<table>
<thead>
<tr>
<th>SR. NO.</th>
<th>PARTICULARS</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AN EFFECTIVE APPROACH FOR LUNG SEGMENTATION IN CT IMAGES</td>
<td>1-5</td>
</tr>
<tr>
<td></td>
<td>T. HEMALA S. RADHARANI</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>INTELLIGENT PLATFORM FOR MOBILE AGENTS</td>
<td>6-11</td>
</tr>
<tr>
<td></td>
<td>M. KAVINILAVU M. SATHYA</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A REVIEW FOR APPLICATION OF SOFT COMPUTING IN INDUSTRIES</td>
<td>12-19</td>
</tr>
<tr>
<td></td>
<td>T. VIGNESH D. SUDHA N. CHANDRAMATHI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PRIORITY BASED PACKET FORWARDING FOR DISTRIBUTED NETWORKS</td>
<td>20-25</td>
</tr>
<tr>
<td></td>
<td>JOHN KENNEDY</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MANET SECURITY WITH OPTIMIZED ROUTING TECHNIQUE USING DYMO ROUTING PROTOCOL</td>
<td>26-31</td>
</tr>
<tr>
<td></td>
<td>MOHAN C VARADHAGANAPATHY S</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>INTRUSION DETECTION SYSTEM FOR MALICIOUS NODE DETECTION IN DYMO PROTOCOL</td>
<td>32-38</td>
</tr>
<tr>
<td></td>
<td>R. SHANTHAKUMARI</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>REDUCING POWER AND DELAY IN WIRELESS SENSOR NETWORKS USING JUMPING ANT ROUTING ALGORITHM AND SPRAY AND FOCUS ALGORITHM</td>
<td>39-49</td>
</tr>
<tr>
<td></td>
<td>A. SINDHUJA K. T. MATHUMITHAA</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>AN EFFICIENT JOB SCHEDULING ALGORITHM FOR GRID ENVIRONMENT</td>
<td>50-55</td>
</tr>
<tr>
<td></td>
<td>KAVIYARASAN.T SURESH.P</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>AN EVOLUTIONARY MODEL FOR OUTLIER DETECTION</td>
<td>55-61</td>
</tr>
<tr>
<td></td>
<td>S. SATNOSHKUMAR P. VISHNU RAJA DR. V. MURALI BHASKARAN</td>
<td></td>
</tr>
</tbody>
</table>
An Effective Approach for Lung Segmentation in CT Images

T.Hemala
M.Phil- Research Scholar,
Sree Narayana Guru College
Coimbatore.

S.Radharani, Asst.Prof,
Sree Narayana Guru College,
Coimbatore.

Abstract

This paper describes a method for extracting lung regions by X-ray Computer Tomography. Mostly Image Processing Techniques are used in medical areas to diagnosis the diseases in the human anatomy. An effective Extraction method is needed to separate the lung region from the other parts of human body. This method mainly helps for detecting the cancer cells and other diseases in the Lung region. The following steps to be followed to extract lung portion 1) identifying and capturing the chest body portion with the background portion 2) removing background portion 3) leaving only lungs portion.

Keywords: image segmentation, computed tomography, Edge Detection.

I Introduction

Lungs are vital part of the respiratory system. They may suffer from many deadly diseases. Many lungs diseases require some form of medical imaging for the diagnostic purpose. Computer Tomography (CT) scan is an important imaging technique for diagnosis of these lungs diseases due to its 3-dimensional capability. A CT image comprised of many 2 dimensional slices. On visual perception, the CT slice image displays four blackish regions and one whitish region. Blackish regions are background, left lung, right lung and trachea. In some images bronchi may be appeared as blackish region. Whitish region is almost a ring like part surrounding and separating the lungs consisting of muscle, fats etc. There is a good contrast between the blackish and whitish region on observation as shown in figure 1.

Structure of Lung

The lungs are a pair of organs located in the chest that performs respiration. Humans...
Fig. 2. have two lungs.
Trachea: Trachea is also known as the windpipe. Air is breathed in through the nose and mouth pulled down through the trachea & into the lungs. Diaphragm: is a Dome shaped sheet of muscle between the chest and the abdomen. when this muscle expands & contracts, it causes the lungs to inflate & deflate. Bronchi: The left and right branch off from trachea & carry oxygen further into the lungs. The right bronchus is wider shorter & steeper than the left. Bronchioles: are thousands of tiny tubes that branch out from 2 bronchi and carry oxygen deep into lungs. Alveoli is available deep inside the lungs it contains the tiny bronchicles & alveoli where the process of extracting oxygen from inhaled air and ridding the body of carbon dioxide occurs.

II. Image Capturing

Chest Region of a human image is captured from Computed Tomography. Computed Tomography (CT), is a helical tomography (latest generation), which produces a 2D image of the structures in a thin section of the body. It uses X-rays. It has a greater ionizing radiation dose burden than projection radiography; repeated scans must be limited to avoid health effects. CT is based on the same principles as X-Ray projections but in this case, the patient is enclosed in a surrounding ring of detectors assigned with 500-1000 scintillation detectors. (fourth-generation X-Ray CT scanner geometry). Previously in older generation scanners, the X-Ray beam was paired by a translating source and detector[4].

III. Preprocessing

Preprocessing is the key step and the starting point for analysis of Images. Here we have two done two steps a) identifying Noise b) Applying Filter.

![Block Diagram of Lung Segmentation](https://example.com/block_diagram.png)

Fig. 3  Block Diagram of Lung Segmentation

a) Removing Noise

Noise Removal is a step needed to remove the noise in the image. High resolution
image is retrieved when the heavy dose passed. This high resolution image is not having noise. When low dosage is passed it increases the image noise and unsharp edges. The Gray scale images are processed in MATLAB. It may contains some noise such as random noise. Noise in the electronic devices also modeled as Poisson noise.

The magnitude of Poisson noise varies across the image, as it depends on the image intensity. It is very difficult to remove such type of noise. Poisson images occur in many situations where image acquisition is performed using the detection of particles[6].

Random Noise
It is a form of random stochastic process, characterized by large number of overlapping transient disturbances occurring at random, such as thermal noise and shot noise. Random noise is characterized by intensity and color fluctuations above and below the actual image intensity. The pattern of random noise changes even if the exposure settings are identical [6].

Harmonic Mean Filter
This Filter is used to remove random noise effectively. The expression is given as

\[ f^\wedge(x,y) = \frac{mn}{\sum_{(x,z)} y_{x,z} \frac{z}{g(x,z)}} \]  (2)

IV. Edge Detection and Extracting Lung Region

It is often important to separate regions of interest from other parts of the image. Many methods are available to perform noise may be overlapping on the information images in additive mode:

\[ I(x, y) = I'(x, y) + N(x, y) \]  --------- (1)

\[ I'(x, y) \] – The Initial Image

\[ N(x,y) \] – The noise

\[ I(x,y) \] - Noise affected Image

PoissonNoise

segmentation .Mainly it is based on the particular application. Image pixels are classified according to the pathological regions and anatomical regions. The purpose of the segmentation of the lung region in the CT image is to achieve a better point of reference in the image. It starts by applying the image slicing algorithm to each DICOM CT image of the raw data. The resulting binary images are then analyzed to choose among them the best image that may help in extracting the lung regions from the raw CT-image data with a certain degree of accuracy and sharpness. To refine the chosen image, other techniques were used for different purposes in a sequence of steps. Erosion, median filter and dilation steps aim to eliminate irrelevant details that may add extra difficulties to the lung border extraction process. The outlining step aims to extract the structure’s borders. The lung border extraction step aims at separating lung structure from all other uninteresting structures. [5]. Adjacent regions are significantly different with respect to the same characteristic(s). We has done this using edge detection and steps for it are:

1. Edge is a set of connected pixels that lie on the boundary between two regions.
2. Edges are detected by Sobel Methods.
3. Accuracy is good in sobel method.

The MatLab Code is given below:

```matlab
bw=rgb2gray(i);
imshow(bw);
b=edge(i,'sobel',0.09);
```

Comparison of edge detection methods are given as Table 1 and Table 2.

V Segmentation of Extracted Lung Region

The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image Segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics [6]. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture[9].

From our segmentation task perspective, spurious white spots corresponding to blood vessels inside the pulmonary regions should be regarded as noise and thus be removed by some suitable method. A morphologic filter [9] was then applied, to eliminate isolated white spots, in a way that the final binary image exhibits a single white connected region. The resulting binary image has only a white region and several dark regions: two of them correspond to the lungs. The main remaining problem consists on rejecting the outer black regions in the binary image. This was simply achieved by a morphologic filter operation using the mask vertices as starting points.

VI Conclusion

The proposed method for lung segmentation in CT images is fast and more accurate to
provide a high-quality of data set for further analysis and visualization procedures.

References


2. Saleem Iqbal, Khalid Iqbal, Lung Segmentation Using Computer Aided Diagnosis, INTERNATIONAL JOURNAL OF ACADEMIC RESEARCH Vol. 3. No. 5. September, 2011,


Intelligent Platform for Mobile Agents

M. Kavinilavu\(^1\) \quad \text{M.Sathy}\(^2\)

Department of Computer Application
Erode Builder Educational Trust’s Group of Institutions.

Abstract- A software agent offers a new computing paradigm in which a program, can suspend its execution on a host computer and can transfer to another agent enabled computer on network so that it could run there. Such mobile agents are used when there is limited capacity in computers. But at times, the systems could not be able to run even the mobile agents. This paper proposes an approach to execute mobile agents on any sort of systems even with limited capacity. A platform that supports various mobile agents depending upon the capacity of the system will be developed. The mobile clients are differentiated into two types depending upon their capacity and the corresponding mobile agent is chosen for the traversal.

Keywords - Mobile agents, Mobile clients.

1 Introduction

Mobile agents traverse between various mobile clients for the execution of certain applications. These mobile agents are used in order to overcome the limited capability problem of mobile clients.

Even with mobile agents, a new problem arose. The mobile clients were unable to run even the mobile agents. The mobile agents usually carry code to execute the application and the data to be executed. This load is minimized for certain mobile clients so that they may execute mobile agents easily.

When the configuration of client is suitable enough to send mobile agent containing both code and data to server, then the same will be sent to server. In the specialized search of a large free-text database, it may be more efficient to move the program to the database server rather than move large amounts of data to the client program. The result will be made available at the client. When the client configuration is minimal, so that mobile agent containing both code and data cannot be sent to the server, then only the state of the client will be sent to server. The code will be stored already in the server. Platform designed will obtain the configuration of client associated. According to the configuration the mobile agents will be sent, either with state alone or with code also.

2 Existing works

Accessing a mobile agent is possible only with a mobile client having sophisticated and powerful resources. Sending a mobile agent to another computer to do the data collection and computation can save battery power. This also means that heavy computations that will take long time on a mobile device can be executed at a more powerful computer with more memory, faster CPU and without any power limitation. In addition, problems with slow and error-prone network connections can be reduced by dispatching the agent to another machine, and get the agent with the result back when finished. Most mobile devices will consume a lot of battery power when transferring data over a network a long period of time. Mobile agents can reduce these problems.

Any application, to be run using mobile agents is done as follows: The data and required code of the application will be put together as mobile agents and will be sent to the server. It will be executed at the
server and the results will be available at the client.

The problem faced here was, executing mobile agents at the client became difficult because of very limited capacity.

3 Motivations

According to the survey in [8], Aglets, Voyager and Grasshopper were among the four best mobile agent platforms; their ranking was: 1) Grasshopper,
2) Jumping Beans, 3) Aglets and 4) Voyager. Tryllian, JADE, Tracy and SPRINGS is also available. The last version of Tryllian has been released recently as open source, JADE is a very popular platform for the development of agent-based systems, the development of Tracy has led to the publication of the most recent book on mobile agents (since some years ago), and SPRINGS has been developed very recently and offers a promising scalability.

4 Proposed System

In this paper, the proposal is made basically to eliminate the limitations with minimal resource mobile agents. Till now the mobile clients with minimal resources cannot access mobile agents and produce results successfully. To overcome this a platform is been designed so that it works in two different ways.

4.1 Differentiating mobile clients

The platform proposed for design will get the configuration of the client. Using it, the mobile clients will be differentiated into thick and thin mobile clients.

4.1.1 Thick mobile client

When the client is said to have enough resources, it is called as thick mobile client. When such a client is in use, both the code and data is combined in mobile agent and sent for execution. The result will be then sent back to the mobile client. Thus the power of mobile device is saved by making the execution done at some other powerful system. The mobile agent used here is called Thick Mobile Agent.

4.1.2 Thin mobile client

When the configuration of the mobile client is not enough for a particular application then it is called as Thin mobile client.

When such a client is in use, only the data called as state will be sent to the server by using mobile agents. The code for the application will be stored already on the server. Thus, even if the mobile client does not support the required resources, the application could be executed. This method avoids the excessive resource usage for dispatching the code to the server. Tailoring for specific devices can also be done to improve the usability of the agent clients based on the screen size, graphical capacities and data entry capabilities of the devices.

5 Architecture

5.1 Thick client

For the thick client the database containing the code (in a database) and the data to be executed will be available.

The client configuration will be sent to the initiator. The initiator will then contact the decision maker. The information sent by the client is extracted and is stored in the database for further verification. The decision maker will use the threshold values and will reply whether it is thick or thin client.

As the reply is thick mobile agent, the code and data will be sent to the dispatcher which will be forwarded to the server for further execution. The result obtained after execution will also be sent to the dispatcher which will then reach the thick client directly.

5.2. Thin client

For the thin client, database will not be available. The code will be present only in the server. Hence the data alone should be sent. The thin clients also follow the same traversal path, except that the code to execute the application will be
present in the server.

In case of thin clients, the dispatcher has some more work of combining and splitting the code and data.

5.3 Platform

The platform will contain the decision maker. The threshold values for decision making will be present in the database already. This will be specified depending upon the efficiency of the clients.

The decision will be made based on the capabilities like memory, processor, java support etc. Extractor is used to extract the client capabilities obtained from the initiator.

A central part of the architecture is the agent system repository where agent information and agents can be stored along with client information such as client device capabilities. We have defined a thin agent client that only deal with the agent state, where the code of the agent will be stored on the agent server. The thin agent client contains code for transmitting and receiving agent data and a simple GUI for manipulating this data. For devices with sufficient memory and CPU, mobile agents will run locally on the device. Above the repository the general agent server services are located that controls the essentials agent services such as registration of agents, locating agents, management of agent lifetimes (initiate agents, clone agents, kill agents) etc. The rest of the architecture is split into two main parts; one for thick agent clients, and one for thin agent clients. The thick agent client is able to execute mobile agents locally, while the thin agent client only is able to manipulate the agent data (state).

There is only one part that distinguishes the architecture of the thick and thin agent client; the Agent joiner/splitter. The Agent joiner/splitter makes it possible to split the data and code of the mobile agents before dispatching the agent to the thin client. This makes it possible to access mobile agents on less capable devices. The Agent initiator is responsible for initiating the agent clients first time, or reconfiguring the agent client based on changes in the mobile agent software or hardware/software changes in the client. Although there are two Agent initiators drawn in the figure 2, only one such service is running (same for thick and thin clients). The last component of the architecture is the Mobile Agent Dispatcher which makes it possible dispatch agents between clients and server. For thin clients the Agent joiner/splitter is used to split the agent before dispatching to a client, and to assemble the agent when received from the client.

