Volume 6 ▪ Issue 5 ▪ September / October 2014

INTERNATIONAL JOURNAL OF

COMPUTER NETWORKS (IJCN)

ISSN : 1985-4129

Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER NETWORKS (IJCN)

VOLUME 6, ISSUE 5, 2014

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 1985-4129

International Journal of Computer Networks (IJCN) is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJCN Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTER NETWORKS (IJCN)

Book: Volume 6, Issue 5, September/October 2014
Publishing Date: 10-10-2014
ISSN (Online): 1985-4129

IJCN Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2014

EDITORIAL PREFACE

The International Journal of Computer Networks (IJCN) is an effective medium to interchange high quality theoretical and applied research in the field of computer networks from theoretical research to application development. This is the *Fifth* Issue of Volume *Six* of IJCN. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. IJCN emphasizes on efficient and effective image technologies, and provides a central for a deeper understanding in the discipline by encouraging the quantitative comparison and performance evaluation of the emerging components of computer networks. Some of the important topics are ad-hoc wireless networks, congestion and flow control, cooperative networks, delay tolerant networks, mobile satellite networks, multicast and broadcast networks, multimedia networks, network architectures and protocols etc.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 6, 2014, IJCN aims to appear with more focused issues. Besides normal publications, IJCN intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJCN give an opportunity to scientists, researchers, engineers and vendors to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in computer networks in any shape.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJCN as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in computer networks from around the world are reflected in the IJCN publications.

IJCN editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCN. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCN provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computer Networks (IJCN)

EDITORIAL BOARD

Dr. Jiang Li
Howard University
China

Dr. Fang Liu
University of Texas at Pan American
United States of America

Dr. Enyue Lu
Salisbury University
United States of America

Dr. Chunsheng Xin
Norfolk State University
United States of America

Dr. Imad Jawhar
United Arab Emirates University
United Arab Emirates

Dr. Yong Cui
Tsinghua University
China

Dr. Zhong Zhou
University of Connecticut
United States of America

Associate Professor Cunqing Hua
Zhejiang University
China

Dr. Manish Wadhwa
South University
United States of America

Associate Professor Paulo de Figueiredo Pires
Federal University of Rio de Janeiro
Brazil

Associate Professor Vijay Devabhaktuni
University of Toledo
United States of America

Dr. Mukaddim Pathan
CSIRO-Commonwealth Scientific and Industrial Research Organization
Australia

Dr. Bo Yang
Shanghai Jiao Tong University
China

Assistant Professor Yi Gu
University of Tennessee at Martin
United States of America

TABLE OF CONTENTS

Volume 6, Issue 5, September/October2014

Pages

Collaborative Re-Localization Method in Mobile Wireless Sensor Network Based on Markov Decision Process

Mona Nasseri *mona.nasseri@rockets.utoledo.com*
Dept. of Electrical Engineering and Computer Science
University of Toledo
Toledo, OH 43606

Robert C. Green *greenr@bgsu.edu*
Dept. of Computer Science
Bowling Green State University
Bowling Green, OH 43403

Mansoor Alam *mansoor.alam2@utoledo.edu*
Dept. of Electrical Engineering and Computer Science
University of Toledo
Toledo, OH 43606

Junghwan Kim *jung.kim@utoledo.edu*
Dept. of Electrical Engineering and Computer Science
University of Toledo
Toledo, OH 43606

Vijay Devabhaktuni *Vijay.Devabhaktuni@utoledo.edu*
Dept. of Electrical Engineering and Computer Science
University of Toledo
Toledo, OH 43606

Wei Cheng *wcheng3@vcu.edu*
Dept. of Computer Science
Virginia Commonwealth University
Richmond, VA 23284

Abstract

Localization in Mobile Wireless Sensor Networks (WSNs), particularly in areas like surveillance applications, necessitates triggering re-localization in different time periods in order to maintain accurate positioning. Further, the re-localization process should be designed for time and energy efficiency in these resource constrained networks. In this paper, an energy and time efficient algorithm is proposed to determine the optimum number of localized nodes that collaborate in the re-localization process. Four different movement methods (Random Waypoint Pattern, Modified Random Waypoint pattern, Brownian motion and Levy walk) are applied to model node movement. In order to perform re-localization, a server/head/anchor node activates the optimal number of localized nodes in each island/cluster. A Markov Decision Process (MDP) based algorithm is proposed to find the optimal policy to select those nodes in better condition to cooperate in the re-localization process. The simulation shows that the proposed MDP algorithm decreases the energy consumption in the WSN between 0.6% and 32%.

Keywords: Mobile Wireless Sensor Network, Markov Decision Process, Mobility Patterns, Time Bounded Essential Localization.

1. INTRODUCTION

A wireless sensor network includes several nodes in a cooperative network that each of them has a power source, processing capability and contains memory. Additionally, each node often has some sensors such as temperature, humidity or velocity sensors. Today, WSNs have become a significant technology for different types of smart devices for various applications including medical, transportation, military and environmental, and an intense research effort is currently proceeding to extend the application of wireless sensor networks [1]. Most of past research assumed that the system is wired, therefore, has an unlimited power supply, has determined resources and is location independent. But for wireless sensor networks, the system is real time and power limited. Sensors have changeable resources, especially for mobile nodes, which their location plays a significant role to choose appropriate resources.

Mobile wireless sensor networks (MWSNs) [2-4] are a particular class of WSN that has become an important area of research for the WSN community. MWSN deployments have considered several challenges that needed to be overcome, including energy consumption, connectivity, bandwidth, coverage, and real time functioning. When there is an uncertainty of the location of some fixed or mobile devices, localization also becomes an important issue. Localization algorithms can deploy obtainable information from the wireless sensor nodes to estimate the position of individual devices.

Sensor nodes may be positioned dynamically or change position during a given experiment time, therefore a method should be used to estimate the location of each node at any given time. For static WSNs, once the node locations have been determined, they are unlikely to change. On the other hand, mobile WSNs must repeatedly estimate their position which is time and energy consuming.

Moreover, all methods which are applicable for static networks and provide high accuracy are not useful for mobile networks due to their need for centralized processing, which is not applicable in a MWSN. At present, the most widely used method for localization is Global Positioning System (GPS). However, there are also several circumstances in which GPS will not work reliably. For instance, GPS requires line of sight to the satellites. As a result, MWSNs in indoor, urban, and underground environments will not be able to use GPS. Furthermore, GPS is relatively expensive, and therefore unattractive for many applications.

Recently, some localization techniques have been proposed to estimate a node's location using information transmitted by a set of nodes that know their own locations, called anchors (these nodes are able to find their location using some resources such as GPS). Additionally, to remove centralized computation, distributed localization methods are proposed in which each node relays the information gained through limited communication with nearby nodes in order to determine its location. These approaches exploit time of arrival (TOA), received signal strength (RSS), time difference of arrival of two different signals (TDOA), and angle of arrival (AOA) to estimate position [5-8].

In this paper, MWSN is studied which allows each node to be used for different purposes such as tracking targets. Principles of a new proposed localization algorithm, Time-Bounded Essential Localization (TBEL) [9], which is focused on achieving localization within a given time-bound through various means is applied to find each node’s location. Yet in the mobile network, nodes must repeatedly re-localize to keep their position information, accurate.

The other issues that arise in MWSN are power consumption and latency. In a large scale network containing mobile nodes it is not possible to recharge nodes whenever power has been drained. Therefore, it is valuable to decrease power consumption in order to increase the network lifetime [10]. One method to satisfy a power efficient network is the Markov Decision Process (MDP) [11] which is applicable to determine the limited number of nodes that contribute to perform localization. Moreover, it could be a method to decrease response time.

The remainder of the paper is organized as follows: A literature review is presented in section 2, Section 3 contains network model and assumptions, including localization method and movement patterns. MDP-Based algorithm is explained in section 4, results and discussion are included in Section 5, and Section 6 concludes the study.

2. LITERATURE REVIEW

Most of the existing research in localization area emphasizes static sensor networks. There are not many studies in Mobile WSN and few algorithms were proposed to work in both static and mobile networks to do localization in the situations which energy and delay are essential factors. There are some surveys that summarize different methods and algorithms for localization in WSN [1, 12]. Various techniques have been proposed to localize nodes in WSN which are based on distance between nodes [13-15]. The most important factors to measure distance are based on RSS, ToA, and AoA [5].

There are other methods which deploy the geometric condition of nodes such as the work in [16] that uses all possible triangles of nodes, so that the location of an unknown node is the center of the intersection of all triangles. In a Gradient Algorithm [17], nodes find the number of hops to all the seeds and apply multilateration technique to find their position. The mentioned algorithms are intended for static networks. For MWSNs, localization should be implemented periodically. In [18], the authors examined how often a localization algorithm should be run in a MWSN, considering the tradeoff between energy and accuracy. In some studies static mobile nodes are used to localize mobile nodes that are located in specific locations [19].

In a wireless sensor network, it is desirable to transmit data at a lower power level while ensuring error-free communication. To reduce power consumption, Transmit-Power Control (TPC) method is a way to save energy, reduce interference and increasing the security [20]. Many existing TPC methods have been proposed for different applications, and surveys of these schemes can be found in [21] and [22]. Energy efficient sensor networks can be improved by deploying localized communication among neighboring sensors and reducing long distance transmissions [23]. In this paper, an MDP based framework algorithm is applied to perform the re-localization process to avoid long distance communications to decrease response delay.

3. NETWORK MODEL AND ASSUMPTIONS

In this study, mobile nodes are moving in the scenario following one of the four particular mobility models which are discussed in section 3.1. Moreover, collaborative groups are formed to localize the mobile nodes. Localization of the mobile node is determined by combining sensed results from different localized sensors. In some random and dense wireless networks, nodes power refilling is not possible, therefore, network lifetime decreases. To overcome this problem, energy efficient methods are preferable. Additionally, applying a higher number of nodes in the localization process imposes a higher amount of information to the network which should be processed. Therefore, response delay and energy consumption increase.

In this work, the square area over which nodes are randomly spread is considered using a mixture of mobile and static nodes. The following assumptions are made regarding each node:

1. All nodes have the same communication range, which is denoted as a circle around the node;
2. Each node can estimate its power consumption to transmit and receive data to and from other nodes or servers that are within its communication range;
3. Each node or server can sense other nodes that are inside their communication range trough signal exchanging;
4. Anchor nodes are aware of their locations and can be either fixed or mobile;
5. Each node is capable of calculating its distance from neighboring nodes in its communication range through distance measurement techniques such as RSSI.

3.1 Movement Patterns

The movement patterns of sensor nodes have an important role in analysis of wireless communication network. It has seen that mobility affects radio communication networks therefore to improve the network performance [24], observing mobility patterns can be helpful.

Main realistic mobility patterns are classified as follows: pedestrians, vehicles, aerial, robots. Pedestrian mobility patterns can be the walking pattern of people or animals. Sensor nodes are attached to moving objects such as pets to track them or animals in herds to be observed by a biologist. The vehicular mobility is the movement pattern of cars, bicycles, trains and etc. Aerial movement pattern shows the flying pattern of birds or any flying objects. Mobility pattern of robots differs based on robot's applications. In some cases it is predictable and some other robots move erratic and unpredictable [24]. To model the realistic mobility patterns, different models are used which the important groups are; cellular mobility models and random trip models [24].

In cellular networks, handoff between cells is the main feature, not the movement details. Random trip mobility is the predominant mobility model for MANETs and is used in this research in simulation. It includes various models such as the Random Waypoint (RWP) pattern that is widely used to evaluate ad hoc network routing protocols. Also Brownian motion is a kind of Gauss-Markov mobility model which has a tuning parameter to change the randomness in movement pattern. Another applied movement pattern in this research is Levy walk pattern that is able to model different movement patterns from people in shopping centers to animals in wildlife [25, 26]. In Modified RWP (MRWP) method, nodes move in a specific direction to reach a target, which could be applicable in military purposes, moreover is practical for hardware implementation.

