Detection of Greedy Nodes in Wireless LAN through Comparing of Probability Distributions of Transmission Intervals

Yuhun Han[1], Seung-Joon Seok[1*], Wang-Cheol Song[2], Duckae Choi[3] and Jee-Wan Huh[4]

[1]*Dpt. of Computer Engineering Kyungnam University*
[2]*Dpt. of Computer Engineering Jeju Nat'l University*
[3]*School of Electronics and Computer Engineering Chonnam Nat'l University*
[4]*Future Internet Research Team Nat'l Institute for Mathematical Sciences*
myhoney02@net.kyungnam.ac.kr, sjseok@kyungnam.ac.kr, siro79@empal.com, dchoi@chonnam.ac.kr, jeewan.huh@gmail.com
**Corresponding Author e-mail: sjseok@kyungnam.ac.kr*

Abstract

IEEE 802.11x CSMA/CA DCF MAC protocol supports that wireless nodes have statistically impartial probabilities of wireless channel access through fair competition. However, there is greedy node problem that maliciously increasing the transmission rates of mobile nodes altering their MAC operation disturbs fair transmissions between wireless nodes. This paper addresses how to find misbehavior greedy nodes. Previous works inspect the operation of DCF MAC protocol by the MAC frame to detect greedy nodes. In this paper, a greedy node detection algorithm using Kolmogorov-Smirnov test is proposed. The algorithm classifies wireless nodes with similar probability distributions of transmission intervals and draws a comparison between groups to find a group of greedy nodes. This paper evaluates the proposed algorithm through simulation and the simulation results shows that the algorithm can accurately detect greedy nodes in the congestion condition.

Keywords: *Greedy Node, Misbehavior, WLAN, Kolmogorov-Smirnov test*

1. Introduction

As the smart phones equipped with IEEE 802.11 Wi-Fi interface have been diffused rapidly, Internet service providers have also expanded their hot spot area. IEEE 802.11x DCF (Distributed Coordinate Function)/CSMA/CA (Carrier Sense Multiple Access/Collision Detection) MAC (Medium Access Control) Protocol provides statistically equal opportunities to transmit data frames to Wi-Fi nodes within a shared wireless channel. Recently, there has been, however, a misbehavior node problem which disrupts equal distribution of the wireless channel resources among Wi-Fi nodes. The misbehavior node problem is further classified into malicious node and selfish node problems. The malicious node interrupts other node's transmission on purpose but the selfish node intentionally transmits more data than other nodes. This paper addresses the selfish node problem.

IEEE 802.11x DCF MAC protocol uses CW (Contention Window) to decide the waiting time before a Wi-Fi node sends a data. The waiting time is the sum of DIFS and back-off delay which is randomly selected integer time slots between 0 and (CW – 1). That is, Wi-Fi nodes which want to send a data wait different back-off delay each other and avoid collisions on wireless channel. However, this can't prevent the transmission

collisions entirely and can only trying to reduce collision probability through temporal dispersion for channel access. When a node decides that its sending data has been collided, the node tries retransmission to three or six times. Since collisions happening mean that the wireless link is congested, DCF protocols distributed in WLAN make CW size twice as large every collision. To enlarge CW size, however, causes longer back-off delay, larger delay and jitter, though it reduces collision probability. Thus, DCF protocol can provide even opportunities to access to wireless channel for Wi-Fi nodes [1].