5.4 The Agent Initiator

When a client connects to the agent server for the first time, it must state the client device capabilities in terms of CPU speed, execution memory, storage, and Java Virtual Machine edition and version. Although a client device has enough memory and fast enough CPU, there could e.g. be no Java virtual machine available that support serialization for the device. This is the reason that Java Virtual machine is also a part of the client capabilities. The capabilities are described in XML. Figure 6.1 shows an example of a description for a client running on a Palm Tungsten T PDA. The client capabilities described in XML consist of five main parts: Device, CPU, Memory, Java and Download. The Device part is used to describe the device in terms of name and operating system running. The next two parts CPU and Memory describe the device capabilities in terms of processing power and executing memory and storage space. The Java part describes the available Java environments on the device in terms of edition and versions. The last part, Download, makes it possible to specify how the agent client should be downloaded to the device. Our client capability description is a small subpart of terminal resource capabilities in Composite Capability/Preference Profiles (CC/PP) tailored for mobile agent purposes. The terminal resource capabilities in CC/PP are tailored for use with Web browsing.

01: <ClientCapabilities>
02: <Device>
03: <Name>Palm Tungsten T</Name>
04: <OS>PalmOS 5.0</OS>
When a mobile agent client is initialized the first time, it follows the following process:

5.4.1. **Install agent initiator:** The agent initiator is a simple Java program that must be installed on the mobile device before installing the agent client. This program is used for extracting device capabilities on the client. The devices capabilities that cannot automatically be detected must be entered manually by the user. In addition, the agent initiator contains software for communicating with an agent server.

5.4.2. **Configure the agent initiator:** The user should look through the device capabilities detected, and add missing, or change faulty information.

5.4.3. **Initiate the agent initiator:** The agent initiator sends the device capability information to the agent server.

5.4.4. **Install agent client:** Based on the device capabilities, a suited agent client will be sent to the device and then installed. The agent client can be sent directly using the agent initiator, or indirectly using transport channels such as email or HTTP- download.

If any characteristics of the client are changed (e.g. a new virtual machine for Java has been released), a new initializing process is initiated. Also if there are major changes of the agent system itself, the agent server may initiate a client update process (similar to the process shown above). The reason for this is that the new version of the mobile agent system could be more CPU and/or memory efficient than the previous one, making it possible to run the full mobile agent system on clients that previously were not powerful enough.

![Diagram of mobile agent architecture](image)

**Figure 2:** A logical view of the mobile agent architecture

5.5 **The Agent Joiner/Splitter**

For clients that cannot run the mobile agents locally (because of client capabilities), we have designed an Agent joiner/splitter for allowing these clients to access the agents anyway. This is done by separating the state (or data) of the agent with the code of the agent. This means that the thin agent client only receives the data part of the mobile agent while the code part resides on agent server. In addition, the client on the mobile device has a graphical user interface (GUI) making it possible to instruct the agents. When the user decides to send an agent from the mobile device, the state of the client.
agent is transmitted to an agent server. This means that both the code and the state of the agent are joined on the agent server. Figure 4 illustrates the process view of a migration of an agent (client application) from the mobile device to the agent server. When an agent migrates to another agent server or thick agent client, both the code and the state are transmitted (as for normal mobile agent systems).

6 Issues to be concentrated

6.1 Agent Separation:
Since the mobile devices still have minimal resources to run mobile agents, it was never an option to run an entire agent on a mobile device. To overcome this problem, the agent application was separated in two parts (code and data), with only the data residing on the mobile device along with the user interface classes.

6.2 Writing to the Memory:
Since an agent can be configured to operate for a long period of time, it was necessary to save the agent ids in the non-volatile memory of the mobile device. This makes it possible to exit the application, and recall the same agent the next time the application is started. J2ME’s Record Management System (RMS) was used to store the agent id.

7 Related Works

7.1 JADE
Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. Although, it is possible to run JADE on all devices that support J2SE, there are currently only few mobile devices that are able to that. This means that JADE only partially solves the problem of running mobile agents on mobile devices, since only the most powerful ones are capable.

7.2 Mobile Information Agents
The aim of project MIA (Mobile Information Agents) is to develop an intelligent information system, which puts information of local relevance from the World Wide Web into the hands of a mobile user. All information needed for such a system is available on the World Wide Web somewhere. The MIA-project implements a system that collects this information, and gives it into the hand of the user. The MIA system also needs to know where the user is. This can be done by using a GPS-receiver or the mobile phone can be tracked in the mobile network.

7.3 Personalized Information Retrieval Service
The design of a mobile-agent system that provides a mobile user with a personalized information retrieval serviceand we describe the implementation of the infrastructure for such a system. This “Personal Agent System” gathers information from the Internet and uses context-aware mechanisms to manage the information according to a mobile user’s needs and preferences. The user’s schedule and location are the context indicators in this system. These indicators are critical in ensuring that users obtain only the information they want, receive information in a form that is most useful for viewing on their mobile device, and is notified of new information in a minimally intrusive manner. The system incorporates a rule-based learning system to enhance the personalization achieved by the system.

8 Conclusions
Mobile devices even with very limited capability will be able to execute any sort of application very easily. All the mobile devices may not have enough capability to run certain large applications,
wherein this mobile agent platform will be much helpful. Not all of these limitations such as the screen size, network bandwidth, battery capacity, and input devices, have much influence on a mobile agent application. There are other more noticeable constraints that limit an agent application. However, these constraints are possible to overcome by this platform.

9 References


[7] Mobile Information Agents for the WWW. MIA Research Group. MIA -


A REVIEW FOR APPLICATION OF SOFT COMPUTING IN INDUSTRIES
T.VIGNESH         D.SUDHA         N.CHANDRAMATHI
Bharathidasan College of Arts and Science, Erode

Abstract

Soft computing (SC) is an evolving collection of methodologies, which aims to exploit tolerance for imprecision, uncertainty, and partial truth to achieve robustness, tractability, and low cost. SC provides an attractive opportunity to represent the ambiguity in human thinking with real life uncertainty. Fuzzy logic (FL), neural networks (NN), and evolutionary computation (EC) are the core methodologies of soft computing. However, FL, NN, and EC should not be viewed as competing with each other, but synergistic and complementary instead. SC has been theoretically developed for the past decade, since L. A. Zadeh proposed the concept in the early 1990s. Soft computing is causing a paradigm shift (breakthrough) in engineering and science fields since it can solve problems that have not been able to be solved by traditional analytic methods [tractability (TR)]. In addition, SC yields rich knowledge representation (symbol and pattern), flexible knowledge acquisition (by machine learning from data and by interviewing experts), and flexible knowledge processing (inference by interfacing between symbolic and pattern knowledge), which enable intelligent systems to be constructed at low cost [high machine intelligence quotient (HMIQ)]. This paper reviews applications of SC in several industrial fields to show the various innovations by TR, HMIQ, and low cost in industries that have been made possible by the use of SC. Our paper intends to remove the gap between theory and practice and attempts to learn how to apply soft computing practically to industrial systems from examples/analogy reviewing many application papers.

Keywords—Chaos computing, computational intelligence, evolutionary computation, fuzzy logic, immune networks, industrial applications, neural networks, soft computing.

I. INTRODUCTION

In 1987, fuzzy control was successfully applied in industrial plants in Japan. In the late 1980s, neuro-control was used for robot arms (including the robot arm of the space shuttle, chemical processes, continuous production of high-quality Manuscript received January 15, 2001; revised April 9, 2001. Y. Dote is with the Muroran Institute of Technology, Department of Computer Science and Systems Engineering, Muroran 050-8585, Japan S. J. Ovaska is with the Helsinki University of Technology, Institute of Intelligent Power Electronics, FIN-02150 Espoo, Finland. Publisher Item Identifier S 0018-9219(01)07606-X. parts, and aerospace applications) in the U.S. [1]. In 1991, the Berkeley Initiative in Soft Computing (BISC) was established as an ILP (Industrial Liaison Program), with L. A. Zadeh as its director. Since the establishment of BISC, researchers throughout the world have been studying soft computing, i.e., the fusion of fuzzy logic (FL), neural networks (NN), and evolutionary computation (EC) [2]. The term computational intelligence, as defined by Zadeh, is the combination of soft computing and numerical processing. This
term was first used in 1990 by the IEEE Neural Networks Council. Three IEEE International Workshops on Soft Computing in Industry have been held in Muroran, Japan, in 1993, 1996, and 1999, with Zadeh as plenary speaker each time [3]–[5]. The first workshop put emphasis on the fusion of neural networks and fuzzy logic. The performance of chaos computing-based communications systems is compared with those of conventional ones in [20].

V. CONSUMER APPLIANCES

A. General View

The field of consumer or home appliances is not a popular research area in the academic community. Almost all such research activities are related to practical product development.

C. Cooling and Heating

The cooling and heating class of applications discussed here consists of three principal products: electric or gas rice cooker, the tastiest cooked rice is supposedly obtained only by using a traditional cooking stove, which generates intense heat. Soft computing control is used to mimic the characteristics and behavior of traditional rice cookers and experienced cooks. Nitta [30] presented the design for an intelligent rice cooker that is based on neurofuzzy reasoning. The necessary intense heat is generated by magnetic induction heating, and the expert cooking process is reproduced by neuro-fuzzy reasoning. During this sensitive multistep process, both the heating temperature and amount of water are finely controlled by rules according to given input information, e.g., preferred rice stiffness, type and amount of rice, and initial amount of water. Shim et al. made a further step toward one-button control and true "heartware" by proposing a rice cooker that can estimate the amount of rice [31]. Neuro-fuzzy reasoning with evolutionary computation-based fine-tuning of estimation rules is used again in their scheme. They also proposed a microwave oven that uses similar methods for optimal control of the entire cooking or defrosting process.

VI. ELECTRIC POWER SYSTEMS

A. General View

Neural networks were applied already in the early 1990s to electric power systems. The first conference on application of artificial neural networks to power systems was held in 1991. In the mid-1990s, fuzzy logic was applied to power system applications [58] such as control, operation, and planning. Soft computing was applied to power systems in the mid-1990s as reported in [59], which describes in detail the ity swings for use in high-speed control on the basis of synchronized phasor measurements. A highly successful prediction rate in real-time was obtained in simulation tests on a sample power systems [44]. Bomfirm et al. developed a method that simultaneously tuned multiple power system damping controllers using modified ECs. They suggested that human expertise will be able to be captured and readily implemented in a more elaborate fitness function [45]. Segal et al. proposed a new approach for real-time tuning of the parameters of a conventional power system stabilizer using a radial basis function neural network (RBFN). The dynamic performance of a system with an RBFN was shown to be quite robust over a wide range of loading conditions and equivalent reactances [46]. Cho et al. developed an expert system using fuzzy relations to deal with uncertainties imposed on fault diagnosis of power systems.
D. Operations
Charytoniuk et al. explored an alternative approach of load forecasting based on indirect demand estimation from available customer data, instead of using time series of load changes and weather factors recorded in the past (conventional method). Neural networks were designed and trained on the basis of the aggregate demands of groups of surveyed customers in different categories [48]. Kim et al. proposed a new method for a reliable short-term load forecasting for special days in anomalous load conditions with a neuro-fuzzy approach, which had not been possible using conventional neural networks due to dissimilar load behaviors of holidays compared with those of ordinary weekdays during the year, and due to insufficient number of training patterns. In their proposed method, special days are classified into five different day-types. Five NN models for each day-type are used to forecast the scaled load curves of special days, and two fuzzy inference methods are utilized to forecast the maximum and minimum loads of those special days. Finally, the results of the both methods are combined to forecast the 24-hour load pattern of special days. Their test results showed very accurate forecasting with an average relative error of 1.78% [49].

Iokibe et al. developed a new method for short-term prediction of daily peak electric power demand. First, a nonlinear chaotic time series was considered method by fuzzy inference was used, and the computation time was reduced remarkably due to fuzzy inference rather than neural networks. Their method was applied to accurate prediction of daily peak electric power demand [73].

MacGill et al. proposed a decentralized coordination framework for operating power systems with dispersed generation and energy storage (for optimizing the dispatch and short-term scheduling). Their method combined elements of dynamic programming with EC. Each power system resource evolved a “future benefit” function that described the impact of its own possible decisions on future power system operation. This “dual evolutionary programming” approach can handle complex resource models and objective functions [51].

E. Planning
Richter et al. provided an EC solution to the profit-based unit commitment algorithm. It was confirmed that EC is a useful tool in searching large discrete solution spaces; and the space of the particular solution was quite large, making EC appropriate for solving the unit commitment problem. It gave more information to users due to the flexibility of the method [54]. Huang et al. developed an approach using an EC-based neural network and dynamic programming to solve power system unit commitment problems. First, a set of feasible generator commitment schedules was formulated by EC-enhanced neural networks. Those pre-committed schedules were then optimized by the dynamic programming technique. By the proposed approach, harmful learning stagnation was avoided. The stability of the neural networks and accuracy were significantly increased. The feasibility and practicability of the proposed method were confirmed experimentally [55].

VII. MANUFACTURING AUTOMATION AND ROBOTICS
A. General View
The term intelligence has been frequently used in this field since robotic technologies that mimic human thinking and behavior of bio-systems have been developed. Contemporary intelligence is sometimes considered to be interactive information processing among human beings, environment,
has become popular in Japan [70]. This technology is needed for the development of human-friendly robots. Other technologies, e.g., fuzzy associative memory and chaotic computation have also been used for developing human-friendly robots (intelligent robots, welfare robots) [68]. Soft computing is widely used in this field.

B. Application Fields
Soft computing has been used in the construction of intelligent robots and manufacturing systems and for solving nonlinear and uncertain problems in the fields of hands and manipulators, mobile robots, multiagent robots, welfare robots, emotional pet robots, and manufacturing systems. It is expected that soft computing will play an increasing role in the realization of human-friendly systems in the future.

Robots and manufacturing systems and their characteristics that have been made possible mainly by soft computing are summarized in Table 6. These applications are discussed in more detail below.

C. Hands and Manipulators
Lin et al. developed compact fuzzy decentralized controllers including sensor fusion schemes and introducing human skills through communication lines for a five-finger robot hand with 17 degrees of freedom. 17 potentiometers, 18 tactile sensors, and 17 actuators were installed in the design.

D. Mobile Robots
Baranyi et al. proposed an improved vector field-based guiding model as an extension of the potential-based guiding model. A simplified neuro-fuzzy approximation algorithm was applied to the realization of models for the guidance of mobile robots [65].

E. Multiagent Robots
Ishiguro et al. developed an architecture for behavior arbitration based on artificial immune networks. Antigens and antibodies in the artificial immune network were used as agents for environment description and decision-making for action, respectively.