Random Waypoint Pattern. In this pattern, the sensors randomly move at various speeds in a zigzag pattern. At each point, every node pauses before it starts moving again. In Figure 1, nodes move under a RWP model. Here nodes move from waypoint P_i to waypoint P_{i+1} with speed v_i. Before moving toward the next waypoint, nodes pause at each waypoint [26].

The number of stops and speed changes in a predefined time depends on the node speed. A node can randomly move to any location within the network bounds, therefore, to update the node position, a random coefficient will be used which is between 0 and 1 according to Equation (1) where X_L and Y_L are the dimensions of a square network area. Also, the velocity is considered as a random value which is attainable by determining MIN_{speed} and MAX_{speed} (minimum and maximum speeds of nodes) according to Equation (2).

$$\begin{aligned} X_{waypoint} &= rand \times X_L \\ Y_{waypoint} &= rand \times Y_L \end{aligned} \tag{1}$$

$$V_i = MIN_{speed} + (MAX_{speed} - MIN_{speed}) \times rand \tag{2}$$

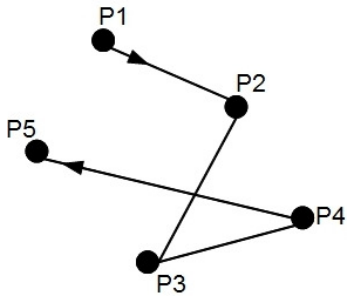

FIGURE 1: Random Waypoint Mobility Pattern for a node that moves from waypoint P_1 to P_5.

Modified Random Waypoint Pattern. In the proposed MWRP pattern, which is defined for nodes that have planned to reach a specific point, at each time step a random change is added to the last point. This is shown in Equation (3) where ω could be any value depending on the purpose of the movement. In this study, it is assumed that ω is 100 due to the environmental dimensions (500×500). This pattern continues until a node reaches a border, in which case the new position is calculated according to Equation (3).

$$\begin{aligned} \mathrm{X}_{\mathrm{waypoint}} &= \mathrm{rand} \times \mathrm{X}_{\mathrm{L}} / \omega + X_{old} \\ \mathrm{Y}_{\mathrm{waypoint}} &= \mathrm{rand} \times \mathrm{Y}_{\mathrm{L}} / \omega + Y_{old} \end{aligned} \tag{3}$$

Brownian Motion. The random movement of particles suspended in a liquid or gas, caused by collisions with surrounding particles is called Brownian motion. In the simulation of a Brownian mobility model, time is divided into N time slots at a predefined interval T, in which a mobile node has a random move at each time slot and the endpoint after N time slots is the cumulative summation of all random moves [22]. Figure 2 shows an example of Brownian motion.

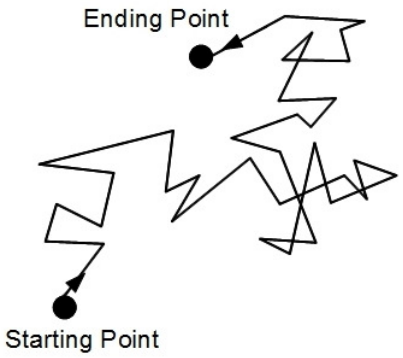

FIGURE 2: A movement example of Brownian Motion for a node.

Levy Motion. A Levy walk is a random walk in which the steps are defined in terms of step-lengths, which have a certain probability distribution, with the directions of the steps being random. As is shown in Figure 3, Levy walks consist of many short flights that are accompanied by long flights [26]. The distribution of the step sizes has a power like tail which is defined in (4) where the a value is between 1 and 3.

$$\Pr(D > d) = O(d^{-a}) \tag{4}$$

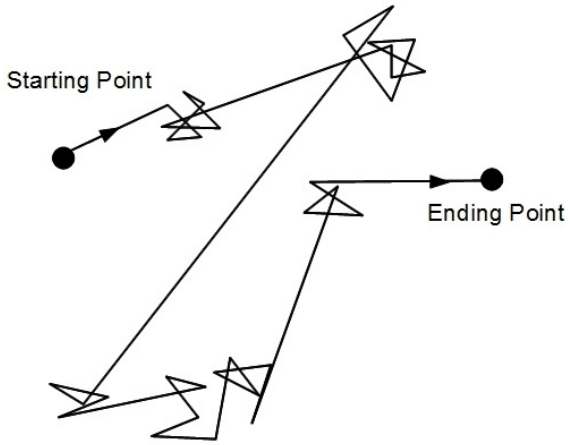

FIGURE 3: A movement example of Levy Walk.

3.2 Localization Method

Localization of sensors in a specific period of time is important in many applications such as battlefield, which message exchange is likely to be detected by enemies. Recently TBEL algorithm proposed a method to localize the network within a specific time bound by introducing k-rounds essential localization, time bounded relative localization and time bounded physical localization terms [9].

In TBEL algorithm, sensors ink rounds of essential localization, localize themselves under their local coordinate system (LCS), during *k* rounds of information flooding. Then sensor nodes relatively localized in *k* rounds of communications if local coordinate systems can be transformed into the same coordinate system for the whole network. Therefore sensor network is physically localizable if, for every pair of sensors, exists an anchor, with which, the pair of nodes are connected through a sequence of coordinate systems.

In this research the idea of TBEL is used to localize the system. Each node can localize itself if it is connected to at least 3 anchor/localized nodes. Then the node's condition would change from un-localized/blind to localized node and will be able to cooperate in the localization process of un-localized nodes. All the nodes in a sensor network transmit messages in predefined *k* rounds of communications, then they stop sending signal until the next re-localization process. The value of *k* depends on network conditions such as area, number of nodes and number of anchors. For example, for a smaller area with higher number of anchors, localization process would be done in a shorter time which means the smaller value of *k*.

In a network where whole nodes are connected, the network completely can be localized. But in cases with some isolated nodes, which are separated and have no connection to other nodes, they remain un-localized. This problem can be removed by providing more anchor nodes in such places.

3.3 Markov Decision Process

A Markov decision process (MDP) is defined by a set of states (S) and the set of actions (A), including transition function (T) and reward function (R) to do specific actions [27]. The transition function describes the probability distribution over the next states as a function of the current state and the agent's action. The reward function determines the reward received after deciding on a taking a certain action in a certain state. According to the Markov Property, the next state and the reward depend on the current state and the action, not on the previous states and actions. An agent or client in the MDP environment alternates between perception and action. The agent detects the state s_t at time t, and selects an action a_t. The agent then receives the

reward that is a function of state and action, $R(s_t, a_t)$, and observes the next state, s_{t+1}, with the probability specified by the transition function $T(s_{t+1}| s_t, a_t)$.

The main objective of an MDP is to find an optimal policy for a client. A policy π: S →A is a mapping function that defines an action in each state. An optimal policy for MDP maximizes some functions of the rewards received by fulfilling the policy.

The value of a policy π or the function value which starts at state 's', with a discount factor α □ [0, 1), is shown in Equation (5) where $E\{r^t|s^0 = s, \pi\}$ is the expected reward received at time t given the initial state 's'. Using this reward formulation, the goal for an agent is to find an optimal policy π^* that maximizes the discounted future reward for all states. By defining the state transition function, T, and the reward function, R, the optimal policy can be calculated using a standard algorithm, e.g., value iteration and policy iteration [27, 28].

$$V^{\pi}(s) = \sum_{t=0}^{\infty} \alpha^t E\{r^t \mid s^0 = s, \pi\}, \tag{5}$$

4. PROPOSED METHOD

4.1 Method Description

Re-localization algorithms in which all neighboring localized nodes cooperate to re-localize un-localized mobile sensor nodes, are both energy and time consuming. The more information a node compiles the more energy and time it consumes. The best way to save energy in a node is by limiting the number of cooperating nodes and exploiting the best nodes to do localization. As a result, smaller amounts of information transfer between the nodes, leading to decreased energy consumption. In this method only the best nodes in the neighborhood are leveraged in the localization process and those nodes that are either too far away or have a low energy level are ignored.

In this study, the MDP method is used to handle the problem of choosing an optimal number of localized nodes, which are also in the best condition energy-wise to cooperate in performing localization for a mobile node. MDP can be used to obtain a tradeoff between energy efficiency and latency; therefore a cost function should be associated with the formulated MDP that is appropriate. This is shown in Equation (6) where P shows power efficiency and D shows delay in receiving signals which is in a direct relation with distance.

$$H = a \times P + (1 - a) \times 1/D \tag{6}$$

Those nodes that contribute to the re-localization process should be the nodes that are within a mobile node's communication range (R). A server that could be an anchor takes part in choosing the optimized number of nodes to collaborate with, based on their conditions according to their distance to the mobile node and the power level. Shorter distances between nodes and anchor/server will lead to lower response delay.

Two states are defined to show the node's circumstances, 'active' and 'passive'. An active node collaborates to do localization, but a passive node remains inactive. The goal is to find the optimized solution for the number of active nodes – this is the policy that is being optimized. Commonly, for an accurate re-localization process, at least three localized nodes must be located within the mobile node's communication range [1]. For a more accurate re-localization process, one can use more than 3 anchors (multilateration localization), depending on the number of available localized nodes, although some nodes do not make any significant changes in accuracy. Additionally, the problem of power consumption arises; therefore it is valuable to consider an upper bound for the number of nodes that may be used in the localization process in order to decrease the required power consumption [11]. Therefore the localized nodes that are

removed in one localization process for a specific node, can save their power to collaborate in the localization process for another node. In Equation (7) N_l shows the number of collaborative nodes, which should be between 3 and N_u, as lower and upper bounds. N_u depends on the number of nodes in the area, size of the area and node's communication range. Note that nodes are not uniformly distributed in the area, therefore some nodes are connected to the higher number of nodes rather than N_u.

$$3 \le N_l \le N_u$$
$$N_u = \text{Node density} \times (\text{communication range})^2 \times \pi \tag{7}$$

In the first phase of the proposed algorithm, the localization will be done for all nodes using the time bounded localization method [9]. The anchors and localized nodes will broadcast packets that contain their location information. They also will collect the information of other anchors. The other nodes will calculate their distances from the anchor, localized and un-localized nodes in their neighborhood and estimate their location according to TBEL algorithm. When an un-localized node changes to localized, it can broadcast its location and collaborate in localization as an active node.

All fixed or mobile localized nodes can be used to calculate the location of un-localized Mobile nodes. The position of mobile nodes can be calculated by collecting and combining the information from different localized nodes.To determine the number of active nodes to collaborate for each area or cluster in a network, a server or head can be used which could also be an anchor. According to the network condition, several servers can be assumed in different locations in which their density depends on the network that can be calculated statistically. The server will check if the mobile node is in its controlling area, by receiving acknowledgement signal from it. They exchange a signal containing; node ID, power level, localization condition- that can be '0' for un-localized node and '1' for localized node- and its location in localized node case. For un-localized node server will take the control of the localization process, including the calculation of the optimal number of active nodes which collaborate. If a mobile node leaves the server's environment, re-localization would be done by other server/anchor node.

A server or anchor will broadcast its decision to track a mobile node. The server should select collaborative localized nodes according to their distances to un-localized node and their energy level. Distances between all pairs of nodes are determined based on RSSI [3] at each node, and calculated distance information will be broadcasted. Therefore, all nodes and servers are aware of distances between nodes in their communication ranges and a server can determine closest nodes to un-localized nodes.

