Clustering is an exploratory tool for analyzing large datasets and has been extensively used in numerous application areas. It is a key step in the analysis of Gene expression data. K. Vivekanandan and P. Krishnakumari provides a framework for the evolution of clustering in Gene expression analysis. When image is transmitted over channel it is often corrupted by impulse noise due to faulty communication or noisy channels. A. Bharathi, K. Anandakumar and A. Shanmugam introduced a novel filtering technique with superior noise removal capacity compared to conventional non-linear filter. Fuzzy set theory has been successfully implanted to control and pattern recognition fields. Heterogeneity is a major characteristic of all modern computer networks as a consequence of the proliferation of different types of OS acting as nodes in such an environment. T. Amudha, Manu Rajan Nair approaches this issue by presenting the solution to a seemingly innocuous problem. That is locating a particular file or files that match a given regular expression on the available network. Reason Qos studies in sensor networks focus on only Qos domain either timeliness or reliability. They are also limited in differentiating services for traffics with different levels of timeliness and reliability requirements. V. Vanitha and D. Palanivel Rajan have analysed the technical issues for supporting Qos constrains traffic in wireless sensor network. As per ISO 8402 standard quality of software is the totality of features and characteristic of a product or services and its ability to satisfies stated or implied needs. K. Paithankar and M. Ingle proposes a generalized classification of quality attribute that leads to provide support through the software development process to produce quality software in view of products process policy plan and service. Interconnection network provide communication between different processing elements and systems memory modules. A. Subramanyam, E. V. Prasad, K. Srinivasulu and T. Harikrishna have shown that the model for the performance evaluation of synchronize multistage interconnection networks using a new queueing model which gives very accurate results and performance analysis and efficient in computation. The scheduling is a methodology of managing multiple queues of processes in order to minimize delay and optimized performance of the system in the environment where queues of processes exist with servers. D. Shukla, Saurabh Jain and Shweta Ojha presents a Markov chain model based in the environment of multilevel queue scheduling with the multiple processes assuming the random movement of scheduler over various processes and different queues.
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Crypto-Compression of Medical Images on Neural Cryptography</td>
<td>1147</td>
</tr>
<tr>
<td>With Queries In Telemedicine System</td>
<td></td>
</tr>
<tr>
<td>N. Prabakaran, C.M. Velu and P. Vivekanandan</td>
<td></td>
</tr>
<tr>
<td>2. A Survey on Genetic Algorithm Based Clustering</td>
<td>1157</td>
</tr>
<tr>
<td>Techniques For Micro Array Gene Data</td>
<td></td>
</tr>
<tr>
<td>K. Vivekanandan, P. Krishnakumari</td>
<td></td>
</tr>
<tr>
<td>3. Weighted Fuzzy Mean Filters For Image Processing</td>
<td>1170</td>
</tr>
<tr>
<td>A. Bharathi, K. Anandakumar, A. Shanmugam</td>
<td></td>
</tr>
<tr>
<td>4. Generalized Classification of Software Quality Attributes</td>
<td>1177</td>
</tr>
<tr>
<td>K. Paithankar, M. Ingle</td>
<td></td>
</tr>
<tr>
<td>5. Qos In Wireless Sensor Networks – Survey</td>
<td>1189</td>
</tr>
<tr>
<td>V. Vanitha, D. Palanivel, Rajan</td>
<td></td>
</tr>
<tr>
<td>6. An Agent-Based Technique For Network-Wide</td>
<td>1199</td>
</tr>
<tr>
<td>File Search In Heterogeneous Peer-To-Peer Systems</td>
<td></td>
</tr>
<tr>
<td>Ms. T. Amudha, Mr. Manu Rajan Nair</td>
<td></td>
</tr>
<tr>
<td>7. A New Model For Evaluating Performance Of Processor Memory Interconnections</td>
<td>1209</td>
</tr>
<tr>
<td>A. Subramanyam, E. V. Prasad, K. Srinivasulu, T. Harikrishna</td>
<td></td>
</tr>
<tr>
<td>8. Analysis of Thread Scheduling With Multiple Processors</td>
<td>1220</td>
</tr>
<tr>
<td>under A Markov Chain Model</td>
<td></td>
</tr>
<tr>
<td>D. Shukla, Saurabh Jain And Shweta Ojha</td>
<td></td>
</tr>
</tbody>
</table>
Crypto - Compression of Medical Images on Neural Cryptography with Queries in Telemedicine System

N. Prabakaran¹, C.M. Velu² and P. Vivekanandan³

ABSTRACT
There is a requisite to secure the medical images from the hacker admittance when the switch over of medical information is taken place among the patients and doctors. We can generate a private key using neural cryptography, which is based on synchronization of Tree Parity Machines (TPMs) by online learning. The random inputs are generated by Pseudo-Random Number Generators (PRNGs). In the proposed TPMs random inputs are replaced with queries are considered. The queries depend on the current state of A and B TPMs. Then, TPMs hidden layer of each output vectors are compared. That is, the output vectors of hidden unit using Hebbian learning rule, left-dynamic hidden unit using Random walk learning rule and right-dynamic hidden unit using Anti-Hebbian learning rule are compared. Among the compared values, one of the best values is received by the output layer. Similarly, the other hidden units, left-dynamic hidden units and right-dynamic hidden units perform the same operations and values are received by the output layer. A private key (256-bit) with padding bits is used for encryption and decryption in Rijndael algorithm and Huffman coding is used for compression of medical images, which is produced as Crypto-Compressed and Encrypted Medical Images (CCEMI). The CCEMI is protected by password, which is a combination of lower layer’s spy unit vector and upper layer’s spy unit vector. We have shown more security of medical images, it is very difficult to break a private key by brute force attack.

Keywords: Neural Cryptography, Medical Images, Rijndael algorithm, Crypto-Compression.

1. INTRODUCTION
Two identical dynamical systems, starting from different initial conditions can be synchronized by a common input values which are coupled to the two systems. Two networks which are trained on their mutual output can synchronize to a time-dependent state of identical synaptic weights. The networks receive a common input vector after calculating their outputs and update their weight vectors according to the match between their mutual outputs in every time step. The input or output relations are exchanged through a public channel until their weight vectors are identical and can be used as a secret key for encryption and decryption of secret messages. The random inputs are replaced by queries in this network. It is based on exchanging inputs between A and B which are correlated to weight vectors of the two networks [15].

The advancement in computer technology and communication encourages health-care provider to provide health-care over the telemedicine. Telemedicine
is the integration of telecommunications technologies, information technologies, human-machine interface technology and medical care technologies for the purpose of enhancing health delivery across space and time. Telemedicine means the use of computer and communications technologies to augment the delivery of health-care services. Telemedicine can improve access to care, increase health care quality and reduce the cost.

The Rijndael algorithm is used for security of telemedical images and crypto-compression technique is used for reducing the size and decreases the quality of the medical images during the transmission by the networks. The Rijndael algorithm is conventional encryption standard since the common private key (256-bit) is used for encryption and decryption of Discrete Cosine Transform (DCT) coefficients, which is produced as CCEMI.

This paper is organized as follows. In Section 2, the basic algorithm for neural synchronization with queries is given. Also lower layer spy unit, upper layer spy unit, definition of the order parameters and transition probabilities of proposed TPMs are explained. In Section 3, the generation of queries is described. Overview of telemedicine system is briefly explained in Section 4. The crypto-compression technique and Rijndael algorithm of medical images are presented in Sec. 5. In Section 6, the security of medical image is discussed. Finally, conclusion is shown in Section 7.

2. NEURAL SYNCHRONIZATION

The weight vectors of the two neural networks start with random numbers, which are generated by Pseudo-Random Number Generators (PRNGs) [10]. In these networks, random inputs are replaced by queries. That is A and B choose alternatively according to their own weight vectors. The partners A and B receive a common input vector at each time; their outputs are calculated and then communicated over public channel [12]. If they agree on the mapping between the current input and the output, their weights are updated according to the learning rule.

2.1 A Structure of Tree Parity Machine

The TPMs consist of K-hidden units, Y-left dynamic hidden units and Z-right dynamic hidden units, each of them being a perceptron with an N-dimensional weight vector \( w \) [4]. The lower layer spy unit (\( \theta \)) is associated with the input units [7]. The upper layer spy unit (\( \xi \)) is associated with the hidden units (\( \sigma \)), left-dynamic hidden units (\( \delta \)), right-dynamic hidden units (\( \gamma \)) and output unit (\( \tau \)).

![Figure 1: A structure of Tree Parity Machine with K=3, Y=3, Z=3, \( \theta=1 \), \( \xi=1 \) and N=4.](image)

The lower layer and upper layer spy units receive the input values from the N-input units, Y-left dynamic hidden units, Z-right dynamic hidden units, K-hidden units and output unit. The network structure of this TPM is shown in Fig.1. The components of the input vectors \( x \) are binary.
\( x_i \in \{-1, +1\}, \quad x_m \in \{-1, +1\}, \quad x_k \in \{-1, +1\} \)  \hspace{1cm} (1)

and the weights are discrete numbers between \(-L\) and \(+L\)

\[
\begin{align*}
    w_0 & \in \{-L, -L+1, \ldots, L, L\}, \\
    w_m & \in \{-L, -L+1, \ldots, L-1, L\}, \\
    w_k & \in \{-L, -L+1, \ldots, L-1, L\}. 
\end{align*}
\]  \hspace{1cm} (2)

where \(L\) is the depths of the weights of the networks.

The TPM reads the input vectors using queries. These input vectors are correlated with the present weight vector \(t_k w\).

At odd time steps, the partner A generates an input vector which has a certain overlap to its weights \(A_k w\).

At even time steps, the partner B generates an input vector which has a certain overlap to its weights \(B_k w\). It is based on the queries to improve the security of the systems.

The index \(i = 1, \ldots, K\) denotes the \(i\)th hidden unit of TPM, \(m=1, \ldots, Y\) left-dynamic hidden unit of the TPM \([6]\), \(k=1, \ldots, Z\) right-dynamic hidden unit of the TPM and \(j =1,\ldots,N\) denotes the \(N\) input units.

The different transfer functions for hidden layer are given below

\[
\sigma_i = \text{sign} \left( \sum_{j=1}^{N} w_{ij} \cdot x_j \right) \hspace{1cm} (3)
\]

\[
\delta_i = \tanh \left( \sum_{m=1}^{N} \sum_{l=1}^{N} w_{im} \cdot x_m \right) \hspace{1cm} (4)
\]

\[
\gamma_i = \arctan \left( \sum_{k=1}^{N} w_{ik} \cdot x_k \right) \hspace{1cm} (5)
\]

where equation (3) is the transfer function of the hidden unit, the equation (4) the transfer function of the left-dynamic hidden unit and the equation (5) the transfer function of the right-dynamic hidden unit.

The transfer functions for lower layer spy unit and upper layer spy unit are given below

\[
\vartheta = -\text{sign} \left( \sum_{i=1}^{K} \sum_{j=1}^{N} x_i g_{ij} \right) \hspace{1cm} (6)
\]

\[
\xi = -\text{sign} \left( \sum_{i=1}^{K} \delta_i \sigma, \gamma_i, \tau \right) \hspace{1cm} (7)
\]

where the equation (6) is the transfer function of the lower layer spy unit and equation (7) the transfer function of the upper layer spy unit.

