilities.

The trends of recent worms are that after the weaknesses of specific systems or operating systems are announced, the hackers attack using these weak points shortly (Zero-Day threat) [3] and that the Self Propagation Code worms[4] which propagate the copy of worms automatically when the weaknesses are exist in the systems are rapidly increasing[5]. If a patch has been ameliorated the system properly, pc usually have survival time of less than 30 minutes.

It is impossible to countermeasure by hands of human because of the reality of faster propagation, RCS worm, and reduction of survival time and vulnerability threat window. Even the automated system cannot countermeasure the attacks properly, due to the many errors that are currently in the system [5].

Especially the increasing number of network and resource hogging RCS worm shows problems of current network solutions such as firewall, intrusion detection system,

Manuscript received February 17, 2009; revised March 4, 2009; accepted March 24, 2009.
Corresponding Author: Doo-Soon Park
* Korea Information Security Agency, Seoul, Korea(nono@kisa.or.kr)
** Division of Computer Science and Engineering, SoonChunHyang University, SinChang-Myeon, Asan-Si, ChungChungNam-Do, Korea (parkds@sch.ac.kr, mhong@sch.ac.kr, leehm@sch.ac.kr)
*** Samsung Electro-Mechanics div., Seoul, Korea (schatze11@hotmail.com)

access control system, packet filter, anomaly network traffic filter.

To countermeasure these new threats, we need to move from prevent, isolate, detect, recover to new security approach that can ensure safety of a network. These researches show this type of commonality in different fields such as biological network, social network, and internet network.

Dynamic network's unique quality is that it can withstand a random attack but it can also bring down the whole network against certain attacks.

In this paper, we suggest an efficient model to prevent and control the spreading of RCS worm. We test and analyze an efficient dynamic network for both a growth network model and the growth and preferential network model using a simulation program. Our simulation results show that the growth and preferential network model is well suitable to prevent the spreading of RCS worm. Therefore, we suggest the characteristics of Power-Law for the growth and preferential network model and the depth distribution of the delivery node [6] to control the spreading of worm. The proposed model dynamically controls the spreading based on degrees of spread of the worm. This paper finds out the optimal depth to effectively restrain the spread of worm through a simulation and confirms that the node in the optimal range could cope with the RCS worm because it could control effectively traffic without heavy overload.

The rest of this paper is organized as follows: the brief overview of RCS worm model is elaborated in section 2. Section 3 elucidates the structures of dynamic network which has characteristics of the Power-Law and provides the simulated results for the robustness of dynamic network under random errors and intended attacks. In section 4, we introduce our new control model of RCS worm using the Power-Law characteristic and provide the analysis of simulated results using our model.

2. Related Works

Code Red, a self-propagation code, spread rapidly through vulnerable hosts in July 2001. Shortly thereafter, a slightly modified Code Red II appeared and spread just as fast. These worms were far more advanced and powerful compared to other existing worms at the time. Slammer Worm appeared in January 2003 and MS Blaster Worm appeared in August 2003, both having even more devastating effects worldwide.

According to CAIDA, Slammer Worm appeared at 05:30, January 25, 2003, and propagated through 75,000 machines in 30 minutes, which accounted for 90% of the world's main hosts.

Recent trends indicate that malicious computer viruses are widely using RCS worms. Since the RCS worm overloads local systems in a minute and spreads globally through the internet by exhausting resources that are able to anesthetize other services, which is an indirect effect of infection that is much more critical. To provide continuous availability of internet connectivity, we need a control model that can dynamically restrain the worm's srpeading after system infection. Fig. 1 shows the cycle of infection and spreading for the RCS worm [7].

