ISSN:2348-2079

Volume-7 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

Network congestion control monitoring system using FWP and SWP with multi hop model

K.P.Uvarajan[1], K.N.Muthu Rathinam[2], S.Santhosh[2], S.Shneha[2], R.Shyamala[2]

[1]AP/ECE ,Nandha Engineering College

[2]Final Electronics and Communication Engineering. Nandha Engineering College

ABSTRACT

In a mixed wireless and wired network environment, packet loss can be caused by various types of due to network congestion as well as wireless errors. However, existing TCP decides that all packet loss is network congestion, which causes congestion control to degrade TCP performance. This project distinguishes network congestions and wireless errors through a deep learning algorithm when a packet loss occurs. In case of loss due to network congestion, congestion control is performed in the same way as existing TCP. In case of loss due to errors, we propose an algorithm that can improve the wireless TCP performance by only retransmitting lost packets without reducing the congestion window. Through the simulation, we show that the proposed algorithm improves the TCP performance by reflecting the result of the pre-learned deep learning algorithm, compared to the "TCP Westwood" or "TCP Veno" proposed for improving the wireless TCP performance by discriminating the congestion and wireless error. In addition, improve the network life time using fixed window protocol and sliding window protocol, each node monitors the forwarding behavior of its neighboring nodes in sliding window mechanism. The node listen the next-hop node with the packets sent at regular intervals. The project 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.

INTRODUCTION

Like the existing system, the proposed system also considers a saturated situation that a node always has data to send when the channel is available. In addition, the project uses two concepts; fixed window model and sliding window model of which the latter produces the best output with slight increased calculation overhead. During monitoring, each node monitors the forwarding behavior of its neighboring nodes. In most cases, a node only monitors its next hop in a route. For that, a three-node segment of a route is considered (with at least two hops) being used to send data packets. If the three nodes are denoted as node 1 (source or the node closer to source), node 2, and node 3 (destination or the node closer to destination), then node 2 is the next hop of node 1 and node 3 is the next hop of node 2. When node 1 transmits a data packet to node 2, it expects to hear node 2's transmission of this packet to node 3 within some specified amount of time. If the fraction of packets not overheard by node 1 exceeds a specified threshold, then node 1 concludes that node 2 is dropping too many data packets and suspects it to be a malicious node. For monitoring purposes, node 1 keeps track of a window of packets that it sent recently to its next hop.

With the fixed windows approach, a malicious node can afford to drop packets at a faster rate, at times. The drawback of the sliding windows approach is that it can lead to higher false positives in noisy environments. In addition, in the proposed

Author for correspondence:
Department of Electronics and Communication Engineering, Nandha Engineering College, Perundurai – 638052

system, the monitoring Node 'A' not only monitors the next hop node's (say 'B') forwarding behavior but also cross-checks the monitoring details of neighbor nodes (senders to 'B') before suspecting the node B.

- False positive rate is less.
- Finding multiple misbehaving nodes in multi-hop environment is possible.
- Finding misbehaving nodes is easier when the network consisting of both TCP and UDP nodes.

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 shrewDDoS attacks using spectral analysis" the authors Yu Chen and Kai Hwang presented a new spectral template-matching 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.

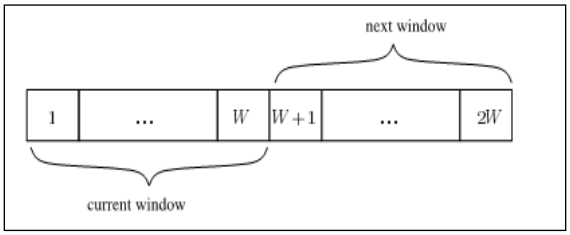
CONGESTION CONTROL

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.

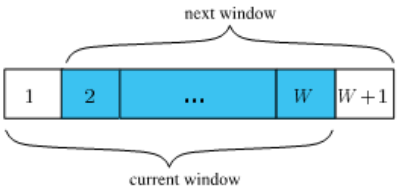
The sliding window approach is free of this deficiency since in any consecutive W-transmitted packets, a malicious node may drop at most L-1 packets without risking suspicion by neighbors. To model the state of sliding window –based monitoring using a discrete-time Markov chain. The purpose of the Markov 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 listbox control.