The K-hidden units of \(\sigma\), Y-left dynamic hidden units of \(\delta\) and Z-right dynamic hidden units of \(\gamma\) define common output bit of hidden layer of the network and are given by

\[
\beta_{\sigma} = \prod_{j=1}^{K} \sigma_i \hspace{1cm} (8)
\]

\[
\beta_{\delta} = \prod_{j=1}^{Y} \delta_i \hspace{1cm} (9)
\]

\[
\beta_{\gamma} = \prod_{j=1}^{Z} \gamma_i \hspace{1cm} (10)
\]

where equation (8) is the output for the hidden units, equation (9) the output for the left-dynamic hidden units and equation (10) the output for the right-dynamic hidden units.

The two TPMs compare the hidden layer’s output bits (hidden, left-dynamic and right-dynamic hidden units) and then update the weight vector to the output unit as well as partners A and B that are trying to synchronize their weight vectors \([8]\).

\[
\psi_{i}^{A,B} = \text{comp}(\beta_{\sigma}, \beta_{\delta}, \beta_{\gamma}) \hspace{1cm} (11)
\]

\[
\phi_{i}^{A} = w_{ij}^{A} x_{ij}^{A} \tau^{\psi_{i}^{A}} \hspace{1cm} (12)
\]

\[
\phi_{i}^{B} = w_{ij}^{B} x_{ij}^{B} \tau^{\psi_{i}^{B}} \hspace{1cm} (13)
\]

where equation (11) represents comparison of the output of hidden, left-dynamic and right-dynamic hidden units of A and B. The equation (12) and (13) represent output of hidden, left and right-dynamic hidden units of A and B respectively.
The equation (14) represents the output vector of the output unit of the TPM.

2.2 Learning Rules

The partners A and B initialize their random number of weight vectors before the start of the training period. At each time step $t$, a public input vector is generated and the bits $\tau^A$ and $\tau^B$ are switched over the public channel.

In the case of indistinguishable output bits $\tau^A = \tau^B$, each TPM adjust those of its weight vectors for which the hidden unit, left-dynamic hidden unit and right-dynamic hidden unit is identical to the output $\phi^{A/B} = \tau^{A/B}$. These weights are adjusted according to a given learning rules. They are

(a) Hebbian Learning rule for hidden units

$$w_i^A(t+1) = w_i^A(t) + x_i \tau^A \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B)$$

$$w_i^B(t+1) = w_i^B(t) + x_i \tau^B \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B)$$

(b) Random walk learning for left-dynamic hidden units

$$w_i^A(t+1) = w_i^A(t) + x_i \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B) \Theta(\tau^A \tau^B)$$

$$w_i^B(t+1) = w_i^B(t) + x_i \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B) \Theta(\tau^A \tau^B)$$

(c) Anti-Hebbian learning for right-dynamic hidden units

$$w_i^A(t+1) = w_i^A(t) - x_i \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B) \Theta(\tau^A \tau^B)$$

$$w_i^B(t+1) = w_i^B(t) - x_i \Theta(\tau^A \phi^A) \Theta(\tau^B \phi^B) \Theta(\tau^A \tau^B)$$

2.3 Order Parameters

The process of synchronization itself can be described by standard order parameters. These order parameters are

$$q_i = \frac{W_i^A \cdot W_i^B}{N}, q_{ij} = \frac{W_i^A \cdot W_j^B}{N}, q_{ijk} = \frac{W_i^A \cdot W_j^B \cdot W_k^B}{N}$$

The equation (18) represents weight distribution of hidden units, left-dynamic hidden units and right-dynamic hidden units of A's TPM.

$$R_{i}^{++} = \frac{W_i^+ \cdot W_i^+}{N}, R_{i}^{+-} = \frac{W_i^- \cdot W_i^+}{N}, R_{i}^{-+} = \frac{W_i^- \cdot W_i^-}{N}$$

The equation (19) represents overlap between two hidden units, two left-dynamic hidden units and two right-dynamic hidden units of A and B respectively.

The distance between two corresponding hidden unit, left-dynamic hidden units and right-dynamic hidden units are defined by the overlap is given below

$$\rho_{i} = \frac{R_{i}^{++}}{\sqrt{q_i}}$$

2.4 Transition Probabilities

A repulsive step can only occur in the $i^th$ hidden unit, $j^th$ left-dynamic hidden unit and $k^th$ right-dynamic hidden unit, if the two corresponding outputs $\phi_i$ are different. The probability for this event is given by the well-known generalization error for the perceptron [13]

$$\rho_{ijk} = \frac{1}{\pi} \cos^{-1}(\rho_{ijk})$$

where equation (21) represents the generalization error for hidden, left-dynamic and right-dynamic unit of two TPMs. The quantity $\rho_{ijk}$ is a measure of the distance between the weight vectors of the corresponding hidden units, left-dynamic hidden units and right-dynamic hidden units these values are independent. The values $\rho_{ijk}$ determine the conditional probability $P_r$ for a repulsive step and $P_a$ for an attractive step between two hidden units, left-dynamic hidden units and right-dynamic hidden units given identical output bits of the two TPMs.
of identical distances $\epsilon_{ijk}^{+} = \epsilon$, the values of K, Y, and Z are found as K=3, Y=3 and Z=3.

$$P_{a}^{\rho} = \frac{1}{2} \frac{(1-\rho x)^{3} + 9(1-\rho x)\epsilon x^{2}}{(1-\rho x)^{3} + 9(1-\rho x)\epsilon x^{2}}$$

(22)

$$P_{b}^{\rho} = \frac{6(1-\rho x)\epsilon x^{2}}{3(1-\rho x)^{3} + 9(1-\rho x)\epsilon x^{2}}$$

(23)

The equation (22) and (23) represent probability of attractive and repulsive steps between two hidden units, two left-dynamic hidden units and two right-dynamic hidden units of A and B respectively.

The attacker E can assign a confidence level to each output $\sigma_{i,j,k}^{+}$, $\delta_{i,j,k}^{+}$ and $p_{i,j,k}^{+}$ of its hidden units, left-dynamic hidden units and right-dynamic hidden units. For this task the local field is given by

$$h_{i,j,k}^{\rho} = \frac{w_{i,j,k}^{+} \cdot x_{i,j,k}^{+}}{\sqrt{N}} + \frac{w_{i,j,k}^{+} \cdot x_{i,j,k}^{+}}{\sqrt{N}}$$

(24)

where equation (24) represents the local field of hidden unit, left-dynamic and right-dynamic hidden units of an attacker’s TPM.

From the above fig. 2, we are able to predict the probability of repulsive steps occur more frequently in E’s TPM than in A’s and B’s for equal overlap $0 < \rho < 1$. So, the partners A and B have a clear advantage over a simple attack in neural cryptography.

Then the prediction error the probability of different output bits for an input vectors ‘x’ inducing a local field $h_{i,j,k}$ is given below [14]

$$e(\rho_{i,j,k}^{\rho}, h_{i,j,k}^{\rho}) = \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{\rho_{i,j,k}^{\rho} - h_{i,j,k}^{\rho}}{\sqrt{2(1-\rho_{i,j,k}^{\rho})} \sigma_{i,j,k}^{\rho}} \right) \right]$$

(25)

where equation (25) represents prediction error of the local field of hidden units, left-dynamic and right-dynamic hidden units of an attacker’s TPM.

3. GENERATION OF QUERIES

As both inputs $x_{i,n}$ and weights $w_{i,n}$ are discrete, there are only $(2L+1)$ possible solutions for the product $x_{i,n} \cdot w_{i,n}$ [11]. Therefore, a set of input vectors consisting of all permutation, which do not exchange $h_{i}$, can be depicted by counting the number $c_{i,n}$ of products with $x_{i,n} \cdot w_{i,n} = f$. Then the local field is given by

$$h_{i,j,k,n}^{\rho} = \frac{1}{L} \left[ \frac{1}{L} \left( \frac{\sum_{j,k} \left( (i_{j,k}^{+} - i_{j,k}^{-}) + (i_{j,k}^{+} - i_{j,k}^{-}) + (i_{j,k}^{+} - i_{j,k}^{-}) \right) }{L} \right) \right]$$

(26)

where equation (26) is the number of inputs and weights in the local field of TPM.

The sum $\eta_{i,j,k,n} = c_{i,j,k}^{+} + c_{i,j,k}^{-} + c_{i,j,k}^{+} + c_{i,j,k}^{+}$ is equal to the number of weights with $\left| i_{j,k}^{+} - i_{j,k}^{-} \right|$ and thus independent of ‘x’. Accordingly, one can write $h_{i,j,k}^{\rho}$ as a function of only L variables, because the generation of queries cannot change ‘w’.

$$h_{i,j,k,n}^{\rho} = \frac{1}{L} \left[ \frac{1}{L} \left( \frac{\sum_{j,k} \left( (i_{j,k}^{+} - i_{j,k}^{-}) + (i_{j,k}^{+} - i_{j,k}^{-}) + (i_{j,k}^{+} - i_{j,k}^{-}) \right) }{L} \right) \right]$$

(27)

where equation (27) is the inputs and sum of current weights vectors of the local field of two TPMs.
The inputs vectors ‘x’ is connected with zero weights which are selected by randomly, because they do not determine the local field. The other input bits $x_{i,m}$ are divided into $L$ groups according to the absolute value $w_{im} = |w_{im}|$ of their corresponding weight. In each group, $c_{i,l}$ inputs are selected randomly and set to $x_{i,m} = \text{sign}(w_{im})$. The remaining $n_{i,j} - c_{i,l}$ input bits are set to $x_{i,m} = -\text{sign}(w_{im})$.

The maximum possibilities of the weight vectors of an attacker’s TPM is given by

$$l_{max} = (4L + 2)^{E - X - Y}$$

Then

$$\ln(l_{max}) = (K + Y + Z) - N \ln(4L + 2)$$

4. OVERVIEW OF TELEMEDICINE SYSTEM

The rapidly evolving telecommunications technologies are creating an environment where individuals will be able to communicate interactively using a variety of media. The application of these technologies to provide medical care is referred to as telemedicine. The medical images for these new technologies is being researched, but it is still to be determined how effective these technologies will be in extending the reach of medical care to geographically and socio-economically isolated populations.

Despite an abundance of physicians and a large, well financed health care system, many areas of the United States still face a chronic shortage of medical providers of all types. Additionally, the ability of most persons living in rural areas to receive the most current specialty or sub-specialty care will be limited geographically under any future system of health care envisioned. These access and provider distribution issues must be addressed in order to achieve quality health care in the medically suburban area. The ability to provide medical care through telemedicine offers a practical solution to this misdistribution.

Telemedicine is the use of electronic communication and information technologies to provide clinical services when participants are at different locations. Videoconferencing, transmission of telemedical images, e-health including patient portals, remote monitoring of vital signs, continuing medical education and nursing call centers are all considered part of telemedicine and tele-health. Telemedicine offers a means to help transform healthcare itself by encouraging greater consumer involvement in decision making and providing new approaches to maintaining a health lifestyle.