Fig. 1. The 3 steps of the RCS worm spreading process

To reduce the rapid spreading of the worm, the proliferation factors in each of the three above steps should be diminished. These factors include advance detection and prevention of weaknesses in step one (susceptible), the reduction of infection rate in step two (infectious) and the increasing cure rate for infected nodes in step three (removed). The traits of the network involved are critical factors in controlling the spread of an RCS worm. There is a static network model as well as a dynamic network model. Since the static network model includes a fixed number of nodes and only can modify link states, it cannot represent a real network system. The difficulty in implementing the Erdos-Renyi model[8] or the Watts-Strogatz model[9], both of which are passive network models, is that the nodes are static while links are the only parts that are changing. This isn't the case in the reality of a network, where both nodes and links cannot be static but are instead always changing. In reality, the networks are dynamic in time. In a static network model, the total number of nodes is static whilelinks are the only dynamic aspect of the system. Within the boundaries of a static network, it cannot depict dynamic aspect of both nodes and links in a real network.

On the other hand, the dynamic network model can change the node-link state and is close to the real network system. To accomplish a dynamic network, Albert et al.[6] introduced a scale-free dynamic network model. Scale-free means there is a conventional scale that is used to describe the size of an earthquake or income range. In a scale-free network model, nodes and links can be added using a two-step algorithm:

1) growth of nodes (g-network) : Start a network at a decent number nodes, and at each hour add nodes that have fewer links. Connect the newly created link to the newly added nodes and the different nodes in the system.
2) preferred connection (p-network): The probability of connecting a new node with a different node.

The scale-free dynamic network model enlarges the network by adding nodes and links that connect these new nodes using the growth of nodes (g-network) and the preferred connection (p-network). An internet connection that has p-network characteristics is able to resist 80% of random malfunctions and it destroys the network at only 18% of intended attacks [6]. However, random attacks by RCS worms demolish the network much more easily due to the multiple and simultaneous increase in resources.

LaBrea project[10] delays the spread of worm using unused IP addresses in the TCP connection, but this project

cannot be used for a non-TCP connection. Williamson [11] proposes the throttling function that controls the transmission rate to be low when each host normally connects additional hosts. However, this method still has some problems since the traits of a worm include abnormal connections. Weaver et al. [12] controls the spread of the worm using strict bandwidth management, but this cannot provide the optimal bandwidth value and a dynamic management system.

3. Network Model Simulation

The spreading rate in the 3 steps of the spreading process is checked to control the propagation of the RCS worm. When the first infection occurs in the RCS model, the infection rate is increased by geometric progression until a certain, critical stage and then it stays in fluent curves since all of systems are already infected. In this paper, we suggest a new control model for the infectious step in the process that can efficiently manage the spread of infection. To minimize the influence of the infectious step, the fast detection of the worm and the efficient limitation of resources that the worm requires to spread after initial infection are essential. The proper network and depth distribution of the delivery node is then selected for our simulation to spread the RCS worm.

A g-network and p-network, which have from 100 to 50,000 nodes, are designed to test the simulation. In the p-network case, we double-checked to ensure that the result should show Power-Law characteristics. If the error boundary exceeds 2% for 1,000 independent networks that each include 10,000 nodes, we shall ignore these simulations. Fig. 2-a and Fig. 3-a show the simulation results of the g-network and p-network after an initial 20 nodes are created. As $t \to \infty$ and the number of nodes is large enough, the g-network and p-network are developed in Fig. 2-b and Fig. 3-b.