H. Manufacturing Technologies
Iokibe et al. proposed a fault diagnosis method using chaos computing for chaotic time series analysis, fuzzy reconstruction of chaos state trajectories and separation of white noise from the trajectories. Their method was applied to fault diagnosis for rotating machine parts, and it was found that the use of this method could greatly reduce the required computational time [73].

Djordjevich et al. developed a system for monitoring tool wear condition using neuro-fuzzy computing. The feed cutting force was estimated by an adaptive neuro-fuzzy inference system based on the measurement of servo motor feed currents [74].

Yen et al. proposed a wavelet-based feature extraction method for monitoring vibration conditions of dynamic systems. In their method, symptom vectors extracted by wavelet transformation were fed into the inputs of the neural network classifier [75].

VIII. POWER ELECTRONICS AND MOTION CONTROL
A. General View
It is well known that I/O mapping by an NN can be approximated by FL. However, an NN has advantageous knowledge acquisition capabilities by learning and more accurate mapping properties. On the other hand, FL can explain the I/O relations and is rich in knowledge representation. Besides, it is suitable for fine-tuning and representation of easily understandable knowledge expressions for human beings with less computation time.

In this field, systems are often nonlinear and uncertain. It is difficult to obtain rigorous mathematical models. Self-tuning (adaptive) capabilities and automated design methods are needed. Soft computing has innovatively solved such real-world problems at low cost.
For hardware realization of the schemes, fast DSPs are widely available [93].

**B. Application Fields**

In all application fields of motion control, welding, induction motor drives, reluctance motor drives, inverters, converters, and diagnosis, there exist high nonlinearities and uncertainties such as current dependent inductance, stray inductance and capacitance, eddy current, temperature dependency effects, skin effect, friction, gear backlash, and compliance, for which rigorous mathematical models cannot be obtained. Self-tuning (adaptive, robust) capabilities and automated designs methods are needed. Soft computing (FL+NN) has innovatively solved these real-world problems [85], [86], [93]. EC is not frequently used in this field for system optimization. Since relatively fast dynamic systems are dealt with, computationally efficient fuzzy neural networks are used. Power electronics and motion control have been the basic implementation technologies for robotics and automation. Human-friendly diagnosis technology in which intelligent information processing is essential is increasingly being developed [93], [95], [96].

Table 7 presents a summary of the power electronics and motion control, and the system characteristics that have been made possible mainly by soft computing, most of which are implemented using commercially available digital signal processors. These applications are described in more detail below.

**C. Motion Control (Including Welding)**

Fahn *et al.* proposed a new method for estimation and control of the speed of a nonlinear servomotor. They used also EC to extract numerical control rules from the input and output data. After an evolutionary process, the resulting numerical rules constituted a lookup table. Then a fuzzy neural network was trained using the numerical data in the lookup table as teaching signals, resulting in automatic generation of fuzzy rules. The NN was mainly used for parameter adjustments. Accurate and robust positioning for the highly nonlinear complex system was achieved [81].

Popovic *et al.* developed pulse torque control, whose torque pulse shape was inferred by FL from the desired disposition data. Such a control scheme was applied to high-precision positioning. Fine-tuning was applied using experimental data.

First the open loop characteristics of nonlinear positioners (static stiffness, hysteresis, drift, frequency response, and coupling effects) were experimentally investigated. Then, a cerebellar model activation NN control algorithm was applied to provide real-time learning and better tracking capability than that of conventional PID control [83]. Cook *et al.* used two NNs for modeling and control of a variable polarity plasma arc welding process, which is highly nonlinear. The input variables to the first NN were the desired crown and root widths. The output variables that were the inputs to the second NN were the torch standoff, forward current, reverse current, and travel speed. Variables were the resulting crown width and resulting root width, which were used for the actual welding. Each NN was trained using experimental data [84].

**IX. PROCESS ENGINEERING**

**A. General View**

Fuzzy logic was first used in the process industry in Japan in 1987. Since processes are usually nonlinear, uncertain and complex, highly skilled operators have controlled process plants. Fuzzy control was devised to mimic skilled operator’s control. In the U.S., neural networks were applied to the chemical process industry in the late 1980s. Since the chemical industry has typically a lot of operation data available,
neural networks are suitable for nonlinear time series analysis. Soft computing offers additional adaptation capability to solve nonlinear and uncertain process engineering problems. Since these processes are large-scale and complex, data mining technology, which has been developed since the late 1980s using heterogeneous methodologies, including soft computing methods based on pattern recognition technology, has recently been used for interpreting and understanding important associations hidden in large process databases [104]. Due to commercial confidentiality reasons, people working in process industries do not usually publish detailed technical papers; their work is focused on the development of practical products. However, data mining software for process industry is now commercially available [111]. Data mining provides the understanding of process and plant performance and, therefore, builds a solid basis for remarkable degree of cost savings and profitability. Data mining technology is being used in the following demanding areas [112]:

- load forecasting and operation guidance for air conditioning systems;
- monitoring of the performance of heating systems;
- inner state estimation for stills (soft sensing);
- quality modeling and quality improvement operation guidance for dissolution processes;
- virtual sensors for the paper industry;

DOTE AND OVASKA: INDUSTRIAL APPLICATIONS OF SOFT COMPUTING: A REVIEW 1257

- virtual sensors for a furnace;
- oil ingredient prediction;
- final quality prediction for chemical reactor process; and
- evaluation of drug effects.

B. Application Fields
The processes in the chemical, paper, and steel industries are all highly nonlinear and uncertain. Also, process systems are usually large-scale and complex.

C. Chemical Process Industry
McAvoy surveyed various kinds of neural networks applied to chemical processes for diagnosis, modeling, feedforward control, soft sensing, and nonlinear model prediction control to optimize plant operation. These NN systems created considerable economic benefits utilizing a lot of process data [97], [101].

Their controller was experimentally applied to a semibatch chemical reactor. A valve controller was designed for the lowest level. On the middle level, a real time adaptive nonlinear controller for a water and air tank was proposed. On the highest level, a supervisory PID controller was devised.

D. Paper Process Industry
Scharcanski et al. devised a simulator (model) for a paper forming process using a neural network. The new model yielded data corresponding to data obtainable along arbitrary scanning lines in planar stochastic fibrous structures, providing profiles, variances, histograms of local area density, and histograms of local free-fiber lengths. These results closely resembled experimental data from commercial paper samples obtained from radiographic or optical transmission images subjected to image analysis [105].

Viljamaa et al. proposed a fuzzy system to compute new target values for low-level controllers during grade changes in a paper machine.

E. Steel Process Industry
Kayama et al. devised a sensor fault detection scheme for a complex, large-scale feedback system using immune networks, Kohonen’s feature maps, and fuzzy inference. The sensors were antibodies connected to each other. Each sensor watched another sensor’s output and informed its abnormality by fuzzy decision
making from learning vector quantizations from other sensors. This method was applied to sensor failure detection of a large scale and complex furnace in steel industry [107].

A. General View
Transportation is a large field with diverse and challenging problems to solve. Since the field of transportation mostly serves ordinary people, passengers, human-orientation and safety in various controls, fault diagnosis, and logistics operations are of considerable importance. It can be seen from Table 1 that nearly 12% of all published conference and journal papers in the field of Transportation contain applications of soft computing. Based on this considerable proportion, it can be concluded that soft computing forms an important collection of methodologies in transportation research and development.

B. Application Fields
Elevators should be comfortable for passengers and their group dispatching control is complex. FL and EC are therefore often used in state-of-the-art elevator control systems.

Soft computing is an efficient means for constructing intelligent vehicles, since the machine, driver, and the driving environment are interacting with each other.

D. Road Transportation
Traffic-actuated fuzzy logic signal group control was developed by Niittymaki et al. It was shown that the proposed method was more efficient than other traffic signal control algorithms [115].

A fuzzy-neuro approach was used to represent the correlation of the attributes with the driver’s route selection. Based on training of the fuzzy neural network for the driver’s actual selections, the route selection function could be made adaptive to the decision-making of the driver [116].

The results of real-time simulation demonstrated that the proposed method could provide passengers with a much more comfortable ride [117].

E. Rail Transportation
Chang et al. proposed a dynamic train coast controller. A genetic algorithm-based method was developed for synthesizing the train coast look-up table before departing from each station for an interstation run. Both indices of punctuality and riding comfort were incorporated successfully into the fitness function in the form of a penalty factor [121]. Liu et al. developed a fuzzy neural network and examined its performance. It was successfully applied to the maintenance of ticket machines.

XI. FUTURE OPPORTUNITIES
The successful applications of soft computing (SC) suggest that SC will have increasingly greater impact in the coming years. Soft computing is already playing an important role both in science and engineering. In many ways, soft computing represents a significant paradigm shift (breakthrough) in the aim of computing, a shift that reflects the fact that the human mind, unlike state-of-the-art computers, possesses a remarkable ability.

XII. CONCLUSION
Soft computing is already a major area of academic research. However, the concept is still evolving, and new methodologies, e.g., chaos computing and immune networks are nowadays considered to belong to SC. While this methodological evolution is taking place, the number of successful soft computing-based products is increasing concurrently. In the majority of such products, SC is hidden inside systems or subsystems, and the end user does not necessarily know that soft computing methods are used in control, diagnosis, pattern recognition, signal processing,
etc. This is the case when SC is mainly used for improving the performance of conventional hard computing algorithms or even replacing them. However, soft computing is very effective when it is applied to real-world problems that are not able to be solved by traditional hard computing. Another class of products uses soft computing for implementing novel intelligent and user-friendly features. Soft computing enables industrial systems to be innovative due to the important characteristics of soft computing: tractability (TR), high machine intelligence quotient (HMIQ), and low cost (LC).
Packet forwarding prioritization (PFP) in routers is one of the mechanisms that is commonly available to network operators. This paper aims to identify the router functionality in forwarding of packets received from normal TCP client nodes and high-speed torrent-like TCP applications and assign priority to regulate and reduce the packet dropping. The proposed algorithm listens to the packets' arrival from both types of applications, i.e., normal and high-speed TCP packet data, calculates the router metric i.e., router’s capability such as packets count in queue, packets time to live in queue, packet resend times and drops the packets only if the incoming packets arrive at a rate more than the limit. Likewise, if the high-speed applications send packets to fill the router outgoing ability, then it won’t reduce the speed even if normal nodes send much packets. At that time, the router application must send an alert message to reduce the outgoing rate of high-speed applications so that normal client nodes will not be affected. This application enables users to discover network policies through measurements of packet losses of normal TCP packets and regulate the flow from both normal and torrent-like applications.

Keywords: Network inference, Network neutrality, Packet forwarding priority, Priority Queuing

I. INTRODUCTION
Packet forwarding prioritization has been available in routers for quite a while, and various models from popular brands. Network operators have come to rely on these mechanisms for managing their networks, for example as a way of rate limiting certain classes of applications e.g., peer-to-peer [1]. PFP can have a significant impact on the performance of applications, beyond those targeted by administrators. PFP can also severely impact the accuracy of measurement tools output and the effectiveness of network troubleshooting procedures. For example, measuring network path characteristics is critical for the diagnosis, optimization, and development of distributed services. PFP settings in routers, an example is forwarding priority based on packets’ protocols or port numbers, however, can potentially introduce a performance dissonance between the view portrayed by measurement tools and what is ultimately experienced by applications.

In this paper, we present an end-to-end approach for packet forwarding priority inference and its associated tool, POPI (Packet forwading Priority Inference). The couple of interesting challenges while designing and implementing POPI in following two ways. First, end-to-end inference accuracy of router properties can be severely affected by background traffic fluctuations. To overcome this challenge, POPI sends relatively large amount of traffic to temporarily saturate bottleneck traffic class capacity.
Second, while most existing inference methods assume certain independence or strong correlation models. An example is back-to-back probe packets, probe traffic of multiple packet types are neither independent nor strongly correlated. Altogether, our approach to PFP inference gives POPI better resistance against background traffic fluctuations and allows it to cope with the characteristics of its measurement traffic.

II. BACKGROUND OF STUDY

This paper indicates that there are three commonly available router mechanisms [2] to enforce priority or link-sharing on traffic classes that usually defined by IP protocol and TCP/UDP port number

i. Priority Queuing
ii. Proportional Share Scheduling
iii. Policing

A. Priority Queuing (PQ)

Priority Queuing allows the assignment of absolute priority among queues. Since queues are served based on their priority, this allows specified packet types to be always sent before other packet types. Various Queuing techniques available are

i. First-In First-Out (FIFO) Queuing
ii. Priority Queuing (PQ)
iii. Flow-based Weighted Fair Queuing (WFQ)
iv. Class-based Weighted Fair Queuing (CBWFQ)

1) FIFO Queuing

This queuing mechanism used to enhance the delivery model by giving specific traffic priority over other traffic. Because FIFO queuing cannot alter the order in which the packets are forwarded, the service experienced by any particular flow is going to depend entirely on the order in which the packets arrive. FIFO also has the potential problem of delaying short frames behind longer frames. This is caused by the longer frame filling the queue.

Figure 2.1.1 FIFO Queuing

Figure 2.1.1 FIFO Queuing shows the node that is ready to process is kept in the queue. Condition is checked when packet arrive and its stored or discarded.

2) Priority Queuing

Priority Queuing (PQ) ensures that important traffic gets the fastest handling at each point where it is used. It was designed to give strict priority to important traffic. PQ can flexibly prioritize according to network protocol incoming interface, packet size, source/destination address, and so on. In PQ, each packet is placed in one of four queues (high, medium, normal, or low) based on an assigned priority. Packets that are not classified by this priority list mechanism fall into the normal queue. During transmission, the algorithm gives higher-priority queues absolute preferential treatment over low-priority queues.

Figure 2.1.2 Priority Queuing

Figure 2.1.2 Priority Queuing is shown which describes packets that are assigned priority class and each has its own queue till the departure

3) Flow-Based Weighted Fair Queuing

For situations in which it is desirable to provide consistent response time to heavy and light network users alike without adding excessive bandwidth, the solution is flow-based weighted
fair queuing (commonly referred to as just WFQ). WFQ is a Cisco premier queuing technique. It is a flow-based queuing algorithm that creates bit-wise weighted fairness by allowing each queue to be serviced fairly in terms of byte count. For example, if queue 1 has 100-byte packets and queue 2 has 50-byte packets, the WFQ algorithm takes two packets from queue 2 for every one packet from queue 1. This makes service fair for each queue: 100 bytes each time the queue is serviced.

WFQ ensures that queues do not starve for bandwidth and that traffic gets predictable service. Low-volume traffic streams (which comprise the majority of traffic) receive increased service, transmitting the same number of bytes as high-volume streams. This behavior results in what appears to be preferential treatment for low-volume traffic, when in actuality it is creating fairness.

4) Class-Based Weighted Fair Queuing

Class-based weighted fair queuing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. CBWFQ enables a network administrator to create minimum guaranteed bandwidth classes. Instead of providing a queue for each individual flow, a class is defined that consists of one or more flows. Each class can be guaranteed a minimum amount of bandwidth.

Figure 2.1.3 Class based Weighted Fair Queuing

Figure 2.1.3 Class based Weighted Fair Queuing shows the packets assigned priority class and given to different queues based on weights assigned. More weight is assigned to high priority work.