As mentioned before, MDP is used to find active nodes that are cooperating in localization. To choose the number of nodes, an energy consumption bound can be considered as shown in Equation (8) where "E_i" is the amount of energy that a node uses to transmit or receive a message and subscript "i" shows the index of a collaborative node. In this paper, N_u is the maximum number of collaborating nodes and therefore the upper bound for energy ($E_{upperbound}$) can be expressed in term of N_u. In the other words $N_u \times E_i$ can define the $E_{upperbound}$ if another energy limitation is not considered.

$$E_{total} = \sum_{i=1} E_i < E_{upperbound} \tag{8}$$

To select some nodes among all possible options (if there are more choices available) a value function is introduced. Nodes with the higher values would be chosen as collaborative nodes according to:

$$V = a \times \frac{1}{d} + b \times (E - E_i)$$
$$b \approx 1 - a \tag{9}$$

Where "E" is the energy level of a node and "d" is the distance between the mobile and localized nodes. 'a' and 'b' are weight factors to define the importance of value function elements.

As mentioned before, in MDP, a transition probability value is considered in selecting a node as a collaborative node. In this paper, the probability of selecting a node to change its state as an active node depends on E_i and its distance to the un-localized node:

$$P = \frac{1}{N} \times \frac{1}{d} \times \frac{1}{E_i} \tag{10}$$

where "N" is the number of active nodes in the communication range of the mobile node. But the problem is how to choose nodes with higher value functions. The upper bound is considered to determine the maximum number of nodes, however, it should be determined if all the nodes are in an appropriate condition to collaborate. Therefore, a condition is defined to evaluate the function value of the node. As mentioned before, at least 3 nodes should collaborate in the re-localization process. If more than 3 nodes are available, the decision on the number of active nodes would be made according to their function values. Nodes with function values comparable to the third node in the descending list of the node's function values, can act as active nodes. For this research, half of the third function value is used as the criteria. That means node with function values higher than criteria can be chosen as an active node. This is described in detail in the section 4.2 as Algorithm 1. But before investigating algorithm 1 which includes all information, Figure 4 shows an overview of whole re-localization process.

4.2 Algorithm Description

In this research, each server/anchor broadcasts a signal which all the nodes in its communication range that receive it, send an acknowledgment signal and their node ID would be saved in a list on server. Each node also broadcasts its information, including, ID, power level and its distances to other nodes. Therefore, each server knows all nodes in its neighborhood and also the other servers' locations. The list later can be used to find localized nodes in the mobile node communication range (potential active nodes). This list can include localized nodes both fixed and mobile or just fixed, that is explained in Algorithm 1 by Function 1. When a server/head/anchor recognizes a mobile node in its area, a message of sensing it would be broadcasted to other anchors and localized nodes in its zone. In the next step collaborative nodes are selected according to their power and their distance to the mobile node. Additionally, distance can be expressed as a delay factor. The optimal number of localized nodes to do re-localization is based on the MDP framework. The probability to choose a localized node to collaborate, which depends on its distance from the mobile node and its energy level, should be considered (10). Finally, the information would be collected to find the location of mobile nodes. The whole the process is described in a pseudo code format in Algorithm 1.

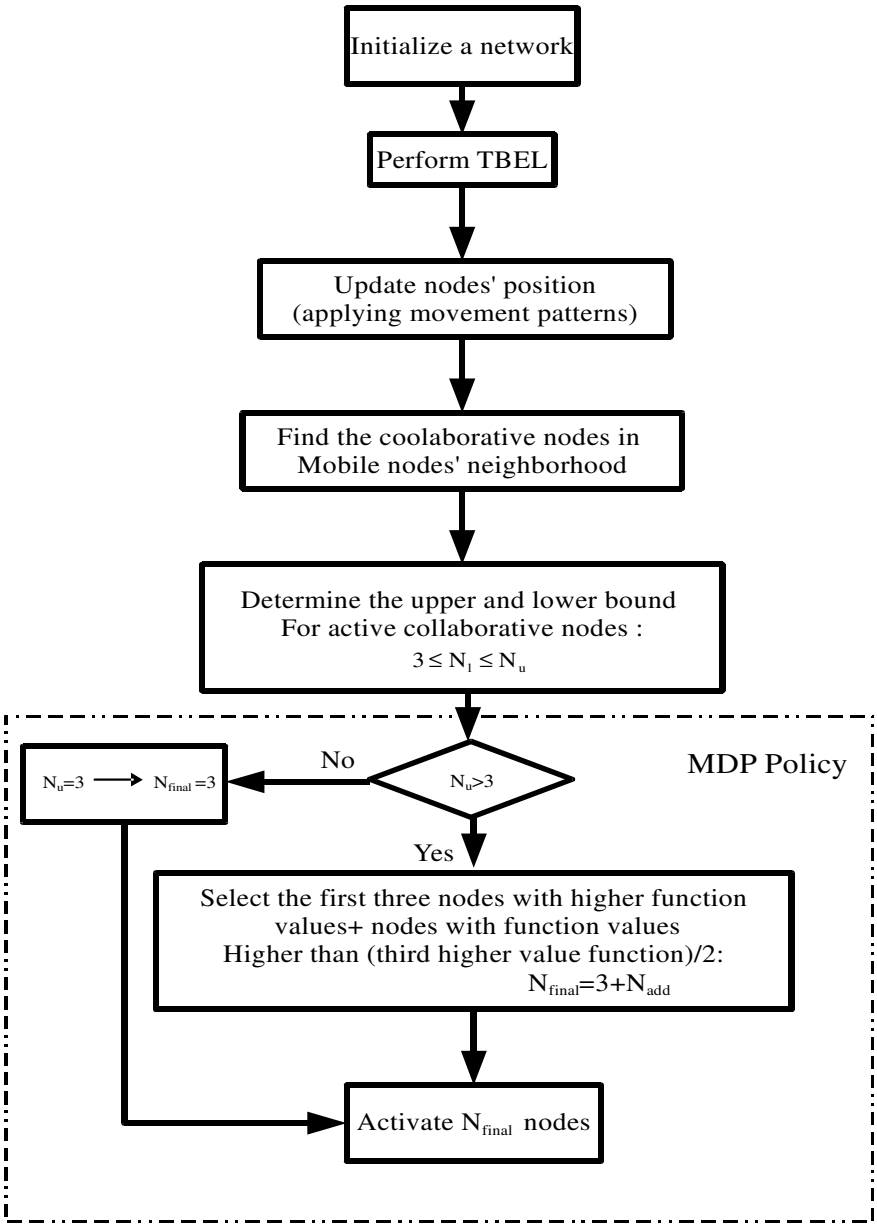

FIGURE 4: Flowchart of re-localization process.

Algorithm. 1

Define E_i, a, b
Determine the nodes by index
Find mobile nodes (M_j) distances from neighbor nodes
Make a list (L_j) of nodes in mobile node's communication range for each M_j
If mobile anchor nodes are allowed to collaborate in the localization process
  Then
    Skip Function 1

Function 1
  Remove mobile node indexes from L_j
End Function 1

Sort L_j by distances from M_j in an ascending manner
Find the length of L_j
Determine power values (E) of the nodes provided by L_j

MDP function
Define Transition probability matrix (T_j)
Find Value function for each node in L_j %%%% including distances from M_j and energy consumption
    V_{ji}= a × 1/distance$_{ji}$ + b × (E-E_i)
Sort V_j in a descending manner
Find upper and lower bounds %%% to determine the no. of active nodes%%%
    Lower bound=3, upper bound=Nu

Policy %%% to select the best nodes to collaborate in the localization process%%%

Select the first 3 nodes with higher value functions
    Counter=3;
  For i=4: N_u
    If V_j(i)>V_j(3)/2
      Counter ++
  End

Select V_j(1)**:**V_j(counter) %%%The counter value determines the finalized number of active nodes%%%

  If length(V_j)≤3
    Counter=length(V_j)
  End

5. RESULTS

The simulations are run in Matlab in which an environment with dimensions of 500 (m) × 500 (m) is considered, including 120 nodes (containing 20 anchors) so that the number of mobile nodes changes from 12 to 60 (10%-50% of nodes). 8000 mw as a maximum power of a node, 0.5 mw for transferring each message that was shown by E_i and a communication range of 60 meters for each node are assumed. These values were selected due to their use in real hardware applications. The MDP method was used to choose the optimal number of active localized nodes to cooperate in the localization process in order to decrease the power consumption and delay in response. The effect of power and response delay factors depends on their coefficients in (9), defined by 'a' and 'b' to show their weights, which are determined according to their importance. For the following results, a is chosen as 3/4. Algorithm 1 is applied for four different movement patterns (RWP, MRWP, Brownian and Levy) and the results in tables 1 and 2, which are the average of 10 runs of simulations show the energy reduction consumption- the difference between power consumption after and before applying algorithm 1, divided by the power consumption before applying MDP- which is calculated to evaluate the algorithm performance.

As mentioned before, localized mobile nodes can act as either active or passive nodes according to the network conditions. Therefore, two experimental conditions are considered. First, the

mobile node is able to operate as an active localized node; therefore multiple localized nodes either mobile or fixed can contribute to estimate the location. Table 1 shows the percentage of power consumption reduction after applying algorithm 1 for four mentioned movement patterns. Power consumption reduction for RWP is almost the same for different numbers of mobile nodes because of node distribution after applying RWP movement. It can be then claimed that node density around the specific mobile nodes is almost fixed or comparable to the last position before applying movement pattern. In Modified RWP, increasing the number of mobile nodes leads to increase in the power consumption reduction and is due to the nature of this movement pattern. When the number of mobile nodes proliferates, more nodes move in a specific direction which causes more localized nodes in the neighborhood. However, saturation occurs because just a limited number of nodes are allowed to contribute in localization. Increasing the number of active nodes has no effect after passing the upper bound (N_u).

Brownian motion results are close for different percentages of nodes and it is due to the short movements of nodes around their last position. On a Levy walk with lower numbers of mobile nodes, nodes have sudden long flights which may put them in a place with lower number of localized nodes. When the number of mobile nodes increases, the probability of having more localized nodes in a neighborhood augments, and therefore the energy consumption reduction increases for environments with higher percentages of mobile nodes.

	Movement Pattern			
Percentage of mobile nodes	**Random Waypoint**	**Modified RWP**	**Brownian Motion**	**Levy Walk**
10%	19.7	9.9	17.41	10.14
20%	21.32	11	16.43	13.04
30%	22.9	32	15.39	17.45
40%	20.54	31.52	18.24	17.89
50%	20.25	30.23	15.42	17.14

TABLE 1: The percentage of power consumption reduction for four movement patterns, considering both fixed and mobile nodes as active nodes.

In some conditions, it is not possible to exploit mobile nodes in the re - localization process due to different reasons such as saving energy for future activities. Therefore, in the second experiment, the mobile nodes are removed from the list of active nodes. That means they are not involved in the localization process, and the number of active nodes decreases. Results in table 2 for all movement patterns show the descending change versus additive number of mobile nodes. This behavior is due to the smaller number of active nodes. By increasing the number of mobile nodes, the number of potential active nodes decreases. Downward trend for all movement patterns is expected, which is endorsed by Table 2 results.

Figure 5 – which was applied for networks including 30-50% of mobile nodes – shows that after applying MDP, considering second experiment assumptions, for a random network topology without observing any special movement pattern, the number of collaborative nodes to do localization decreases which is the reason for lower energy consumption. Additionally, as mobile nodes are removed from active node lists, incrementing the number of mobile nodes decreases the number of active nodes. Therefore, as it is demonstrated in Figure 5, for a higher percentage of mobile nodes, the number of active nodes before and after MDP implementation is closer or almost the same.

