REAL-TIME MISBEHAVIOR DETECTION IN WIRELESS NETWORKS WITH FALSE POSITIVE RATE REDUCTION

[1]Pavithran A, [2] Raja A
[1]Final Year PG Scholar, [2] Assistant Professor
[1]Master of Computer Applications, [2] Master of Computer Applications
[1]Kongu Engineering College, Perundurai, Tamil Nadu, India, [2]Kongu Engineering College, Perundurai, Tamil Nadu, India.

Abstract In wireless network, every node accesses the network in a cooperative manner and randomly delays transmissions to avoid collisions by following a common backoff rule. However, in such a distributed environment without a centralized controller, a malicious node may deliberately choose a smaller backoff timer and selfishly gain an unfair share of the network throughput at the expenses of other normal nodes channel access opportunities. The distributed nature of the CSMA/CA-based wireless protocols allows malicious nodes to deliberately manipulate their backoff parameters and, thus, unfairly gain a large share of the network throughput. This paper designs a real-time backoff misbehavior detector, termed as the fair share detector (FS detector), which exploits the nonparametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any a priori knowledge of the statistics of the selfish misbehavior. While most of the existing schemes for selfish misbehavior detection depend on heuristic parameter configuration and experimental performance evaluation, the papers develop a Markov chain-based analytical model to systematically study the performance of the FS detector in real-time backoff misbehavior detection. The paper quantitatively computes the system configuration parameters for guaranteed performance in terms of average false positive rate, average detection delay, and missed detection ratio under a detection delay constraint. We present thorough simulation results to confirm the accuracy of the theoretical analysis as well as demonstrate the performance of the developed FS detector. In addition, this paper systematically studies the generic scenario with multiple misbehaving nodes in a multihop wireless network.

Keyword: Backoff, Fair share detector, Markov chain based Analytical Model, Multihop.

I.INTRODUCTION

The first step in the software development life cycle is the identification of the problem. As the success of the system depends largely on how accurately a problem is identified. Wireless ad hoc networks have attracted much attention due to their mobility and ease of deployment. However, the wireless and dynamic natures render them more vulnerable to various types of security attacks than the wired networks. The challenges, in this paper first design a real-time backoff misbehavior detector, termed as the fair share detector (FS detector), which exploits the nonparametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any a priori knowledge of the statistics of the selfish misbehavior. The proposed work develops a robust detector for backoff misbehavior detection based on the Kolmogorov-Smirnov (K-S) test, without a priori knowledge of the misbehavior strategy either. The detector resorts to estimating the collision probability of a transmission to establish the distribution of the idle time between two consecutive successful transmissions from a tagged node. The collision estimation is, however, done with an approximate method for short detection delay; such an approximation in fact negatively impacts the performance in both false positive rate and detection delay, to be discussed in detail in Section 7. In our preliminary work and adopt the nonparametric CUSUM test for the backoff misbehavior detection as well. The detector directly counts the number of successful transmissions of a tagged node within an observation window to obtain a sample. The observation window needs to linearly increase with the number of nodes in the network to fairly count transmissions from each node, which as a result will increase the detection delay. The FS detector newly developed in this paper takes each successful transmission over the network as its observation sample. Such a sampling method is independent of the network size and turns out to result in good performance in both false positive rate and detection delay. Also, the FS detector does not require any modification to the protocols, and can be implemented by any node assuming the role of the detection agent that monitors the network. The proposed system develops an analytical model for the FS detector, which can provide quantitative performance analysis and theoretical guidance on system parameter configuration. Specifically, use a discrete-time Markov chain to model the behavior of the detector, because the detector's next state depends only on its current value and the coming observation sample. This Markov chain-based model enables us to conduct rigorous quantitative analysis of the FS detector on three fundamental metrics: average false positive rate, average detection delay, and missed detection ratio, and further compute the system configuration for guaranteed performance. In particular, the Markov chain modeling the FS detector takes different transition probabilities under the normal traffic condition and under the abnormal condition with misbehaving nodes present, respectively. The Markov chain obtained from the normal traffic condition can be used to directly calculate the average false positive rate and also provide the initial states for misbehavior analysis. Based on these initial states, we can then use the Markov chain under the abnormal conditions to analyze the average detection delay and the missed detection ratio. Note that the missed detection ratio is not often considered in the context of the CUSUM test due to its "nonstop until detection" property. In this paper, we examine a missed detection ratio under a detection delay constraint, which is of importance regarding real-time detection. The objective of the paper is:

- The false positive problem cannot be observed by simulating the same network using popular ad hoc network simulators.
- Exact quantitative evaluations of false positives in monitoring-based intrusion detection for ad hoc networks.
- Previous studies showed that the simulated network exhibits the aggregate false positive behavior but not much similar to real time high false positives.
- In Wireless networks that provide resilience to Byzantine failures caused by individual or colluding nodes
- To verify that a public key belongs to an individual and to prevent tampering and forging
- To provide secure communications
- To mitigate malicious attacks on the network.
- To identified the attack as soon as possible.