Protocol: Formal Specification

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.

Sender

- Input: sequence [Frames]
- First: nat,
- Ftsend: nat,
- 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).

Receiver:

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.

Frame Message

- Snumber: {0... 1…K-1}
- Frame: Frames,
- Timeout: Time

ACK Message

- Snumber: {0... 1…K-1}
- 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.

State

- Sender: Sender,
- Receiver: Receiver,
- Fchannel ⊆ FrameMessage
- Achannel ⊆ AckMessage
- Time: Time

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.

InitialState:
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) $time = 0$

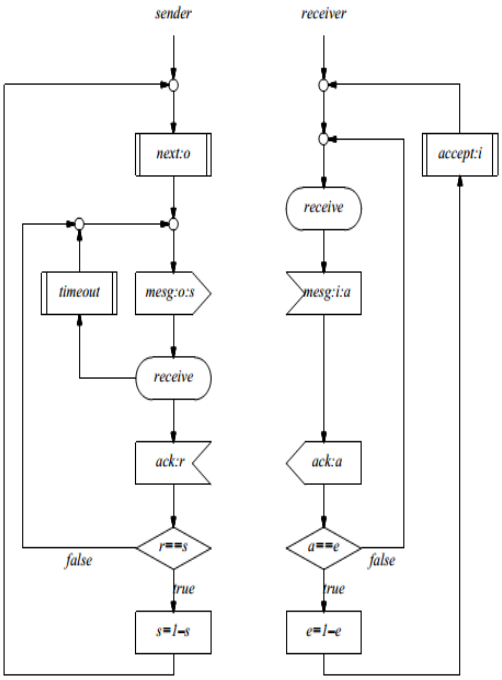


Fig 3.1 protocol process- fwp

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.

The Delay Action

Delay(t)

Precondition:

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

Effect:

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

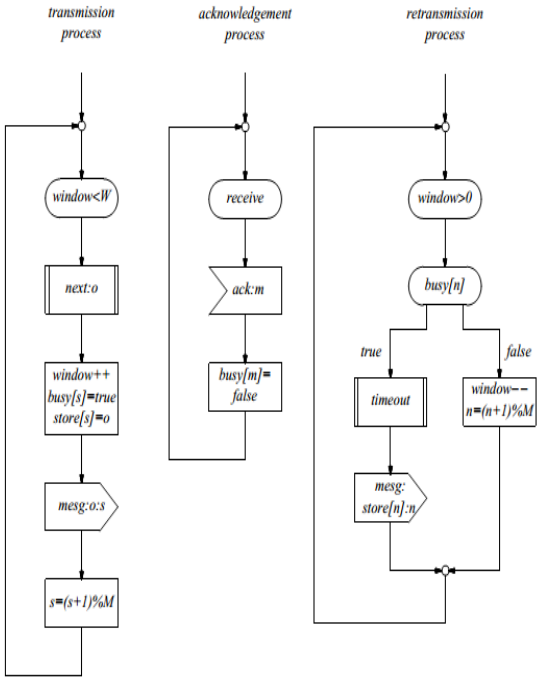


Fig 3.2 Protocol Process- SWP

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.

Actions of The Sender

The precondition ReuseZeroPre of action Send ex-presses 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

$$
\begin{aligned}
&ReuseZeroPre(s0) = \\
&\quad (rem(K)(ftsend(s0)) = 0 \;\&\; ftsend(s0) > 0) \Longrightarrow \\
&\qquad (ftsend(s0) = first(s0) \;\& \\
&\qquad time(s0) > tackmax(s0) + Lmax) \\
&\text{IN} \\
&\quad first(s0) \leq ftsend(s0) < first(s0) + N, \\
&\quad ReuseZeroPre(s0)
\end{aligned}
$$

$$
\begin{aligned}
&\text{Effect:} \\
&\text{LET} \\
&SendNS(s0) = s0\ with \\
&\quad [ftsend := ftsend(s0) + 1] \\
&\text{IN} \\
&\quad s1 = SendNS(s0)\ with \\
&\qquad [fchannel := fchannel(s0) \cup \\
&\qquad\quad \{(rem(K)(ftsend(s0)), \\
&\qquad\quad input(s0)(ftsend(s0)), \\
&\qquad\quad time(s0) + Lmax)\}] \\
&\quad \vee\ s1 = SendNS(s0)
\end{aligned}
$$