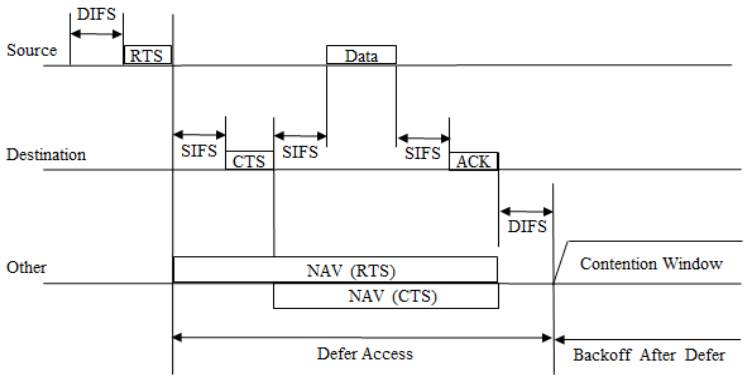


Figure 1. Shows IEEE 802.11x DCF MAC Operation

Selfish nodes artificially juggle with the parameters of DCF MAC protocol, such as DIFS, NAV (Network Allocation Vector), and CW, to occupy wireless channel more frequently than other nodes complying with the standard DCF. Also, Selfish nodes may increase CW size through intentionally making data of normal nodes collided. Thus, this selfish operations result to increase its throughput and reduce throughputs of normal nodes. The selfish node breaks no squares in uncongested conditions but we will face the serious selfish problem in network environments where more and more people use Wi-Fi devices now and forever.

In this paper, we propose an algorithm to pick out selfish nodes among nodes sharing a shared wireless channel in Wi-Fi network. Many previous approaches to solve the selfish problem use analysis of MAC operations in frame level. Some recent literatures, however, consider throughputs of nodes as a macroscopic point of view. It is because selfish nodes evidently get higher throughput than normal nodes and otherwise there is no considerable problem. Different statistics theories have been used to compare thoughts of Wi-Fi nodes. This paper also proposes a new statistical algorithm to detect selfish nodes. The algorithm firstly compares probability distributions of transmission intervals among all nodes using Kolmogorov-Smirnov test and then divides the nodes into groups by test results. Finally the algorithm tries to find the greedy node groups through comparing characteristics among groups.

1.1. Frame Level Approaches

An inspector, which an access point device generally work as, traces all frames generated by nodes and examines the MAC operations of all nodes in different points of view. First criteria, which can be considered to pick out selfish nodes, is back-off delay or the time interval between two continuative data frames sent by each node [2]. To reduce CW size or not to increase CW size after collision is an easy method which selfish nodes exploit. Since

normal nodes frequently experience collision events in congested WLAN, the inspector can detect selfish node with small back-off intervals after comparing the intervals among Wi-Fi nodes. Figure 2 shows how to measure the back-off interval at inspector.

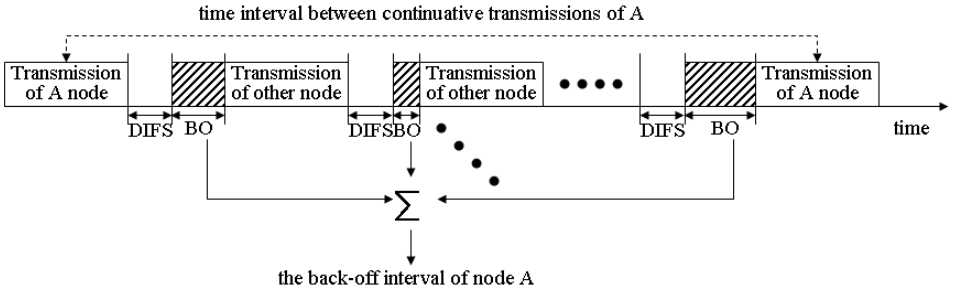


Figure 2. Measuring Back-off Intervals for Each Node [2]

The inspector can trace the sequence of frames between nodes in another way [2]. In normal case, RTS, CTS, DATA, ACK frames sequentially exchanged between sender and receiver nodes. However, a selfish node may intentionally make CTS, DATA, or ACK frames sent by normal nodes crashed after normal RTS. Selfish nodes are more sensitive to channel signal power for CCA (Clear Channel Assessment) than normal nodes. When an inspector broadcasts low-power probe messages, selfish nodes surely respond to the message [3].

1.2. Statistically Analyzing Approaches

Statistical analysis based approaches means that an inspector node periodically gathers the performance information from Wi-Fi network and then processes it. These can be used in real network, irrespective of selfish node's operations and also processing load can be reasonable. [6] proposes an algorithm using the frequency of data transmission. An inspector analyzes the frequency by a statistical method and then picks out selfish nodes. Figure 3 depicts a clear example of comparing the transmission frequency among Wi-Fi nodes. In this method, the only issue is the criterion to separate selfish nodes.