II. Related Works

In the paper "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems" the authors Tao Peng, Christopher Leckie, And Kotagiri Ramamohanarao presented a survey of denial of service attacks and the methods that have been proposed for defense against these attacks. In this survey, they analyzed the design decisions in the Internet that have created the potential for denial of service attacks. They reviewed the state-of-art mechanisms for defending against denial of service attacks, compare the strengths and weaknesses of each proposal, and discuss potential countermeasures against each defense mechanism.

In the paper "PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks" [12] the Distributed Denial-of-Service (DDoS) attacks are a critical threat to the Internet. The paper introduces a DDoS defense scheme that supports automated online attack characterizations and accurate attack packet discarding based on statistical processing. The key idea is to prioritize a packet based on a score which estimates its legitimacy given the attribute values it carries. Once the score of a packet is computed, this scheme performs score-based selective packet discarding where the dropping threshold is dynamically adjusted based on the score distribution of recent incoming packets and the current level of system overload. The paper described the design and evaluation of automated attack characterizations, selective packet discarding, and an overload control process.

In the paper "Defense Against Spoofed IP Traffic Using Hop-Count Filtering" the authors Haining Wang Cheng Jin and Kang G. Shin stated that IP spoofing has often been exploited by Distributed Denial of Service (DDoS) attacks to (1) conceal flooding sources and dilute localities in flooding traffic, and (2) coax legitimate hosts into becoming reflectors, redirecting and amplifying flooding traffic. Thus, the ability to filter spoofed IP packets near victim servers is essential to their own protection and prevention of becoming involuntary DoS reflectors

In the paper "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis" the authors Yu Chen and Kai Hwang presented a new spectral templatematching approach to countering shrew distributed denial-of-service (DDoS) attacks. These attacks are stealthy, periodic, pulsing, and low-rate in attack volume, very different from the flooding type of attacks. They are launched with high narrow spikes in very low frequency, periodically. Thus, shrew attacks may endanger the victim systems for a long time without being detected. In other words, such attacks may reduce the quality of services unnoticeably.

In the paper "Robust and efficient detection of DDoS attacks for large-scale internet" the authors Kejie Lu, Dapeng Wu, Jieyan Fan, Sinisa Todorovic and Antonio Nucci stated that in recent years, distributed denial of service (DDoS) attacks have become a major security threat to Internet services. How to detect and defend against DDoS attacks is currently a hot topic in both industry and academia. In the paper, they proposed a novel framework to robustly and efficiently detect DDoS attacks and identify attack packets. The key idea of their framework is to exploit spatial and temporal correlation of DDoS attack traffic. In this framework, we design a perimeter-based anti-DDoS system, in which traffic is analyzed only at the edge routers of an internet service provider (ISP) network.

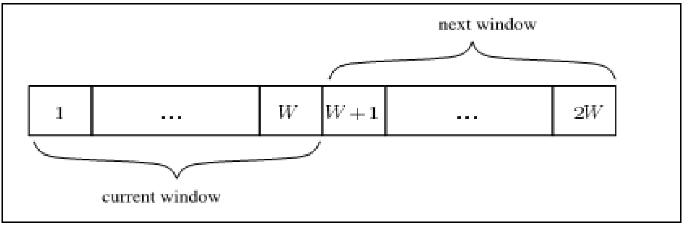
III. *METHODOLOGY*

Fixed Window Protocol

In monitoring-based intrusion detection, each node monitors the forwarding behavior of its neighboring nodes. In most cases, a node only monitors its next hop in a route. For monitoring purpose, nodes keep track of a window of packets that it sent recently to its next hop. Two types of window can be used to keep track of monitoring: fixed window or sliding window. To understand the similarities and differences between the fixed and sliding windows, assume that noise does not impact the overhearing of transmission within a node's radio range. In such a scenario, a malicious node can drop up to L-1 packets out of W on the average without risking suspicion by neighbors. The temporary drop rates can be different.