Resend(i)

$$
\begin{aligned}
&\text{Precondition:} \\
&\quad i \geq first(s0), \\
&\quad i < ftsend(s0)
\end{aligned}
$$

$$
\begin{aligned}
&\text{Effect:} \\
&\quad s1 = s0\ with \\
&\qquad [fchannel := fchannel(s0) \cup \\
&\qquad \{(rem(K)(i),\ input(s0)(i),\ time(s0) + Lmax)\}] \\
&\quad \vee\ s1 = s0
\end{aligned}
$$

Receiveack(am)

Precondition:

$am \in achannel(s0)$

Effect:

LET

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

AND

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

IN

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

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:

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

Receive(fm)

Precondition:

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

Effect:
LET
$fset(s0, fm) = \{ j \mid j < N \,\&$
$snumber(buffer(s0)(j)) = snumber(fm) \,\&$
$frn(buffer(s0)(j)) = \varepsilon \,\&$
$snumber(buffer(s0)(j)) \geq j \}$
AND
$RNS(s0, bn, fr) = s0$ *with*
$[buffer := buffer(s0)$ *with*
$[(bn) := (snumber(buffer(s0)(bn)), fr)]]$
IN
if $fset(s0, fm) \neq \emptyset$ then
$s1 = RNS(s0, choose(fset(s0, fm)),$
$frame(fm))$ *with*
$[fchannel := fchannel(s0) \setminus \{fm\}]$
$\vee\, s1 = RNS(s0, choose(fset(s0, fm)),$
$frame(fm))$
else $s1 = s0$ *with*
$[fchannel := fchannel(s0) \setminus \{fm\}]$
$\vee\, s1 = s0$

Sendack

Precondition:
$acklastdel = TRUE$

Effect:
LET
$SendackNS(s0) = s0$ *with*
$[acklastdel :=$ if $lastdel(s0) = K - 1$
then $FALSE$ else $acklastdel(s0)]$
IN
$s1 = SendackNS(s0)$ *with*

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).

Table 4.1 Comparison of Packet Drop count of 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

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.

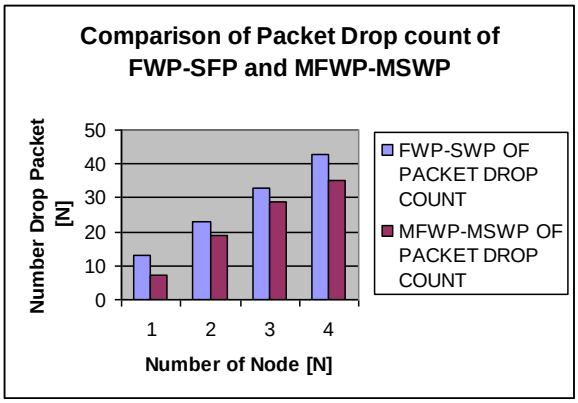


Fig 4.1 Comparison between Packet Drop counts

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.