In Figure 6 the tradeoff between energy consumption and response delay can be found. As mentioned before the response delay is in a direct relation to the distance of the mobile and active nodes. Figure 6 shows as the distance between nodes increases, value function decreases and there is no connection for distances more than 60. On the other hand, increasing the energy level increases the function value as well.

Percentage of mobile nodes	Movement Pattern			
	Random Waypoint	Modified RWP	Brownian Motion	Levy Walk
10%	20.8	2.08	15.26	7.31
20%	16.47	3.54	12.5	9.08
30%	13.11	1.3	8.2	9.6
40%	7.7	1.18	5.1	4.9
50%	4.9	0.67	2.2	3.5

TABLE 2: The percentage of power consumption reduction for four movement patterns, considering fixed nodes as active nodes.

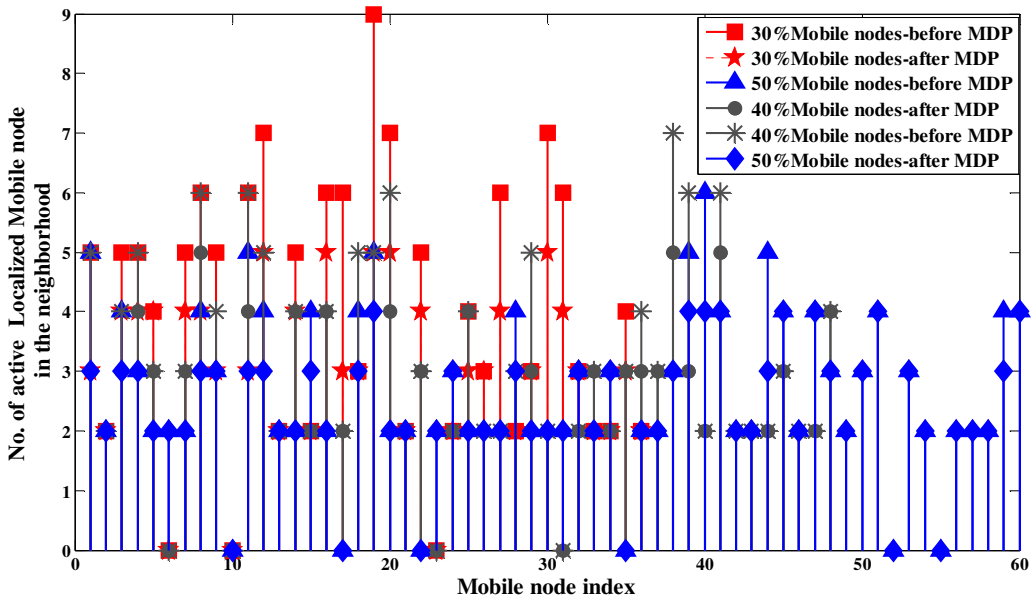

FIGURE 5: The effect of MDP algorithm on the number of active localized nodes for environment, including 30-50 percent of mobile nodes.

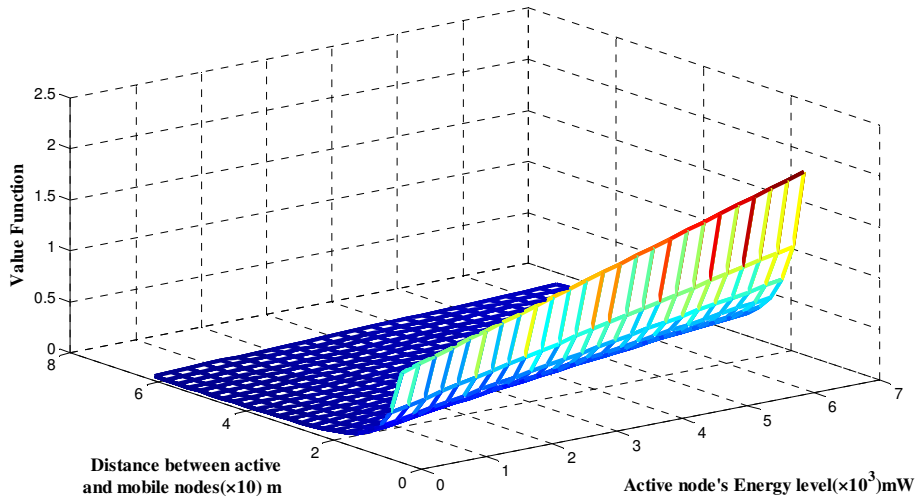

FIGURE 6: The tradeoff between energy levels and response delay to evaluate the value function.

6. CONCLUSION

In this work, a localization algorithm is proposed for energy constrained WSNs. The proposed scheme selectively activates nodes to collaborate in localization. The activation of nodes depends on the node value function coupled with an MDP approach. Results show that proposed algorithm is capable of reducing the total energy consumption of the network in the localization process. The algorithm was simulated observing four movement patterns (WRP, MWRP, Brownian motion and Levy walk) and varying the number of mobile nodes. In the proposed scheme, collaborative/active nodes are selected according to their instant power and their distance to the mobile nodes, in which distance can describe the delay factor. Based on the MDP framework, the optimal number of localized nodes to do re-localization was found and the MDP-based policy selects the best nodes among neighboring nodes. For the proposed algorithm, 0.6 to 32 percent, energy consumption reduction was obtained. As a future work the proposed algorithm will be simulated in Contiki software which can communicate with actual hardware. In the next phase of this research, the algorithm will be applied in a hardware platform, including several wireless Z1 Zolertia motes, to show the applicability of the proposed method.

7. Acknowledgements

This material is based upon work supported in part by the National Science Foundation under Grant No. CNS-1248381.

8. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "*Wireless sensor networks: a survey*." Elsevier, Computer Networks, vol. 38, Issue 4, pp. 393-422, Mar. 2002.

[2] I. Amundson, and X. Koutsoukos. "A Survey on Localization for Mobile Wireless Sensor Networks." in Mobile Entity Localization and Tracking in GPS-less Environnments. vol. 5801, pp. 235-254, 2009.

[3] S. A. Munir, et al. "Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing." in Advanced Information Networking and Applications Workshops, 2007. pp. 113-120.

[4] F. Aiello, et al. "Using Mobile Agents as Enabling Technology for Wireless Sensor Networks." in Sensor Technologies and Applications, SENSORCOMM '2008. pp. 549-554.

[5] S. Pandey and P. Agrawal. "A survey on localization techniques for wireless networks." Journal of the Chinese Institute of Engineers, vol. 29, pp. 1125-1148, Nov. 2006.

[6] P. Rong and M. L. Sichitiu. "Angle of Arrival Localization for Wireless Sensor Networks." in Sensor and Ad Hoc Communications and Networks, SECON '06, 3rd Annual IEEE Communications Society, 2006. pp. 374-382.

[7] S. K. Meghani, et al. "Localization of WSN node based on Time of Arrival using Ultra wide band spectrum." in Wireless and Microwave Technology Conference (WAMICON), 2012. pp. 1-4.

[8] D. Qiao-ling, et al. "TOA-Based Location Estimation Accuracy for 3D Wireless Sensor Networks." in Wireless Communications, Networking and Mobile Computing, WiCom 2009, pp. 1-4.

[9] C. Wei, et al. "Time-Bounded Essential Localization for Wireless Sensor Networks." Networking, IEEE/ACM Transactions on, vol. 21, pp. 400-412, 2013.

[10] J. Li, et al. "Power-Efficient Node Localization Algorithm in Wireless Sensor Networks." Advanced web and Network Technologies and applications Lecture notes in computer science, vol. 3842, pp. 420-430, 2006.

[11] S. Misra, S. Singh. 2012. “Localized Policy-Based Target Tracking Using Wireless Sensor Networks.” ACM Transactions on Sensor Networks. vol. 8, no. 3, Article 27, Jul. 2012.

[12] I. Amundson, X. D. Koutsoukos. “A Survey on Localization for Wireless Mobile Networks.” In Proceedings of the 2nd international conference on Mobile entity localization and tracking in GPS-less environments (MELT'09), Springer-Verlag, Berlin, Heidelberg, vol. 5801, 2009, pp. 235-254.

[13] F. Reichenbach, et al. “Comparing the Efficiency of Localization Algorithms with the Power-Error-Product (PEP).” in Distributed Computing Systems Workshops, ICDCS '08, 28th International Conference on, 2008. pp. 150-155.

[14] D. Moore, J. Leonard, D. Rus and S. Teller. “Robust distributed network localization with noisy range measurements.” In Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004. pp. 50-61.

[15] N. Patwari, and I. Alfred. “Using proximity and quantized RSS for sensor localization in wireless networks.” In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, San Diego, CA, USA, 2003.

[16] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. “Range-free localization schemes for large scale sensor networks.” In Proceedings of the 9th annual international conference on Mobile computing and networking (MobiCom), 2003. pp. 81–95.

[17] R. Nagpal, H. Shrobe, and J. Bachrach. “Organizing a global coordinate system from local information on an ad hoc sensor network.” In Second International Workshop on Information Processing in Sensor Networks (IPSN), 2003. pp. 333–348.

[18] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang. “Dynamic localization control for mobile sensor networks.” In 24th IEEE International Performance, Computing, and Communications Conference (IPCCC), 2005. pp. 587–592.

[19] P. Bergamo and G. Mazzimi. “Localization in sensor networks with fading and mobility.” In The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. pp. 750–754.

[20] H. Ren and M. Q. H. Meng. “ Power Adaptive Localization Algorithm for Wireless Sensor Networks Using Particle Filter.” IEEE Transactions on Vehicular Technology, vol. 58, no. 5, pp. 2498-2508, Jun 2009.

[21] S. Koskie and Z. Gajic, “Signal-to-interference-based power control for wireless networks: A survey, 1992–2005.” Dyn. Continuous, Discrete Impulsive Syst. B: Appl. Algorithms , vol. 13, no. 2, pp. 187–220, 2006.

[22] V. Kawadia and P. Kumar, “Principles and protocols for power control in wireless ad hoc networks.” IEEE J. Sel. Areas Communication. vol. 23, no. 1, pp. 76–88, Jan. 2005.

[23] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. “Next century challenges: Scalable coordination in sensor networks.” In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, Washington, USA, Aug. 1999, pp. 263–270.

[24] C. Schindelhauer. “Mobility in Wireless Networks,” SOFSEM: Theory and Practice of Computer Science, 2006, pp. 100-116.

[25] Christian Bettstetter. 2001. “Mobility modeling in wireless networks: categorization, smooth movement, and border effects.” Newsletter, ACM SIGMOBILE, Mobile Computing and Communication, pp. 55-66, Jul. 2001.

[26] W. Pu and I. F. Akyildiz. “Effects of Different Mobility Models on Traffic Patterns in Wireless Sensor Networks.” In Global Telecommunications Conference, GLOBECOM 2010, pp. 1-5.

[27] A. Munir and A. Gordon-Ross. “An MDP-based Application Oriented Optimal Policy for Wireless Sensor Networks.” CODES+ISSS, Proceeding of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis, 2009. pp. 183-192.

[28] M. Nasseri, M. Alam and R. C. Green. “MDP based optimal policy for collaborative processing using mobile cloud computing.” Cloud Networking (CloudNet), IEEE 2nd International Conference on, pp. 123-129, San Francisco, USA, Nov. 2013.