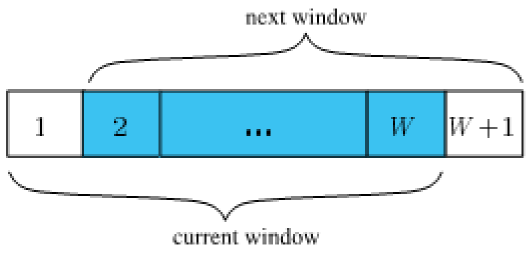
The sliding window approach is free of this deficiency since in any consecutive W-transmitted packets, a malicious node may drop at most L-1 packets without risking suspicion by neighbors. To model the state of sliding window –based monitoring using a discrete-time Markov chain. The purpose of the Markov model is to determine analytically the expected time to suspect its next hop by a monitoring node. Markov models are commonly used to analyze the expected time to encounter a bug in a software system. The fixed window protocol monitors the packet drops detection by checking the front and rear side of the packet. So the drop detection can be finding effectively. The sliding window protocol monitors the packet drop detection in the sequence of packets. It is possible to reduce the number of false positive due to monitoring by having higher threshold values, allowing a node to exceed the not-overheard threshold multiple times before labeled as suspicious, or both. This will mitigate the false positive problem in normal networks without attacks.

Fixed Window protocol Based Sender Form



In this form, the current system IP Address is fetched and used as sender node. The numbers of packets being sent every time, the drop packets threshold count are selected using combo box controls. In addition, this form contains option to listen the packets dropped by the receiver node. The packets dropped at the starting and ending side of same frame are taken into consideration during packet drop listening.

Slice Window Based Sender Form



In this form, the IP address is automatically filled in the textbox control during the form load event. Then the destination IP is given in the textbox control. Then by clicking the packets send button, the packets are sent to the receiver application and the details are stored in the PacketsIn table. Then by clicking the listen packet button the drop detection can be displayed in the list box control.

Protocol: Formal Specification

A. DATA STRUCTURES

First we define the data structure of the protocol. For the sender, the window "slides" over the infinite input sequence input. They are not specify the nature of the frames in the input sequence. Variable first denotes the first frame in the sending window, ftsend is the first frame that is not sent yet, and we always have first (s_0) ≤ ftsend (s_0) ≤ first (s_0) + N. Thus, at any moment of time, frames with indices from first to ftsend-1 (if any) are already sent but not yet acknowledged, and frames with indices from ftsend to first + N-1 (if any) are in the sending window but not yet sent. Variable tackmax expresses the time when received the acknowledgment with the maximum sequence number K-1 for the last time. As a time domain Time, take the set of non-negative real numbers.

<u>Sender:</u>

1) Input: sequence [Frames]
2) First: nat,
3) Ftsend: nat,
4) Tackmax: Time

For the receiver, output is the output sequence, buffer is a record with two fields snumber and frn, that represents the receiving window with N elements (which are either frames or empty spaces, denoted by ε), last del is the last delivered sequence number, acklastdel is a Boolean variable which tells whether they are allowed to send the acknowledgment for lastdel to the sender, delmax is a boolean variable which tells whether we already delivered the maximum sequence number K - 1 at least once, and variable tdelmax expresses the time when we delivered the frame with the maximum sequence number K -1 for the last time (the importance of variables tackmax and tdelmax is explained in subsection II-A).

<u>Receiver:</u>

1) $output$: $finite\_sequence[Frames]$,
2) $buffer$: $\{0, 1, \ldots N-1\} \longrightarrow$
   $(snumber : \{0, 1, \ldots K-1\},$
   $frn : Frames \cup \{\varepsilon\})$,
3) $lastdel$: $\{0, 1, \ldots K-1\}$,
4) $acklastdel$: $bool$,
5) $delmax$: $bool$,
6) $tdelmax$: $Time$

The frame channel and the acknowledgment channel are represented by its contents, namely a set of frame messages and a set of acknowledgment messages, respectively. Besides a sequence number and possibly a frame, each message includes its timeout, i.e. the latest time when it must be removed from the channel. When a message is sent, we assign as its timeout the current time plus Lmax, where Lmax is the maximum message lifetime. Although timeout is formally a part of a message, it is never used by the recipient of this message.

<u>Frame Message:</u>
1) Snumber: {0... 1…K-1}
2) Frame: Frames,
3) Timeout: Time

<u>ACK Message:</u>
1) Snumber: {0... 1…K-1}
2) Timeout: Time

The complete state of the protocol consists of the sender, the receiver and the two channels fchannel and achannel, together with the variable time, indicating the current time.