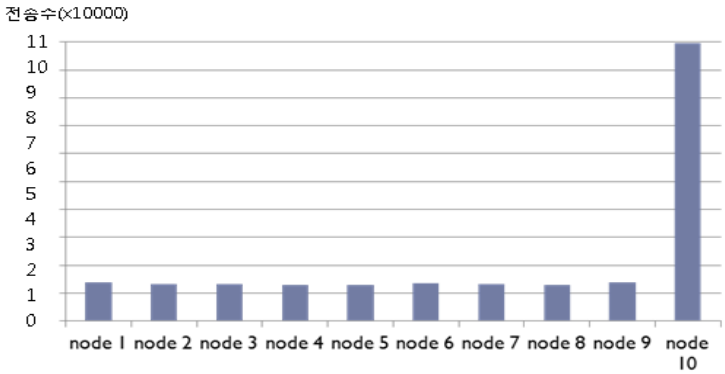


Figure 3. Comparing the frequencies of data transmission

[4] tests whether the probability distribution of first stage back-off interval follows uniform distribution or not through Kolmogorov-Smirnov test. This method considers only un-collided continuative data frames showed in Figure 2. An inspector traces the probability distribution continuously and periodically estimates the distribution to pick out selfish node

which doesn't follows to the normal back-off operation. However, this can't detect other selfish nodes complying with the normal back-off operation.

2. Kolmogorov-Smirnov Test

Kolmogorov-Smirnov (K-S) test is to comparing two cumulative distribution functions to find out whether two populations have the same probability distribution. To achieve this, K-S test uses a criteria D which is the maximum difference between two cumulative probabilities. If D value is lower than a predetermined threshold value (*Th*), we can conclude that below hypotheses H_0 is denied and H_1 is accepted.

$$D = Max\,|F(X) - G(X)|$$

Hypothesis 1: If (F(X) – G(X)) > 0, H_0: "F(X) = G(X) for all X" is correct.
Otherwise, H_1: "F(X) > G(X) for any X" is correct.
Hypothesis 2: If (F(X) – G(X)) > 0, H_0: "F(X) = G(X) for all X" is correct.
Otherwise, H_1: "F(X) < G(X) for any X" is correct.
Hypothesis 3: If D > *Th*, H_0: "F(X) = G(X) for all X" is correct.
Otherwise, H_1: "F(X) ≠ G(X) for any X" is correct.

3. A Group Based Algorithm using Transmission Interval Distribution

Greedy nodes make its probability distribution of transmission interval dense because they have shorter transmission interval than normal nodes. We use this fact to make an algorithm to pick out selfish nodes in Wi-Fi network. Also we assume that an inspector which works in access point node can monitor all frames within Wi-Fi network. The inspector periodically carries out the procedure of the proposed algorithm (Figure 4). First the inspector extracts time intervals between continuative data frame for each node and then makes probability distribution of the time intervals for each node. Next, our group based algorithm to detecting selfish nodes is carried out and the result is reported to another module which denies selfish nodes to access to wireless channel.