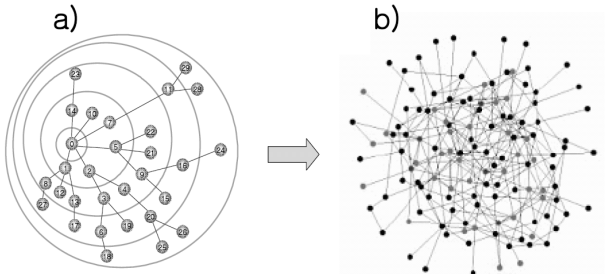


Fig. 2. The g-network growth

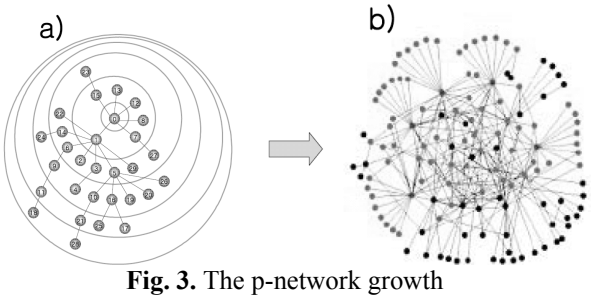


Fig. 3. The p-network growth

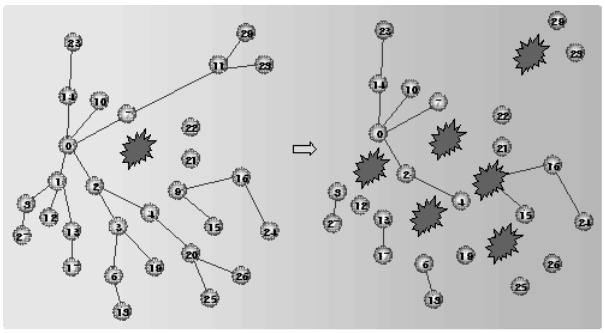


Fig. 4. The separation of network in g-network

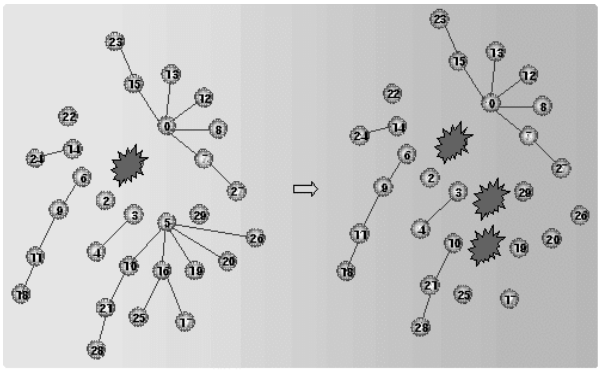


Fig. 5. The separation of network in p-network

In Fig. 4, the damage of 6 nodes can cause the separation of all nodes except a root node, which is a 0 node in the g-network that includes 20 nodes. In Fig. 5, when node 1 lost its functionality, around 40% of network separation occurred and the whole network fails because of damage to only 4 nodes. Thus, unlike the g-network, the p-network causes more serious problems in the event of a specific node attack.

The g- and p-network simulations are performed under the following limitations, which are used to select the proper dynamic network for controlling the spread of an RCS worm.

1) The proposed model is a macroscopic model, so it ignores the properties of network protocol and services.
2) Non-terminal nodes are not infected during the transmission while the final terminal nodes can be infected.
3) The properties of the growth network are not affected by the number of links m for initial node m_0 and additional nodes [6], so $m_0 = m$ is set up with 1 for easier analysis.

The noticeable difference between the p-network and the g-network is an existence of heavy-tail distribution in the node-link connection. In heavy-tail distribution, the number of links at a specific node is much higher than in an exponential distribution. Fig. 6 depicts the node-link distribution for the theoretical Power-Law function ($P(k) \sim k^{-r}$) at $r = 3$ using a logarithmic conversion. The node-link distribution and its logarithmic conversion graphs are shown in Fig. 7 and Fig. 8 using g-and p-networks that are tested 10 times with 10,000 nodes in each case.

Since Fig. 6 and Fig. 8 show similar simple decline in

their logarithmic conversion graph, these are identical networks. The maximum link in g-network only has around 15 nodes. In contrast, the heavy-tail distribution, which has around 90~400 links, is found in the p-networks. Because Fig. 6 and Fig. 8 show similar simple decline in their logarithmic conversion graph, these are identical networks. Table 1 shows the average distribution of terminal and non-terminal (transmission) nodes in both networks. This result is achieved by 1,000 repetitions for around 10,000 nodes.