5. CRYPTO-COMPRESSION OF MEDICAL IMAGE

The medical image is grouped into 8x8 blocks, changed from unsigned integers to signed integer and input to the Forward DCT (FDCT). At the output from the decoder, the inverse DCT outputs 8x8 sample blocks to
form the reconstructed image. Then, each DCT coefficients is separated by its corresponding constant in a standard quantization table and succeeded by rounding to the closest integer. This output value is normalized by the quantizer step size. Dequantization is the inverse function which returns the result to a representation appropriate for input to the IDCT [2]. All of the quantized coefficients are ordered into the ‘zig-zag’ sequence. This ordering helps to facilitate entropy coding by placing low-frequency coefficients before high-frequency coefficients.

The medical images, which specifies entropy coding method is Huffman coding. The 2-step process of entropy encoding converts the zig-zag sequence of quantized coefficients into an intermediate sequence of symbols and converts the symbols to a data stream in which the symbols no longer have externally identifiable boundaries.

The HVS (Human Visual system) is very sensitive to lower frequencies than to higher ones [9].

The Rijndael algorithm reads the input of medical image data block from the FDCT coefficients and execute various transformations for encryption and decryption. These are constituted as two-dimensional array of bytes [9]. The total number of rounds is 24 for encryption and decryption of medical image of FDCT coefficients block using private key which is generated by two TPMs. To encrypt a block of medical image of FDCT coefficients in AES (Advanced Encryption Standard), we first perform an Add Round Key step (XORing a sub-key with the block) by itself. There are regular rounds that involve 4 steps

(ii) SubBytes Transformation

The SubBytes transformation is a non-linear byte substitution that operates independently on each byte of the state using a substitution table (S-Box), which is invertible, is constructed by composing two transformations:
1. The multiplicative inverse in the Galois Field ($2^8$) with the irreducible polynomial is $m(x) = x^8 + x^4 + x^3 + x + 1$. The element $\{00\}$ is mapped to itself.

2. Apply the affine transformation (over GF($2^8$)):
   The inverse of SubBytes transformation which is needed for decryption, is the inverse of the affine transformation followed by the same inversion as the SubBytes transformation

(iii) ShiftRows Transformation
The ShiftRows transformation rotates each row of the input state to the left shift and then offset of the rotation corresponds to the row number. The inverse of this transformation is computed by performing the corresponding rotations to the right shift.

(iv) Mixcolumns Transformation
The MixColumns transformation operates on the state column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomial over GF($2^8$) and multiplied modulo $(x^4 + 1)$ with a fixed polynomial $a(x)$, given by

$$a(x) = \{03\} x^3 + \{01\} x^2 + \{01\} x + \{02\}$$

The coefficient of $a(x)$ is also elements of GF($2^8$) and is represented by the hexadecimal values in this equation. The inverse mixcolumn transformation is the multiplication of each column with

$$a^{-1}(x) = \{0B\} x^3 + \{0D\} x^2 + \{09\} x + \{0E\}$$

modulo $(x^4 + 1)$ for decryption process.

6. THE SECURITY OF MEDICAL IMAGES
Here, we determine that queries to increase the security of the neural key-exchange protocol and synchronization time. The CCEMI is based on the password-protection with padding bits, which is a combination of lower layer’s spy unit vector and upper layer’s spy unit vector. The medical images are encrypted and decrypted using 256-bit private key in Rijndael algorithm and compressed with Huffman coding. If the attacker wants to break the password then he has to decode the CCEMI and decrypt using 256-bit private key. An attacker tries to find out all possibilities of the right key. The attacker will take $10^{56}$ years to break the secret key using brute force attack.

7. CONCLUSION
In the proposed TPMs, we trained the three transfer functions in the hidden layer. That is, hidden unit using Hebbian learning rule, left-dynamic hidden unit using Random walk rule and right-dynamic hidden unit using Anti-Hebbian learning rule included with queries. Also, the queries increase the probability of repulsive steps for an attacker during the synchronisation. In addition, the method receives a new parameter, which can be adapted to give optimal security. The CCEMI has more secured due to password-protection, crypto-compression using Huffman coding and private key (256-bit) in Rijndael encryption. A private key is generated by two TPMs. The CCEMI reduces the time for transmission and the space on disk. For the attacker to find out all possibilities of password keys and private keys, it will take trillion years against the brute force attack.

REFERENCES


VivekanandanPeriyasamy received his Master of Science in Applied Mathematics from Madras University in 1978 and Doctor of Philosophy from Anna University in 1987. Also, he obtained his postgraduate degree in Master of Engineering in Computer Science and Engineering from Anna University in 1995. He is working as Professor of Mathematics, Department of Mathematics in Anna University from 1978. He visited Singapore, Malaysia, Bangladesh, Sultanate of Oman, Thailand, JAPAN and USA three times for presenting research papers and Chairing Sessions. He has published more than 77 research papers in national and international journals. His areas of research are Neural Networks, Internet Security and Software Reliability. Also, he is Director, KNowledge Data Centre, CPDE Building, Anna University, Chennai-600 025, India.
Prabakaran N, received M.Sc in Computer Science from Anna University. He is doing research under the guidance of Dr. P. Vivekanandan. His areas of interest are network security, visual programming, computer networks, neural networks and C# programming. He had published seven papers in International Journals and four papers in international conferences.

C. M. VELU received his Master of Science in Operations Research and Statistical Quality Control from Sri Venkateswara University, Tirupathi in 1985 and M.S in Computer Systems and Information from BITS, PILANI in 1994. He is currently doing research in Computer Science in Anna University under the guidance of Dr. P. Vivekanandan. He has visited UAE as a Computer faculty. He served as faculty of Computer Science and Engineering for more than two decades. He has published seven research papers in international journals. His areas of research are Artificial Neural Network, Digital Image Processing, Pattern Recognition, Data Compression, Data Warehousing and Data Mining. He has published about five papers in national and international conferences.
A Survey on Genetic Algorithm based Clustering Techniques for Micro array gene data

K.Vivekanandan 1        P.Krishnakumari 2

ABSTRACT
Clustering is a key step in the analysis of gene expression data, and in fact, many classical clustering algorithms are used, or more innovative ones have been designed and validated for the task. Despite the widespread use of artificial intelligence techniques in bioinformatics and, more generally, data analysis, there are very few clustering algorithms based on the genetic paradigm, yet that paradigm has great potential in finding good heuristic solutions to a difficult optimization problem such as clustering. In this paper the nature of microarray data is discussed briefly and a survey on genetic algorithm based clustering techniques for micro array gene data is presented. Some preliminary concepts that form the basis for the development of clustering algorithms are introduced. Finally, some of the most popular clustering techniques like GenClust, HGACLUS, hybrid method using EM algorithm, multiobjective genetic clustering algorithm are discussed. As such, the study provides a framework for the evaluation of clustering in gene expression analysis.

Keywords: clustering, gene data, genetic algorithm, microarray data analysis

1. MICRO ARRAY DATA AND ITS COMPLEXITY
The recently developed gene expression micro array technique measures the expression levels of thousands of genes in a single experiment. This large amount of data is something of a gold mine, from which a number of things can be found. Gene-expression micro array data have been explored in a variety of ways including gene clustering, gene selection and many others. Gene expression micro arrays are a prominent experimental tool in functional genomics which has opened the opportunity for gaining global, systems-level understanding of transcriptional networks. Micro array platforms for measuring the expression levels of most or all genes of an organism are available for a variety of organisms ranging from yeast to human. Experiments that use this technology typically generate overwhelming volumes of data, unprecedented in biological research, which makes the task of mining meaningful biological knowledge out of the raw data a major challenge. Hence, exploitation of gene expression data is fully dependent on the availability of advanced data analysis and statistical tools. Many clustering [9,25] algorithms and software tools for analysis of microarray data were developed in recent years. Clustering algorithms applied to gene expression data partition the genes into distinct groups according to their expression patterns over the probed biological conditions. Such partition should assign genes with similar expression patterns to the same cluster (keeping the homogeneity merit of the clustering solution) while retaining the distinct expression pattern of each cluster (ensuring the separation merit of the solution). Cluster analysis eas
the interpretation of the data by reducing its complexity and revealing the major patterns that underlie it. Clustering is the task of organizing a set of objects into meaningful groups. These groups can be disjoint, overlapping, or organized in some hierarchical fashion. The key element of clustering is the notion that the discovered groups are meaningful. Clustering is an exploratory tool for analyzing large datasets, and has been extensively used in numerous application areas. Clustering has a wide range of applications in life sciences and over the years it has been used in many areas ranging from the analysis of clinical information, phylogeny, genomics, and proteomics. For example, clustering algorithms applied to gene expression data can be used to identify co-regulated genes and provide a genetic fingerprint for various diseases. The primary goal of this article is to provide an overview of the various issues involved in clustering large datasets, describe the merits and underlying assumptions of some of the commonly used clustering approaches, and provide insights on how to cluster datasets based on genetic algorithm paradigm. The article is organized as follows. The sections 2 to 5 describe the various types of clustering algorithms developed over the years, similarity measures, limitations of the conventional clustering algorithms and dimensionality reduction. The sections 6 and 7 focus on genetic algorithm, criteria for evaluating clustering algorithms. The section 8 describes microarray technology and focuses on the problem of clustering data arising from microarray experiments. Finally, section 9 provides a brief introduction to the GA based clustering techniques like GenClust, HGACLUS, hybrid method using EM algorithm, multiobjective genetic clustering algorithm.

2. Types of Clustering Algorithms
The topic of clustering has been extensively studied in many scientific disciplines and a variety of different algorithms have been developed [18, 30, 31]. Two recent surveys on the topics [15, 17] offer a comprehensive summary of the different applications and algorithms. These algorithms can be categorized along different dimensions based either on the underlying methodology of the algorithm, leading to partition or agglomerative approaches; the structure of the final solution, leading to hierarchical or nonhierarchical solutions; the characteristics of the space in which they operate, leading to feature or similarity approaches.