<u>State:</u>
1) Sender: Sender,
2) Receiver: Receiver,
3) Fchannel $\subseteq$ FrameMessage,
4) Achannel $\subseteq$ AckMessage,
5) Time: Time

<u>InitialState:</u>

1) $sender$
   1) $input = arbitrary\ sequence\ of\ frames$,
   2) $first = 0$,
   3) $ftsend = 0$,
   4) $tackmax = 0$
2) $receiver$
   1) $output = empty\ sequence$,
   2) $buffer = \forall (i : \{0, 1, \ldots N-1\}) :$
      $(snumber = i, frn = \varepsilon)$,
   3) $lastdel = 0$,
   4) $acklastdel = FALSE$,
   5) $delmax = FALSE$,
   6) $tdelmax = 0$
3) $fchannel = \emptyset$,
4) $achannel = \emptyset$,
5) $time = 0$

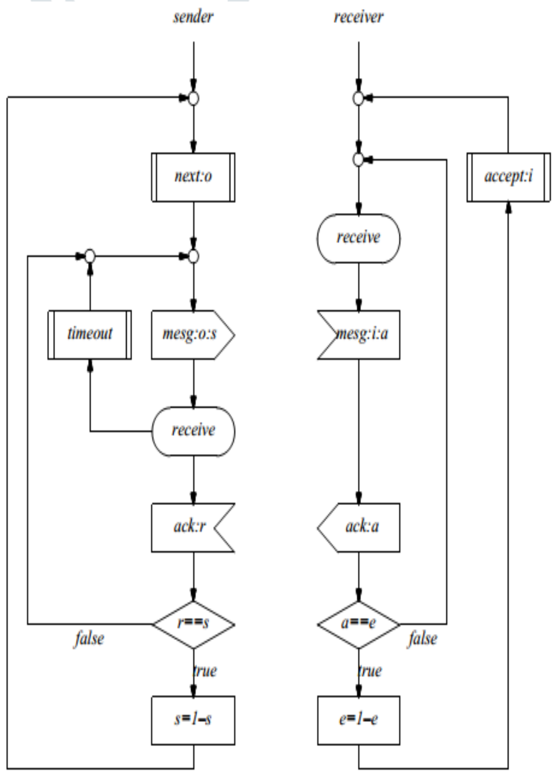


FIG 3.1 PROTOCOL PROCESS- FWP

The initial state of the protocol is defined below in a rather obvious way. The only subtlety is the values of tackmax, lastdel and tdelmax; they are initialized as 0, but we can easily determine from other variables that these values are initial and should not be used.

The protocol is specified by a state machine with 7 atomic actions: 1 general, 3 for the sender and 3 for the receiver, where some actions have a parameter. Below we show the precondition and the effect of each of them, using some abbreviations and PVS-like notation

B. THE DELAY ACTION

Delay(t)

Precondition:

$$\forall fm:\ fm \in fchannel(s0) \Longrightarrow$$
$$time(s0) + t \le timeout(fm),$$
$$\forall am:\ am \in achannel(s0) \Longrightarrow$$
$$time(s0) + t \le timeout(am)$$

Effect:

$$s1 = s0\ with\ [time := time(s0) + t]$$

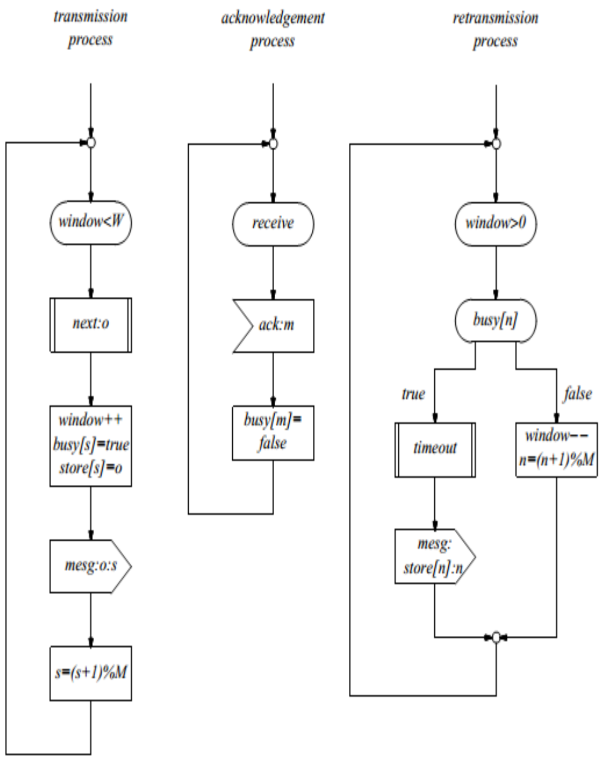


Fig 3.2 Protocol Process- SWP

Action Delay (t) expresses the passing of t units of time. The precondition of this action, using a "time-lock" construction, ensures that any message in a channel is removed from the channel before its timeout expires.