The ratio of terminal to non-terminal in the g-network is almost evenly distributed at 50:50, but the p-network has 67:33 ratios. The maximum ratio of variation is bounded at 1.2% and it satisfies the assumptions set for the p-network. Therefore, the distribution of terminal and non-terminal

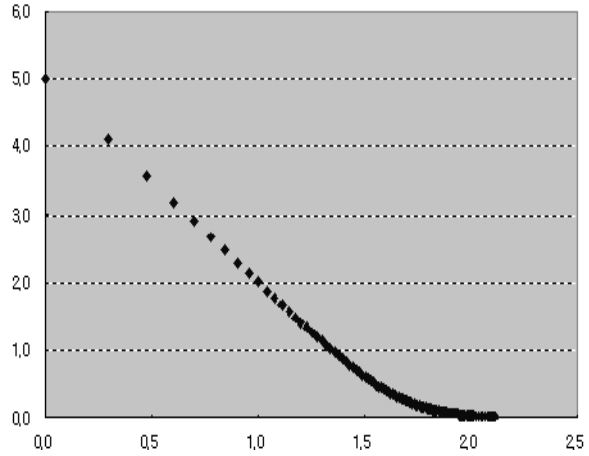


Fig. 6. The theoretical Power-Law function at $r = 3$ using a logarithmic conversion

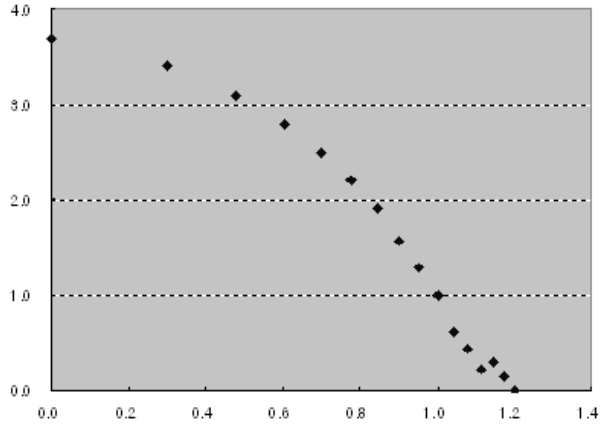


Fig. 7. The node-link distribution and its logarithmic conversion for g-network

Table 1. The distribution of terminal and non-terminal transmissions for the g-network and p-network

# of tested samples (1,000)	G-terminal	G-non-terminal	P-terminal	P-non-terminal
Average	5000.8	4999.2	6697.8	3302.2
Variation	28.6	28.6	40.8	40.8
Ratio of variation	0.6%	0.6%	0.6%	1.2%

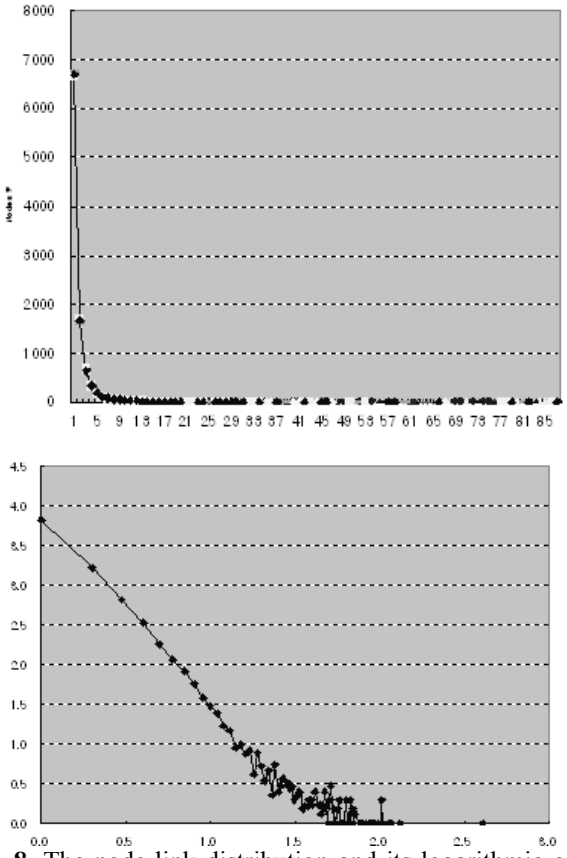


Fig. 8. The node-link distribution and its logarithmic conversion for p-network

demonstrates that the chance of the p-network being readily infected by malicious worms is approximately 33.9% higher than that of the g-network. In addition, the DDOS virus which attacks several transmission nodes can readily break down the network with trivial efforts. However, because the number of non-terminal transmissions is small, it is possible to reduce the spread of the worm when an efficient model is provided. As a result of the simulation, although the p-network includes heavy-tail distribution that can be easily destroyed by specific attacks and the distribution of terminal and non-terminal transmissions shows the ease of such infection, the p-network is a much more feasible approach to control the spread of worm infection with minimal effort when we provide an efficient and appropriate method.