2.1. Agglomerative And Partitional Algorithms
Partitional algorithms, such as K-means [19], K-medoids [16], probabilistic [6], graph partitioning based [13], or spectral based [16], find the clusters by partitioning the entire dataset into either a predetermined or an automatically derived number of clusters. Partitional clustering algorithms compute a k-way clustering of a set of objects either directly or through a sequence of repeated bisections. A direct k-way clustering is commonly computed as follows. Initially, a set of k objects is selected from the datasets to act as the seeds of the k clusters. Then, for each object, its similarity to these k seeds is computed, and it is assigned to the cluster corresponding to its most similar seed. This forms the initial k-way clustering. This clustering is then repeatedly refined by recalculating the new seed so that it optimizes a desired clustering criterion function. A k-way partitioning through repeated bisections is obtained by recursively applying the above algorithm to compute two-way clustering (i.e., bisections). Initially, the objects are partitioned into two clusters, and then one of these clusters is selected and is further bisected, and so on.
This process continues $k-1$ times, leading to $k$ clusters. Each of these bisections is performed so that the resulting two way clustering solution optimizes a particular criterion function. Criterion functions used in the partitional clustering reflect the underlying definition of the “goodness” of clusters. The partitional clustering can be considered as an optimization procedure that tries to create high-quality clusters according to a particular criterion function. Many criterion functions have been proposed [28]. Criterion functions measure various aspects of intracluster similarity, intercluster dissimilarity, and their combinations. These criterion functions use different views of the underlying collection, by either modeling the objects as vectors in a high-dimensional space or by modeling the collection as a graph. Hierarchical agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criterion is met. The three basic criteria to determine which pair of clusters to be merged next are single-link, complete-link, and group average (UPGMA - unweighted pair group method with arithmetic mean) [16]. The single-link criterion function measures the similarity of two clusters by the maximum similarity between any pair of objects from each cluster, whereas the complete-link criterion uses the minimum similarity. In general, both the single-link and the complete-link approaches do not work very well because they either base their decisions to a limited amount of information (single-link) or assume that all the objects in the cluster are very similar to each other (complete link). On the other hand, the group average approach measures the similarity of two clusters by the average of the pair wise similarity of the objects from each cluster and does not suffer from the problems arising with single-link and complete link. In addition to these three basic approaches, a number of more sophisticated schemes have been developed, such as CURE [22], ROCK [23], and CHAMELEON [18] that has been shown to produce superior results. CURE [22] is more robust to outliers, and identifies clusters having non-spherical shapes and wide variances in size. ROCK [23] is a robust hierarchical clustering algorithm for categorical attributes that employs links and not distances when merging clusters and provides good scalability. CURE and ROCK are designed to find clusters that fit some static models. These algorithms can breakdown if the choice of parameters in the static model is incorrect with respect to the data set being clustered. But CHAMELEON [18] measures the similarity of two clusters based on a dynamic model. In the clustering process, two clusters are merged only if the inter-connectivity and closeness (proximity) between two clusters are high relative to the internal inter-connectivity of the clusters and closeness of items within the clusters. The merging process using the dynamic model presented facilitates discovery of natural and homogeneous clusters. Finally, hierarchical algorithms produce a clustering solution that forms a dendogram, with a single all-inclusive cluster at the top and single-point clusters at the leaves. In contrast, in nonhierarchical algorithms there tends to be no relation between the clustering solutions produced at different levels of granularity.

3. SIMILARITY MEASURES

In most microarray clustering applications the goal is to find clusters of genes or clusters of conditions. A number of different methods have been proposed for computing these similarities, including Euclidean distance-based similarities, correlation coefficients, and mutual information. The use of correlation coefficient-based similarities is primarily motivated by the fact that while clustering gene expression datasets, the expression levels
of different genes are related under various conditions. The correlation coefficient values between genes is estimated by the Pearson correlation coefficient, which is given by

$$\text{sim}(i, j) = \frac{\sum_{n=1}^{m} (y_i - \bar{y}_i)(y_j - \bar{y}_j)}{\sqrt{\sum_{n=1}^{m} (y_i - \bar{y}_i)^2 \sum_{n=1}^{m} (y_j - \bar{y}_j)^2}}$$

and it can be used directly or transformed to absolute values if genes of both positive and negative correlations are important in the application. An alternate way of measuring the similarity is to use the mutual information between a pair of genes. The mutual information between two information sources \(A\) and \(B\) represent how much information the two sources contain for each other. D’Haeseleer et al [8] used mutual information to define the relationship between two conditions \(A\) and \(B\). A feature common to many similarity measures used for microarray data is that they almost never consider the length of the corresponding gene or condition vectors, which is the actual value of the differential expression level, but focus only on various measures of relative change or how these relative measures are correlated between two genes or conditions [29]. The reason for this is twofold. First, there are still significant experimental errors in measuring the expression level of a gene, and is not reliable to use it “as is.” Second, in most cases interest is shown on how the different genes change across the different conditions (i.e., either upregulated or downregulated) and the interest is not shown in the exact amount of this change.

4. LIMITATIONS OF THE CONVENTIONAL CLUSTERING SCHEMAS

Since the early days of the development of the microarray technologies, a wide range of existing clustering algorithms have been used, and novel new approaches have been developed for clustering gene expression datasets. The most effective traditional clustering algorithms are based either on the group-average variation of the agglomerative clustering methodology, or the \(K\)-means approach applied to unit-length gene or condition expression vectors. Agglomerative solutions are inherently suboptimal when compared to partitional approaches, which allow for a wider range of feasible solutions at various levels of cluster granularity. However, despite this, the agglomerative solutions tend to produce reasonable and biologically meaningful results, and allow for an easy visualization of the relationships between the various genes or conditions in the experiments. The ease of visualizing the results has also led to the extensive use of self-organizing maps (SOM) for gene expression clustering [25]. However, as the dimensionality of these datasets continues to increase (primarily by increasing the number of conditions that are analyzed), requiring consistency across the entire set of conditions will be unrealistic. These algorithms tend to produce local solutions and hence genetic algorithms are integrated to provide good heuristic solutions.

5. DIMENSIONALITY REDUCTION

Many new algorithms have been proposed recently to tackle the problem of clustering gene expression data with high dimensionality. The basic idea of global dimension reduction is to compress the entire gene/condition matrix to represent genes by vectors in a compressed space of low dimensionality, such that the biologically interesting results can be extracted [20]. Alter et al [2] proposed to use singular value decomposition (SVD) to compress the data and then apply traditional clustering algorithms (such as \(k\)-means). They also showed that their algorithms can find meaningful clusters on cancer cells, leukemia dataset and yeast cell cycle
dataset. Finding clusters in subspaces tackle this problem differently by redefining the problem of clustering as finding clusters whose internal similarities become apparent in subspaces or clusters that preserve certain expression patterns among the dimensions in subspaces [13]. The various algorithms differ from one another in how they model the desired clusters, the optimization algorithm and clustering algorithm that generate the desired clusters, and whether the algorithms allow genes that belong to more than one cluster (i.e., overlapping clusters). Cheng and Church [7] assume each expression value in the matrix as the addition of three components: the background level, the row effect, and the column effect. Thus, they use minimum mean squared residue as the objective function to find clusters in subspaces that have small deviations with respect to the rows in the cluster, the columns in the subspace, and the background defined by the cluster. Correlation clustering groups the data sets into subsets called correlation clusters such that the objects in the same correlation cluster are all associated to a common hyperplane of arbitrary dimensionality. The prominent application for correlation clustering is the analysis of gene expression data. Gene expression data contain the expression levels of thousands of genes, indicating how active the genes are, according to a set of samples. A common task is to find clusters of co-regulated genes, i.e. clusters of genes that share a common linear dependency within a set of their features. The first approach that can detect correlation clusters is ORCLUS [1] that integrates PCA into k-means clustering. The algorithm 4C [5] integrates PCA into a density-based clustering algorithm. Elke Achtert [10] proposed correlation clustering algorithm COPAC (Correlation PArtition Clustering) that aims at improved robustness, completeness, usability, and efficiency.

6. GENETICAL ALGORITHMS (GA)

In GAs, the search space of a problem is represented as a collection of individuals [11]. The individuals are represented by character strings, which are referred to as chromosomes. A collection of such strings is called the population. The purpose is to find the individual from search space with the best genetic material. The quality of an individual is measured with an objective function or the fitness function. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned to a number of copies that go into the mating pool. Biologically inspired operators like crossover and mutation are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till the termination condition is satisfied.

7. CRITERIA FOR EVALUATING CLUSTERING ALGORITHMS

Considering the characters of gene expression data, and the particular applications in functional genomics, the optimal algorithms for analysis of gene expression data need the following properties suggested by Jain et al and Han et al [17,14]

- Scalability and efficiency: Algorithms should be efficient and scalable considering the large amount of data to be handled.
- Irregular shape: Algorithms need to be able to identify a dense set of points which forms a cloud of irregular non spherical shapes.
- Robustness: The clustering mechanisms should be robust against large amounts of noise and outlier.
- Order insensitivity: Algorithms should not be sensitive to the order of input. That is, clustering results should be independent of data order.
• Number of Clusters: The number of clusters inside the data set needs to be determined by the algorithm itself and not prescribed by the user.

• Parameter estimation: The algorithms should have the ability to estimate any parameters required by the algorithm from the data set, and no domain knowledge input is required from the user.

• Dimensionality: Algorithms need the ability to handle data with high dimensionality or the ability to find clusters in subspaces of the original space.

• Stability: No data object will be classified into different clusters for different running of the algorithm.

• Incrementability: Algorithms should be able to incrementally handle the addition of new data or the deletion of old data instead of re-running the algorithms on the new data set.

• Interpretability: The clustering results of the algorithms need to be interpretable. That is, clustering may need to be tied up with specific biological interpretations and applications.

8. OVERVIEW OF MICROARRAY TECHNOLOGIES

DNA microarrays measure gene expression levels by exploiting the preferential binding of complementary, single-stranded nucleic acid sequences. cDNA microarrays, developed at Stanford University [Stanford University Genomic Resources: http://genome-www.stanford.edu] are glass slides, to which single-stranded DNA molecules are attached at fixed locations (spots) by high-speed robotic printing. Each array may contain tens of thousands of spots, each of which corresponds to a single gene. mRNA from the sample and from control cells is extracted and cDNA is prepared by reverse transcription. Then, cDNA is labeled with two fluorescent dyes and washed over the microarray so that cDNA sequences from both populations hybridize to their complementary sequences in the spots. The amount of cDNA from both populations bound to a spot can be measured by the level of fluorescence emitted from each dye. For example, the sample cDNA is labeled with a red dye and the control cDNA is labeled with a green dye. Then, if the mRNA from the sample population is in abundance, the spot will be red; if the mRNA from the control population is in abundance, it will be green; if sample and control bind equal the spot will be yellow; if neither binds, it will appear black. Thus, the relative expression levels of the genes in the sample and control populations can be estimated from the fluorescent intensities and colors for each spot. After transforming the raw images produced by microarrays into relative fluorescent intensity with some image processing software, the gene expression levels are estimated as log-ratios of the relative intensities. A gene expression matrix can be formed by combining multiple microarray experiments of the same set of genes but under different conditions, where each row corresponds to a gene and each column corresponds to a condition (i.e., a microarray experiment) [27, 3]. The Affymetrix GeneChip oligonucleotide array contains several thousand single-stranded DNA oligonucleotide probe pairs. Each probe pair consists of an element containing oligonucleotides that perfectly match the target (PM probe) and an element containing oligonucleotides with a single base mismatch (MM probe). A probe set consists of a set of probe pairs corresponding to a target gene. Similarly, the labeled RNA is extracted from sample cell and hybridizes to its complementary sequence. The expression level is measured by determining the difference between the PM and MM probes. Then, for each gene (i.e., probe set) average difference or log average can be calculated, where average difference is defined as the average difference
between the PM and MM of every probe pair in a probe set and log average is defined as the average log ratios of the PM/MM intensities for each probe pair in a probe set.

9. Genetic Algorithm Based Clustering for Microarray Data

9.1 Genclust

Genclust is a genetic algorithm for clustering gene expression data proposed by Vito Di Gesù et al. It has two key features: (a) a novel coding of the search space that is simple, compact and easy to update (b) it can be used naturally in conjunction with data driven internal validation methods. It is experimented with the FOM methodology, specifically conceived for validating clusters of gene expression data. The validity of the algorithm has been assessed experimentally on real data sets, both with the use of validation measures and in comparison with other algorithms like Average Link, Cast, Click and K-means. GenClust is experimentally competitive with K-means, Click [21] and Cast [4]. Moreover, the algorithm is well suited for use in conjunction with data driven internal validation methodologies and in particular FOM, which has received great attention in the specialized literature. It defines clustering as an optimization problem.