C. ACTIONS OF THE SENDER

The precondition Reuse Zero Pre of action Send expresses that sequence number 0 can only be reused after more than Lmax time units has passed since the last acknowledgment of sequence number K -1 (and only if all preceding frames have already been acknowledged). The precondition of action Resend allows resending any frame in the sending window that has been already sent. In the effect of action Receiveack, a set is a set of indices of frames in the sending window that are acknowledged by the acknowledgment message am. It is easy to see that this set consists of at most one index

$$ReuseZeroPre(s0) =$$
$$(rem(K)(ftsend(s0)) = 0\ \&\ ftsend(s0) > 0) \Longrightarrow$$
$$(ftsend(s0) = first(s0)\ \&$$
$$time(s0) > tackmax(s0) + Lmax)$$

IN

$$first(s0) \le ftsend(s0) < first(s0) + N,$$
$$ReuseZeroPre(s0)$$

Effect:

LET

$$SendNS(s0) = s0\ with$$
$$[ftsend := ftsend(s0) + 1]$$

IN

$$s1 = SendNS(s0)\ with$$
$$[fchannel := fchannel(s0) \cup$$
$$\{(rem(K)(ftsend(s0)),$$
$$input(s0)(ftsend(s0)),$$
$$time(s0) + Lmax)\}]$$
$$\vee\ s1 = SendNS(s0)$$

Resend(i)

Precondition:

$$i \ge first(s0),$$
$$i < ftsend(s0)$$

Effect:

$$s1 = s0\ with$$
$$[fchannel := fchannel(s0) \cup$$
$$\{(rem(K)(i),\ input(s0)(i),\ time(s0) + Lmax)\}]$$
$$\vee\ s1 = s0$$

Receiveack(am)

Precondition:

$am \in achannel(s0)$

Effect:

LET

$aset(s0,\ am)\ =\ \{\ j\ |\ j \geq first(s0)\ \&$
$\quad j < ftsend(s0)\ \&$
$\quad rem(K)(j) = snumber(am)\ \}$

AND

$RANS(s0,\ i,\ bk) = s0\ with$
$\quad [first := i,$
$\quad tackmax :=\ \text{if } bk = K - 1 \text{ then } time(s0)$
$\qquad\qquad \text{else } tackmax(s0)]$

IN

$\text{if } aset(s0, am) \neq \emptyset \text{ then}$
$\quad s1 = RANS(s0, choose(aset(s0, am)) + 1,$
$\qquad snumber(am))\ with$
$\qquad [achannel := achannel(s0) \setminus \{am\}]$
$\quad \vee\ s1 = RANS(s0,\ choose(aset(s0, am)) + 1,$
$\qquad snumber(am))$
$\text{else } s1 = s0\ with$
$\qquad [achannel := achannel(s0) \setminus \{am\}]$
$\quad \vee\ s1 = s0$

D. ACTIONS OF THE RECEIVER

The precondition of action Receive ensures that can only receive messages after more than Lmax time units have been passed since the last delivery of a frame with sequence number K - 1. In the effect of the Receive action, f set is a set of indices in the receiving window into which the frame from message f m can be inserted. It is easy to see that this set consists of at most one index.

Here for a frame fr, one (fr) denotes the sequence of frames of length one with the only element fr; o is the operator that performs concatenation of two finite sequences of frames; and shift is the operator that removes the first element of a buffer and adds another element to its end, i.e. for a buffer buff with N elements and a buffer element be, the expression shift (buff, be) is defined as follows:

$shift(buff, be)\ =\ \forall\,(i\ :\ \{0,\ 1,\ \ldots\ N-1\}):$
$\quad \text{if } i < N-1 \text{ then } buff(i+1) \text{ else } be$

Receive(fm)

Precondition:

$fm \in fchannel(s0),$
$delmax(s0) \Longrightarrow time(s0) > tdelmax(s0) + Lmax$

Effect:

LET

$fset(s0,\ fm)\ =\ \{\ j\ |\ j < N\ \&$
$\quad snumber(buffer(s0)(j)) = snumber(fm)\ \&$
$\quad frn(buffer(s0)(j))\ = \varepsilon\ \&$
$\quad snumber(buffer(s0)(j)) \geq j\ \}$

AND

$RNS(s0,\ bn,\ fr) = s0\ with$
$\quad [buffer := buffer(s0)\ with$
$\qquad [(bn) := (snumber(buffer(s0)(bn)),\ fr)]]$

IN

$\text{if } fset(s0, fm) \neq \emptyset \text{ then}$
$\quad s1 = RNS(s0, choose(fset(s0, fm)),$
$\qquad frame(fm))\ with$
$\qquad [fchannel := fchannel(s0) \setminus \{fm\}]$
$\quad \vee\ s1 = RNS(s0,\ choose(fset(s0, fm)),$
$\qquad frame(fm))$
$\text{else } s1 = s0\ with$
$\qquad [fchannel := fchannel(s0) \setminus \{fm\}]$
$\quad \vee\ s1 = s0$

Sendack

Precondition:

$acklastdel\ =\ TRUE$

Effect:

LET

$SendackNS(s0)\ = s0\ with$
$\quad [acklastdel := \text{if } lastdel(s0) = K - 1$
$\qquad \text{then } FALSE \text{ else } acklastdel(s0)]$

IN

$s1\ = SendackNS(s0)\ with$

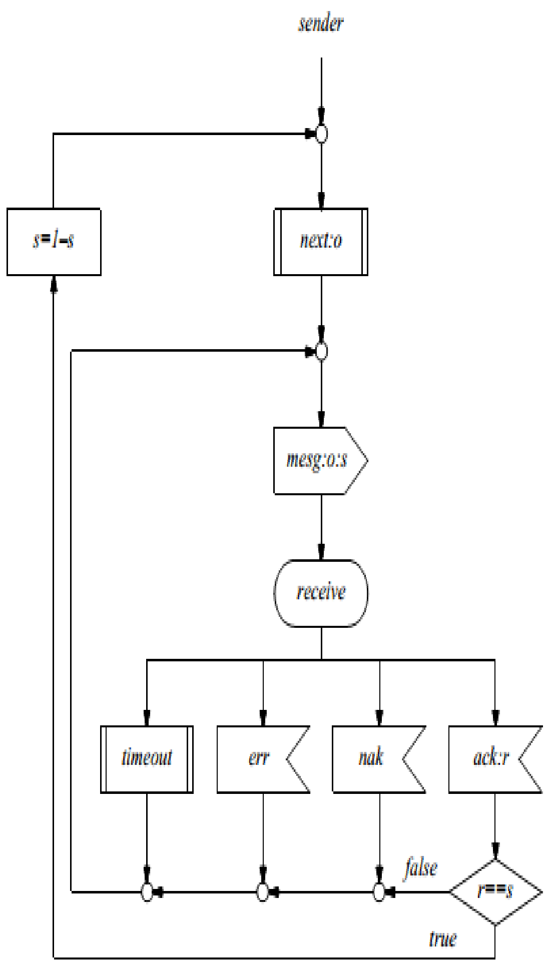


Fig 3.3 Protocol Process- MFWP-MSWP Model

IV EXPERIMENTAL RESULTS

Comparison of Packet Drop between hope and multi hope protocols

The Packet drop count of existing hope base fixed and sliding window protocol is compared with the proposed Marko chain model for Multi-Hop Wireless Networks. The Packet drop count of existing protocol is drop count threshold 88 (ex: single hope). The packet drop count for the proposed protocol is 31(ex: multi hope). The comparison of packet drop count in FWP-SFP and MFWP-MSWP

PACKETS	FWP-SWP OF PACKET DROP COUNT	MFWP-MSWP OF PACKET DROP COUNT
25	13	7
50	23	19
75	33	29
100	43	35

Table 4.1 Comparison of Packet Drop count of FWP-SFP and MFWP-MSWP

The Packet drop count of existing hope base fixed and sliding window protocol is compared with the proposed Marko chain model for Multi-Hop Wireless Networks.