Improving Firewall Performance by Eliminating Redundancies In Access Control Lists

Ajay Krishna Vasu *ajay_krishna_v@yahoo.co.in*
Computer Science Department
Sri Venkateswara College of Engineering
Pennalur, 602117, India

Ashwin Ganesh *ariel_ash@yahoo.com*
Computer Science Department
Sri Venkateswara College of Engineering
Pennalur, 602117, India

Priya Ayyappan *appy178@gmail.com*
Computer Science Department
Sri Venkateswara College of Engineering
Pennalur, 602117, India

Anirudhan Sudarsan *anirudhan.sudarsan@gmail.com*
Computer Science Department
Sri Venkateswara College of Engineering
Pennalur, 602117, India

Abstract

A firewall is a network security device that works to protect an organization's internal network from both unauthorized and malicious users. It functions by examining all packets that enter any one of its incoming interfaces and comparing the structure of the packet against a set of predefined rules. Each rule specifies if a packet corresponding to the rule is to be permitted or denied. This set of rules is called an access control list (ACL) and it forms the basis of a firewall's policy. Incorrect configuration of the firewall can lead to redundant rules which cause performance degradation. We propose an algorithm to identify and eliminate redundant rules in an access control list during the configuration phase. The proposed work defines an access control list as a linked list data structure. A comparison of the proposed work and the conventional approach is also presented.

Keywords: Firewall, Access Control List, Network Security, Firewall Configuration, Firewall Policy.

1. INTRODUCTION

Firewalls serve as perimeter defence devices for private corporate networks ranging from small to huge. Such networks are under constant threat from attackers who attempt to penetrate them in view of financial or other forms of personal gain. Firewalls are hence installed to prevent such occurrences. These firewalls prevent packets both from being routed into the network and from being routed out of the network unless the source host device has the necessary permissions to access a particular resource.

A firewall achieves this by acting as a gatekeeper for the network i.e. they act as the entry or exit point of the network. All packets, whether incoming or outgoing have to pass through the firewall [1].

The firewall is best implemented through packet filtering technology [2]. A packet can be considered to be a structure with various attributes such as source IP address, destination IP address, source port and destination port. Each packet will correspond to a rule defined in the ACL or will not be mapped onto to any rule in which case, it will be mapped onto an implicit deny. A packet filtering firewall functions by comparing the attributes of each incoming packet against the rules defined in the access control list [3]. The firewall then takes a decision to either route the packet into the network or to drop the packet. Today's firewalls also include additional options to log the incoming packets for later analysis [4].

1.1 Overview of Various Firewall Technologies

The main reason why firewalls are utilized is because of the inefficiency of encryption algorithms in containing malicious packets from being routed into the private network [2]. Some of the common firewall technologies are presented below,

Stateful packet filter- A stateful packet filtering firewall inspects the state of existing network connections in order to make a decision on whether to forward a packet or filter it. If the incoming packet is a legitimate request for a new or part of an existing connection, it is routed through to the internal network [2]. The working of stateful packet filter is based on the concept that packets from the same source need not be examined repeatedly as long as they belong to the same connection. It is known as a dynamic packet filtering technology [5]. The advantage of this type of firewall is that it considerably reduces the average number of comparisons required before a packet is matched with the firewall rules. It provides a greater level of security and is also easy to utilize [3].

Stateless packet filter- A stateless packet filtering firewall is the simplest firewall to implement from the point of view of implementation complexity as well as functionality. It is the most widely used firewall technology. These firewalls do not store any connection related information. Each packet is treated as an independent entity [2]. The firewall examines each incoming packet and decides to either route it or drop it.

Application Gateways- Application gateways basically play the role of a proxy. They process service requests from external clients. An application level gateway performs more thorough inspections on a packet than the average packet filter. It functions even at the application layer by examining the format of the application contained in the packet. The application level gateway can hence detect and block packets if they carry viruses and other malicious code fragments, in addition to blocking them based on IP addresses [2]. When acting as a proxy, the application level gateway provides authentication mechanisms such as username password combinations. It can also provide a detailed log of all the actions it has taken on various packets. The drawbacks of this type of firewall are the implementation complexity which leads to both a complex as well as a slowed down operation as well as a lack of support for new applications and protocols [3 and 6].

Circuit Gateways- Also known as circuit level gateways, these firewalls predominantly function at the transport layer. They make the decision to route packets or filter them based on both IP addresses as well as the port numbers contained in the transport layer header. This header can be either a TCP or a UDP header. When combined with a packet filtering firewall, it is termed as a dynamic packet filter [2]. Circuit level gateways can examine and validate the formation of TCP connections by observing the three way handshake occurring [7]. It also maintains the connection state just like a stateful packet filter and permits packets only if they belong to one of the existing connections [2].

Despite all the technological advancements in firewall technologies, firewalls continue to have a few limitations including the following,

1) They do not deal with threats fully. They basically prevent only unauthorized users and applications from entering the trusted network while providing access to permitted users. A firewall fails, if an unauthorized user has already entered the network. In such cases, it

cannot prevent malicious activities from being carried out. There is a pressing need to also implement additional security measures such as sniffing and encryption [7].

2) An application gateway can identify viruses and malicious code, but by itself, it cannot destroy the source of the attack [5].

3) Modern networks are typically large, i.e. they contain several hundreds if not thousands of network devices including hosts, switches and routers. There is an increasing chance that an attack can be carried out by hosts on the internal network itself. Traditional firewalls are powerless in such scenarios [7].

There are several firewalls that are available in the market today such as the Checkpoint SPLAT, Cisco ASA and the freely available OpenBSD packet filter. The three aforementioned firewalls do a very good job of protecting the private network from intruders.

Performance testing of these three firewalls in a lab environment has indicated that the Cisco ASA exhibits the best performance for several indicators such as HTTP throughput, TCP throughput and UDP throughput. The HTTP throughput of the Cisco ASA was found to be nearly twice as much as that of the SPLAT and the OpenBSD packet filter. However, for indicators such as Concurrent Connections and Connections per second, the BSD was found to be the best performer.

There is a strong need to improve firewall technology as well as the performance due to an increasing amount of regulations such as the CobiT framework, the Payment-Card Industry Data Security Standard and the NIST standard [4].

Hereon, we use the term “access list” to refer to an access control list unless specified otherwise.

1.2 Working of An Access List In Stateless Packet Filtering Firewall

There are two types of access lists that can be created to define a firewall’s policy. They are,

- Standard Access List
- Extended Access List

We first define a simplified structure for an IP packet as follows,

struct packet {
Source IP address
Destination IP address
Source port number
Destination port number
}

The structure of the packet contains a source IP address, destination IP address, source port number and destination port number (transport layer). In reality, a packet will contain many more attributes such as version, internet header length, total length, identification, flags, time to live, etc. However, such additional fields have been ignored in the above defined structure as they play a very negligible role in the working of a stateless packet filter.

A standard access control list is one which uses only the *Source IP address* attribute in the packet’s structure to decide whether to forward the concerned packet or to filter it. The *Source IP address* of the packet is compared with each rule defined in the access list until a match is obtained. The other attributes are ignored. An extended access control list on the other hand, uses all four attributes defined in the packet’s structure to determine whether to forward the packet or not. Similar to the working of a standard access list, here also, the packet’s attributes are compared sequentially against the rules in the access list, until a match is obtained. In this

case, a match occurs if and only if every condition defined in the rule is matched successfully with the packet's corresponding attributes [8].

Comparisons usually occur until a rule is matched with the packet or none of the rules match with the packet. In case none of the rules can be successfully matched, then, an implicit deny is applied to the packet by default. This means that any packet not explicitly permitted by any rule in the access list will be denied by default. For the purpose of this paper's analysis, a Cisco based standard access list was considered.

FIGURE 1 shows an example of a network. It must be noted that the figure is not indicative of a production network. The cloud denotes an untrusted network such as the internet. The router r1 serves to represent a perimeter security device such as a firewall. The other devices are a part of the trusted network.

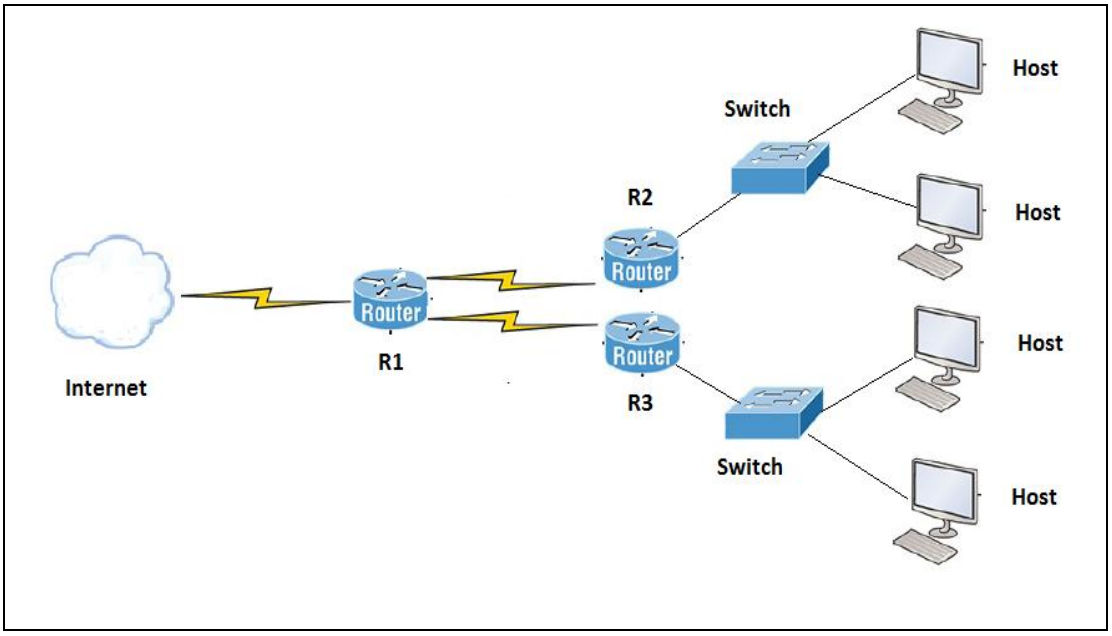

FIGURE 1: Example of a Network.

Private networks usually employ RFC 1918 addressing on their internal network devices and implement network address translation to get onto the internet. All class A, B and C addresses that are not a part of the RFC 1918 ranges are public IP addresses. The range of private address is as follows,

Class A: 10.0.0.0 /8 to 10.255.255.255 /8.
Class B: 172.16.0.0 /12 to 172.31.255.255 /12
Class C: 192.168.0.0 /16 to 192.168.255.255 /16

The command format for defining a standard ACL based on a Cisco router is as follows:
access-list [access-list number 1-99] [permits or deny] [source IP address] [wild card mask]

Once it has been defined, the access list must be applied to an interface of a firewall or a router which can also act as one. The access list can be applied in two directions in each interface- either outbound or inbound. When it is applied to the outbound interface, then the packet is first routed to that interface before it is compared against the rules in the access list. If the access list is applied on an inbound interface, the packet is first compared with the rules in the ACL before

being routed or dropped. The firewall policy is considered functional only after being applied to the interface [8]. The following is an example of an ACL.

1. access-list 10 permit 154.10.5.1 0.0.0.0
2. access-list 10 deny 154.10.0.0 0.0.255.255
3. access-list 10 permit 132.15.4.0 0.0.0.255
4. access-list 10 deny 132.15.0.0 0.0. 7.255
5. access-list 10 permit 15.248.27.4 0.0.0.0
6. access-list 10 permit 17.24.142.0 0.0.0.255

The above access list functions as follows,

a) Packets from the host address 154.10.5.1 are permitted but packets from the rest of the 154.10.0.0 /16 network are denied.
b) Packets from the 132.15.4.0 /24 network are permitted but packets from the remaining addresses in the 132.15.0.0 /21 network are denied.
c) Packets from the host address 15.248.27.4 are permitted.
d) Packets from the 17.24.142.0 /24 network are permitted.
e) An implicit deny is enforced by default at the end of the access list.