Given a subset \( Y = \{y_1, y_2, ..., y_m\} \) of \( X \), let \( c(Y) \) denote the centroid of \( Y \) and let its variance be

\[
\text{VAR}(Y) = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{d} (y_{ij} - c(Y))^2.
\]

Given an integer \( k \), we are interested in finding a partition of \( X \) into \( k \) classes \( C_0, C_1, ..., C_k-1 \) so that the total internal variance is

\[
\text{VAR}(P) = \sum_{i=0}^{k-1} \text{VAR}(C_i).
\]

The algorithm proceeds in stages, producing a sequence of partitions, each consisting of \( k \) classes, until a halting condition is met. Following the evolutionary computational paradigm, a population evolves by means of genetic operators, i.e., cross-over, mutation and selection, resulting in a random walk in cluster space, where the fitness function gives a drift to the process towards a local optimum. Its performance is measured by adjusted Rand index and FOM. Table 1 [26] shows the adjusted Rand index and FOM values for the rat RCNS data set of various algorithms.

Table 1: Performance of the Algorithms for RCNS Rat

<table>
<thead>
<tr>
<th>Method</th>
<th>Adjusted Rand</th>
<th>FOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenClust-random</td>
<td>0.168</td>
<td>3.00</td>
</tr>
<tr>
<td>Mix-Kmeans-random</td>
<td>0.444</td>
<td>3.81</td>
</tr>
<tr>
<td>Mix-Inverse-Random</td>
<td>0.259</td>
<td>3.81</td>
</tr>
<tr>
<td>Cast</td>
<td>0.120</td>
<td>3.98</td>
</tr>
<tr>
<td>Kmeans-Simulated-Anneal</td>
<td>0.187</td>
<td>3.71</td>
</tr>
<tr>
<td>Click</td>
<td>0.191</td>
<td>4.05</td>
</tr>
<tr>
<td>GenClust-AvLink</td>
<td>0.181</td>
<td>4.00</td>
</tr>
</tbody>
</table>

The table shows that GenClust-AvLink is to be preferred to GenClust-Random. Moreover, GenClust-AvLink seems to take better advantage of the output of Average Link than K-means. It also appears that GenClust-AvLink is competitive with Average Link and K-means, Cast and Click.

9.2 Hgclus

HGACLUS is suggested by Haiyan Pan et al [12]. In this paper, the parallelism searching capability of GAs is used to design a clustering schema (HGA-CLUS) combining merits of the Simulated Annealing to find an optimal or near-optimal set of medoids whose size was predefined. According to this optimal set of medoids, each observation was allocated to the nearest medoid and the best \( k \) clusters were then constructed. Each string was evaluated using the following fitness function

\[
f(s) = \frac{\text{trace}B/(k - 1)}{\text{trace}W/(n - k)}
\]
where \( n \) and \( k \) are the total number of points and the number of clusters in the partition, respectively. \( B \) and \( W \) are the covariance matrices of between-cluster sums and the pooled within-cluster sums of squares, respectively.

\[
p(s_h) = \frac{\exp\left(\frac{f(s_h)}{T}\right)}{\sum_{h=1}^{p} \exp\left(\frac{f(s_h)}{T}\right)}, \quad h = 1, 2, \ldots, p.
\]

where \( T > 0 \) is a cooling temperature. The cooling schedule function is

\[
T(g) = \frac{G - g}{G} T_0, \quad g = 0, 1, \ldots, G - 1.
\]

The process of fitness computation, selection, crossover, and mutation was executed for \( G \) generations. Variance Ratio Criterion (VRC) and Silhouette Width are the validation indices used by the algorithm. With regard to the criteria of external isolation and internal consistency, HGACLUS appeared to perform as well as or better than any of the other mentioned methods for multi-class clustering. The fig 1 shows [12] the results for the datasets.

9.3 Hybrid Method Using EM Algorithm

Zeke et al [32] suggest the hybrid method using EM algorithm. In this paper a framework is proposed that hybridizes Evolutionary Computation, in particular Genetic Algorithm with a local-learning algorithm to perform optimal clustering of time-course gene expression data. The hybrid algorithm combines the strengths of GA and the local-learning algorithm (EM) by using the former to select subset of data as initial cluster centers and the later to perform fast local optimization to achieve the final centers from these initial centers. In this way, the optimality of the final centers returned by the local-search algorithm can be used as the objective function for GA, which searches for the globally optimal subset of data as initial cluster centers. In other words, rather than beginning the local optimization from data points that are randomly chosen, the hybrid algorithm begins the local optimization from data points that are globally optimal, therefore increasing the consistency of the final clustering solution. The hybrid algorithm is applied to

![Figure 1: The average VRC and Silhouette Width Values of Five Clustering Methods for Models 1(3 sets of cancer genes) and Model 2 (5 sets of cancer genes)](image)

the human fibroblasts time course data that requires clustering of 512 useful genes and shows superior performance over using EM alone. Results show that although the hybrid algorithm requires higher computational cost, it performs consistently better in clustering accuracies. The hybrid framework is applied
A Survey on Genetic Algorithm based Clustering Techniques for Micro array gene data

The experimental results on gene expression time course data available on the public domain show the advantages of the hybrid GA-EM approach when compared with the standard approach of using random initialization EM algorithm only.

9.4 Multi Objective Genetic Clustering Algorithm
Sanghamitra Bandyopadhyay, Anirban Mukhopadhyay and Ujjwal Maulik [24] proposed a two-stage clustering algorithm, that employs a recently proposed variable string length genetic scheme and a multi objective genetic clustering algorithm. It is based on the novel concept of points having significant membership to multiple classes. An iterated version of the well known Fuzzy C-Means is also utilized for clustering. The significant superiority of the proposed two-stage clustering algorithm as compared to the average linkage method, Self Organizing Map (SOM) and a recently developed weighted Chinese restaurant based clustering method (CRC), widely used methods for clustering gene expression data, is established on a variety of artificial and publicly available real life data sets. The biological relevance of the clustering solutions is also analyzed. The Table 2 shows [24] maximum selectivity of different clusters produced by different algorithms on Yeast Sporulation data. The following Table 3 shows the comparison of various GA based methods used for gene data clustering.

Figure 2: The Human Fibroblasts To Serum Data (517 Genes).
A typical GA run is shown in Fig. 3 [32]. The fitness increases uni-directionally, which is the characteristic of the elitist selection scheme.

Figure 3: Evolution Of The Best-Fitness Found In The Population.
Table 3: Comparison Of GA Based Clustering Algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Approach</th>
<th>Validation Indices</th>
<th>Data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. GenClust</td>
<td>GA based partitional clustering</td>
<td>FOM &amp; Rand index</td>
<td>Rat RCNS</td>
</tr>
<tr>
<td>2. HGAclust</td>
<td>GA based simulated annealing concept</td>
<td>VRC &amp; silhouette width</td>
<td>Embryonal CNS data</td>
</tr>
<tr>
<td>3. Hybrid EM</td>
<td>GA based on mixture of MLR's</td>
<td>Maximum log likelihood / BIC</td>
<td>Human fibroblast to serum data</td>
</tr>
<tr>
<td>4. Multi objective clustering</td>
<td>GA based Fuzzy C means</td>
<td>ARC &amp; silhouette width</td>
<td>Yeast Sporulation data</td>
</tr>
</tbody>
</table>

10. Conclusion

In this survey several genetic algorithm based clustering techniques for micro array gene data are presented. Some preliminary concepts that form the basis for the development of clustering algorithms are discussed. This paper provides a framework for the evaluation of clustering in gene expression analysis.

References


10. Elke Achtert et al., “Robust, Complete, and Efficient Correlation Clustering”, In Proc. SIAM Int. Conf. on Data Mining (SDM), Minneapolis, MN,PP.413-418, 2007.


Author’s Biography

Dr. K. Vivekanandan received the Ph.D. degree in Computer science from Bharathiar University, India. He is currently Reader in Bharathiar University, India. He has a total teaching experience of 21 years. He has published 17 papers in international and national journals. He has produced 5 Ph D’s and 11 MPhil scholars in computer science. His research interests include data mining and knowledge discovery, and Management Information System.
P. Krishnakumari received the MPhil degree in Computer science from Bharathiar University, India. She is currently pursuing her Ph.D in computer science and has a total 13 years of teaching experience. At present she is a senior lecturer in the Department of computer science of Sri Ramakrishna college of Arts and Science for women, Coimbatore. She has presented research papers in international and national conferences and published papers in international journals. Her research interests include data mining and knowledge discovery, genetic algorithms, Bioinformatics, image compression, networking.
Weighted Fuzzy Mean Filters For Image Processing

A.Bharathi 1 K.AnandaKumar 2 A.Shanmugam 3

ABSTRACT
A New fuzzy filter for the removal of heavy additional impulse noise called the Weighted Fuzzy Mean (WFM) filter is proposed. In this WFM filter we generate five fuzzy sets for an image such as dark (DK), median (MD), bright (BR), very dark (VDK) and very bright (VBR). The WFM-filtered output signal is the mean value of the corrupted signals in a sample matrix and these signals are weighted by a membership grade of an associated fuzzy set stored in a knowledge base. The knowledge base contains a number of fuzzy sets decided by experts or derived from the histogram of a reference image. When noise probability exceeds 0.3, WFM gives very superior performance compared with conventional filters when evaluated by mean square error (MSE), peak signal-to-noise-rate (PSNR). In this mean filter we are using the method fuzzy logic. This splits the image into blocks. By comparing the pixel value in each block we can increase the resolution of the image based on the SNR values.

Keywords: Weighted fuzzy mean filter, Image processing, Knowledge base, Histogram, Impulse noise, fuzzy estimator

INTRODUCTION
When image is transmitted over channels, it is often corrupted by impulse noise due to faulty communications or noisy channels. Impulse noise consists of very large positive or negative spikes of short duration. A positive spike has a value much larger than those of background signals and appears like a bright spot on the image. On the other hand, a negative spike has a value much smaller than those of background signals and appears like a dark spot on the image. They both are easily detected by the eyes and degrade the image quality. Hence their removal is an important task in image processing. Although the generalized mean filter and nonlinear mean filter have been proposed for removing impulse noise from image, they suffer from an inability to remove positive and negative spikes simultaneously. The median filter is the best performing and most popular stack filter in heavy noise situation, but its performance deteriorates rapidly when spike probability exceed 50%. In this paper, we introduce a novel filtering technique with superior noise removal capacity compared to conventional nonlinear filter. Fuzzy set theory has been successfully applied to control and pattern recognition fields. It is suitable for dealing with problems containing high levels of uncertainty to which class pattern recognition or image processing problems usually belong. WFM is a powerful tool for removing additional impulse noise from images especially when noise probability is larger than 0.3.

KNOWLEDGE BASE SUPPORTED IMAGE NOISE REMOVAL PROCESS
Our WFM is supported by a simple knowledge base which can be a static knowledge base built by human experts or by referring to sample images or a dynamic knowledge base built? in run time. A 1-10 mins set-up time is initially required for WFM static knowledge base,
but this is an one-time process. The dynamic knowledge base is a constant automatic delay of less than 1s. It is equivalent to the knowledge base of our WFM is supported by a simple knowledge base which can be a static knowledge base built by human experts or by referring to sample images, or a dynamic knowledge base built in run time. A 1-10 mins set up time is initially required for WFM static knowledge base, but this is a one time process. The dynamic knowledge base has a constant automatic delay of less than 1s. its equivalent of knowledge base set-up.