Table 4.2 Average of attacker node finding in Existing System

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

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

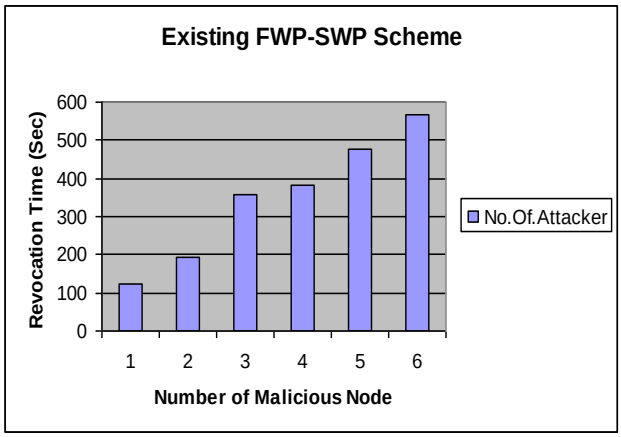


Fig 4.2 Existing FWP-SWP- Number of Attacker

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

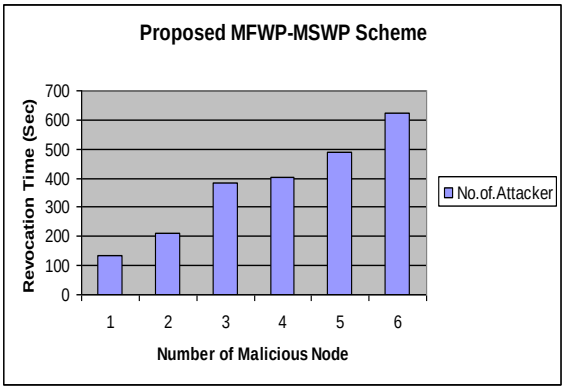


Fig 4.3 MFWP-MSWP - Number of Attacker

CONCLUSION

The new system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. In this project, major issues to ensure secure communications for mobile ad hoc networks, namely, certificate revocation of attacker nodes are solved. In contrast to existing algorithms, we propose a cluster-based certificate revocation with vindication capability scheme combined with the merits of both voting-based and non-voting based mechanisms to revoke malicious certificate and solve the problem of false accusation. The process of preparing plans had been a new experience, which was found use full in later phases of the project is completed. Efforts had been taken to make the system user friendly and as simple as possible. However at some points some features may have been missed out which might be considered for further modification of the application. The new system become useful if the below enhancements are made in future.

- Any attack should be identified as soon as possible.
- To mitigate malicious attacks on the network.

In future, the system is more effective and efficient in revoking certificates of malicious attacker nodes, reducing revocation time, and improving the accuracy and reliability of certificate revocation.

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, 3(2), 2006.

[2]. Michael T. Goodrich, "Probabilistic Packet Marking for Large-Scale IP Traceback" IEEE/ACM TRANSACTIONS ON NETWORKING, 10(10), 2007.

[3]. Shui Yu,Wanlei Zhou, and Robin Doss," Information Theory Based Detection Against Network Behavior Mimicking DDoS Attacks" IEEE COMMUNICATIONS LETTERS,12(4), 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, 20(5), 2009.

[5]. Akash Mittal, Prof. Ajit Kumar Shrivastava, Dr. Manish Manoria"A Review of DDOS Attack and its

[6]. Countermeasures in TCP Based Networks" International Journal of Computer Science & Engineering Survey (IJCSES) 2(4), 2011.

[7]. Tao Peng, Christopher Leckie, And Kotagiri Ramamohanarao "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems" ACM Computing Surveys,39(1), 2007.

[8]. 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, 6(12), 2015.

[9]. LuKaszApiecionek, Jacek M. Czerniak, and Wojciech T. Dobrosielsk "Quality of Service Method as DDoS protection Tool" D.Fileve et al. (eds.), Intelligent Systems' Advances in Intelligent Systems and Computing 323, Springer International Publishing Switzerland 2015.

[10]. 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 3(2), 2010.

[11]. 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,2 (4), 2011, 1574-1580.

[12]. 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 Article ID 532434, 3 pages, 2009.

[13]. 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, 4(5), 2017.

[14]. Mr.SandeepShinde, Dr.J.W.Bakal "Traceback Mechanism for DDoS Attack Using Local Flow Monitoring in MANET" IJCSET (www.ijcset.net) 5(8), 301-303.ISSN: 2231-0711.

[15]. K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP raceback under Denial of Service Attack," Proc. IEEE INFOCOM, 2001.