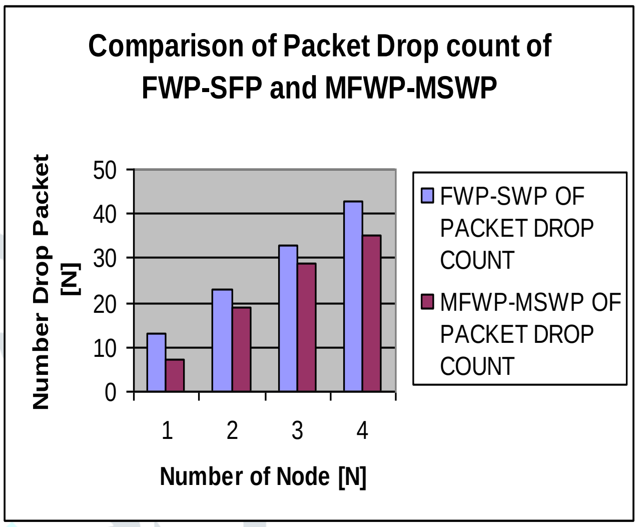


Fig 4.1 Comparison between Packet Drop counts

5.2.2. Performance analysis for existing system Cluster base Revocation Certification

The **Table 5.2** represents experimental result for existing system. The finding malicious node and revocation node process within second details and Mines details as followed.

S.NO	REVOCATION TIME (SEC)	NO.OF ATTACKER NODES	AVERAGE OF ATTACKER PER MINS (Throughput) (%)
1	100	125	3.68
2	200	195	10.67
3	300	356	25.38
4	400	384	38.22
5	500	475	60.41
6	600	566	90.63

Table 4.2 Average of attacker node finding in Existing System

The **Figure 4.2** represents experimental result for existing system. The finding malicious node and revocation node process within Minis details as followed.

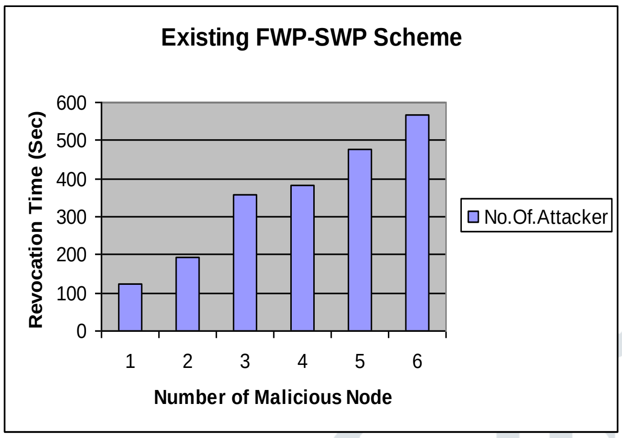


Fig 4.2 Existing FWP-SWP- Number of Attacker

The **Figure 5.3** represents experimental result for proposed system. The finding malicious node and revocation node process within Minis details as followed.

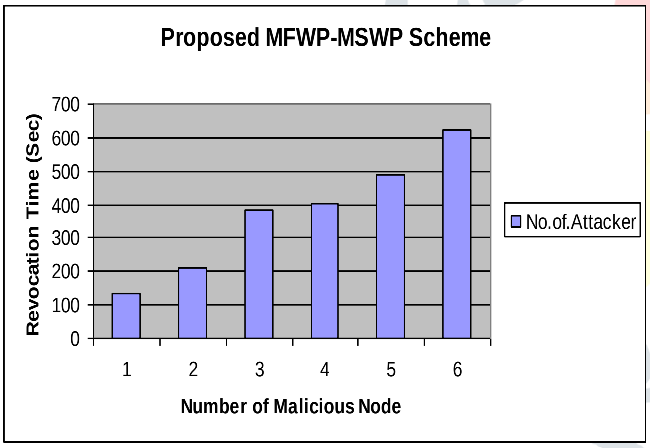


Fig 4.3 MFWP-MSWP - Number of Attacker

V. CONCLUSION

This paper proposes a novel fair share detector for real-time backoff misbehavior detection in wireless networks. While most existing work for backoff misbehavior detection depends on heuristic parameter configuration and experimental performance evaluation, we are able to use our model for a quantitative study to achieve guaranteed detection performance in terms of average false positive rate, average detection delay, and missed detection ratio. The paper also presents quantitative evaluations of false positives in monitoring-based intrusion detection for ad hoc networks. It showed that, even for a simple three-node configuration, an actual ad hoc network suffers from high false positives. However, this problem of false positives cannot be observed by simulating the same three-node network using popular ad hoc network simulators, because they do not simulate the noise seen in actual network environments. By implementing the fixed window and the sliding window protocol the packet drop detection can be overcome. The proposed system eliminates the problem in the existing system. And the proposed protocol is easy to implement the thesis. The proposed is very cost effective. So the proposed system overcomes the false positive problem. It is possible to retransmit the lost packet and immediately it sends the lost packet during the drop detection. It sends the acknowledgement for retransmitting the lost packet. So the false positive problem and the packet loss problem can be overcome in this paper.