The image transmission process is shown in fig 1. This process consists of two phases: sender phase and receiver phase. The sender phase sends the source image to the receiver. The static knowledge base is pre-established by image experts or by referring to sample image, which may be independent of the image to be filtered.

The proposed noise removal process uses a dynamic knowledge base. This process also consists of a sender phase and receiver phase but the knowledge base must be transmitted from the sender side to the receiver side along with the image to be filtered. Fig2 (a) describes the image transmission process when WFM is applied to remove noise.

**Figure 1: Transmission Process**

**Figure 2 (a): Noise Removal Process**

The generation of the dynamic or static base is as follows: consider a noise free image S sized N1xN2 pixels with L gray levels. For convenience we denote it as $S = \{s(i,j)\}_{N1xN2}$, where $s(i,j)$ is a pixel of the source image without any noise for $0 \leq i \leq N1-1$ and $0 \leq j \leq N2-1$. Then some fuzzy subsets defined on the universe of discourse $[0, L-1]$ can be built. Each of the Fuzzy subsets represents an abstract concept for the gray level of image pixels.

**Figure 2 (b): Sender Phase**

The sender phase is shown fig 2 (b). The knowledge base is updated with the source image after histogram processing. This phase also called knowledge base building phase the completed knowledge base consists of few fuzzy sets specifying the gray-level features of the noise free source image and is referred by WFM. When removing impulse noise during the receiver phase as shown in fig 2(c), the receiver phase filter receives corrupted images by invoking WFM and referring to the information stored in the knowledge base.

**Figure 2 (c): Receiver Phase**
Membership grade

Fig:3 Example pf membership functions for the fuzzy sets DK, MD, and BR the membership grade usually a value in the range \([0, 1]\) where “1” denotes a full membership and “0” denotes no membership. According to fig 3, we know that for instance, pixels \((i, j)\) with the gray level 160 has the intensity property: “not dark”, “quite median”, and “poorly bright”.

The fuzzy sets describing the intensity features of a noise free source image can be derived from the histogram of the source image, and they together constitute the proposed knowledge base. The histogram of a digital image with gray levels in the range \([0, L-1]\) is a discrete function. Where \(S_k\) is the kth gray level of image \(S\), \(n_k\) is the number of pixel with the kth gray level in \(S\), \(n\) is the total number of pixel in \(S\), and \(k=0,1,2,\ldots,L-1\). In other words \(p(\text{sk})\) gives an estimate of the probability of occurrence of gray level \(sk\). Before the algorithm for generating fuzzy sets to be stored in the knowledge base, membership function type has to be defined first.

**CONSTRUCTION ALGORITHM OF FUZZY SETS STORED IN KNOWLEDGE BASE**

**Step 1:** Decide the intervals of \([DK_{\text{begin}}, DK_{\text{end}}], [MD_{\text{begin}}, MD_{\text{end}}]\) and \([BR_{\text{begin}}, BR_{\text{end}}]\) for the fuzzy sets DK, MD and BR, respectively.

**Step 1.1:** Set \(DK_{\text{end}} = \left[ \frac{L-1}{N_f} \right], BR_{\text{begin}} = (N_f - 1)\) left overlap, and

\[MD_{\text{begin}} - DK_{\text{end}} - \text{left overlap}, \text{ and} \]

\[MD_{\text{end}} = BR_{\text{begin}} + \text{right overlap}, \text{ where } N_f \text{ is the number of fuzzy sets and left overlap and right overlap denote the overlapping range of the fuzzy sets.}\]

**Step 1.2:** Set \(DK_{\text{begin}}\) be the first \(s_k\) such that \(n_k > t\) from 0 to \(DK_{\text{end}}\), where \(t\) is a threshold;

**Step 1.3:** Set \(BR_{\text{end}}\) be the last \(s_k\) such that \(n_k > t\) from \(BR_{\text{begin}}\) to \(L-1\).

**Step 2:** Find a point \(s_k\) with the maximum value of \(p(\text{sk})\) in the interval of \([DK_{\text{begin}}, DK_{\text{end}}]\), then generate the membership function \(f_{DK}\) of fuzzy set DK by the following sub steps:

**Step 2.1:** \(m_{dk} \rightarrow S_k\),

**Step 2.2:** \(\alpha_{dk} \rightarrow M_{dk} - DK_{begin}\),

**Step 2.3:** \(\beta_{dk} \rightarrow DK_{end} - M_{dk}\).

**Step 3:** Find a point \(s_k\) with the maximum value of \(p(\text{sk})\) in the interval of \([MD_{\text{begin}}, MD_{\text{end}}]\), then generate the membership function \(f_{MD}\) of MD by the following sub steps:

**Step 3.1:** \(M_{md} \rightarrow S_k\).
Weighted Fuzzy Mean Filters For Image Processing

Step 3.2: $\alpha_{MD} \rightarrow M_{MD}$ - DK begin,

Step 3.3: $\beta_{MD} \rightarrow M_{MD}$ - MDend.

Step 4: Find a point $S$ with the maximum value of $p(S_i)$ in the interval of $[BR_{begin}, BR_{end}]$, then generate the membership function $f_{BR}$ of $BR$ by the following sub steps:

Step 4.1: $m_{BR} \rightarrow S$.

Step 4.2: $\alpha_{BR} \rightarrow BR_{begin}$.

Step 4.3: $\beta_{BR} \rightarrow BR_{end}$.

Step 5: Stop

Weighted Fuzzy Mean Filter

The proposed WFM is basically a mean filter operating with fuzzy members. Conventional mean filter are inefficient for heavy tailed additive noise, but WFM can remove such kind of noise efficiently and simply, WFM adopts a 3x3 sample window to determine the gray level value of each filtered signal, and the pixel to be filtered stands in the central cell of the sample window. Let $X(i,j)$ be the original input image and $Y(i,j)$ be the filtered output image, respectively. In the $X$ each entry $x(i,j)$ may be corrupted by noise $n(i,j)$ so that has the gray level $X(i,j) = s(i,j) + n(i,j)$

WFM decision process then determines the final filtered output of each pixel. This decision is made by referring to a fuzzy estimator which is derived from fuzzy interval stored in the knowledge base.

The maximum-likelihood estimator (MLE) has been widely used in statistics. Now we propose a new estimator called a fuzzy estimator (FE) which is similar to MLE but more powerful for the removal of noise.

**Definition**

A fuzzy interval $l$ is of L-R Type if there exists two shape function $L$ and $R$ and four parameters $(m_l, m_r)$.
\[ f_{LR}(x) = \begin{cases} 
L \left( \frac{m_1 - x}{\alpha} \right) & \text{for } x \leq m_1, \\
1 & \text{for } m_1 \leq x \leq m_1, \\
R \left( \frac{x - m_1}{\beta} \right) & \text{for } x \geq m_1. 
\end{cases} \]

**Definition**

If \( I \) is the fuzzy interval stored in the knowledge base, then a fuzzy estimator can be produced by following formula

\[ f_{LR}(X(i,j)) = \frac{\sum_{n_k = 0}^{N} \sum_{n_l = 0}^{N} f_{LR}((i+k,j+l),x(i+k,j+l))}{\sum_{n_k = 0}^{N} \sum_{n_l = 0}^{N} \sum_{n_j = 0}^{N} f_{LR}(x(i+k,j+l))} \]

Where \( X(i,j) \) is an \( n_1 \times n_2 \) sample matrix centered at the input pixel \( x(i,j) \). In our experiment we let the fuzzy interval \( I \) be follows:

\[ f_{LR}(x) = \begin{cases} 
0 & \text{for } x \leq m_1, \\
1 & \text{for } m_1 \leq x \leq m_1, \\
0 & \text{for } x \geq m_1. 
\end{cases} \]

**Decision process of WFM Filter**

**Begin**

If \( y_{DK}(i,j) - f_{LR}(X(i,j)) < y_{MD}(i,j) - f_{LR}(X(i,j)) \)

Then \( y(i,j) \leftarrow y_{DK}(i,j) \);

Else \( y(i,j) \leftarrow y_{MD}(i,j) \);

**If** \( y_{BR}(i,j) - f_{LR}(X(i,j)) < y(i,j) - f_{LR}(X(i,j)) \)

Then \( y(i,j) \leftarrow y_{BR}(i,j) \); **End**

**EXPERIMENT RESULT**

In our experimental noise model the source image is corrupted by additive impulsive noise with probability \( p \).

\[ x(i,j) = \begin{cases} 
s(i,j) + n(i,j), & \text{with probability } \frac{1}{2}p, \\
s(i,j) - n(i,j), & \text{with probability } \frac{1}{2}p, \\
s(i,j), & \text{with probability } 1 - p. 
\end{cases} \]

It should be remembered that the major drawbacks in the use of generalized mean and nonlinear mean filter for impulse noise removal is that they cannot remove positive and negative spikes, simultaneously. However median filter have been extensively used for mixed impulse noise removal. In this project comparison between WFM and the median filter are emphasized.
Some Conclusions Can Be Drawn From Our Experiments

1) WFM with FE is always better than WFM with MLE.
2) Conventional filters, including the median filter, are unsuitable for cases where additive impulse noise is heavy, especially when noise probability is larger than 30.
3) WFM shows good stability for the full range of impulse noise probability and for the MSE and PSNR criteria.
4) WFM removes mixed spikes noise very well but most conventional mean filter cannot.

Discussion and Conclusion
A new type filter WFM applying the fuzzy modeling technique is proposed and analyzed in this project. This major operation in WFM’s computational complexity is less than that of the median filter. Moreover according to the experimental results, the former is superior to the latter in terms of various noise removal evaluation criteria. A knowledge base supporting technique is also developed for WFM. The knowledge base can be either a static knowledge base produced by consulting experts or referring to a set of sample image or dynamic knowledge base produced dynamically from each clean source image. Certainly the dynamic knowledge base scheme has the best noise removal performance, but the extra needs overhead to generate and transmit the knowledge base for each image. Subjective evaluation of WFM also shows high quality restoration of filter image.

References


Author’s Biography

Dr. A. Shanmugam received his Bachelor of Engineering Degree from PSG College of Technology in 1972, Coimbatore, Master of Engineering Degree from College of Engineering, Guindy, Chennai in 1978 and Doctor of Philosophy in Electrical Engineering from Bharathiar University, Coimbatore in 1994. He was the Professor and Head of Electronics and Communication Engineering Department at the time of relieving. He is currently the Principal, Bannari Amman Institute of Technology, Sathyamangalam. His Professional activities include…Guided Eight Ph.Ds and guiding fifteen Ph.Ds in the field of Wired & Wireless Networks. Published and presented more than 125 Papers in International and National Journals and also in Conferences.

Mrs. A. Bharathi received her Bachelor of Engineering Degree from Kongu Engineering College in 1998, Perunudurai, Master of Engineering Degree from Bannari Amman Institute of Technology, Sathyamangalam, ai in 2007 and she is doing Doctor of Philosophy in Computer Science and Engineering from Anna University, Coimbatore. She had 10 years of teaching experience currently she is working as Assistant Professor, Department of IT, Bannari Amman Institute of Technology, and Sathyamangalam. His Professional activities include…Guided Ten UG projects and guiding Seven UG and Three PG projects. Published and presented 6 papers in International and National Conferences.