B. Proportional Share Scheduling (PSS)

PSS such as Weighted Fair Queuing (WFQ) and Weighted Round-Robin (WRR) enables the assignment of bandwidth limit to traffic classes.

C. Policing

Policing restricts the maximum rate of a traffic class. Traffic that exceeds the rate parameters is usually dropped. The traffic class cannot borrow unused bandwidth from others. In the first mechanism, PQ sets absolute priorities between traffic classes. The other two mechanisms do not impose such absolute model; i.e., the loss experienced by one class depends on its allocated bandwidth and its traffic rate.

III. PACKET-FORWARDING PRIORITY INFERENCE

There may be several candidate metrics to conclude packet forwarding priority, such as packet loss, delay or out-of-order events. In this paper, we only use packet loss as the inference metric because it is the most direct consequence of a priority configuration.

PFP in routers are set in a per-interface basis. Prioritization of packets does not become evident until the associated link (or a sublink for a traffic class) is saturated, at which point the configured router will begin to drop packets based on its settings. This simple observation defines the basis of the approach used in POPI: In order to reveal packet-forwarding priorities, one needs to saturate the path available bandwidth for a given class to produce loss rates difference among different classes.
Figure 3.1 shows router A which has the bottleneck link with 10Mbps of available bandwidth, while router B has a total of 91Mbps of available bandwidth [6][8]. If, for a particular packet type, router B is configured to only allow 1Mbps using policing, we can saturate the sublink and detect the priority setting on router B. On the other hand, if PQ or PSS is configured at router B, this approach could not detect the priority setting unless the whole link B is saturated. Thus, POPI adopts a two-step approach to PFP inference:

(i) Saturate the link with relatively large amount of traffic and
(ii) Cluster packet types based on their loss rate ranks.

PFP can have a significant impact on the accuracy of network measurements, the performance of applications and the effectiveness of network troubleshooting procedures. Despite its potential impacts, no information on PFP settings is readily available to end users. In this paper, it uses packet loss as the inference metric because it is the most direct consequence of a priority configuration. PFP in routers are set in a per-interface basis. This observation defines the basis of the approach used in POPI: In order to reveal packet forwarding priorities, one needs to saturate the path available bandwidth for a given class to produce loss rates difference among different classes. Assuming the existence of a PFP mechanism in routers such an approach will succeed at uncovering priority settings in routers along a path if the available bandwidth for the controlled class is lower than the bottleneck available bandwidth of the path.

For every burst Figure 3.2 shows, loss rate ranks are computed by first sorting packet types in ascending order according to their packet loss rates in that burst and then assigning ranks in order. On observation, Identifying whether there is consistent difference among k ranks over n observations. Based on ranks packets are grouped. Grouped packets are assigned priority on loss basis and priority is inferred at user level.

POPI works in two probe modes, End-to-End Probe (EEP) and Hop-by-Hop Probe (HHP). In both modes, the sender sends multiple packet types toward the receiver. The receiver feedbacks certain information of every received packet to the sender, which is used by the sender to measure the end-to-end losses and reordering events along the path. EEP mode works exactly as the way described in above figure1. In this paper, it has demonstrated that POPI, an end-to-end priority inference tool, is able to accurately infer the router’s packet forwarding priority using loss statistics [4].

IV. DESIGN OF POPI

In this section, we present the design of the POPI tool which has four steps

1. Forwarding packets from sender to receiver
2. Measuring link loss rate
3. Change Queue Priority based on port based load sharing
4. Alerting nodes which send more packets
A. Forwarding packets from sender to receiver

The router metric such as incoming bit rate of normal TCP nodes, high speed torrent like application’s port’s incoming bit rate and time to live in queue are stored first. The client application sends packets to server application which acts as router, the packets if arrived has reached the maximum limit of router then they are added in queue. The packets remain there up to their time to live. If the traffic is less enough, then the packets are routed to destination nodes otherwise if time exceeds they are simply dropped.

B. Measuring link loss rate

The packets lost details are logged in a database table to measure the loss rate. The details are analyzed whether the high speed port applications cause the more loss rate. Packet dropping start time and end time are kept in log so as to measure the loss rate.

C. Change Queue Priority based on port based load sharing

The packets lost details are measured and queue priority is maintained between normal queue and high speed port’s queue. Details are gathered such that the incoming packets arrived quickly from high speed port’s application are responsible for packet loss in normal traffic.

D. Alerting nodes which send more packets

The link loss rate is measured and decided whether the high speed torrent ports are responsible for that loss. If the condition is met, then the loss amount as well as rate is calculated and nodes that containing high speed torrent applications are notified to reduce the packet sending speed. The new approach includes maximum throughput algorithm to serve the nodes better. Maximum throughput scheduling is a procedure for scheduling data packets in a packet-switched best-effort communications network, typically a wireless network, in view to maximize the total throughput of the network, or the system spectral efficiency in a wireless network. This is achieved by giving scheduling priority to the least "expensive" data flows in terms of consumed network resources per transferred amount of information.

V. CONCLUSION

This paper keeps log the packet queues and drop details. The continuous packet drops are easily notified and alerting procedure is invoked to reduce the loss rate. The new approach helps in efficient packet forwarding in the router. The new system uses maximum throughput scheduling algorithm so as to serve high speed as well as normal TCP packets to flow efficiently. The difficulty in eliminating continuous packet drop in the router is eliminated by using this application. It reduces the router overhead in queue priority management. The user interface assists in analyzing the router efficiency effectively. The application is tested well and end users satisfaction is found to be more. The application works well for given tasks in network environment. Any node with .Net framework installed can execute the application and identifies the best site.

The system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The system is very fast and any transaction can be viewed or retaken at any level. Error messages are given at each level of input of individual stages. This software is very particular in reducing the difficulty in analyzing the router algorithms. The router application if designed using web services, then it can be used across all platforms. The application if developed as web site can be used from anywhere. The multiple queues for same packet type can be implemented in router to regulate multiple homogeneous packets traverses along multiple paths. The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.
REFERENCES


MANET SECURITY WITH OPTIMIZED ROUTING TECHNIQUE USING DYMO ROUTING PROTOCOL

Mohan C
M.E-Computer & Communication Engineering
Department of Information Technology
Kongu Engineering College, Perundurai, Erode,
Tamil Nadu, India

Varadhaganapathy S
Professor
Department of Information Technology
Kongu Engineering College, Perundurai,
Erode, Tamil Nadu, India

Abstract- Mobile Ad hoc Network (MANET) is defined as a self organizing dynamic topology network formed by a group of wireless mobile nodes. In Mobile Ad hoc Network (MANET), nodes can move around arbitrarily, so any nodes can join or leave the group at any time. The security is important for basic network functions like packet forwarding and routing. Here authentication is used to provide security to Ad hoc networks. Authentication can be provided by using Double Hash Authentication Technique (DHT) to protect the routing information. The Route Optimization is provided by Self Healing Optimized Routing Technique (SHORT). The Dynamic MANET On Demand (DYMO) routing protocol used for simulation which performs well under high mobile environment.

Keywords: Mobile Ad hoc Network (MANET), Double Hash Authentication Technique (DHT), Self Healing Optimized Routing Technique (SHORT), Dynamic MANET On Demand (DYMO).

I. INTRODUCTION

A Network is a group of nodes logically connected for the sharing of information among themselves. Based on the medium used for communication, the network is classified into two types; they are wired networks and wireless networks. Wired networks use coaxial, twisted pair or fiber optic cable for establishing connection between the nodes to share information. Wireless network uses radio frequency (RF) for transmission and reception of data in air medium instead of using physical cables. There are currently two variations in wireless networks; they are infrastructure based networks and infrastructure less ad hoc networks.

The infrastructure networks are also known as Cellular network, have fixed and wired gateways. They have fixed base stations which acts as an infrastructure through which the mobile nodes can communicate with each other and send information. The infrastructure less networks are also known as Ad hoc Networks, as they have no fixed base stations. All nodes within their communication range communicate directly with one another and share information. In Mobile Adhoc networks, nodes can move around arbitrarily, causing the network topology to change dynamically.

Due to the dynamic topology of the network frequent path breaks occur in MANET which causes more utilization of resources like bandwidth. In Mobile Ad hoc Network, a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. The routing protocols used to find path and forward the packets in traditional wired networks are not directly used in mobile ad hoc networks due to highly dynamic topology of mobility of nodes. On demand routing protocol is used in this approach because it finds the path and forwards the packets only whenever it is required by using connection establishment process [3]. On demand routing protocols incurs less overhead to keep track the mobility in the network when compared to other protocols in
MANET. Dynamic MANET On demand (DYMO) routing protocol is used in this approach because it is more suitable for high mobile environment.

It is hard to provide security to MANET as the topology of the network varies constantly. To secure the ad hoc networks the attributes like availability, authentication, integrity and non repudiation are to be considered. Here authentication is considered which is used to verify the identity of node or a user [6]. Every transmitting or receiving node has its own signature which it uses to authenticate that the packets has been transmitted or received by the legitimate node which it is intended to receive.

II. RELATED WORKS

In MANET security is an important in functions for packet forwarding and routing because the mobile nodes are more prone to security threats. To provide security attributes like authentication, confidentiality, availability, integrity and non repudiation should be considered.

A. KEY DISTRIBUTION

In key distribution a common public key is shared between the node and its two hop node group. This key is kept secret against the one hop node group. The source node now generates random key Ks and encrypts it with the public key of the two hop node group that it has already shared. On receiving the encrypted key Ks the receiver node decrypts with the corresponding private key. Every node will share a random key Ks with its corresponding two hop node group. Due to mobility some new node can join or leave the two hop node group so the public key of the two hop nodes should be changed periodically using key redistribution method.

B. PROVIDING AUTHENTICATION TO THE NODES

The Authentication can be provided using Double Hashing Technique (DHT) [8], here Hash function is used to generate a fixed length output for a variable length message. Here Public one way hash function is used to authenticate the RREQ twice. So the routing packets also include two hash values along with RREQ they are (H1, H2). H1 is used to prevent the current node that is the sender from modifying the packet. H2 is used to check whether the received routing packet has been modified by checking with the H1 value. The Double Hash Technique involves the following algorithm [2].

- Generate RREQ packet from source node. It includes RREQ={S,L,H,R} where S- Source Identity, L- sequence number (RREQ), R- Routing information. H- Hop count
- Source node then multicasts the RREQ with Hash value{S, L+1, H, R, H1, 0} to its multicasts group
- Any intermediate node within this group can verify the authenticity of packet. H2=0(from source node); H1 = H(SL+1H1HRK) K- Secret key shared by two-hop node and source
- Before forwarding the packet increment the hop count by 1 and copy H1 value to H2 and calculate the new H1 value. i.e. H1=H (S\(L+1\)H1\(H\)R\(K\)); H2=H(S\(L+1\)H1R\(K\)) where \(K\) is common secret key between intermediate node and two hop node
- Forward the Routing packet to its Multicast group
- On Receiving {S\(L+1\)H1\(H\)R\(1\)H2\(H\)} nodes within the group can use (S, L+1, H+1, R) and public hash function to calculate H(S\(L+1\)H1\(R\)\(K\))
- Compare this value with H2 and validate whether routing packet is modified by intermediate node
- If intermediate node wants to modify the packet it has to forge the H2 value before forwarding the packet

The same concept is applied for RREP from destination to source [7].

C. OPTIMIZATION OF ROUTE LENGTH

The route generated by the on demand routing protocol may deviate from the optimal path due to lack of knowledge about the global topology and also due to mobility of the nodes in Mobile Ad hoc Networks. This affects the network performance and consumes more resources in terms of bandwidth battery power etc. Here Self Healing optimized Routing Technique (SHORT) is used to optimize the routing process. In SHORT every node monitors the route continuously and tries to optimize it whenever a better route to destination is available. Thus the SHORT algorithm enhances the performances in terms of bandwidth without incurring any additional cost. On demand routing protocol is used in this approach which is more efficient for
large networks because they establish the route whenever it is needed [1].

The Path aware SHORT algorithm is used in this approach used will work well with any underlying routing protocol. In Path aware SHORT algorithm Hop count field and Hop comparison array are used. Hop count field is initialized to 0 for sender and incremented at each hop. The Hop comparison array which holds the source, destination address, hop count and neighbor node address. In Path aware SHORT node transmit the packet within its transmission range, the receiver of the packet checks the header for source, destination, neighbor node address and hop count. This information’s are stored in hop comparison array and used to find the shortest path to the destination. The old entries are replaced by the newer entries. The delay in finding a shortest path is reduced by using SHORT algorithm and Double Hash Technique together. The figures 1,2,3 gives the complete description of the working of SHORT algorithm in the mobile environment [7].

![Figure 1: Routing Path](image1)

![Figure 2: Path change due to Mobility](image2)

**III. SECURED AND OPTIMIZED ROUTING USING DYMO ROUTING PROTOCOL**

To provide authentication based security and optimized routing for mobile ad hoc network using Dynamic MANET On demand (DYMO) routing protocol. The authentication can be provided using Double Hash Technique (DHT) and the route to the destination can be optimized by using Self Healing Optimized Routing Technique (SHORT). The Hash function is used to provide authentication to the mobile nodes. The

![Figure 3: After applying SHORT](image3)
The formula used to calculate the Hash value in this approach is

\[ H = h(S \cdot L + 1 \cdot H \cdot R) \mod 10 \]  

(1)

For calculating \( H_2 \) the hop count is reduced by one hop [as the route request packet will route from one node to its neighbor node the hop count increases by one hop so it is reduced by one hop to get the result \( H-1 \)] and the hash value \( H_2 \) is calculated by using the same formula.

The Dynamic MANET On demand routing protocol (DYMO) is used in this approach [4]. DYMO is an On demand routing protocol which is an extension of AODV and performance much better in high mobile environment.

![Figure 4: Path Accumulation Feature of DYMO](image)

The DYMO which performance much better in high mobile environment than any other On demand routing protocols because it has path accumulation feature within it. With path accumulation feature DYMO holds all the information about the intermediate nodes which it passes through. This information is more useful during the selection of route under high mobility where there is a frequent path break occurs. This information helps the local nodes to establish the route if it has the route to the destination. Major advantage of DYMO over other on demand routing protocol is that, it has high packet delivery ratio, low end to end delay and low control over head under high mobility environment.

IV. SIMULATION PARAMETERS

Evaluation of the performance of the DYMO protocol was done different for different scenarios by varying speed of the nodes for packet transfers. This simulation has been done using QualNet as the simulator.