4. The proposed dynamic control model for the RCS worm and its performance analysis

4.1 The proposed dynamic control model for the RCS worm

A detection and control module is in Fig. 9. Total router traffic is compared to the flow pattern database using a collection and analysis procedure. When abnormal traffic flows into the control model, the collection and analysis

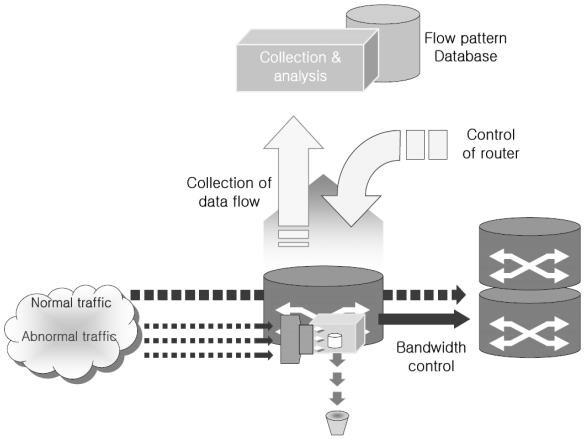


Fig. 9. The detection and control module

procedure detects and sends a command into the router to control the bandwidth. Therefore, the router reduces the bandwidth for the transmissions, which can reduce the spread of the RCS worm.

The proposed dynamic control model for the bandwidth is in Fig. 10. The p-network is the growth and preferred network model, and according to our simulation results, when the nodes are numbering at more than 20, heavy-tail distribution always appears. Relatively thick heavy-tail nodes are mostly deployed in the initially created nodes such as m_0 , so the greater part of them are in $\lim_{i \to 0} Depth(n_i)$.

When the number of total packets in m_0 is over a predefined critical value, the analysis module in the proposed spreading control model compares the total number of packets between the nodes in $Depth(n_i)$ and in m_0 and decides whether or not it is necessary to control spreading. When the traffic flow follows the RCS model or the total number of packets is over critical value during a certain period or certain time, the analysis module sends feedback to $Depth(n_i)$ to control bandwidth.

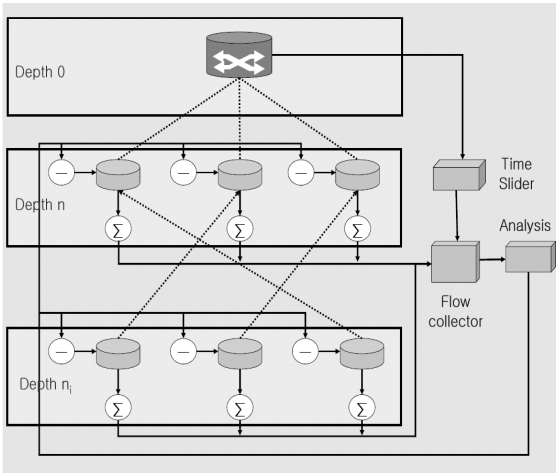


Fig. 10. The proposed dynamic control model for p-network

4.2 Performance Analysis

Fig. 11 shows the normalized result graph when the spread constant is 5 times higher (a purple curve) than the other one (a orange curve) under the same bandwidth and with the infection speed increased 6 times. When we restrict the bandwidth in 1/2 times under the same spread constant, there are 40 times the delaying effect for worm spreading in Fig. 12. Essentially, the spread constant K is the basic constant for the RCS worm, and the control of bandwidth is highly effective for a big K . The p-network, which has the characteristics of the growth and preferred network, consists of heavy-tail distribution that has high frequency for a low value spot. The average depth of the p-network, which has 10,000 nodes, is around 12, and the average number of links for each node is around 486.