Mr. K. Ananda Kumar was born in TamilNadu, India on March 1975. He received the B.Sc Degree in Physics from Bharathiar University in 1995. He received his MCA Degree in Computer Applications from Bharathiar University in 1998. He received his M.Phil from Periyar University in 2006 and he is doing Doctor of Philosophy in Computer Science and Engineering from Bharathiar University, Coimbatore. He had 10 years of teaching experience. Currently he is working as HOD in Computer Applications Department, Dr.SNS Rajalakshmi College of Arts and Science College, Coimbatore. His Professional activities include…Guided Fifteen PG projects and Five M.Phil and guiding Seven PG and Three M.Phil projects. Published and presented 8 papers in International and National Conferences.
ABSTRACT
As per ISO 8402 standard, quality of software is the totality of features and characteristics of a product/service and its ability to satisfy stated/implied needs. Quality attributes possess the characteristics associated with the system and act as the measures to determine the quality of the software. Each quality attribute has been pertaining to specific feature or property of software and has unique effect. Software quality attributes contribute equally for successful software development even though all sub-characteristics do not equally affect software product and process. Though several issues related to the quality attributes and the respective classification have been discussed through variety of models and standards, each classification has cited with a specific view. Therefore a strong need has emerged to systematically categorize quality attributes in an integrated manner. This paper proposes a generalized classification of quality attribute that leads to provide support throughout the software development process to produce quality software in view of product, process, policy, plan and service.

K. Paithankar¹, M. Ingle²

I. INTRODUCTION
As per ISO 8402 standard, quality of software is the totality of features and characteristics of a product/service and its ability to satisfy stated/implied needs. Therefore, quality is defined to be a complex and multifaceted concept [1]. In the context of software engineering, software quality is a measurement of wellness of software design and conformance of the design so as to provide substantial benefit to the end users. Quality attributes possess the characteristics associated with the system and act as the measures to determine the quality of the software. Also, these attributes guide the software engineering process for conformance of all the aspects of software development. It is mandatory for any software development organization to produce quality software; particularly in this competitive era and quality attributes play a major role in determining the quality of software. Each quality attribute has been pertaining to specific feature or property of software and has unique effect. Hence, classification of quality attributes with different perspectives is required for the successful software development thereby providing guidance and ease to conform quality of product, process and service. We discuss various views of software quality attributes and existing classification models of software quality attributes in Section 2. Section 3 describes the proposed quality classification with six major classes based on some specific criteria. These classes of quality attributes have been further divided into subclasses and so on. Finally,
we conclude with the results and discussions in Section 4.

2. LITERATURE REVIEW

Many researchers have presented certain issues related to the quality attributes and the respective classification. As per International Standards Organization (ISO), quality standards cover all the aspects of quality. ISO 9126 is an international standard for the evaluation of software quality and has been covered with four major aspects namely quality model, external metrics, internal metrics and quality in use metrics respectively. Quality model has been described in ISO/IEC 9126-1:2001. ISO/IEC TR 9126-2:2003 has covered the external quality metrics. Internal metrics has been defined in ISO/IEC TR 9126-3:2003. ISO/IEC TR 9126-4:2004 used to define quality in use metrics. In addition ISO/IEC 25000:2005 has also been used as guide for Software product Quality Requirements and Evaluation (SQuaRE). As per ISO 9126-1, the classification of software quality explores in terms of a structured set of characteristics as functionality, reliability, usability, efficiency, maintainability and portability, which have further been divided into quality attributes.

There exists a variety of quality classification models such as McCall’s model, Boehm’s model, model of Software Assurance Technology Center (SATC), Software Quality Institute model, common subsets model etc. It is observed that there exist three important aspects of software product such as operational characteristics (also known as product operations), ability to undergo changes (product revision) and adaptability to new environment (product transition) [2]. Operations aspect of the system covers correctness, reliability, usability, integrity and efficiency whereas maintainability, flexibility and testability considered being revision aspect elements of a system. Portability, reusability and interoperability attributes deal with adaptability of system. Another model represents a multilevel classification of quality attributes [3] [4]. General utility is at the root level representation which has further been expanded through quality factors maintainability, as-is utility and portability.

Maintainability quality factor covers sub-factors modifiability, understandability and testability. Human engineering, efficiency and reliability have been encapsulated within as-is-utility whereas device independence and self-containedness considered being the sub-factors of portability. Further subdivision of these quality sub-factors includes many crosscutting attributes. The model views the decomposition of quality attributes into source code characteristics. The quality can be viewed in terms of goals such as requirement quality, product quality, testing effectiveness and implementation effectiveness [5]. Ambiguity, completeness, volatility, understandability, traceability, structure, maintainability, reusability, internal/ external documentation, resource usage, completion rates and correctness are the attributes identified to achieve the stated goals. One improvement in ISO-9126 classification includes reusability as the top level attribute with additional subordinate properties. Functionality, reliability, efficiency, usability, maintainability, portability and reusability are defined as the primary classes of attributes. Further subdivision of these classes includes the attributes to be used in other environments [6]. The quality attributes in the form of subsets may be applicable for a specific phase of software development. A group of certain attributes such as Reliability, Availability, Serviceability, Usability, and Installability (RASUI) has been referred for effectiveness of the system, whereas a collection of attributes such as Functionality, Usability, Reliability, Performance, and
Supportability (FURPS) has been proved important for software requirements conformance. Reliability, Availability, Scalability, and Recoverability (RASR) is another set of quality attributes recommended for assessing quality of the database. Some quality attributes such as Reliability, Availability, Maintainability and Safety (RAMS) are clustered together to deal with safety-critical systems [2].

Our literature survey reveals that each classification has cited with a specific view. Coverage of limited attributes and their usage have been primarily observed limitations. Further, it has been noticed that software quality attributes contribute equally for successful software development even though all sub characteristics do not equally affect software product and process [7]. Also, certain quality attributes such as functionality, reliability, availability and usability have been common in most of the classifications, but have been composed in different manner. In addition, many more quality factors are identified which have not yet been classified relevantly [8] [9]. In the perspective of literature study, there exists a scope of generalized classification of attributes with systematic categorization of attributes in an integrated manner.

3. Proposed Classification

We present a generalized classification of quality attributes with six major classes namely; Runtime attributes, Non-runtime attributes, Business Oriented attributes, Architecture Oriented attributes, Domain Specific attributes and Impact Oriented attributes in view of product, process and service as shown in Fig. 1. We describe each of these classes and its further classification as follows:

### Runtime Attributes

This class of quality attributes is strictly concerned with the execution time. These attributes are required to be measured at the time of system execution. At run time it is important to know about the working of the software and its behavior. It highlights on the fundamental process of transformation that software and hardware components of the system perform on inputs to produce outputs. For example response of the system is considered to be of great importance and hence at the time of user-system interaction the time and functional aspects of the software have to be thought about. Therefore, the quality attributes concern with the software in this manner has taken up in runtime class. As shown in Fig. 2, there exists secondary attributes such as security, functionality, interoperability performance, accessibility, usability, availability, and traceability that are used to assess the run time behavior of the system. Security is the first sub-class deals with the ability of the system to resist unauthorized attempts of usage/ behavior modification while providing service to users. It protects information from theft or corruption, or the preservation of availability. One more class of runtime attributes is interoperability. It is the ability of system to cooperate with other systems while in execution and hence concerned with run time behavior of the system. Yet another class of run-time attributes is accessibility, which is the ability to access the functionality of the system, and to get possible benefit of the system, when the system is in operation. Traceability is the ability to verify the history, location, or application of an item by means of documents at the time of system execution.

Functionality is another sub-class of runtime attributes and is the ability of the system to accomplish intended work. It has been further classified into understandability,
Understandability refers to the capability of system being understood and accepted under the circumstances to accomplish any task. Conciseness is concerned with expressing more in few words. Consistency is the property of uniformity of successive results or events and is considered as part of functionality. Clarity refers to the ability of the system that clearly visualizes the concepts, as in thought, understanding and mind. Another subclass of run time attributes is performance which is the ability of the system to be timely used as desired. It has been further classified into attributes such as utility, response time, throughput behavior of the system, timeliness and structure (URTTS). Utility is the capacity of the system to work as needed. Response time is the time elapsed from submitting an instruction till the first response of the system. Throughput behavior of the system is defined as the amount of work done in a given time. To improve performance of the system maximum throughput is expected. Timeliness being the property of on time plays a major role for better system performance. To get the desired performance of the system, sufficient storage of data and results has necessarily to be provided. Subsequently usability is another subclass of run time attributes which is a measure of convenience and practicability of the product for intended users. Ease of use and ease of training to the end users pertaining to the system proved to be of great importance. Thus, usability has to be conformed at run time. It has further been subdivided into set of attributes working to have usable software namely; learnability, efficiency, access control, unambiguity, validity, resilience, customizability, practicability and operability (LEAUVRCPO). Learnability is the ability to know the details of the system without any external support. Efficiency being the property of a system to fulfill the requirement perfectly in short runs of time desired to be attribute of usability. Access control deals with the protection and security of the system when in use. Repeatability defined as the variability of the measurements obtained by one person while measuring the same item repeatedly. Another attribute in usability subset is unambiguity that is the ability of the system being interpreted in the same manner by all the users. It makes the users understand the intended purpose of the system. Validity refers to the logical, analytical or necessary trueness of the system at the time of use. Resilience is considered to be usability attribute being the property of system to energize itself when deformed elastically and then, recover. The software is used by many users for different purposes and satisfaction of the users is important. Customizability is the ability of the software to be changed by the user or programmer as per the need and to provide satisfaction. Practicability is the ability that makes the software usable for a specified purpose. Operability is the major aspect of usability and defined as ability to keep a system in a functioning and operating condition.

Availability is also a subclass of runtime quality attributes and described as the measure of time when the system is up and running correctly. It is the elapsed time between failures and the time needed to resume operation after a failure. Availability branches into set of attributes such as reliability, sustainability and anomaly management (RSA). Reliability is essential to confirm availability since it is the capability of software to maintain its level of performance under stated conditions for a stated period of time. Sustainability also plays an important role to measure availability as it is the property to uphold the system for the required function. Anomaly management deals with anomalies that may exist while system is running.
Non-runtime Attributes

Attributes of this class do not concern with run time behavior of the system but play a major role in determining quality in offline manner. Also, it has been noticed that the conformance of run time quality depends on the conformance of non-run time quality. It specifies criteria that can be used to judge the operation of a system and the way the system will do it. For example, software performance requirements, software external interface requirements, software design constraints are considered to be nonfunctional requirements of the system and are embodied in the static structure of the software system. Nonfunctional requirements are difficult to test; therefore, they are usually evaluated subjectively with overall characteristics. Fig. 3 depicts such attributes that can evaluate non-runtime behavior of the system namely; documentation, manageability, completeness, portability, accuracy, integrity, reusability and testability.

The documentation sub-class of non-runtime attributes refers to the process of preparing and providing evidences in the form of communicable material for system investigation. Successful system development has the essential requirement of well documentation. Completeness is concerned with implementing all the capability in terms of sufficient data items, functions, interface and code and hence it is considered as a class of non-runtime attributes. The sub-class accuracy has been defined as the degree of conformity of a measured or calculated quantity to its actual value. Integrity is also a non-runtime attribute and defined as the ability to separately develop components and make them work together correctly. Components are functioning in a specific manner to achieve objectives, at the same time collectively working to achieve a common goal of the system. Yet another class of non-runtime attributes is testability and has given a great importance as it provides error handling capability to the system. Testing of the system has to be accomplished prior to the execution of the system, thus testability is the major non-runtime attribute.