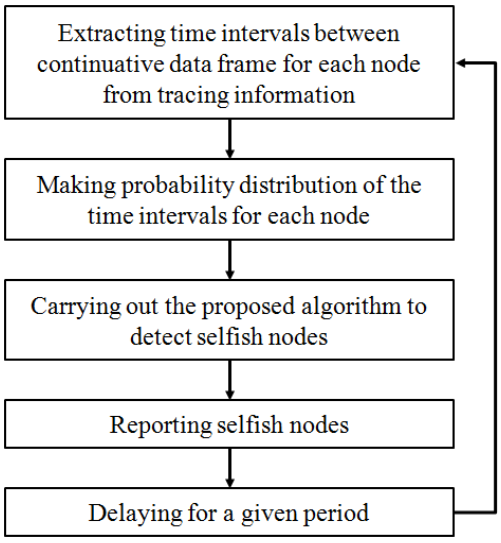


Figure 4. Periodic Procedure used to Detecting Selfish Nodes

Figure 5 is the proposed group-based algorithm using Kolmogorov-Smirnov (K-S) test among all nodes within Wi-Fi network. First the algorithm checks the collision rate of wireless link. The reason is because normal nodes are not disturbed by selfish nodes in case of under-provisioned link. Next, nodes with very low throughput are excluded from a node list because selfish nodes clearly get very high throughput. Then a node's transmission interval distribution is compared with every other's through K-S test. All nodes of the list can be classified into one or more groups according to the results of K-S test. Next, the algorithm calculates average throughput and average standard deviation of for each group. Finally the algorithm checks if the average standard deviation value of each group is lower than a threshold value in order from the group with the largest throughput to with the smallest average throughput. If it is, nodes within the group are considered as selfish nodes. Otherwise, the algorithm is finished immediately and all remaining node are considered as normal node.