When the ratio of concentration to link is increased, the overload to process the packets is also increased. The distribution effect of traffic is achieved because the high ratio of concentration nodes are located beneath depth. In Fig. 13 and Fig. 14, although we tested with different numbers of nodes, our simulated results show similar packet process rates for whole curves. This result proves that the proposed model can select an optimal depth according to the spread effect and the overload of nodes for depth. When the node has the same depth, the g-network provides a packet process rate that is 9 times higher , so the p-network is superior in controlling the spread of the worm.

Fig. 15 shows the number of nodes and accumulated number of nodes for each depth. The maximum distribution of nodes is $n_i = 5$ and high density $\lim_{i \to \infty} Depth(n_5)$.

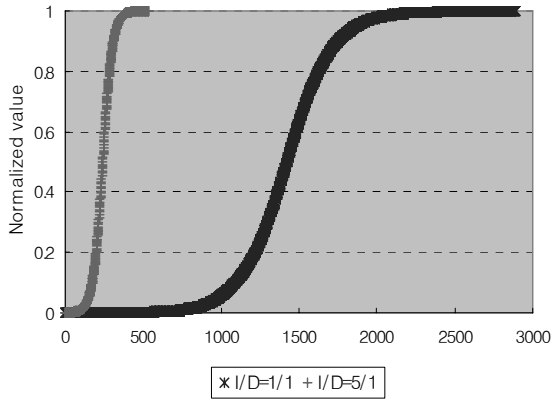


Fig. 11. The spread graph using different spread constants (K)