An important concept related to access list creation is rule ordering which has been the subject of a lot of significant research recently. Optimizing the ordering of the rules will lead to a much better performance by the firewall in terms of reduction in the number of comparisons required before a rule is matched to the packet. However, optimizing the cost of the packet matching process is an NP hard problem [9].

When configuring an access list, care must be exercised in ensuring that the rules are ordered correctly so as to prevent problematic behaviour by the firewall. This is based on the fact that a firewall works by comparing packets against the access list rules in a sequential manner [8]. One such a problem is illustrated below. The access list defined above has been modified by interchanging rules 1, 2 and rules 3, 4. However, the intent behind defining this access list is still the same. The altered access list is shown below,

1. access-list 11 deny 154.10.0.0 0.0.255.255
2. access-list 11 permit 154.10.5.1 0.0.0.0
3. access-list 11 deny 132.15.0.0 0.0. 7.255
4. access-list 11 permit 132.15.4.0 0.0.0.255
5. access-list 11 permit 15.248.27.4 0.0.0.0
6. access-list 11 permit 17.24.142.0 0.0.0.255

The altered access list functions as follows,

a) All packets from the 154.10.0.0 /16 network are denied.
b) All packets from the 132.15.0.0 /21 network are denied.
c) Packets from the host address 15.248.27.4 are permitted.
d) Packets from the 17.24.142.0 /24 network are permitted.
e) An implicit deny is enforced by default at the end of the access list.

As mentioned above, both access list 10 and 11 were defined with the same intent. However, access list 11 does not permit packets from the host address 154.10.5.1 while a packet from the same address is permitted by access list 10. This is because, when access list 10 is applied, the packet is first compared with the rule *access-list 10 permit 154.10.5.1 0.0.0.0* which routes the packet into the private network. However, when access list 11 is applied, the packet is compared with the rule *access-list 11 deny 154.10.0.0 0.0.255.255* before the rule *access-list 11 permit 154.10.5.1 0.0.0.0* which cause this packet to be dropped.

This logic can also be extended to rules 3 and 4 of the access list 10. When a packet arrives from an address in the 135.15.4.0 /24 network, it will be routed through to the internal network by access list 10. The same packet will however be dropped when access list 11 is applied as the packet will first be compared to the rule *access-list 11 deny 132.15.0.0 0.0.7.255* before the rule *access-list 11 permit 132.15.4.0 0.0.0.25*.

2. RELATED WORK

A measurable amount of research has been carried out in relation to firewalls, especially firewall performance. The stateless packet filter compares the structure of the incoming packets against rules in the access list sequentially until there is a match. This method is very inefficient as the worst case time complexity will become directly proportional to the number of rules in the access list thereby affecting its scalability. The number of rules in the firewall can grow larger due to complex user requirements and different networked applications. Some of the research work has focused on utilizing specialized data structures while others have attempted to suggest hardware based solutions [9 and 10].

[11] proposes a method based on the histograms of packet filtering to monitor firewall performance in real time and to predict the patterns of packet filtering in terms of rule order as well as rule field order. The rule order, the rule field order and the characteristics of packet flow have a significant impact on packet filtering time. This paper suggests an approach to optimize early acceptance and rejection path. This method uses histograms of both packet matching rule and packet not matching rule fields. The presented algorithm calculates the histograms in terms of packet matching and non-matching probabilities on a real time segment basis. The packet processing time is saved by 123% when compared to static rule ordering mechanism and 104% when compared to dynamic rule ordering mechanism.

[10] presents a method that performs early rejection of unwanted flows without affecting other traffic flows. The main aim here is to have a minimum number of early rejection rules but maximum discarding effect. This adaptive technique relies on the construction of a tree structure based on packet flow and packet field values. The constructed trees for each field are combined to create an optimal statistical matching tree of all rules defined in the policy. A packet entering a network is compared to a look up table of values containing packet field characteristics of the tree structure. Until a rule is matched, subtrees of the tree are recursively called. If there is no rule match, default filtering action takes place. The reduction in matching is maximal when the upcoming traffic distribution over field values matches the distribution of the constructed tree. However this is unlikely, because some flows start and others terminate with passage of time. Hence two types of updates are performed- those triggered by performance and those done periodically.

[12] suggests that the performance of a firewall based network can be substantially improved when the traffic behavior of packets is optimized. This paper attempts to solve the optimal rule ordering problem (ORO), which is to find a rule ordering among a given set of rules, where the relations with the previous rules has been preserved in such a way that only a minimum number of packets are matched to the rules. This paper considers the ORO problem as a Binary Integer Problem and thus solves it by using the 'branch and bound' problem along with gradient projection method, thus reducing the effect of combinatorial explosion. The space complexity of the proposed algorithm is linear and the time complexity is polynomial.

In [13], the firewall rule ordering is modified dynamically to obtain maximum efficiency by analyzing network traffic behavior and using packet matching statistics. This optimization process analyzes the traffic behavior and dynamically modifies the order of the filtering rules. The traffic is segmented into P segments of L packets and the matching ratio, mean and variance are calculated. Then, a quantity called the match ratio is calculated and the rules are dynamically reordered based on a matching rate coefficient. It is seen through simulations that the dynamic re-ordering reduces the filtering processing delays found in static rule order systems and

improves the performance of the firewall. The technique does not require any additional overhead and is scalable and easy to implement.

[14] proposes a method to optimize the early acceptance and rejection paths. It uses a set of statistical splaying filters with a binary search on prefix length (SSF-BSPL) technique to reduce packet filtering times. This technique combines three levels of filtering called the statistical Policy Filtering Level, field filtering level and cascaded filtering level. These three filtering levels are combined to enhance packet processing time. This multi-level packet filtering also helps to counter DOS attacks which target the default rule.

3. IMPLEMENTATION

The access control list is implemented in the form of a singly linked list data structure. However, we would like to make a point that using a binary search tree or an AVL tree would provide better results in terms of reducing the average number of comparisons per packet [15].

Each node in the linked list corresponds to one rule in the access list. When a packet arrives at the interface where this access list is applied, its structure will be compared with the attributes of each node until there is a match. The aim here is to reduce the number of comparisons by eliminating redundant rules. A rule in a firewall is redundant if removing that rule does not alter the functioning of the firewall. A more precise definition for a redundant rule is given in [1] - A rule r is redundant in a firewall **f** if and only if the resulting firewall **f'** after removing rule r is equivalent to **f**. There are two possible types of redundant rules that can be considered which are as follows,

Backward redundant rules- A rule R is said to be backward redundant if there exists a rule P appearing earlier than R in the access control list such that R is a subset of P.

Forward redundant rules- A rule R is said to be forward redundant if there exists a rule P appearing after R in access control list such that R is a subset of P and R and P perform the same actions while any rule Q defined between R and P is disjoint from R or performs the same action as R [16].

It has been observed in [16] that 7.8% of the rules are backward redundant while 7.2% of the rules are forward redundant leading to an overall redundancy of 15% on an average.

The algorithm that we propose in this section can be implemented internally in the firewall to affect the final configuration while the firewall is being configured. The working of this algorithm need not be known to administrators thus eliminating any potential administrative hassles.

3.1 Proposed Algorithm

We define the following variables and types for our algorithm,

List - access list defined as a linked list
Rule – A rule which is defined as a node in the **List**.
Function - An attribute of each **Rule** that takes two values- 0 and 1 corresponding to deny or permit respectively.
IP - An attribute of each **Rule** defined as an array that specifies the list of IP addresses for which the corresponding **Function** is defined.
List L - The defined access list.
Rule R1- New rule being inserted into List L.
Rule R- A rule already defined in the List L.

Algorithm **redundant-insert** (**Rule** R1, **List** L) *{*
check=0
For each Rule R in L {
If((R.Function ⊕ R1.Function)=0)

```
{
Array IP[] = R.IP[] U R1.IP[]
if (IP[]=R.IP[]) {
        Check=0
        Break
        }
else if (IP[]= R1.IP[]) {
        Remove(R)
        Insert (R1)
        Check=0
        Break
        }
else {
        Check=1
        Continue;
        }
} //end of if
} //end of Loop
if (check=1)
insert (R1)
} //end of code
```

This algorithm takes as input the rule to be added (Rule R1) and the access list to which the rule is to be added (List L). Rule R1 is then compared with each rule R defined previously in List L. The algorithm then checks if R1 and R both perform the same function. If the function performed is the same then, the IP addresses are compared.

If Rule R has defined IP addresses that include the ones defined in R1 as well as some additional addresses, R1 is considered redundant and not added to List L. If R1 contains all IP addresses contained in R and also a few more, then Rule R is removed from the list and R1 is inserted at the end. If neither of the cases occurs, then the loop proceeds until either of the cases occur or there are no more rules left to compare. If neither of the cases occurs till the very end, then the Rule R1 is inserted at the end of the list.

We define the following access list to provide a practical illustration of the algorithm. The rules are numbered for the purpose of ease of understanding.

1. access-list 25 deny 100.25.24.0 0.0.0.255
2. access-list 25 permit 154.63.0.0 0.0.255.255
3. access-list 25 deny 201.23.24.25 0.0.0.0
4. access-list 25 permit 220.0.25.0 0.0.255.255
5. access-list 25 permit 154.63.128.0 0.0.127.255
6. access-list 25 permit 154.0.0.0 0.255.255.255

FIGURE 2 represents this access list with six rules where rules 2, 5 are redundant as rule 6 performs their function.

FIGURE 2: Access list with 6 rules.

When the **redundant-insert** algorithm is implemented then the access list creation will proceed like in FIGURE 3 and FIGURE 4. When there is an attempt to insert rule 5 into the access list, it is

not added as rule 2 performs the function of rule 5 while also covering a wider range of addresses. When rule 6 is inserted into the access list, rule 2 is removed as rule 6 performs the same function as rule 2 while covering a larger range of addresses.

FIGURE 3: Access list after attempting to insert rule 5.

FIGURE 4: Access list after inserting rule 6.

3.2 Testing Setup

The following ACL with redundant rules was defined for the testing phase-

access-list 40 deny 164.27.48.0 0.0.7.255
access-list 40 permit 131.126.128.0 0.0.127.255
access-list 40 permit 8.8.16.0 0.0.0.255
access-list 40 permit 7.4.25.36 0.0.0.0
access-list 40 deny 27.0.0.0 0.15.255.255
access-list 40 deny 164.27.32.0 0.0.31.255
access-list 40 permit 18.14.0.0 0.0.255.255
access-list 40 deny 96.24.32.0 0.0.31.255
access-list 40 permit 180.64.72.0 0.0.3.255
access-list 40 permit 7.4.0.0 0.0.255.255
access-list 40 permit 195.254.124.128 0.0.0.127
access-list 40 permit 206.121.64.192 0.0.0.63
access-list 40 deny 96.24.0.0 0.0.127.255
access-list 40 deny 175.10.128.0 0.0.63.255
access-list 40 permit 100.1.1.1 0.0.0.0

The **redundant-insert** (Rule, List) algorithm was not used when defining this list. The following rules in access-list 40 are redundant. Only one of each pair is required for the firewall to function correctly.

a) *access-list 40 deny 164.27.48.0 0.0.7.255* and *access-list 40 deny 164.27.32.0 0.0.31.255*
b) *access-list 40 permit 7.4.25.36 0.0.0.0* and *access-list 40 permit 7.4.0.0 0.0.255.255*
c) *access-list 40 deny 96.24.32.0 0.0.31.255* and *access-list 40 deny 96.24.32.0 0.0.31.255*

In each of the above three cases, the second rule makes the first one redundant. The presence of three additional rules in the access list takes a considerable toll on the overall performance of the firewall in terms of the average number of comparisons required per packet before a rule is matched successfully. However, when the same access list is defined using the **redundant-insert** algorithm, we get the following completed ACL.

access-list 41 permit 131.126.128.0 0.0.127.255
access-list 41 permit 8.8.16.0 0.0.0.255
access-list 41 deny 27.0.0.0 0.15.255.255

access-list 41 deny 164.27.32.0 0.0.31.255
access-list 41 permit 18.14.0.0 0.0.255.255
access-list 41 permit 180.64.72.0 0.0.3.255
access-list 41 permit 7.4.0.0 0.0.255.255
access-list 41 permit 195.254.124.128 0.0.0.127
access-list 41 permit 206.121.64.192 0.0.0.63
access-list 41 deny 96.24.0.0 0.0.127.255
access-list 41 deny 175.10.128.0 0.0.63.255
access-list 41 permit 100.1.1.1 0.0.0.0

The access list 41 defined using the redundant-insert algorithm has only twelve rules as against fifteen rules in access list 40. This should in theory, lead to a twenty percentage improvement in firewall performance in terms of average number of comparisons for a packet match.