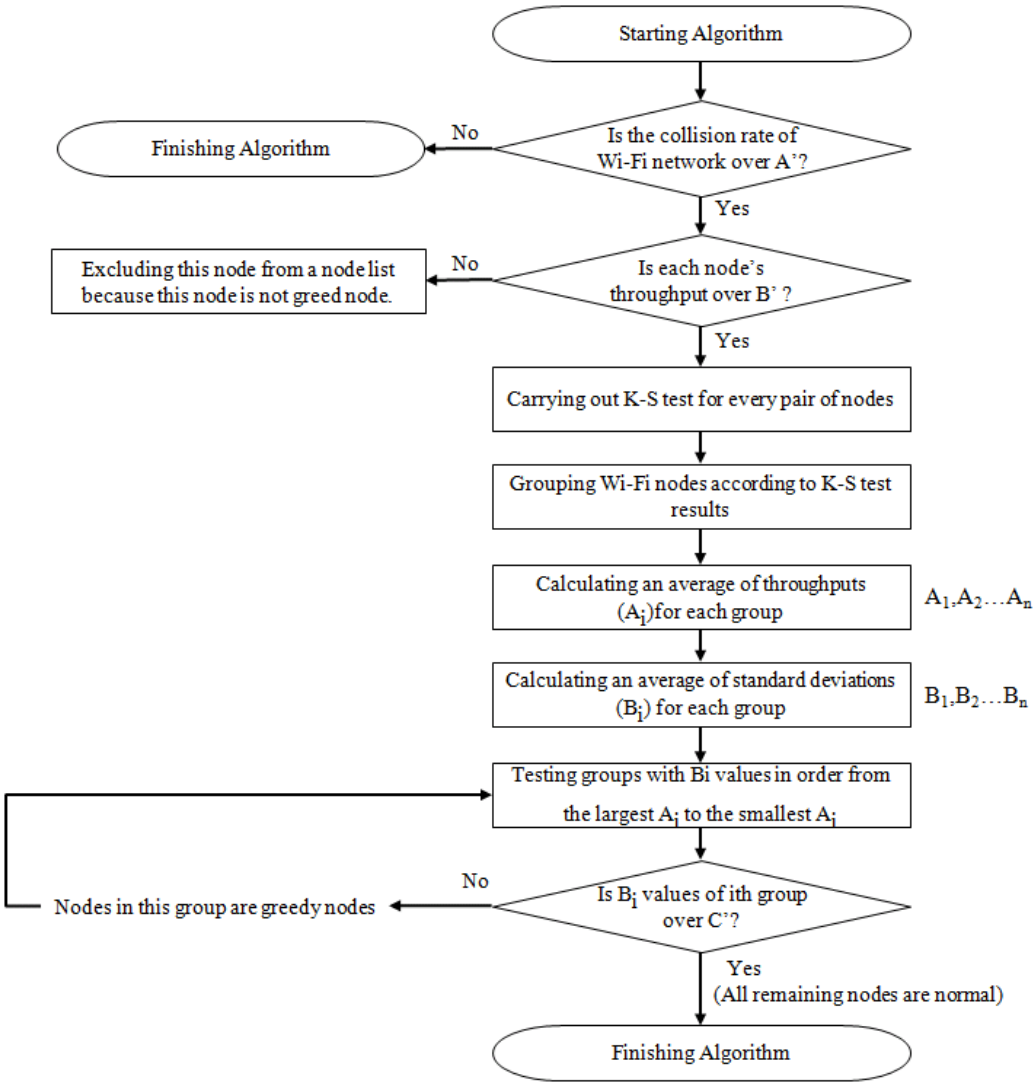


Figure 5. Proposed Group based Algorithm to Detecting Selfish Nodes using Kolmogorov-Smirnov Test

4. Performance Test

This paper evaluates the proposed algorithm, which picks out selfish nodes in Wi-Fi network, using simulation. Simulator software is ns2 (network simulator version 2) and our simulation experiments are fulfilled for a single Wi-Fi network where several Wi-Fi nodes takes communications with wired nodes via one AP (Access Point). One or more Wi-Fi nodes work for selfish ones and the orders are normal ones. Our selfish node detection algorithm is implemented in the AP. In simulation experiments, we make selfish nodes of reducing their contention window (CW) size. Three types of CW set (CWmin, CWmax), such as (8, 8), (16, 16), and (32, 32), are considered to make differential selfish degrees. We augment the number of nodes to take various congestion conditions into consideration.

The simulation results show that the proposed algorithm can pick out selfish ones among Wi-Fi nodes with considerable accuracy. In particular, our algorithm can assort selfish nodes according to selfish levels. However, the AP doesn't try to find selfish nodes in uncongested link conditions which influence to normal nodes. When the number of Wi-Fi nodes is less or equal to 5, we can consider network is uncongested.

Table 1. Simulation Results in Case of one Selfish Node: (CWmin:8, CWmax:8)

Selfish Node's CW set (#1 node)	The number of normal nodes	Simulation results	Rate of Success
(8, 8)	1 (#2 node)	Group1 = {1,2}	N/A
(8, 8)	4 (#2 ~ #5 nodes)	Group 1 = {1 ~ 5}	N/A
(8, 8)	9 (#2 ~ #10 nodes)	Group 1 = {1} Group 2 = {2 ~ 10}	100%
(8, 8)	14 (#2 ~ #15 nodes)	Group 1 = {1} Group 2 = {2 ~ 15}	100%
(8, 8)	19 (#2 ~ #20 nodes)	Group 1 = {1} Group 2 = {2 ~ 20}	100%

Table 2. Simulation Results in Case of Two Selfish Nodes: (CWmin:8, CWmax:8), (CWmin:16, CWmax:16)

Selfish Node's CW set (#1, #2 nodes)	The number of normal nodes	Simulation results	Rate of Success
(8, 8), (16, 16)		Group1 = {1,2}	N/A
(8, 8), (16, 16)	3 (#3 ~ #5 nodes)	Group 1 = {1 ~ 5}	N/A
(8, 8), (16, 16)	8 (#3 ~ #10 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3 ~ 10}	100%
(8, 8), (16, 16)	13 (#3 ~ #15 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3 ~ 15}	100%
(8, 8), (16, 16)	18 (#3 ~ #20 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3 ~ 20}	100%

Table 3. Simulation Results in Case of Three Selfish Nodes: (CWmin:8, CWmax:8), (CWmin:16, CWmax:16), (CWmin:32, CWmax:32)

Selfish Node's CW set (#1, #2, #3 nodes)	The number of normal nodes	Simulation results	Rate of Success
(8, 8), (16, 16), (32, 32)		Group1 = {1~3}	N/A
(8, 8), (16, 16) , (32, 32)	2 (#4 ~ #5 nodes)	Group 1 = {1 ~ 5}	N/A
(8, 8), (16, 16) , (32, 32)	7 (#4 ~ #10 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3} Group 4 = {4 ~ 10}	100%
(8, 8), (16, 16) , (32, 32)	12 (#4 ~ #15 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3} Group 4 = {4 ~ 15}	100%
(8, 8), (16, 16) , (32, 32)	17 (#4 ~ #20 nodes)	Group 1 = {1} Group 2 = {2} Group 3 = {3} Group 4 = {4 ~ 20}	100%