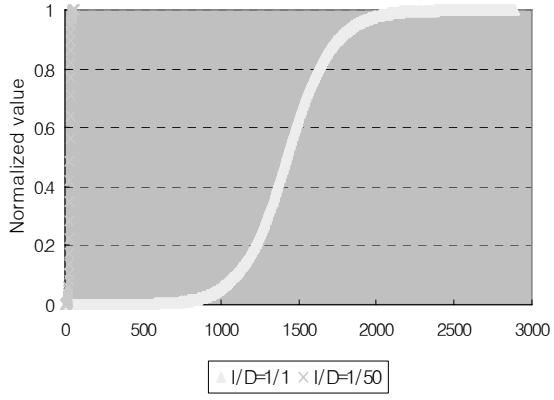


Fig. 12. The spread graph using different bandwidth

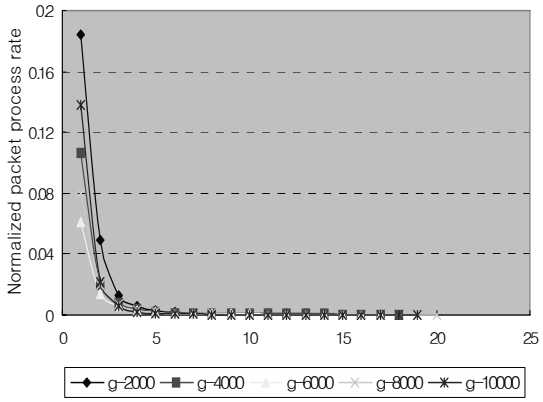


Fig. 13. The packet process ratio per node depth in g-network

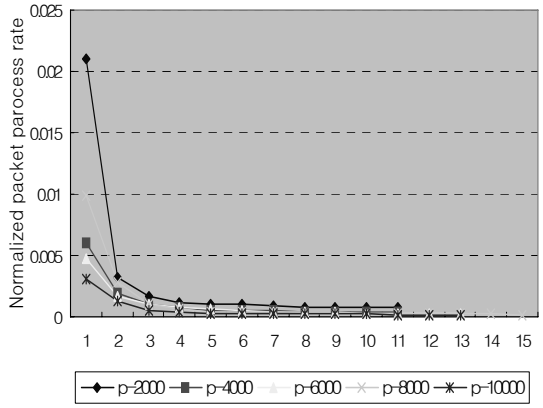


Fig. 14. The packet process ratio per node depth in p-network

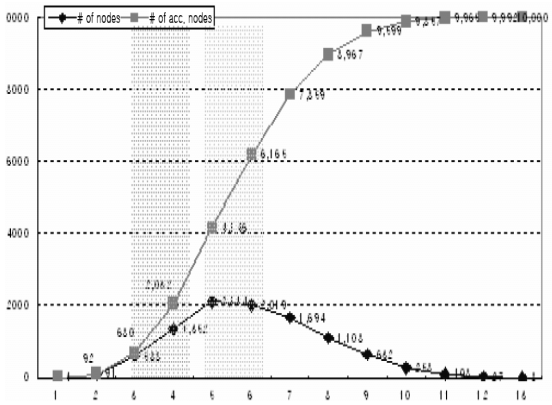


Fig. 15. The number of nodes per depth

The range of good efficiency for $Depth(n_i)$ is represented using the shadow effects in Fig. 16 and it was at a depth of $1 \le n_i \le 4$. Thus, $0 \le Depth(n_i) \le (Max(Dpeth(n_i)))_{33\%}$ is the most efficient depth for our simulated result. When we select a $Depth(n_i)$ larger than $(Max(Dpeth(n_i)))_{33\%}$, the control effects on the worm's spread is significantly reduced. In this depth range, the number of nodes for bandwidth control is 2,030 nodes (around 20%).

The measurement of overload per node is shown in Fig. 17. Node overload is relatively high at the range of $Depth(n_i) = 1$ while low at the range of $(Max(Dpeth(n_i)))_{16\%} \le Depth(n_i)$

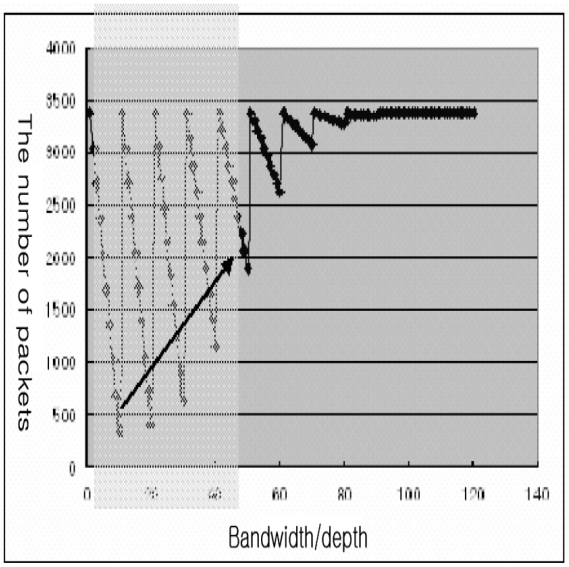


Fig. 16. The effect of spread control for bandwidth and depth

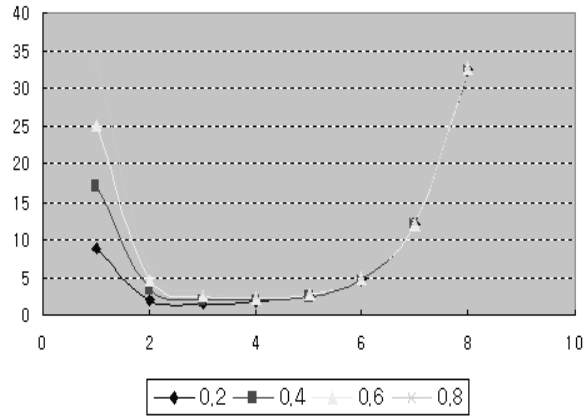


Fig. 17. The overload of nodes per depth

$\le (Max(Dpeth(n_i)))_{42\%}$. In this depth range, the number of nodes needed to control the bandwidth is around 4,100(about 41%), so it requires a relatively large number of nodes. When the packet process ratio per depth, overload of nodes and the number of nodes that should be controlled are considered in the optimal depth range is when $Depth(n_i)$ is $2 \le n_i \le 4$. Thus, the depth range is $(Max(Dpeth(n_i)))_{16\%} < Depth(n_i) < (Max(Dpeth(n_i)))_{33\%}$ and the number of nodes is around 20%.