QualNet is a discrete event simulator. It is derived from Glomosim simulator. It provides a GUI based environment to work with various routing protocols. It has four major components Architect, Analyzer, Packet Tracer and File Editor. The Simulation parameter includes the flowing which is required to perform the following security functions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terrain Dimension</td>
<td>1500*1500</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>50</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random way point</td>
</tr>
<tr>
<td>Propagation model</td>
<td>Two-ray Rayleigh fading</td>
</tr>
<tr>
<td>MAC protocol</td>
<td>IEEE 802.11 DCF</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>DYMO</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300 sec</td>
</tr>
<tr>
<td>Antenna type</td>
<td>Omni directional</td>
</tr>
<tr>
<td>Transmission range (m)</td>
<td>200</td>
</tr>
<tr>
<td>Node speed (m/s)</td>
<td>0.5, 10, 15, 20</td>
</tr>
<tr>
<td>Traffic type</td>
<td>CBR</td>
</tr>
<tr>
<td>Traffic rate</td>
<td>10 packets/s</td>
</tr>
</tbody>
</table>

V. RESULTS AND DISCUSSION

The Figure 5 shows the Packet Delivery ratio of DYMO for different mobility speed of node in MANET. For different speed the total packet received at the destination is plotted in a graph.
The Figure 6 shows the control overhead involved in using DYMO routing protocol under high mobile environment. For different mobility speed the number of control packets transmitted is calculated.

The Figure 7 shows the end to end delay involved in using DYMO routing protocol under high mobile environment. For different mobility speed the number of the end to end delay involved in using DYMO is calculated.

VI. CONCLUSION

Thus this system which uses DYMO protocol performs much better than other on demand routing protocol by offering higher packet delivery ratio and low control overhead. The Authentication is provided by using hash function on Double Hash Authentication Technique (DHT) and the route can be optimized using Self Healing Optimized Routing Technique (SHORT). This study analyses the performance of DYMO routing protocol with the application of Double Hash Technique in association with Self Healing Optimized Routing Technique (SHORT). This system offers more security even in the presence of malicious nodes and performs better under high mobile environment using DYMO routing protocol by applying SHORT algorithm.

REFERENCES


Master Thesis, Department Of Computer Science University Of Aarhus Denmark.


Intrusion Detection System for Malicious Node Detection in DYMO Protocol

R.Shanthakumari M.E
Assistant Professor
Department of Information Technology
Kongu Engineering College
Perundurai, Erode, Tamilnadu.

Abstract – Mobile ad hoc networks is a collection of wireless nodes or mobile links that can dynamically form a network to exchange information without using any centralized fixed network infrastructure. In such a network the nodes are freely move and coordinate themselves in a random manner. Due to the frequently changing topology the MANET are vulnerable to various attacks by the malicious node. The presence of malicious node, it affects the network performance and reliability. A novel approach for enhanced intrusion detection system for malicious node detection in dynamic MANET on demand routing protocol is used. The proposed approach employs a method for gives set of condition under which malicious node should be monitored. This approach effectively detects the malicious node with minimal energy consumption and also this approach leads to less consumption of control packets and less link breakage in mobile ad hoc routing.

Keywords: MANET, Security, Energy Consumption, DYMO.

I. INTRODUCTION

Mobile ad hoc network is the wireless network formed by the wireless nodes without any help of centralized infrastructure. In such a network the nodes are freely move and organize themselves in an arbitrary manner and each nodes act as a router. They play a vital role in the route discovery and maintenance from the source to destination or from one node to another node. The new nodes are join or leave in the network at any time in MANET. Owing to the dynamic nature, connections and positions are frequently changing. The main technique used is multi-hopping, which increases the overall network capacity and performance. Due to the lack of infrastructure and dedicated nodes that provide network management operation like traditional routes in the fixed network poses many new challenges.

The first challenge is how to maintain the connectivity in the network. The routing problem is solved by some of the routing protocols like AODV (Ad hoc On Demand Distance Vector), DSR (Dynamic Source Routing). The second challenge is security. Due to the characteristic such as frequently changing topology, heterogeneous and decentralized control, ad hoc networks are vulnerable to various attacks including spoofing, modification of packets, distributed denial of service, etc. To secure adhoc networks, the prevention based approach is not enough, it is necessary to develop intrusion detection mechanism.
Normally the intrusion detection involves run time gathering of data from system operation, and the subsequent analysis of data.

The data can be audit logs generated by an operating system or packets sniffed from a network. The mobile devices are dependent on battery power, it is important to minimize the energy consumption [4].

In this paper we proposed an enhanced intrusion detection system to detect the malicious node in mobile ad hoc network using DYMO routing protocol. Dynamic MANET On Demand routing is the reactive routing protocol and it is extension of AODV routing protocol.

II. RELATED WORK

A. Enhanced IDS Model

In ad hoc networks, routing is serious weakness if some of the nodes operating maliciously. The enhanced intrusion detection system is used to detect the malicious nodes in the mobile ad hoc network. This scheme contains the record of all nodes present in the network. This model employs a method; it gives the set of condition under which the malicious node should be monitored. Energy is consumed by adjusting the transmission range of each node and also identifies the malicious node based on trust level and complete data rate [5].

B. Routing Protocols

In an ad hoc network routing protocols are classified into two types proactive and reactive. The proactive routing protocols maintain the up-to-date routing information from each node to every other node in the network. It all the times to discover and maintain the routes but in the reactive routing protocol the routes are established on demand only.

Enhanced intrusion detection system concept is already implemented in the AODV routing protocol as EIDS-AODV. Dynamic MANET On demand routing protocol is the extension of AODV routing protocol and it combines the features of both AODV and DYMO routing protocol. So the EIDS model is combined with the DYMO routing protocol to detect the DYMO accumulation feature so that it effectively utilizes the energy.

III. EIDS MODEL

EIDS model consists of four activities:
(i) scan all nodes present in the network.
(ii) Maintenance of routing table
(iii) Maintenance of sequence number
(iv) malicious node detection.

Initially, scan all the nodes present in the network by comparing transmission range of a node with transmission range of a network. If a node is moving outside during data transmission then minimize its energy accordingly.

Maintenance of routing table is performed by comparing the neighbor ip and the originator ip, if it is equal then initializes the routing table by new entity sequence number with originator sequence number. Otherwise initialize the routing table with new entity sequence number with zero.

Maintenance of Sequence number is performed by comparing the routing table sequence number for destination ip with RREQ sequence number. If it is equal then maintain the routing table. Otherwise assign routing table sequence number to RREQ.
sequence number.

Identification of malicious node is initially, a node receives a packet. Then, we will check for duplicate packets and also status of sequence number in the routing table during rebroadcast RREQ. If it is equal then node is non-malicious otherwise we will check its confidence values as well as packet transfer rate by this node. Based on this process, behavior of node can be easily identified. By this process, we can identify number of malicious nodes.

A. Calculating Complete Data Rate

Capacity of the node depends on percentage of the packets forwarded by that node. Node capacity is calculated by using the formula

\[ \text{Complete data rate} = \frac{\text{ACK}_i}{\text{RTS}_i} \]

Where \( \text{ACK}_i \) is the number of acknowledgement received by the node \( i \) in \( T \) seconds. Assume three confidence axioms which are as follows:

1. \( \text{ACK} \leq \text{RTS} \)
2. Complete data rate is calculated for every \( T \) seconds.
3. Reset \( \text{ACK} \) and \( \text{RTS} \) values for every \( T \) seconds

\[ \text{CDR}_i = \alpha \times \text{CDR}_i \text{ old} + (1 - \alpha) \times \text{CDR}_i \text{ sample} \]

\( \text{CDR}_i \text{ old} \) maintains the previous value of CDR and \( \alpha = 0.3 \) is constant. \( \alpha \) is weight in exponential weighted moving average method. This represents the weight of old value. \( \text{CDR}_i \text{ sample} \) represents new calculated value. At routing level, suppose there is a route \( r_i = n_s, n_1, n_2 \ldots n_d \), where \( n_s \) is the source and \( n_d \) is the destination.

\[ \text{CDR}_i = \pi \text{CDR}_j \]

\( n_j \in r_i, n_j \neq n_s, n_d \)

B. Calculating Confidence Value

When the information is equally distributed in the entire network, then confidence level of information can be computed by using the following formula

\[ C(i, j) = c(i, j) \times \text{DCT}(i, j) + [1 - c(i, j)] \times \text{ICT}(i, j) \]

\( C(i, j) \) is the confidence level which is totally based on direct confidence trust \( \text{DCT}(i, j) \) and indirect confident trust \( \text{ICT}(i, j) \). The computation of \( \text{DCT}(i, j) \) and \( \text{ICT}(i, j) \) can be easily calculated by using the formula,

\[ C(i, j) = \text{number of interactions} / (\text{number of interactions}+1) \]

Where number of interactions evaluates number of direct Interactions between nodes \( i \) and node \( j \), and this information is also kept in trust table. Node \( i \) is confident of all calculated trust value of node \( j \).

C. Energy Consumption

It is assumed that each node can transmit or receive data within a maximum radius \( R \), but both operations cannot occur simultaneously. Also, energy power is initialized to each node. Based on these assumptions; an enhanced intrusion detection system for transmission can be designed as follows:

1. Source transmits data to destination within the maximum radius \( R \).
2. When destination \( D \) is moving out of the transmission range then calculates the closest position from radius \( R \) and evaluates its transmission power (Tp) at this position. When a node position exceeds from \( R \) then reduce its transmission power until it is not equal to \( Tp \).
3. If above step is satisfied then data transmission will be continued till it accomplishes. Otherwise transmission will not be initiated.

Edge depends on two main factors:

1. Node position:

The node position is very difficult to calculate control topology by varying transmission radius \( R \).

2. Communication range:

Communication range is calculated by using the following formula

\[
E = \{(u, v) \in V^2 | u = v \land d(u, v) \leq R\}
\]

Where \( R \) is the maximum range of communication. Neighbour set \( N(u) \) of vertex \( u \) is given by,

\[
N(u) = \{v | (u, v) \in E\}
\]

D. Identification of Malicious Node

Initially all nodes are performed well in nature. A node is suspicious by considering such cases. (i) If the node is moving out of transmission range after route discovery process (ii) If the communication breakage occurs during data packet transmission. Case (i) can be easily rectified and overcome as mentioned in the section 3.3. So, the further work is deals with the secondary case. The proposed algorithm will verify whether the sequence number of rebroadcasted RREQ is equal to the sequence number of same RREQ that is already stored in the routing table of current node. If the sequence number is different, then it is considered as an abnormal activity. Then calculate its confidence value and node capacity. If results are not satisfactory in terms of confidence value and node capacity then suspicious node will be marked as malicious node.

IV. ROUTING PROTOCOL

The AODV and DYMO routing protocols are come under the category of on demand routing protocol.

A. AODV Routing Protocol

In AODV when a source wants to communicate with the destination node means it is prepares the RREQ (Route request) packet and floods the RREQ packet in the network. The neighbours in turn broadcast packets to their neighbours tills it reaches the destination node. A node discards its packet if it has been received already. Once the RREQ packet reaches the destination node, then the destination node responds with the RREP (route reply) packet. After the source node receives the RREP packet then the route is established between source and destination [3].

In the route maintenance the source or intermediate node detects any link failure, and then it sends link failure to its upstream nodes. So source can reinitiate route discovery if necessary.

B. DYMO Routing Protocol

It is the enhanced version of AODV routing protocol. There are two basic operations are involved in the DYMO routing protocol, they are (i) Route Discovery (ii) Route Maintenance. In the route discovery process source node prepares the RREQ packet and it floods the RREQ packet in the network. The intermediate nodes are stores the route to the source node and transmit the packet to its neighbours until it reaches the destination.
node. Once the RREQ packet reaches the destination node it responds with the RREP packet [1].

Route maintenance is the process of responding the changes in topology after route discovery process. To maintain the routes, it continuously monitors the active links and updates the valid timeout period entry in its routing table each and every time of sending and receiving the data packets. If the node receives the data packet for the destination node and it does not have a valid route to the destination means it prepares the REER (Route Error) packet.

C. Difference between AODV and DYMO

The main difference between the AODV and DYMO routing protocol, DYMO protocol uses the path accumulation feature. In the AODV routing protocol the RREQ packet contains the information about next hop node and destination node. Whereas in the DYMO routing protocol, the intermediate nodes append their information in the RREQ packet. So that, the nodes are well aware of another nodes present in the network. HELLO messages are discarded in the DYMO routing protocol [2].

V. EXPERIMENTAL RESULT

QUALNET simulator is used to evaluate the performance of AODV and DYMO routing protocol. Common simulation parameters are used in the both AODV and DYMO routing protocol there are,

<table>
<thead>
<tr>
<th>SIMULATION PARAMETERS</th>
<th>AODV</th>
<th>DYMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Simulation time</td>
<td>300 s</td>
<td>300 s</td>
</tr>
<tr>
<td>Traffic Model</td>
<td>CBR</td>
<td>CBR</td>
</tr>
<tr>
<td>Packet Size</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Area</td>
<td>1000 * 1000</td>
<td>1000 * 1000</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>250 m</td>
<td>250 m</td>
</tr>
<tr>
<td>Transmission</td>
<td>0.28183815</td>
<td>0.28183815</td>
</tr>
<tr>
<td>Tx power</td>
<td>0.173</td>
<td>0.173</td>
</tr>
<tr>
<td>Rx power</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>Pause Time</td>
<td>200 s</td>
<td>200 s</td>
</tr>
<tr>
<td>Malicious nodes</td>
<td>35</td>
<td>35</td>
</tr>
</tbody>
</table>

A. Performance Metrics

Packet Delivery Ratio

It is the ratio of total number of packets transmitted in the sender to the total number of packets received by the receiver.

Average End to End Delay

Average time taken between the data packet send by the sender and the data packet received by the receiver.

Relative Overhead

It is the ratio of total number of control packets to the number of data packet.

In the existing enhanced intrusion detection system there two types of test are conducted. First by changing the node speeds and, second one is to increase the malicious nodes and the results are shown EIDS-DYMO.
shows better performance over other routing protocols.