Table 4. Simulation Results in Case of Six Selfish Nodes: (CWmin:8, CWmax:8), (CWmin:16, CWmax:16), (CWmin:32, CWmax:32), (CWmin:8, CWmax:16), (CWmin:8, CWmax:32), (CWmin:16, CWmax:32)

Selfish Node's CW set (#1 ~ #6 nodes)	The number of normal nodes	Simulation results	Rate of Success
(8, 8), (16, 16) (32, 32), (32, 32)		Group1 = {1~4}	N/A
(8, 8), (16, 16) (32, 32), (8, 16)	2 (#7 ~ #8 nodes)	Group 1 = {1, 4} Group 2 = {2} Group 3 = {3} Group 4 = {7, 8}	100%
(8, 8), (16, 16) (32, 32), (8, 16) (8, 32), (16, 32)	7 (#7 ~ #13 nodes)	Group 1 = {1, 4} Group 2 = {5} Group 3 = {2} Group 4 = {6} Group 5 = {3} Group 6 = {7 ~ 13}	100%
(8, 8), (16, 16) (32, 32), (8, 16) (8, 32), (16, 32)	12 (#7 ~ #18 nodes)	Group 1 = {1} Group 2 = {4} Group 3 = {5} Group 4 = {2} Group 5 = {6} Group 6 = {3} Group 7 = {7 ~ 18}	100%
(8, 8), (16, 16) (32, 32), (8, 16) (8, 32), (16, 32)	17 (#7 ~ #23 nodes)	Group 1 = {1} Group 2 = {4} Group 3 = {5} Group 4 = {2} Group 5 = {6} Group 6 = {3} Group 7 = {7 ~ 23}	100%

5. Conclusion

This paper presents a novel algorithm that can pick out selfish nodes in hot spot area with considerable accuracy. The algorithm compares the probability distributions of transmission interval each other using Kolmogorov-Smirnov test and divides Wi-Fi nodes into several groups according to results of the comparisons. Next the algorithm estimates the characteristics of each group and then decides whether all nodes in the group are selfish nodes or not. This paper evaluates the proposed algorithm using ns2 simulator and the simulation results shows that the algorithm can accurately detect greedy nodes in the congestion condition.

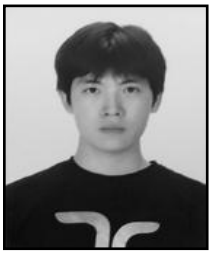
Acknowledgements

This work was supported by the National Institute for Mathematical Sciences (NIMS) grant funded by the Korea government (No. A21201)