5. Conclusion and future work

The damage of malicious worms have recently become a serious problem for internet networking. In this paper, we analyzed the spread trends of an RCS worm model that exhausts resources, and we proposed a new dynamic control model for RCS worms using depth distribution characteristics. Our simulation result provides the optimal depth range as $0 < n_i D < (Max(n_i D))_{35\%}$. In addition, using only 38% of nodes, traffic can be controlled without

overload in the same range, and the range of bandwidth was a maximum of 1/7. For future work, although $n_i D$ was applied for the whole bandwidth in this paper, we may achieve enhanced performance when we select certain ranges of bandwidth for some nodes.

References

[1] M.C. Motwani, M.C. Gadiya, R.C. Motwani, "Survey of Image Denoising Techniques", Proceedings of GSPx, Santa Clara, CA., Sep., 2004.

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

[3] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in Your Spare Time", *Proc. of the 11th USENIX Security Symposium*, pp.3-10, 2002.

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

[5] 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.

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

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

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

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

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

[11] M. Williamson, "Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code", *18th Annual Computer Security Applications Conf.*, pp.6-7, Dec, 2002

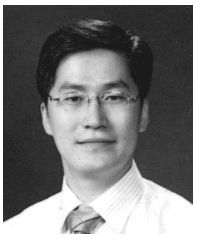
[12] N. Weaver, I. Hamadeh, G. Kesidis, V. Paxson, "Preliminary Results Using Scale Down to Explore Worm Dynamics", *Proc. of the 2004 ACM workshop on Rapid Malcode*, pp.3-6, Oct. 2004.

Byung-Gyu No

Dr. No holds a Bachelor and a Master Degree in Computer Science from Chungnam National University, Korea. In August 2005, Dr. No obtained his PhD from the Department of Computer Science at Soonchunhyang University. From 1988 to 1997, he was a senior technical researcher at the Electronics & Telecommunication Research Institute (ETRI), and he has been in his current post as a vice president of KISA since 2005, and he is also a technical expert group member of the Korea Communication Commission and expert advisory committee member of the Financial Security Agency.

Doo-Soon Park

He received his Ph.D. from Korea University, Korea. He is a professor in the Division of Computer Science and Engineering and the Director of Culture Technology at Soonchunhyang University in Korea. From 2004 to 2005, he was a visiting professor at the University of Colorado. From 2002 to 2003, he was a Dean at the Engineering College of Soonchunhyang University in Korea. Since 2000, he has been a Director in the Korean Multimedia Society. Since 2009, he has been a Director in the Korean Information Processing System. His research interests include Cloud Computing, Parallel Processing, Data Mining, and Multimedia Information Processing.



Min Hong

He is an assistant professor in the Division of Computer Science and Engineering at Soonchunhyang University. His research interests include physically-based modeling and simulation for medical applications. Min Hong received an MS in Computer Science from the University of Colorado at Boulder and a Ph.D. in Bioinformatics from the University of Colorado at Denver and the Health Sciences Center.

HwaMin Lee
She received her BS, MS and Ph.D. in Computer Science from Korea University, Seoul, in 2000, 2002 and 2006 respectively. She is currently a full-time lecturer in the Division of Computer Science & Engineering at Soonchunhyang University in Korea. Her research interests are in Cloud Computing, Grid Computing, Distributed Computing, Mobile Computing, Fault-tolerant Systems and Multi-agent System.

Yoon Sok Park
He received his BS (1999), MSEE (2002), and Ph.D. (2007) qualifications in Electrical and Computer Engineering from the University of Texas at Austin. His Ph.D. work involves the design and fabrication of MEMS-based electro/chemical sensors for detecting explosives utilizing chemiluminescence. He is currently a senior engineer at Samsung Electro-Mechanics div., and his research interests include the development of inkjet print-heads and their application in the field of printed electronics.