B. RESULTS BY VARYING NODE SPEEDS

TABLE 2 MOBILITY SPEED: 5 MPS

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Packets received</th>
<th>Throughput (bits/s)</th>
<th>End-to-end delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>282</td>
<td>3612</td>
<td>.0163</td>
</tr>
<tr>
<td>DYMO</td>
<td>286</td>
<td>3903</td>
<td>.0271</td>
</tr>
</tbody>
</table>

TABLE 3 MOBILITY SPEED: 10 MPS

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Packets received</th>
<th>Throughput (bits/s)</th>
<th>End-to-end delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>261</td>
<td>3612</td>
<td>.01675</td>
</tr>
<tr>
<td>DYMO</td>
<td>276</td>
<td>3773</td>
<td>.0399</td>
</tr>
</tbody>
</table>

TABLE 3 MOBILITY SPEED: 20 MPS

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Packets received</th>
<th>Throughput (bits/s)</th>
<th>End-to-end delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>249</td>
<td>3422</td>
<td>.02732</td>
</tr>
<tr>
<td>DYMO</td>
<td>260</td>
<td>3395</td>
<td>.06453</td>
</tr>
</tbody>
</table>

The tabulated values are shown in form of graph

Fig. 1. Packets Received in DYMO and AODV

Fig. 2. Throughput Vs Node mobility

VI. CONCLUSION

The above results shows that by varying mobile node speed, the DYMO routing protocol produce the good result compared to AODV because of its path accumulation feature. In the DYMO contains the existing topology information so that the data packets send in the network based on the path accumulation feature. Hence the DYMO routing protocol is well suited in the high mobility environment. The EIDS-DYMO protocol is implemented in the future it will shows the better performance.
REFERENCES


REDUCING POWER AND DELAY IN WIRELESS SENSOR NETWORKS USING JUMPING ANT ROUTING ALGORITHM AND SPRAY AND FOCUS ALGORITHM

A.SINDHUJA  K.T.MATHUMITHAA

Department of Computer Science Engineering
Dr. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY

Abstract — Wireless sensor Networks (WSNs), is one of the most rapidly growing scientific domain. This is because of the development of advanced sensor nodes with extremely low cost. One of the characteristic feature of WSNs compared to the traditional wireless communication networks, is the power awareness, due to the fact that the batteries of the sensor nodes have restricted lifetime and are difficult to be replaced. This is why we focus on “power awareness”. Due to its working environment and the mobility of sensor node, this kind of sensor network is very much essential to reduce power utility. We propose a jumping ant routing algorithm (JARA) which combines the advantages of reactive and proactive routing to speed up the route discovery time and reduce the route discovery overhead in sensor network, thereby reducing power. JARA, a population based algorithm provides, natural and intrinsic way of exploration of search space in optimization settings in determining optimal data aggregation. The simulation results shows improvement in energy efficiency depends on number of source nodes in sensor network which is 45% energy efficiency using optimal aggregation compared to approximate aggregation schemes in moderate number of sources. To reduce “delay” in such networks we introduce replication or “spraying” methods that can reduce the overhead of flooding-based schemes by distributing a small number of copies to only a few relays, whereas 20% energy efficiency in large number of source nodes. To route messages efficiently in such networks, we propose a scheme that also distributes a small number of copies to few relays. However, each relay can then forward its copy, instead of naively waiting to deliver it to the destination itself. This scheme exploits all the advantages of controlled replication, and could deliver the message faster thereby reducing delay. Simulation results for traditional mobility models, as well as for a more realistic “community-based” model, indicate that our scheme can reduce the delay 20 times compared to existing techniques.

Keywords—Wireless sensor networks, JARA (Jumping Ant Routing Algorithm), Spray and Focus Routing, cluster head, Cluster nodes.

I. INTRODUCTION

A Wireless Sensor Network consists of a group of spatially distributed sensor nodes which are interconnected without using wires. Each of the distributed sensor nodes typically consists of one or more sensing elements, a data processing unit, communicating components and a power source,
which is usually a battery. The sensed data is collected, processed and then routed to the desired end user through a designated sink point, referred as base station. Now it has become feasible to construct multifunctional sensor nodes with advanced capabilities. Such sensor nodes are relatively of smaller size, lower cost and lesser power consumption. WSNs are originally motivated for the use in military applications, such as border monitoring. Now a days it is mainly focused on civilian applications such as environment monitoring, object tracking and bio-medical applications. A wireless sensor network operates with limited computational and sensing capabilities capable of sensing, computing and wirelessly communicating. Since the sensor nodes have irreplaceable, batteries with limited power capacity, it is essential that the network be energy efficient in order to maximize the life span of the network. Large number of sensor nodes have to be networked together, direct transmissions from any specified node to a distant base station is not used, as sensor nodes that are farther away from the base station will have their power sources drained much faster than those nodes that are closer to the base station. On the other hand, minimum energy multi-hop routing scheme will result in rapidly drain energy resources of the nodes, since these nodes engage in the forwarding of a large number of data messages (on behalf of other nodes) to the base station.

Related Work

WSNs consist of hundreds of even thousands of sensor nodes which may be sparsely distributed in remote locations. A typical sensor node consumes most of its energy during communication. However, energy expenditure takes place while performing sensing and data processing too. Hence each and every protocol should be so designed, that minimum energy should be consumed during sensing, processing and communication. The application of a aggregation approach helps reduce the amount of information that needs to be transmitted by performing data fusion at the aggregate points before forwarding the data to the end user.

Ant Routing Algorithm:

Ant Routing Algorithms are inspired by the behavior of real ant colonies. Many studies have discussed the use of this algorithm to solve various problems. Since it is reliable, survivable and dynamic, the optimum solution for this algorithm is determined by creating artificial ants. The artificial ants search the solution space as real ants search their environment for food. The probabilistic movement of ants in the system allows the ants to study new paths and to re-explore old visited paths. The strength of the pheromone deposit directs the artificial ants towards the best paths, while the pheromone evaporation lets the system forget old information and avoid quick convergence to sub-optimal solutions. The probabilistic selection of the paths enables searching large numbers of solutions. Ant routing is a self-configured, self-built protocol, which can reduce the number of broadcast messages that need to be sent and which maintains several multi paths. When a node wishes to find and maintain a path to its destination, it sends forward ants searching for this destination. A forward ant moves in the network searching for the destination using the intermediate nodes’ probability routing tables and the local heuristic information. Forward ants collect information about paths and intermediate nodes local information as they travel along the path. When a forward ant reaches its destination, the information carried by this forward ant is graded. The forward ant is then killed, and a backward ant is generated.
The backward ant carries its corresponding forward ant’s grade and the identities of the intermediate nodes in the path. The backward ant is sent back along the reverse path of its corresponding forward ant. As backward ants move in the reverse path, the intermediate nodes modify their pheromone table based on the path grade carried by the backward ant, and accordingly update their pheromone table probability. Finally, the source node receives the backward ants, updates its tables and kills the backward ant.

**Zone routing protocol:**
Proactive and reactive protocols both have particular flaws. The Zone Routing Protocol (ZRP) combines the advantages of both into a hybrid scheme, utilizing a proactive mechanism to discover a node’s local neighborhood, and applying a reactive protocol to communicate between neighborhoods. As mentioned earlier, the ZRP provides a framework for their routing protocols. The separation of nodes local neighborhood from the global topology of the entire network allows the application of different methods, thus exploiting each technique’s features in given situation. These local neighborhoods are called zones. Each node may be within multiple overlapping zones, and each zone may be of a different size. The “size” of a zone is not determined by geographical measurement, as might be expected, but instead is given by a radius of length $q$, where $q$ denotes the number of hops to the perimeter of the zone.

II.JUMPING ANT ROUTING ALGORITHM
This investigation presents a jumping ant routing algorithm (JARA) that combines the advantages of both Ant Routing Algorithm and Zone Routing Protocol, while also employing jumping mode to reduce the pro-active overhead. The algorithm is discussed in two parts. The first part relates to how each node uses proactive routing protocol to maintain the topology of $q$ hops. The intra-network is assumed to have been already established by one of proactive routing protocols, and thus to have already generated an intra-network table. The other part concerns how each node applies ant routing to discover paths outside its zone. Each node has its own zone, and each ant obtains a route within $q$ hops. Hence, each ant jumping $q$ hops distance is considered as one movement. This work explains and simulates the proposed algorithm, using $q = 2$. The setting $q = 2$ was chosen because it is sufficient to demonstrate the predominance of the proposed algorithm. Each node in our algorithm can discover detailed information of neighboring nodes within $q$ hops. These neighboring nodes can be organized into a zone called the intra-network. Those nodes within a zone are classified into boundary and interior nodes.

![Fig.1: Zone of node A with $q = 2$.](image)

The minimum distance of a boundary node minimum distance to the central node is exactly equal to the zone radius $q$. Nodes with minimum distances of less than $q$ are called interior nodes. Fig. displays an example of a zone where $q = 2$. The central node in the figure is node A. Nodes B, D, E, H, F and J denote the boundary nodes; nodes C, G and I are interior nodes, and node K and L are outside the routing zone. Each ant in node A adopts the pheromone table to choose a boundary...
node as the next node. If the ant specifies node B, then it must move to node B via node C. Node C only relays packets from the central node A to a boundary node. Hence, the JARA can speed up the route discovery and find a better path. The subsection explains route discovery, then discusses the effect of changes in network topology.

A. Route discovery:

Ants are classified as forward, backward and guide ants. Forward and backward ants are responsible for collecting path information and updating pheromone. A guide ant constructs an optimal path when all the backward ants have arrived at source a node, or when the network topology has changed. Every node also has a pheromone table.

The movement of route discovery is as follows:

1. The source node creates several forward ants to search for destination. The ants gather path information as they travel along the path.
2. A node creates a backward ant when a forward ant arrives there.
3. The backward ants are sent back following the reverse path, and update the pheromone table.
4. The guide ant is generated when all backward ants have arrived at the source. It updates the routing table along the optimized path, and constructs an optimal path.

B. Forward ant:

Every node in the network can be considered as a source, destination or intermediate node. A node that wants to find a path to a destination sends forward ants to search for this destination and obtain path information. When a forward ant is generated by source node, it adopts the pheromone table to obtain the next visiting node and record the path information. According to the routing principle, the next visiting node of an ant depends only on the probability in the pheromone table. In the proposed algorithm, ants prefer to move to a node that has not been visited. Such behavior is introduced to prevent ants from being enticed into the same route, thus losing the advantage of exploration. The values of the probability are calculated according to local heuristic information. A forward ant moving to an intermediate node utilizes the probability of pheromone table, adds the next destination node to intermediates node stack, and obtains local heuristic information to update path information of the forward ant packet. If the forward ant moves to the interior node, then the interior node need not do anything, but only relays the forward ant to the next destination node. The forward ant is killed when it arrives at the destination node, and a backward ant is then created. The destination node also employs path information to obtain a grade to assign to the backward ant.

C. Backward ant:

When the backward ant is received, if this node is intermediate node of the backward ant’s stack, then the node collects the grade from the backward ant’s stack and then updates the pheromone table using the grade of the backward ant, and sends the ant to the next intermediate node. If the node is an internal node, then the node looks up the routing table and transmits the ant to the next destination. The backward ant is killed when it arrives at the source node. The pheromone updating function increases the pheromone value on the incoming link, and decreases the values on other links, using the following function:

For destination D, at intermediate node i,
Where \( f(q) \) is the evaporation function, and \( g(q) \) denotes the enforcement function. The aim of the evaporation function is to help the system forget the old information quickly. A higher value of \( q \) implies a faster evaporation. Generally, the evaporation function should be given a small value. The enforcement function helps the system increase the amount of pheromone on the edges. When the grade \( q \) increases, the enforcement function should be increased. This section considers why the guide ant is adopted, and begins by considering how this algorithm indicates where to deliver packets without the guide ant. Either of the following two methods described below. The first method is to determine the routing when the packet has already flowed on networks, and the other method is to utilize the backward ant. Using the first approach, the route discovery is processed earlier. Determining the routing while packets are transmitted on networks is time-consuming, and may degrade cause the performance of transmission. Routing depends on each backward ant when using the second method. Such a method increases the frequency of the routing update, may even cause records to be sent along an unnecessary path. The guide ant can carry data, so does not create additional packets. The operations are described in detail later. The guide ant also repairs broken routes to a destination. The source node generates a guide ant to construct an optimal path. The guide ant updates the routing table, and can determine whether the resident node is a boundary or interior node. If the node is a boundary node, then the guide ant updates the routing table by referring to the pheromone table and then choosing the maximum probability node in the pheromone table. Otherwise, if the node is an interior node, then the guide ant updates the routing table and the node acts simply as a forwarder.

III. PROPOSED SOLUTION

Network topology changed:

A guide ant is generated when the network topology changes. The guide ant can carry data and choose a new optimal path. This section discusses two cases, where the changed node is an interior node, when it is a boundary node.

Changed node is interior node:

When an interior node is (removed or deleted), a new path is built using the intra-network table, and is adopted by the guide ant to update the routing table. Therefore, route discovery does not need to be performed again. The guide ant is killed when it arrives at a node in the original path, restoring the general packet transmission. Fig. 2., illustrates an example of this case. Node A has a path to node C via node B, and node B is the interior node of node A. When node B is broken, node A finds another node to node C using ‘‘looking for intra-network table’’. In Fig. 3, node A discovers a new path to node C via node D, and therefore generates a guide ant to guide this path. The guide ant updates the routing table in nodes.

![Fig. 2: Interior node move.](image-url)
In a heterogeneous environment, things are not that simple. Some nodes may be “better” relays for a given destination. Such, for example, could be nodes that tend to see the destination more often (e.g. work in the same building, or in general belong to the same social network [9, 13]).

D. Data Aggregation:

The application model considered for this work consists of a single destination (base station) and multiple sources. Since the nodes are wirelessly connected which communicates to neighbors in vicinity, therefore multi-hop communication is used to reach the destination. It is assumed that density of nodes gives a connected node graph. For a application setting, data aggregation is applied in the network. “Data aggregation merges message data in-network while traversing through network” it is also termed as data fusion. The aggregation gain can be measured as (original – aggregated) original in the given application message size. The aggregation suffers from delay termed as aggregation delay. There is a tradeoff in delay and gain in aggregation. The simulation study reveals that energy-efficiency is related to number of source nodes in correlated sensing.

IV. SPRAY AND FOCUS

Existing spraying schemes [16, 17], generate and distribute (“spray”) a small, fixed number of copies or “forwarding tokens” to a number of distinct relays. Then, each relay carries its copy until it encounters the destination or until the TTL (time-to-live) for the packet expires. By having multiple relays looking independently and in parallel for the destination, these protocols create enough diversity to explore the sparse connectivity graph more efficiently, and can discover a short path-over-time to the destination. Although such schemes have been shown to perform well in some scenarios [16, 17], they require a high amount of mobility by network nodes to achieve this performance. However, in many practical situations, the mobility of each node is limited to a
small local area for the majority of time. An example where such local mobility might arise could be, for example, that of a university campus, where most people tend to stay or move locally within their buildings for long stretches of time [11]. To make our point more clear, consider for example the “Spray and Wait” scheme [16, 17] in such a scenario. This scheme consists of two phases: in the first phase it distributes a fixed number of copies to the first few relays encountered, and in the second phase each of these relays waits until it encounters the destination itself (i.e. “Direct Transmission” [19]). It is easy to see that, here, this scheme would spread all its copies quickly to the node’s immediate neighborhood, but then few if any of the nodes carrying a copy might ever see the destination [3]. What is more, if the network is not too sparse, there might exist partial paths over which a message copy could be transmitted fast to a node closer to the destination. Yet, in schemes like Spray and Wait a relay with a copy will naively wait until it moves within range of the destination itself. This problem could be solved if a sophisticated single copy scheme is used to further route a copy after it’s handed over to a relay, a scheme that takes advantage of transmissions (unlike Direct Transmission). With this in mind, we propose Spray and Focus, which in the second phase (“focus” phase) rather than waiting for the destination to be encountered, each relay can forward its copy to a potentially more appropriate relay, using a carefully designed utility based scheme. In the next few sections, we describe our protocol in detail.