3.3 Testing Process

We define the term “unmatched packets” as the incoming packets that are not matched to any rules in the access list which means that such rules will be handled by the implicit deny in the access list.

Both C and Java programming were used for our implementation. The results obtained were found to be similar in both implementations. We simulated the generation of 10000 packets some of them that could be matched with a rule in the access list and some of them which did not match any rule in the list. The comparison is performed between two simulated firewalls- one that follows a conventional approach and one that implements the **redundant-insert** algorithm.

First the average number of comparisons per packet was calculated for the access list 40 that was not optimized by the **redundant-insert** algorithm. We also varied the percentage of unmatched packets from zero to fifty, with the aim of calculating the average number of comparisons under more realistic or practical scenarios were we would most likely witness several incoming unmatched packets.

Then, the average number of comparisons per packet was calculated for access list 41 which was defined using the **redundant-insert** algorithm. The optimized access list in theory would hold a cutting edge in terms of performance when compared to the access list 40. We again varied the percentage of unmatched packets from zero to fifty.

For both the non-optimized and the optimized access lists, the total number of comparisons was first calculated and then the average number of comparisons per packet was calculated and averaged out over fifty iterations of the simulation. The formula used to calculate the average number of comparisons was as follows,

Average number of comparisons per packet= Total number of comparisons/ Total number of packets
We represent the average number of comparisons as C_{avg}.

3.4 Results

The following results were observed after the implementation.

a) For the non-optimized access list 40, C_{avg} was found to be 7.988 with no packets going unmatched. When the percentage of unmatched packets was increased to 12.5, C_{avg} jumped to 8.824. When the number of unmatched packets was at 16.66%, C_{avg} was 9.17. When the unmatched packets percentage was increased to 25, C_{avg} increased correspondingly to 9.75. When the unmatched packet percentage was further increased to 33.33, C_{avg} increased to 10.324. C_{avg} increased to a final value of 11.497 when the unmatched packet percentage was amplified to 50.

b) For the optimized access list 41, C_{avg} was found to be 6.499 when there were no unmatched packets. When the unmatched packet percentage was increased to 12.5, the C_{avg} value proportionately increased to 7.281. The C_{avg} further underwent an increase to 7.44 with the rise of percentage of unmatched packets to 16.66. C_{avg} further increased to take values of 7.875, 8.348 and 9.25 as the percentages of unmatched packets grew to 25, 33.33 and 50 respectively.

The percentage increase in performance degradation in terms of an increasing C_{avg} was also observed and tabulated. ΔC_{avg} represents the percentage increase in performance degradation relative to no unmatched packets.

Unmatched packet percentage	C_{avg} – access list 40	C_{avg} – access list 41
0	7.988	6.499
12.5	8.824	7.281
16.66	9.171	7.44
25	9.75	7.875
33.33	10.324	8.348
50	11.497	9.25

TABLE 1: C_{avg} value for both the non-optimized and the optimized access list.

Percentage of unmatched packets	ΔC_{avg} for access list 40	ΔC_{avg} for access list 41
12.5	10.465	12.032
16.66	14.809	14.479
25	22.058	21.172
33.33	29.243	28.450
50	43.00	42.32

TABLE 2: ΔC_{avg} due to increase in unmatched packet percentage.

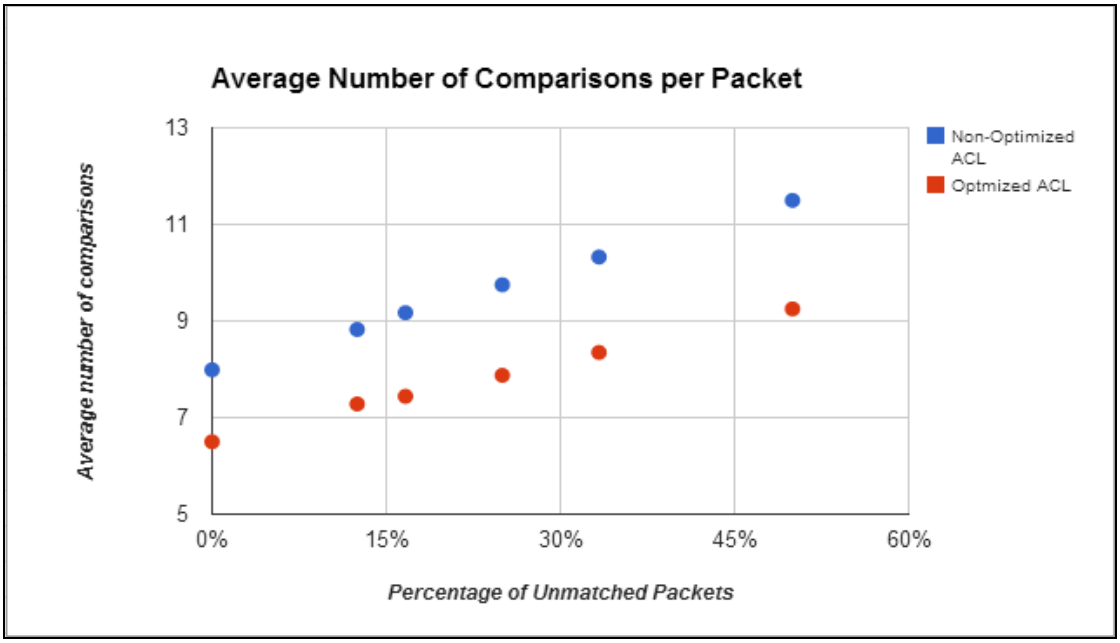

FIGURE 5: C_{avg} for various percentages of unmatched packets for both access lists.

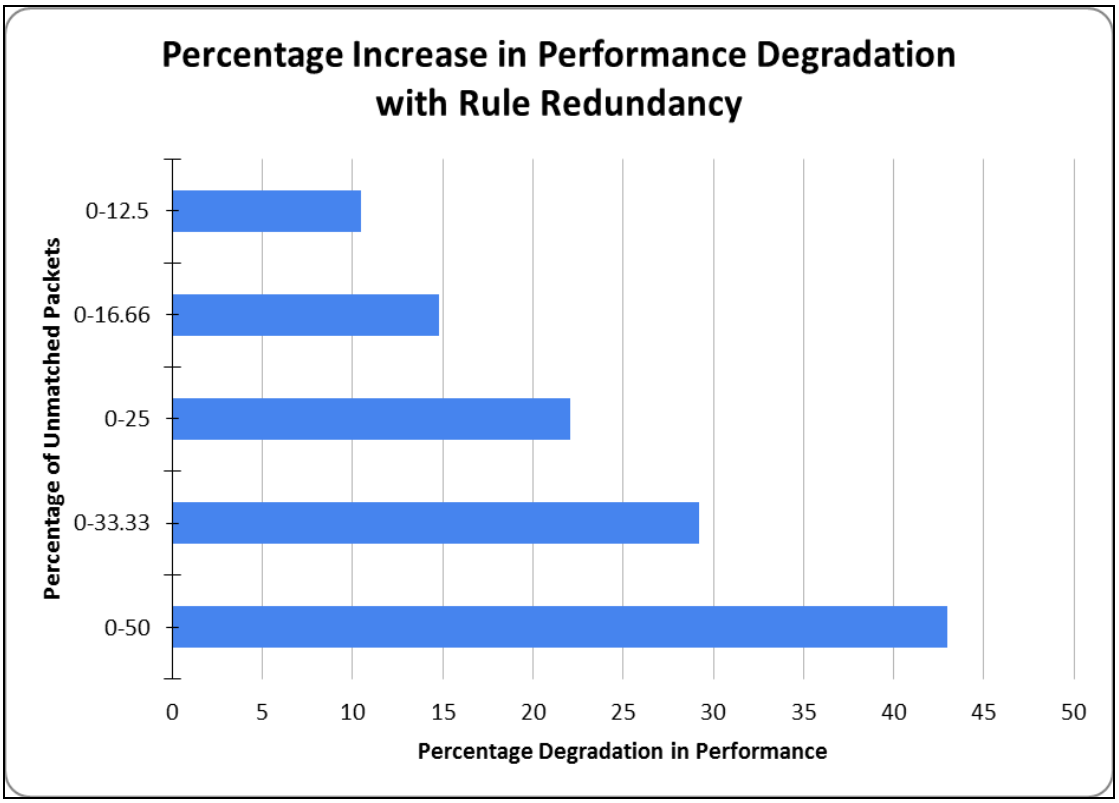

FIGURE 6: Increase in percentage degradation of C_{avg} for non-optimized access list 40.

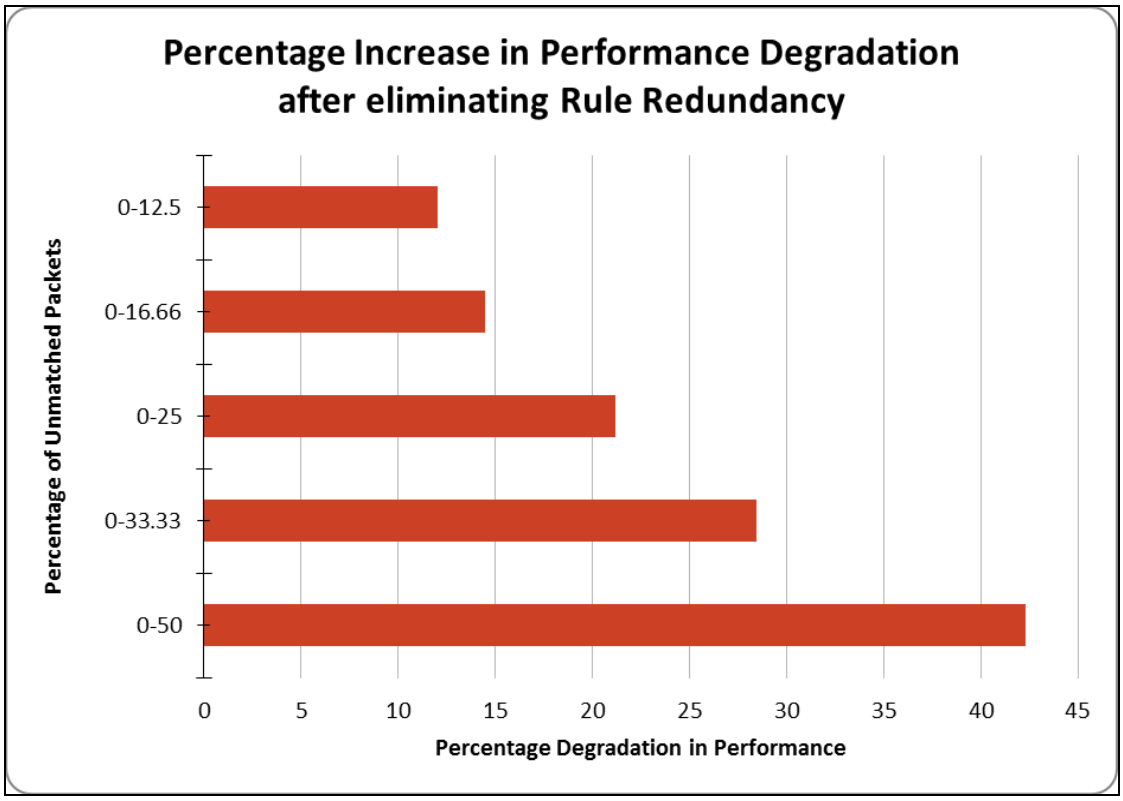

FIGURE 7: Increase in percentage degradation of C_{avg} for optimized access list 41.