References

[1] IEEE 802.11 Working Group, "IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", **(2007)** June.
[2] M. Raya, J. P. Hubaux and I. Aad, "DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspot", IEEE Transaction on Mobile Computing, vol. 5, Issue 12, **(2006)** December, pp. 1681-1705.
[3] K. Pelechrinis and G. Yan, "Detecting Selfish Exploitation of Carrier Sensing in 802.11 Networks", Proceeding of the IEEE INFOCOM 2009, **(2009)** April, pp. 657-665.
[4] P. Serrano, A. Banchs, V. Targon and J. F. Kukjelka, "Detecting Selfish Configurations in 802.11 WLANs", IEEE Communications Letters, vol. 14, Issue 2, **(2010)**, pp. 142-144.
[5] M. Cagalj, S. Ganeriwal, I. Aad and J. P. Hubaux, "On Selfish Behavior in CSMA/CA Networks", Proceeding of the IEEE INFOCOM 2005, **(2005)** March, pp. 2513-2514.
[6] C. E. Shannon, "Prediction and entropy of printed English," The Bell System Technical Journal, vol. 30, pp. 50-64, **(1951).**
[7] P. Kyasanur and N. Vaidya, "Selfish MAC Layer Misbehavior in Wireless Networks", IEEE Transaction on Mobile Computing, vol. 4, Issue 5, **(2005)** September, pp. 502-516.
[8] L. Guang and C. Assi, "MAC Layer Misbehavior in Wireless Networks: Challenges and Solutions", IEEE Wireless Communications, vol. 15, Issue 4, **(2008)**, pp. 6-14.
[9] O. Queseth, "The effect of selfish behavior in mobile networks using CSMA/CA", Proceeding of the IEEE 61st Vehicular Technology Conference, **(2005)** May, pp. 2157-2161.
[10] C. Wang, B. Li and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF", IEEE Transaction on Vehicular Technology, vol. 53, no. 4, **(2004)** July, pp. 1235-1243.
[11] A. Khaladj, M. Rahgozar and N. Yazdani, "The effect of decreasing CW size on performance IEEE 802.11 DCF", Proceeding of the 7th Malaysia International Conference on Communication, **(2005)** November, pp. 521-525.
[12] S. Ci and H. Sharif, "Evaluating Saturation Throughput Performance of the IEEE 802.11 MAC under Fading Channels", Proceeding of the 2nd International Conference on Broadband Networks, **(2005)** October, pp. 726-731.
[13] G. No and I. Ra, "An Efficient and Reliable DDoS Attack Detection Using a Fast Entropy Computation Method", Proceeding of the 9th International Symposium on Communications and Information Technology, **(2009)** September, pp. 1223-1228.
[14] Q. Qian, H. -Y. Che and R. Zhang, "Entropy Based Method for Network Anomaly Detection", Proceeding of the 15th IEEE Pacific Rim International Symposium on Dependable Computing, **(2009)** November, pp. 189-191.
[15] A. Ziviani, A. Tadeu, A. Gomes, M. L. Monsores and P. S. S. Rodrigues, "Network Anomaly detection using Nonextensive Entropy", IEEE Communication Letters, vol. 11, Issue 12, **(2007)**, pp. 1034-1036.

Authors



Yuhun Han

Yuhun Han received the B.S. degree and M.S. degree in computer engineering from Kyungnam University in 2008 and 2010, respectively. Since 2010, he has been a researcher of mirae corporation. His research area covers Internet protocols, network security, and wireless network.

Seung-Joon Seok

Seung-Joon Seok received his B.S. degree in electronics engineering from Konkuk University, Seoul, Korea in 1997 and the M.S. and Ph. D degrees in electronics engineering from Korea University, Seoul, Korea, in 1999 and 2003, respectively. Since 2004, he has worked for computer engineering department, Kyungnam University. His research interests include quality of service model of Internet, TCP performance in wireless Internet, traffic engineering, and MAC of wireless network, future Internet.

Wang-Cheol Song

Wang-Cheol Song received the B.S. degree in Food Engineering and Electronics from Yonsei University, Seoul, Korea in 1986 and 1989, respectively. And he received his M.S. and PhD in Electronics from Yonsei University, Seoul, Korea, in 1991 and 1995, respectively. Since March 1996, he has been a professor of Department of Computer Engineering, Jeju National University, Korea. His research interests include VANETs and MANETs, Future Internet, Network Security, and Network Management.

Deokjai Choi

Deokjai Choi received the B.S. degree in Computer Engineering from Seoul National University, Seoul, Korea in 1982, and his M.S. from KAIST in Computer Science in 1984, and Ph. D in Computer Science and Telecommunication Program from University of Missouri-Kansas City in 1995. Since March 1996, he has been a professor of Department of Computer Engineering, Chonnam National University, Korea. His research interests include Context Aware Computing, Future Internet, Network Security, and Network Management.

Jee-Wan Huh

Jee-Wan Huh received his M.S and Ph.D degree from Jeju National University all in Computer Engineering. Since 2010, he has joined National Institute for Mathematical Sciences (NIMS) Korea as a Resercher. His major research interests include Future Internet, Nature-inspired routing and Information-Centric Network, and its applications in networks.