E. Spraying Phase:

When a new message gets generated at a source, and needs to be routed to a given destination, Spray and Focus first enters the “Spray phase” for this message. When a new message is generated at a source node it also creates L “forwarding tokens” for this message. A forwarding token implies that the node that owns it, can spawn and forward an additional copy of the given message, according to the following rules:

- each node maintains a “summary vector” with IDs of all messages that it has stored, and for which it act as a relay; whenever two nodes encounter each other, they exchange their vectors and check which messages, they have in common (as in epidemic routing).
- if a node (either the source or a relay) carrying a message copy and n > 1 forwarding tokens encounters a node with no copy of the message, it spawns and forwards a copy of that message to the 2nd node; it also hands over \( \frac{n}{2} \) forwarding tokens and keeps \( \frac{n}{2} \) for itself; (Binary Spraying [17])
- when the node has a message copy but only one Forwarding token for this message, then it can only forward this message further according to the rules of the “Focus phase”.

F. Spraying mechanism:

Another interesting question is how the number of copies (or forwarding tokens) should be distributed to different relays. In a homogeneous environment (i.e. IID node movement) it is beneficial to spread messages as quickly as possible, as all nodes are statistically equivalent [1]. What matters mainly there is how many relays are looking in parallel. It is proven in [17] that, if node movement is IID and nodes forward copies only to new nodes, the algorithm that minimizes spraying time is Binary Spraying [17]. In a heterogeneous environment, things are not that simple. Some nodes may be “better” relays for a given destination. Such, for example, could be nodes that tend to see the destination more often (e.g. work in the same building, or in general...
belong to the same social network [9, 13]). Ideally, we would like to be able to choose as relays the L nodes that most frequently encounter the destination. In an offline version of the problem, we could potentially formulate a stochastic version of the linear program described in [6] and solve for the minimum expected delivery time. However, in the online version of the problem, waiting for a “better” relay incurs a cost, because it means that opportunities to spread extra copies are forfeited. In other words, there are two conflicting strategies that can potentially reduce delay when a new node is encountered,

(1) Spawn and forward an extra copy right-away to increase parallelism, and
(2) Defer forwarding until a relay that is much more Correlated” with the destination is encountered

G.Node encounters:
First, when we refer to a “node encounter”, we assume that nodes periodically transmit beacons to recognize each other’s presence. We expect that the period of this beacon would have some effect on the performance of our protocol (if beacons are not sent often enough, some forwarding opportunities might be missed [3]). However, we assume that this is an issue that is handled by the underlying media access (MAC) protocol [5], and that, ideally, nodes “encounter” each other as soon as they come within communication range. Furthermore, there is an overhead involved in the exchange of the message summary. However, each message’s ID is expected to occupy only to have faster transfer, thereby reducing the delay. These node encounters are essentially needed for reducing the delay.

H.Number of copies used:
Another interesting question is how many copies to use per message, i.e. what the value of L should be. In general, we would like this number to be only a small percentage of the total number of nodes. For the case of simple controlled replication, or Spray and Wait, analytical expressions exist that can calculate the number of copies needed to achieve an average delay that is α times the optimal one [17].

I. Focus Phase:
When a relay for a given message has only one forwarding token left for that message, it switches to the “Focus phase”. Unlike Spray and Wait, where in the Wait phase messages are routed using Direct Transmission (i.e. forwarded only to their destination) [16,17], in the Focus phase a message can be forwarded to a different relay according to a given forwarding criterion. Specifically, these forwarding decisions are taken based on a set of timers that record the time since two nodes last saw each other. Although the use of last encounter timers has been proposed in the past (e.g. [4]), we argue that any scheme that takes advantage of these timers needs to be carefully designed for the specific environment in hand (e.g. sparse networks with stochastic mobility) in order to achieve good performance. Let us turn then our attention to these timers. (Due to lack of space we omit some of the details of our utility-based mechanism.

J.Age of last encounter timers with transitivity:
A number of different utility functions could be envisioned for this purpose. These could also take into account other relevant information (e.g. GPS position, speed, history of encounters, etc.) in addition to the timer values. However, it is beyond the scope of this paper to evaluate all these
options, and we defer this for future work. Some efforts towards the design of multi-parameter utility functions can be found in [12]. Here, for simplicity, we will assume that these timers is our utility function (i.e. messages get forwarded to nodes with smaller and smaller timer values for the destination). A gradient-based scheme can then be used to maximize the utility function for the destination. We summarize here the functionality of the Spray and Focus protocol. Each node maintains a vector with IDs of all messages that it has stored, and for which it acts as a relay; whenever two nodes encounter each other, they exchange their vectors and check which messages they have in common; each message also carries a TTL (time-to-live). When a new message is generated at a source node create $L$ “forwarding tokens”, with $L$ chosen, if a node (either the source or a relay) carries a message copy and (i) $n > 1$ forwarding tokens - perform Binary Spraying. (ii) $n = 1$ forwarding token - perform Utility-based Forwarding according to the last encounter timers used as the utility function. Hence by using this technique the delay can be reduced to a maximum. Although node mobility is still IID (uncorrelated), each node now takes a very long time to move from one side of the network to another, and thus carrying a message is not as beneficial as in the Random Waypoint case. Figure 6 depicts the number of transmissions and the average delay of all schemes for this scenario.

Here, the few copies are spread locally, and then each message relay takes a very long time to traverse the network and reach the destination. Even if the number of copies were increased, the delay of the spraying phase would still dominate performance, since new nodes are found very slowly. On the other hand, Spray and Focus can overcome these shortcomings and excel (unless the network is too sparse), achieving the smallest delay with only a few extra transmissions. Note also that, despite using the same utility function as Spray and Focus, Utility Flooding is still plagued by its flooding nature. This problem was even more pronounced when other existing utility functions were used [10]. This implies that in disconnected networks, the use of a utility function is not enough by itself to improve performance, but rather has to be combined with controlled replication.

V. IMPLEMENTATION & RESULTS:

The Jumping Ant Routing Algorithm is simulated in MATLAB with a setting of sensor network of 50 nodes. The neighborhood is obtained from the random topology. Set of source nodes 20, 30&40 and a destination is considered for generating optimal routing tree using this algorithm. On varying the routing parameters and weights of nodes, shortest distance and correlation, the optimal routing tree in sensor network is obtained. The fig.7 below shows the convergence of optimal solution globally in the setup after 1000 iterations.
VI. CONCLUSION

A Jumping ant routing based solution for the Optimal routing Problem has been implemented and investigated. Extensive simulation is carried out for correctness of algorithm. It is observed that routing energy efficiency depends on the number of sources. The results of simulation reveal that this optimal routing algorithm save energy up to 45% for moderate number of source nodes. The proposed Algorithm shortens the route discovery time and reduces the route discovery overhead, especially in dense topologies. The simulation results shows that, using spray and focus, the delay can be reduced to 20 times compared to the existing techniques. Hence this “dual” mechanism improves energy efficiency and minimizes delay.

VII. REFERENCES


AN EFFICIENT JOB SCHEDULING ALGORITHM FOR GRID ENVIRONMENT

Kaviyarasan.T
M.E-Computer & Communication Engineering
Department of Information Technology
Kongu Engineering College, Perundurai, Erode,
Tamil Nadu, India
arasu247@gmail.com

Suresh.P
Assistant Professor (Sr.Grade)
Department of Information Technology
Kongu Engineering College, Perundurai, Erode
Tamil Nadu, India
sureshme@gmail.com

Abstract - Grid computing deals with the effective utilization of the globally distributed computer resources to solve large problems. Effective scheduling algorithms are used for the effective utilization of the unused CPU cycles that are distributed geographically. The existing algorithms mainly focus on system performance rather than the user satisfaction. In this paper we have presented a new adaptive job scheduling algorithm that focus on satisfying user deadline and minimizing the makespan. This algorithm also considers the application demand and the communication demand of the particular job. The proposed algorithm improves user satisfaction and effective resource utilization and also reduces the makespan.

Keywords: Grid scheduling, User satisfaction, Resource utilization, Communication demand, Makespan.

I. INTRODUCTION:

Grid computing is an emerging computing model which is the interconnection of computer systems where the machines utilize the same resources collectively. In grid computing, the information and tasks are distributed to a group of networked computers to accomplish a common goal. Grid computing provides a framework and deployment environment that enables resource sharing, accessing, aggregation and management [8]. It allows resource and coordinated use of various resources in dynamic, distributed virtual organization. A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. It is a shared environment implemented through the deployment of a persistent, standards-based service infrastructure that supports the resource sharing within distributed communities. Resources can be computers, storage space, instruments, software applications that are connected through the Internet and a middleware software layer that provides basic services for security, monitoring, resource management and so forth. A grid is a system of high diversity, which is rendered by various applications, middleware components, and resources. To achieve the promising potentials of tremendous distributed resources, effective and efficient scheduling is fundamentally important. Scheduling refers to the way the processes are assigned to run on the available CPUs. This assignment is carried out by the software known as scheduler. Grid scheduling is a sophisticated decision making system that operates at different levels of grids. The scheduling process is generalized in the three stages such as resource discovering and filtering, resource selecting and scheduling according to certain objectives of job submission. Job scheduling on computational grids is gaining importance due to the large scale
application of the grid. Effective job scheduling in grid requires modelling the available resources on grid nodes and computation requests of jobs determine the current load of the system and predict the job execution time.

Basically, a grid scheduler receives applications from grid users, selects feasible resources for these applications according to acquired information from the Grid Information Service (GIS) module, and finally generates application-to-resource mappings based on objective functions and predicted resource performance. Information about the status of available resources is collected by GIS for grid schedulers [7]. Several open source or proprietary schedulers have been developed for clusters of servers, including portable batch system and load sharing facility. This type of schedulers typically run in batch mode and can be customized to specific policies in an attempt to balance the load among the various servers. However, the primary objective of most existing approaches is to improve overall system performance. In grid computing, many unique characteristics make the design of scheduling algorithms more challenging. The main factor to be considered is heterogeneity and autonomy. The utilization of resource will not be constrained within the same system itself.

II. RELATED WORKS:

There are various algorithms that are already proposed for grid scheduling. These algorithms mainly focus resource utilization and makespan. Makespan is the maximum time taken for completing all the submitted jobs.

A. Min-min

This heuristic algorithm considers the set of unmapped jobs and calculates the expected completion time for all the jobs in the set in all the machines. For each job, the machine with minimum expected completion time is identified. Finally the job with minimum expected completion time among all the other jobs in the set is allocated to the machine which has the MCT for that job. This process is repeated for the remaining unmapped tasks. In this algorithm, the makespan is comparatively improved but the idleness of the machine remains unsolved [11,12].

B. Max-min

This proceeds as the Min-min algorithm in calculating the expected completion time for all the unmapped jobs in the set and finding the machine with minimum expected completion time for all jobs. The job with maximum expected completion time is allocated to the machine which has MCT for that job. This improves the makespan and balances the load to some extent and performs best for jobs with longer execution time [13].

C. Genetic algorithm

In SGA, a new generation of individuals is created by replacing only a portion of the population with the new individuals. Unlike other replacement strategies, in SGA, a new individual replaces the individual that is the most similar to it only with new individual obtains a better fitness value than the one to be replaced.

The genetic algorithm is an optimal searching technique which searches a near-optimal solution in large solution spaces. Genetic algorithm is high level algorithm that integrates other methods and genetic operators. Genetic algorithm-based schedulers for computational grids and most of genetic algorithm operators are implemented and compared, to find the best genetic algorithm scheduler [11].

D. IPDT fuzzy scheduler

IPDT-FUZZY scheduler considers the demands of grid applications with such uncertainties. The scheduler uses fuzzy optimization, and both computational and communication demands are expressed as fuzzy numbers. Its performance was evaluated and it was shown to be attractive when communication requirements are uncertain. Its efficacy is compared, via simulation, to that of a deterministic
counterpart scheduler and the results reinforce its adequacy for dealing with the lack of accuracy in the estimation of communication demands [1, 2].

E. Priority based scheduling algorithm

In this algorithm a new parameter named "priority" has been taken into consideration. The algorithm classifies the jobs into high, medium and low categories based on their priority. The priority assignment is done by considering the computational power of job and level of parallelism. The value for level of parallelism is assigned based on the amount of parallelism exhibited by the job and the amount of parallelism exhibited by the available resources. Generally, a job, which needs high computational power and exhibits low parallelism, is given a high priority [7].

F. Application demand aware scheduling algorithm

Application demand aware scheduling was based on application demand, through which, most users are satisfied by guaranteeing their submitted jobs to be completed within their expected completion times without sacrificing system utilization more. This is the novelty of this algorithm, and it is beneficial for job scheduling and resource sharing in heterogeneous environment [5].

G. Security-driven scheduling algorithm

Security-sensitive applications that access and generate large data sets are emerging in various areas including bioinformatics and high energy physics. Data grids provide such data-intensive applications with a large virtual storage framework with unlimited power. However, conventional scheduling algorithms for data grids are unable to meet the security needs of data-intensive applications.

Security-driven scheduling algorithm addresses the problem of scheduling data-intensive jobs on data grids subject to security constraints. Using a security- and data-aware technique, a dynamic scheduling strategy is proposed to improve quality of security for data intensive applications running on data grids. To incorporate security into job scheduling, it introduce a new performance metric, degree of security deficiency, to quantitatively measure quality of security provided by a data grid [9].

H. Probabilistic and adaptive scheduling algorithm

Probabilistic and adaptive scheduling algorithm is based on system generated predictions for Inter-Grid resource sharing keeping collaborating Grid systems autonomous and independent. First it use system-generated job runtime estimates without actually submitting jobs to the target Grid system. Then this job execution estimate is used to predict the job scheduling feasibility on the target system. Algorithm adapts itself to the actual resource behaviour and performance [4].

I. Multisite co-allocation scheduling

Cooperation of multi-domain massively parallel processor systems in computing grid environment provides new opportunities for multisite job scheduling. At the same time, in the area of co-allocation, heterogeneity, network adaptability and scalability raise the challenge for the international design of multisite job scheduling models and algorithms.

Multisite co-allocation scheduling algorithm represents multisite job scheduling schema through the introduction of multisite job scheduling model and the performance model under the grid environment. It introduces two job multisite and cooperative scheduling models and algorithms with the core of the optimal and greedy-heuristic resource selection strategies [10].

III. PROBLEM DEFINITION

In this adaptive job scheduling algorithm, the jobs are scheduled by considering all the parameters described below. The user deadline is considered as the user demand. The Execution time
is considered as the application demand and the data rate is considered as the communicational demand. The size of instruction that can be executed is represented as Million Instruction per second (MIPS). This MIPS is the measure of the computational power of the particular processor. It may vary from processor to processor.

Each machine executes a single task at a time. In grid environment, the size of tasks and the number of machines are known priori. Expected Time to Compute (ETC) of each job in every machine is calculated by using the formula (1). ETC (Ji, Rj) is the estimated execution time of job Ji on machine j. The ETC matrix is calculated from the above values.

$$ETC (J_i, R_j) = \frac{L (J_i)}{M (R_j)} \quad (1)$$

Where L (Ji) is the length of the instructions of job Ji and M (Rj) is MIPS of the resource Rj.

The time taken for data transfer for that particular resource T (Ji, Rj) is calculated by the formula (2)

$$T (J_i, R_j) = \frac{S (J_i)}{B (R_j)} \quad (2)$$

Where S (Ji) is total size of the files to be transferred for the job Ji and B (Rj) is Bandwidth of the particular resource (Rj).