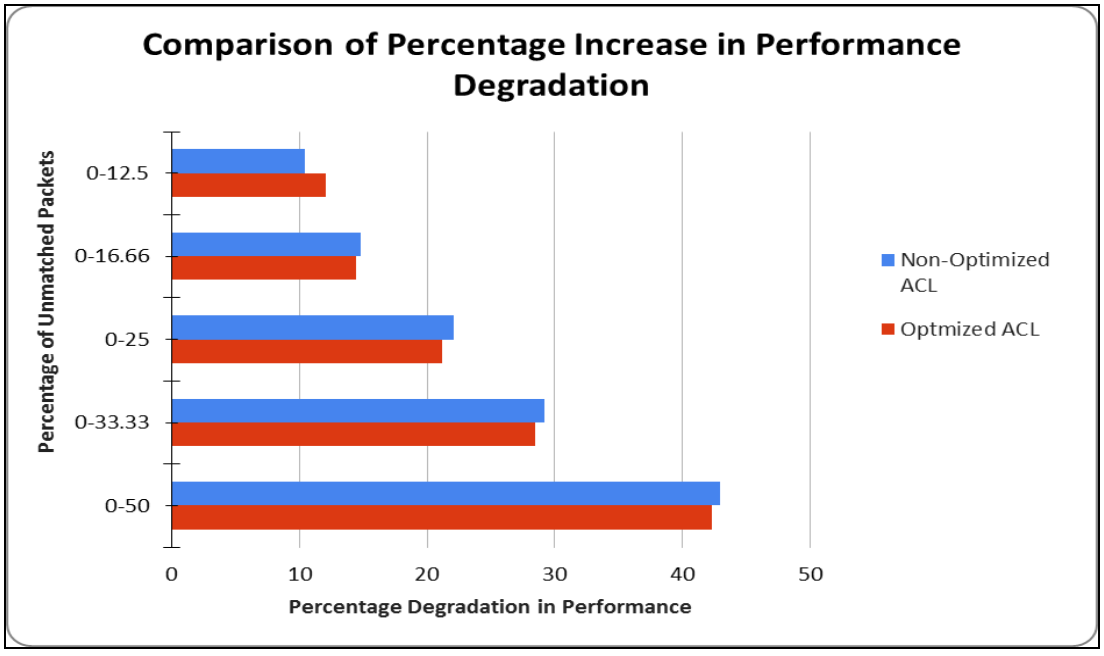

FIGURE 8: Comparison of increase in percentage degradation of C_{avg} for both access lists.

4. ANALYSIS AND DISCUSSION

It is extremely important to improve the performance of existing firewall technologies. This is because the firewall acts as the network's protector and prevents intruders from breaking in. However, the firewall's presence also causes inconvenience to devices within the network because of its negative impact on the network's performance. All packets attempting to enter the network have to go through the firewall. The rate of transmission is slowed because each packet has to wait until packets before it in the queue are matched to a rule in the firewall policy. This problem is further exacerbated by the presence of redundant rules. This is predominantly because in case there is one redundant rule, each packet that enters the network may have to be compared against at least one extra rule which leads to slower transmission.

While this negative impact may not be evident when the number of incoming packets is minimal, it becomes much more apparent when the number of incoming packets is numbering in the millions. Hence, eliminating even one redundant rule will have a visible impact on the firewall's performance because the lesser number of rules leads to lesser number of comparisons. For the sake of simplicity, if we assume that there are hundred rules in the access list which includes two redundant rules and each comparison takes one second, then removing the two redundant rules leads to performance improvement as it takes less time to compare against 98 rules than a hundred rules. In reality however, a comparison will require only a millisecond or even lesser. So the improvement will be estimable when the number of incoming packets is very high.

Also, it must be noted that only packets that match to rules defined after the redundant rules will be affected by their presence. Hence, any performance improvement will have an effect only on those packets. If all the redundant rules are at the end of the access list, keeping or removing them will have absolutely no impact on the firewall's performance.

The immediate observations show unambiguously that the optimized access list 41 defined by implementing the **redundant-insert** algorithm shows a considerably better performance in terms of a considerably reduced value for the average number of comparisons required per packet before a rule is matched.

However, as the percentage of unmatched packets increase, it is noticed that the rate of performance degradation in terms of the average number of comparisons per packet expressed as a percentage value is almost the same for both the optimized and non-optimized access lists which leads to a possible conclusion that the degradation in performance, which is expressed as $\mathbf{\Delta}\ C_{avg}$ is independent of the number of rules defined in the access list.

5. FUTURE WORK

The security of a network plays a huge role in its daily functioning. The firewall is the most fundamental security device in a network. Hence it is crucial to maximize the firewall's performance. There are several aspects of the firewall's functioning that can be given serious thought for research such as rule reordering, redundant rule elimination, impact of burst traffic and rule combination. This will be the subject of our future work.

6. CONCLUSION

Firewalls play an increasingly important role in network security across the world in thousands of enterprise networks today. The amount of packet traffic entering any enterprise network is ever increasing and hence there is considerable incentive in improving firewall performance. In this paper, we proposed an algorithm to identify and eliminate redundant rules in access lists during the configuration phase itself. The results of our implementation strongly indicated that a considerable amount of improvement in firewall performance could be obtained by eliminating redundancy in access control lists.

7. REFERENCES

[1] A. Liu, M. Gowda. “Complete Redundancy Detection in Firewalls." In the proceedings of the 19th annual IFIP WG 11.3 working conference on Data and Applications Security, 2005, pp. 193-206.

[2] H. Ling-Fang. “The Firewall Technology Study of Network Perimeter Security.” In Proceedings of the IEEE Asia-Pacific Services Computing Conference, 2012, pp. 410-413.

[3] L. Zhu, H. Mao and H. Qin. “A case study on Access Control Rules Design and Implementation of Firewall.” In Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing, 2012, pp. 1-4.

[4] C. Sheth and R. Thakker. “Performance evaluation and Comparative Analysis of Network Firewalls.” In Proceedings of the International Conference on devices and communication, 2011, pp.1-5.

[5] H. Mao, L. Zhu and M. Li. “Current State and Future Development Trend of Firewall Technology.” In Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing, 2012, pp. 1-4.

[6] M.Z.A Aziz, M.Y Ibrahim, A.M Omar, R.A Rahman, M.M.M Zan, & M.I Yusof. “Performance analysis of application layer firewall.” In Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA), 2012. pp. 182-186.

[7] A. Krishna and A. Victoire. “Simulation of Firewall and Comparative Study.” In Proceedings of the 3rd International conference on Electronics Computer Technology, 2011, pp. 10-14.

[8] T. Lammle. CCNA Routing and Switching Study Guide. Indianapolis, Indiana: Sybex, 2013, pp. 501-528.

[9] I. Mothersole and M. Reed. “Optimizing Rule Order for a Packet Filtering Firewall.” In Proceedings of the Conference on Network and Information Systems Security (SAR-SSI), 2011, pp. 1-6.

[10] H. Hamed, A. El-Atawy & E. Al-Shaer. “Adaptive Statistical Optimization Techniques for Firewall Packet Filtering.” In Proceedings of the 25th IEEE International Conference on Computer Communications, 2006, pp. 1-12.

[11] Z. Trabelsi, L. Zhang & S. Zeidan. “Packet flow histogram to improve firewall efficiency”, In Proceedings of the 8th International Conference on Information, Communication and Signal Processing, 2011, pp. 1-5.

[12] H. Hamed and E. Al-Shaer. “Dynamic Rule-ordering Optimization for High-Speed Firewall Filtering.” In Proceedings of the ACM symposium on Information, computer and communications security, 2006, pp. 332-342.

[13] Z. Trabelsi. Z. Sayed, H.E & Zeidan. “Firewall packet matching optimization using network traffic behavior and packet matching statistics.” In Proceedings of the Third International Conference Communications and Networking (ComNet), 2012, pp. 1-7.

[14] Z. Trabelsi & S. Zeidan. “Multilevel Early Packet Filtering Technique based on Traffic Statistics and Splay Trees for Firewall performance improvement.” In Proceedings of the IEEE International Conference on Communications (ICC), 2012, pp. 1074-1078.

[15] A. Sudarsan, A. Vasu, A. Ganesh, D. Ramalingam and V. Gokul. “Performance Evaluation of Data Structures in implementing Access Control Lists.” International Journal of Computer Networks and Security, vol. 24, issue 2, pp. 1303-1308, 2014.

[16] P. Gupta. “Algorithms for routing lookups and packet classifications.” PhD thesis, Stanford University, 2000.

INSTRUCTIONS TO CONTRIBUTORS

The International Journal of Computer Networks (IJCN) is an archival, bimonthly journal committed to the timely publications of peer-reviewed and original papers that advance the state-of-the-art and practical applications of computer networks. It provides a publication vehicle for complete coverage of all topics of interest to network professionals and brings to its readers the latest and most important findings in computer networks.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCN.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting from Volume 7, 2015, IJCN aims to appear with more focused issues. Besides normal publications, IJCN intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJCN LIST OF TOPICS

The realm of International Journal of Computer Networks (IJCN) extends, but not limited, to the following:

- Algorithms, Systems and Applications
- ATM Networks
- Cellular Networks
- Congestion and Flow Control
- Delay Tolerant Networks
- Information Theory
- Metropolitan Area Networks
- Mobile Computing
- Multicast and Broadcast Networks
- Network Architectures and Protocols
- Network Modeling and Performance Analysis Network
- Network Security and Privacy
- Optical Networks
- Personal Area Networks
- Telecommunication Networks
- Ubiquitous Computing
- Wide Area Networks
- Wireless Mesh Networks
- Ad-hoc Wireless Networks
- Body Sensor Networks
- Cognitive Radio Networks
- Cooperative Networks
- Fault Tolerant Networks
- Local Area Networks
- MIMO Networks
- Mobile Satellite Networks
- Multimedia Networks
- Network Coding
- Network Operation and Management
- Network Services and Applications
- Peer-to-Peer Networks
- Switching and Routing
- Trust Worth Computing
- Web-based Services
- Wireless Local Area Networks
- Wireless Sensor Networks

CALL FOR PAPERS

Volume: 7 - Issue: 1

i. Submission Deadline : November 30, 2014 **ii. Author Notification:** December 31, 2014

iii. Issue Publication: January 2015

CONTACT INFORMATION

Computer Science Journals Sdn BhD

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627

Fax: 006 03 6204 5628

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2014
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627
FAX: 006 03 6204 5628
EMAIL: cscpress@cscjournals.org