The paper is fortified with the noise model to study the impact of monitoring-based intrusion detection on larger ad hoc networks. Our results indicate two potential problems with monitoring-based IDT:

- IDT may reduce performance of a normal network, especially when the network is not dense, and
- IDT may not improve the network throughput since any mitigation of packet dropping by malicious nodes is offset by suboptimal paths used owing to false positives.
- This can increase overhead intrusion detection and may deter its use. In light of that our results indicate a fundamental problem with monitoring based IDTs: the key technique used is unreliable, and any detection process based on it is likely to be error prone.
- In future, it intends to investigate the effectiveness of probing techniques in the presence of colluding attackers. It also would like to develop new intrusion detection techniques that avoid the problems of passive monitoring.
- If it is developed as website it can be accessed from anywhere and the monitoring is improved by implementing the advanced protocol to find the

intrusion activities. Then it improves the efficiency of the network for sending the packets.

- The advanced algorithm is chosen for monitoring the packet drop detection and also to retransmit the lost packet. So the cost of the algorithm can be reduced.
- To avoid the packet loss in prior.
- In future, we plan to study the specific scenario with multiple misbehaving nodes in a multihop wireless network with false positive rate taken into consideration.

REFERENCES

1. Yoohwan Kim,Wing Cheong Lau,Mooi Choo Chuah and H. Jonathan Chao “PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks” IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 3, NO. 2, APRIL-JUNE 2006.
2. Michael T. Goodrich, “Probabilistic Packet Marking for Large-Scale IP Traceback” IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. X, NO. X, JANUARY 2007.
3. Shui Yu,Wanlei Zhou, and Robin Doss,” Information Theory Based Detection Against Network Behavior Mimicking DDoS Attacks” IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 4, APRIL 2008.
4. Yang Xiang, Wanlei Zhou, and MinyiGuo “Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 5, MAY 2009.
5. Akash Mittal, Prof. Ajit Kumar Shrivastava, Dr. Manish Manoria”A Review of DDOS Attack and its Countermeasures in TCP Based Networks” International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011.
6. Tao Peng, Christopher Leckie, And Kotagiri Ramamohanarao “Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems” ACM Computing Surveys, Vol. 39, No. 1, Article 3, Publication date: April 2007.
7. N. Syed Siraj Ahmed and D. P. Acharjya “Detection of Denial of Service Attack in Wireless Network using Dominance based Rough Set” (JACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 12, 2015.
8. LuKaszApiecionek, Jacek M. Czerniak, and Wojciech T. Dobrosielsk “Quality of Service Method as DDoS protection Tool” D.Fileve et al. (eds.), Intelligent Systems’ 2014, Advances in Intelligent Systems and Computing 323, Springer International Publishing Switzerland 2015
9. Chu-Hsing Lin1, Jung-Chun Liu2, Chien-Ting Kuo3 “Priority Queue-based Scheme to Maintain Quality of Service for Normal Users Suffering from Large DDoS Attacks” International Journal of Future Generation Communication and Networking Future Generation Communication and Networking Vol. 3, No. 2, June, 2010 Vol. 3, No. 2, June, 2010
10. Santosh Kumar , Abhinav Bhandari and A. L. Sangal “Comparison of Queuing Algorithms against DDoS Attack” Santosh Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1574-1580.
11. Yang Xiao,1 Hui Chen,2 Shuhui Yang,3 Yi-Bing Lin,4 and Ding-Zhu Du5 ” Wireless Network Security” Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking Volume 2009, Article ID 532434, 3 pages
12. Salima Rashid Al Dhabooni , Hothefa Shaker Jassim , Zeyad T. Sharef and Baraa T. Sharef “Survey: Security Attacks in Wireless Sensor Networks” International Advanced Research Journal in Science, Engineering and Technology ISO 3297:2007 Certified Vol. 4, Issue 5, May 2017.
13. Mr.SandeepShinde, Dr.J.W.Bakal “Traceback Mechanism for DDoS Attack Using Local Flow Monitoring in MANET” IJCSET (www.ijcset.net) | Vol 5, Issue 8,301-303 ISSN: 2231-0711.
14. K. Park and H. Lee, “On the Effectiveness of Probabilistic Packet Marking for IP raceback under Denial of Service Attack,” Proc. IEEE INFOCOM, 2001.