The Expected Completion time (ECT) for each task ti on each machine Rj is given by the formula (3)

$$ECT (J_i, R_j) = ETC (J_i, R_j) + T (J_i, R_j) \quad (3)$$

The makespan which is defined as the maximum time taken to complete all the jobs is given by the formula (4)

$$Makespan = \max (ECT (J_i, R_j)) \quad (4)$$

The problem of scheduling meta tasks to resources must include the following.

1. The number of meta tasks that are to be scheduled.
2. The number of resources available in the grid for processing the meta tasks.
3. The processing capacity of each resource in MIPS
4. The size of jobs in millions of instructions.
5. ETC matrix of size R×J where J is the number of jobs and R is the number of resources.

Let us consider the mathematical representations to denote the relationships between the resources and jobs and also to introduce the parameters involved in our algorithm such as execution time, completion time, ready time, etc.

$$ECT_{ij} \quad \text{Completion time of the job or task} \ J_i \ \text{in the resource} \ R_j$$

$$DCT_{ij} \quad \text{Difference in time between the deadline given by the user and the calculated completion time for the job in available resources}$$

$$\text{MinDCT}_{ij} \quad \text{The minimum value from the difference values DCT}_{ij} \ \text{for the given job}$$

$$\text{UT}_i \quad \text{User requisition time or the deadline given by the user for the jobs in U}$$

$$\text{ECT}_{ij} \quad \text{Expected Completion Time of job} \ J_i \ \text{in resource} \ R_j$$

The resource set is represented as R={R1,R2,R3,.....,Rm}. As we consider the static environment both the jobs submitted and the resource available are taken as fixed and they do not change over time. The jobs submitted can be enclosed within the job set which is represented as R={J1,J2,J3,.....,Jn}. The jobs submitted are considered as the independent tasks that can be executed in parallel with other available tasks. Also the tasks submitted are fixed and they do not change with time. The users submit the jobs with the requisition time i.e. within which the job needs to be completed which can also be called the demanded deadline of the user for the submitted jobs.
IV. PROPOSED ALGORITHM

In the proposed algorithm, first the Expected Completion Time (ECT) is calculated for all the available resources in the grid by using the computational power of the resources. For a single job more number of expected completion times for different resources has been calculated. The time taken for data transfer to the resources is also considered during the calculation of ECT. The user deadline for that particular job is matched with calculated expected completion times. The difference between the expected completion time and user deadline is calculated. Then allocate the job to the resource that has the minimum difference value. Then remove the allocated job from the job set. Then the waiting time of the resource is changed and the ETC matrix is recalculated for the remaining unmapped jobs. Then continue the above steps until all the jobs are scheduled. Thus both the user satisfaction and system performance can be taken into consideration effectively in this algorithm. Now we will give the detailed algorithm description.

1) While there are jobs to be allocated do
2) For each job \( J_i \) to be scheduled \( J_i \in U, 1 < i < n \),
   Do
   For each resource \( R_j \) available, \( R_j \in R, 1 < j < m \),
   Do
   \[ ETC(J_i, R_j) = L(J_i)/M(R_j) \]
   \[ T(J_i, R_j) = S(J_i)/B(R_j) \]
   \[ ECT(J_i, R_j) = ETC(J_i, R_j) + T(J_i, R_j) \]
   End for
   End for
3) \( Job = \text{Min} (UT_i), \text{where } 1 \leq i \leq n \)
4) For each job \( J_i \) in \( U \)
   Do
   If \( J_i = \text{job} \) then
   Begin
   For each resource \( R_j \) in \( R \)
   Do
   \[ DCT_{i,j} = \text{a maximum value} \]
   End if
   End for
   \[ \text{MinDCT}_{i,j} = \text{Min} (DCT_{i,j}) \text{ Where } 1 \leq j \leq m \]
   Allocate the job \( J_i \) to the resource \( R_j \)
   Remove \( J_i \) from \( U \)
   Update the ready time of the resource \( R_j \)
   End if
End for
5) Repeat, until all the jobs are assigned to the resources
6) End while

V. CONCLUSION AND FUTURE WORK

The proposed algorithm schedules the job which has the earliest deadline to the most suitable resource among the available resources. It can guarantee most jobs to be completed ahead of their expected completion time without losing good system performance in scheduling process and this can make most of the users to get satisfied, which is the primary aim of scheduling process. This algorithm is more beneficial with respect to the user satisfaction and has minimal makespan when compared with Min-min algorithm and has good system performance when compared with the application demand aware scheduling algorithm. In the proposed system, only the independent tasks are considered. So this work can be extended to suit for dependent tasks. The future work is to study the function and feasibility of dynamic scheduling, to make some possible improvements on the current scheduling algorithm and make it more flexible and efficient in actual application.

REFERENCES


AN EVOLUTIONARY MODEL FOR OUTLIER DETECTION

S.Santhoshkumar¹, P.Vishnu Raja², Dr.V.Murali Bhaskaran³

¹PG Scholar, Department of CSE, Kongu Engineering College, Perundurai, Tamilnadu, India.
²Assistant Professor, Department of CSE, Kongu Engineering College, Perundurai, Tamilnadu, India.
³Principal, Paavai College of Engineering, Pachal, Namakkal, Tamilnadu, India.

Abstract- Outlier detection problem has important applications in the field of fraud detection, network robustness analysis and intrusion detection. Fundamental issue is that the notion of objects that are outliers typically varies between users, problems, domains or even datasets. Various researches have been carried out with different methods for outlier detection problems. The objective of the Outlier Detection problem is to find small groups of data objects that are exceptional when compared with rest large amount of data. In this paper, we propose Multi-populated Parallel Genetic Algorithm for outlier detection problem. Testing on numerous examples from the related work indicates that MPGA is an appropriate tool for solving outlier detection problem. More importantly, MPGA provides high quality solutions, even to large size problems in a reasonable computational time.

Index Terms – outlier mining, outlier detection problem, genetic algorithm

1. Introduction

Outliers are the data objects that do not comply with the general behavior or model of the data, which are grossly different from or inconsistent with the remaining set of data. It can be caused by measurement or execution error. For example, the display of a person’s age as 999 could be caused by a program default setting of an unrecorded age. Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This could result in the loss of important hidden information because one person’s noise could be another person’s signal. In other words, the outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus outlier detection and analysis is an interesting data mining task referred to as outlier mining.

The problem of defining outliers is nontrivial. If a regression model is used for data modeling, analysis of the residuals can give a good estimation for data “extremeness”. When multidimensional data are analyzed, not any particular one but rather a combination of dimension values may be extreme. For nonnumeric (i.e., categorical) data, the definition of outliers requires special consideration.

Outlier mining can be described as follows: With given a set of n data points or objects and k, the expected number of outliers, we must find the top k objects that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data.
2. Related Work

The outlier detection problem, in its most general form, is difficult to solve due to a number of challenges [2]. The boundary between normal and outliers behaviour is often not precise. In many problem domains, normal behaviours keep evolving and the exact notion of an outlier varies with different tasks. Numerous techniques have been proposed to detect outliers for different applications. These techniques can be categorized into several approaches [2, 4, 9]: statistical approach, clustering based approach and distance based approach. In the distance based approach, a simple distance or similarity measure between every two instances/points in the data set is calculated and the points whose distances are longer than a particular radius (threshold) are considered outliers. Compared with the clustering based approach, the distance based approach is much simpler to use. Unlike the statistical based methods, the distance based methods make no prior assumptions about the data distribution model and are more suitable for multi-dimensional data sets. Due to these advantages, the distance based approach is widely used in outlier detection.

In the distance based approach, there are two fundamental methods on which many later techniques are based on [1, 7]. The first one builds upon the work of Knorr et al. [5]. In this work, a data point is defined as a Distance- Based outlier DB(β, r), where β is specified by the user based on the actual situation and r is the distance radius acting as an outlier threshold. While β is relatively easy to specify as outliers usually have a small neighbourhood, the value of r is usually very difficult to estimate and typically requires trial and error via hand-crafting of empirical search [19].

The second fundamental method develops the work from Breunig et al. [2]. This work uses local outlier factors (LOFs) rather than global distances [5]. A data point is given an outlier-ness score based on its relative density with respect to the nearest neighbour points. This method can detect outliers in data sets that have regions of varying densities, which cannot be easily handled by the Knorr’s algorithm [5]. However, this method requires specifying the number of neighbourhood points (MinPtn) a priori, which typically needs hand-crafting and trial and error.

Another potential disadvantage for the distance based methods is the computational cost. For relatively small data sets this is not a problem. However, for larger data sets these methods typically require a large computational effort since the calculation of the distances between a large number of data instances/points is costly [4].

This paper aims to convert the outlier detection problem to an optimization problem and develops an efficient genetic algorithm for outlier detection. Instead of using a manual handcrafting, trial and error process, this approach will automatically evolve good values for the important parameters.

3. Genetic Algorithm

Genetic Algorithms (GA) are adaptive heuristic search algorithms based on the evolutionary concept of natural selection and genetics. It follows the Darwin’s principles of “Survival of the fittest” where the fit best individuals retain their positions overtaking the weaker individuals in a group which they compete for the limited resources. Genetic algorithm is robust when compared to other searching mechanisms. Even though it is generally said to be random process, it is not actually random. Instead, it chooses the best individuals in each iteration thereby moving fast towards the stable optimal solution from the initial random population of individuals chosen.
In GA, the potential solution to a problem can be represented by a set of parameters called genes and the genes are combined together to form a structure called chromosome. N chromosomes are collectively called as population. In genetic terms, genes are called as genotype and chromosomes are called as phenotype. Initially, the chromosomes in the population are chosen at random. It then applies recombination genetic operators to these structures so as to proceed towards final solution.

By calculating the fitness value for all chromosomes, it evaluates these structures and allocates reproductive opportunities in the next generation in such a way that those chromosomes which provide a better solution to the target problem are given more chances to reproduce than those chromosomes which represent poor solutions.

3.1 Initialize Population

Initially many individual solutions are randomly generated to form an initial population, covering the entire range of possible solutions (the search space). Each point in the search space represents one possible solution marked by its value. Binary encoding scheme is used for the initialization of the population.

The encoded chromosome for the solution can be shown as,

Node Assignments:

\[ x = \{a, b, c, d, e, f, g, h\} \]
\[ n = \{1, 1, 1, 2, 2, 2, 3\} \]
Groups: G: \{1, 2, 3\}

3.2 Selection

Parents for this reproduction are selected using Roulette Wheel selection mechanism. Although all selection mechanisms have the final target of choosing the best chromosomes for the next crossover phase, they differ in the way they use to evaluate them. They retain the best individuals over the iterations by replacing the worst individuals which have the low probability of being carried over to the next generations. The selection of the chromosomes are done by calculating the fitness function,

\[ f = \frac{2r}{N(N + 1)} \]  

Where \( f \) is the fitness value and \( r \) is the rank of the chromosome.

3.3 Reproduction (Crossover)

During Reproductive phase, the chromosomes with high fitness values are recombined to form new chromosomes which constitute the individuals for the next generation. Crossover takes two individuals, cuts them at random positions to get head and tail segments. The tail segments are then swapped between two parents to form two new full length chromosomes or offsprings. The offsprings thus produced inherit the genes and their characteristic from both parents which may eventually turn out to be the optimal solution when they are subjected to the same process in further generations.

3.4 Mutation

Mutation is the process of randomly flipping a bit in the entire chromosome with some fixed small probability. It is done to introduce some form of diversity among the chromosomes without sacrificing the characteristics of the parents. In the process of genetic algorithm, sub-optimal solutions may be attributed to selection pressure in traditional selection mechanisms which ignore the weaker individuals to a larger extent. If they are given some better chance of survival in the next generation, there will be an improved chance for them to converge to an optimal solution. The proposed approach is a new variant of the standard genetic algorithm which incorporates the advantageous feature of clustering in selection process of genetic algorithm.
Step 1: Generate Initial Population

Step 2: Evaluate Fitness of each Individual

Step 3: Sort the Individuals based on their Fitness values

Step 4: Divide the Population into n groups

Select Individuals

Apply Genetic Operators

Evaluate Fitness of Offspring

Add offspring to the Same Group

Select Individuals

Apply Genetic Operators

Evaluate Fitness of Offspring

Add Offspring to the Same Group

Select Individuals

Apply Genetic Operators

Evaluate Fitness of Offspring

Add Offspring to the Same Group

Step 6: Combine Groups

Step 7: Allow Migration of Individuals based on Fitness

Step 8: Repeat the process from Step 5 to required number of Generations

Step 9: Sort the Individuals based on fitness

Figure 3.1 Flow chart of Multi-populated Parallel Genetic Algorithm
A fitness function must be formulated for each problem that is to be solved by genetic algorithm. For a particular chromosome, a fitness function or objective function returns a single numeric fitness value or “figure of merit” which is proportional to the “utility” or “ability” of that chromosome in the entire population consisting of n chromosomes. For many problems like functional optimization, objective function value is enough to attain a solution. But for complex combinatorial problems, a combination of performance measures relating to that specific problem will only drive towards the optimal solution.

4. Experimental Results
In this section we have evaluated the proposed algorithm for the ability to find the number of outliers present in the dataset. In MPGA, when compared to single point crossover two point crossover gives better offsprings. Thus the proposed system provides high quality solutions in a reasonable computing time there by limiting population size and number of generations. Figure 4.1 shows the performance under various executions and Figure 4.2 shows the offsprings generated under various executions of MPGA and SGA.

4.1 Impact of variation of crossover operators
Here, the influence of various crossover operators on the proposed algorithm is studied. In general one point and two point crossovers are applied and their impact on the profit of the proposed algorithm is listed in the table 4.1 and pictured in figure 4.3

<table>
<thead>
<tr>
<th>No of Executions</th>
<th>Single point Crossover</th>
<th>Two point Crossover</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>368767</td>
<td>368739</td>
</tr>
<tr>
<td>2</td>
<td>352543</td>
<td>376700</td>
</tr>
<tr>
<td>3</td>
<td>300320</td>
<td>367938</td>
</tr>
<tr>
<td>4</td>
<td>367610</td>
<td>368601</td>
</tr>
<tr>
<td>5</td>
<td>333974</td>
<td>341777</td>
</tr>
<tr>
<td>6</td>
<td>355391</td>
<td>374541</td>
</tr>
<tr>
<td>7</td>
<td>293888</td>
<td>307481</td>
</tr>
<tr>
<td>8</td>
<td>318918</td>
<td>319498</td>
</tr>
<tr>
<td>9</td>
<td>393925</td>
<td>389609</td>
</tr>
</tbody>
</table>

Figure 4.1 Performance comparison of MPGA with SGA
Figure 4.2 Number of generations in SGA and MPGA
5 Conclusion And Futurework

In this paper, we have proposed a new algorithm for outlier detection. The experimental results have shown that the proposed genetic algorithm is good in calculating the number of outliers in a particular period of time. For future work it has been planned to implement the proposed system in distributed environment and to improve the processing speed and performance of the algorithm.

References