Manageability is also a sub-class of non-runtime attributes and is the ability of the system to plan, organize resources, direct, administrate and control the overall process of development. Manageability has further extended in a group of attributes, modifiability, maintainability and flexibility (MMF) with equal share of each attribute for the intended purpose. Modifiability is the ease with which a software system can accommodate required changes. Maintainability facilitates updates to satisfy new requirements. The software product that is maintainable is simple, well-documented, and should have spare capacity for processor and memory usage. Flexibility is defined as the ability to adapt to different circumstances.

Another sub-class of non-runtime attributes is portability. Portability is the ability of a system to run under different computing environments. The environment is a combination of hardware and software both. A set of attributes, machine independence, system independence, replaceability, installability, adaptability and data commonality has been covered under portability and abbreviated as MSIADR. Machine independence refers to the ability of the software to be used on any machine (i.e. hardware). System independence is the ability of the software to be installed, operated and modified on any system (i.e. hardware and operating system). Presence of these two attributes generates possibility of installability which is the capability of the software product to be installed in a specified environment. Installability is further extended to distributability that is the ability to share the common resources for applications.
and users. Adaptability leads to improve portability as it is the ability of the system to be modified by circumstances. Data commonality refers to the availability of common data for all the users and applications. Replaceability is the capability of the system to retain itself after replacement of data structure, function, module or program.

Reusability is referred to be the degree to which existing applications can be reused in new applications and hence identified as a sub-class of non-run time attributes. It has further been classified into a cluster of attributes recommended to measure reusability, such as representation independence, application independence, data encapsulation, function encapsulation and interfaceability (RADFI). For reuse, system has to be independent of representation and application used. Data and function encapsulation are necessary for reusability as it refers to hiding the details about data and function. A major characteristic of reusability is interfaceability, which deals with exchanging information across the common boundary shared by the components.

Business Oriented Attributes

There exist many non-software attributes that influence other software or non-software quality attributes. These attributes, attempt to conform quality in view of business policies. Business objectives are specific statements that give projections about growth or development to companies. these are the stated, measurable targets to achieve business aims. The effective business objectives have to be specific, measurable, agreed realistic and time specific. For example, a business objective could be to increase sales of the product by next year. A combination of attributes such as cost and schedule, economy, marketability, appropriateness for organization, and localization are observed to be quality attributes that affect business system as highlighted in Fig. 4. Cost and schedule is the ability to account cost of the system with respect to time and market, expected project lifetime, and utilization of systems. It plays significant decision making role for planning and executing business policies. Another class of business oriented attributes has termed as marketability. It is the capability to use the system with respect to market competition and to use feedback to increase profitability of the business.

Yet another sub-class of business oriented attributes is appropriateness for organization that has been explored for availability of the human input, allocation of expertise, and alignment of team and software structure. Also, it facilitates business process re-engineering to provide adaptable system. Generality and commonality (GC) are the factors that contribute to measure appropriateness of organization. Generality refers to the availability of the system for the majority of people while commonality is the ability of the software to be used with common features for distinct users. Localization is stated to be the major sub-class of business oriented attributes as any business system has to be localized to realize benefits.

Architecture Oriented Attributes

The quality attributes used to measure structural aspect of the system are known as architecture oriented attributes. There are many common ways of designing computer software modules and their communications. For example, blackboard, Client-server, Database-centric architecture, Distributed computing, Event Driven Architecture, Implicit invocation, Monolithic application, Peer-to-peer, Pipes and filters, Plug-in, Representational State, Transfer, Structured (module-based), Service-oriented, Search-oriented, Space based, Shared nothing, Three-tier model are some of the available architectures.
can be used while system development. These are also non software attributes. Fig. 5 illustrates attributes that are considered for assurance of the quality of structural design of the system namely; conceptual integrity, correctness, structured, scalability, extensibility, supportability and self-containedness. Conceptual integrity has been defined as the integrity of the overall structure composed from variety of small architectural components. Therefore, it plays a vital role for building quality software. Accountability to satisfactorily fulfill all requirements of the system is expressed through correctness. Realization of requirements has dependency on design; hence correctness has been covered under architecture oriented attributes. The structured aspect of the system is the capacity to organize activities in well manner. Scalability deals with handling growing amounts of work in a graceful manner, or to be readily enlarged. Extensibility is a system design principle where the implementation takes future growth into consideration. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Another class of architecture oriented attributes is supportability that has been defined as affectivity of system structure to provide service to the intended users. It has been proved that the basis of a desired system is a good architecture and hence essential to provide support of use. Self-containedness is the ability of the system to contain its projected purpose in its definition and hence considered as architecture oriented attribute.

**Domain Specific Attributes**

This class of quality attributes deals with the specific business and application domain. The objective of defining the domain is developing mass-customized products that reduce the costs, delays, and inflexibility characteristic of software and systems. Also it increases the ability to align business activities to the needs and produce solutions to cater the specific needs. It has further been categorized in three sub-classes that can contribute for the stated purpose such as sensitivity, calibrability and stability as represented in Fig. 6.

Sensitivity is described as the degree to which a system component can pick up something being measured. It has dependency on the area of application and development. Further, calibrability is defined as the ability of the system to recalibrate itself to some specific working range. It has been associated with unique standards and measurements against the standards for evaluating product and process. Stability is the sub-class of domain specific attributes and is the degree to which software can be run over periods of time without crashing or otherwise malfunctioning. It has termed as a characteristic of application area and the plateform used to build the system.

**Impact Oriented Attributes**

This class of quality attributes is mainly concerned with determining the overall effect of the system from users view point. User satisfaction is the primary goal of any software and has to be cultivated at conception stage of the system. Therefore, software has to be available in such a manner that influences the user without a need of redressal. Hence, Simplicity, communicativeness, maturity, self-descriptiveness, fault tolerance, effect and helpfulness are the major factors contribute to evaluate this aspect of the system as illustrated in Fig.-7.

One class of impact oriented quality attributes is simplicity that is defined as the property of being simple or uncombined i.e. the availability of the system without complexity. Also, a simple system is easy to understand
and use, and hence proved effective. Another class of impact oriented attributes is maturity. It is the state of the system being sensible and fully grown. Hence, maturity has considered to be directly proportional to effectivity. Yet another class of attributes is fault tolerance that is the ability of a system to continue performing operations properly, even in case of failure of some of its components. Since it has been desired for proper functioning of the system, it is considered to be the impact attribute. Installability is also a sub-class of impact oriented attribute as it is the capability of the system of being loaded and used effectively. Affect is the measurement of affectivity of the system over organization wide operations. Any decision has effect over the organizational process. Therefore, affect has been considered to have significant share in view of totality of the system. Communication is another class of impact attributes as it is the ability to convey desired and expected information to the intended users, organizational process or within the system. Finally, we have a class of impact oriented attributes as self descriptiveness. It is interpreted as the capability of the system to sufficiently describe itself and hence shows suitability with impact oriented class.

We consider Helpfulness as further sub-class of impact oriented attributes. It is the degree with which system is ready to help users and hence justify itself to be the attribute of this category. Helpfulness is further classified into a pair of attributes, visibility and survivability and is denoted by VS. Visibility is the transparency of system in context of interface, data and communication. Survivability refers to the ability of the system to persist in the same mode till use.

4. RESULTS AND DISCUSSION
In the proposed classification, we have made efforts to reconcile the limitations of the existing classification approaches with the stated aim. Certain attributes such as functionality, performance, usability, portability, reusability have been reorganized in perspective of suitability for improvisation of quality in respective class. Also, the proposed quality classification exclusively focuses on business and economy view of the system. In addition, impact oriented attributes have been incorporated to extend the overall quality of the system. Thus, the classification proves to be beneficial to produce quality software, as it is a unified approach to make use of the attributes for the product, process, policy, plan and service. The attributes have explicitly classified with defined role, but does not account for the crosscutting attributes. Therefore, in view of crosscutting attributes, still there remains a scope of further improvements.

REFERENCES
Griffith University, Nathan, Brisbane, QLD 4111 AUSTRALIA, Software Quality Institute 09/03/98.


Author’s Biography

Dr. Maya Ingle is Professor & Senior System Analyst at School of Computer Science and Information Technology, Devi Ahilya University, Indore since 22 years. She is also Chair Assessment & Research committee. She is Ph. D. in computer science and M. Tech in Computer Science from IIT, Kharagpur. Her research interest includes Software Engineering, NLP, Speech Recognition, Usability Engineering and Agile Computing. She has published more than 60 papers in national and international journals and conferences.

Kshama Paithankar is Assistant Professor at Shri Vaishnav Institute of Technology & Science, Indore. She is pursuing Ph. D. in computer science and has 14 years of teaching experience. Her research interest includes Software Engineering and Usability Engineering. She has published more than 05 papers in national journals and conferences.

Figure 1: Major Quality Attribute Classes
Quality Run Time Attributes

- Understandability
- Conciseness
- Consistency
- Clarity
- Utility
- Response timing
- Throughput behavior
- Timeliness
- Storage
- Reliability
- Sustainability
- Anomalousness
- Learnability
- Efficiency
- Access control
- Usability
- Validity
- Robustness
- Comprehensibility
- Practicability
- Operability
- Repeatability

Quality Non-Run Time Attributes

- Documentation
- Comprehensibility
- Accuracy
- Portability
- Functionality
- Manageability
- Mortality
- Maintainability
- Flexibility
- Machine independence
- System independence
- Interoperability
- Data commonality
- Replaceability
- Modifiability
- Maintainability
- Flexibility
- Representation independence
- Application independence
- Data encapsulation
- Function encapsulation
- Interfaceability

Usability

- Usability
- Testability
- Distributability
- Completeness
- Accuracy
- Integrity
- Reusability
- Documentation
- Testability
- Distributability
- Completeness
- Accuracy
- Integrity
- Reusability
- Documentation

Traceability

- Repeatability
- Customizability
- Practicability
- Operability
- Repeatability
- Customizability
- Practicability
- Operability
- Repeatability
- Customizability
- Practicability
- Operability

Figure 2: Run Time Attributes

Figure 3: Non Run Time Attributes
Generalized Classification of Software Quality Attributes

Quality

Business Oriented Attributes

- Cost & Schedule
- Economy
- Marketability
- Localization
- Appropriateness of Organization
  - Generality
  - Commonality

(A)
Figure 4 : Business Oriented Attributes

Architecture Oriented Attributes

- Conceptual Integrity
- Supportability
- Correctness
- Structured
- Scalability
- Self-containedness
- Extensibility

Figure 5 : Architecture Oriented Attributes
Quality

Domain Specific Attributes

Stability
Calibrability
Sensitivity

Impact Oriented Attributes

Simplicity
Maturity
Affect
Fault tolerance
Communicativeness
Helpfulness
Self-descriptiveness

• Visibility
• Survivability

Figure 6: Domain Specific Attributes

Figure 7: Impact Oriented Attributes
QoS in Wireless Sensor Networks – Survey

V. Vanitha1 D. Palanivel Rajan2

ABSTRACT
Sensor networks are distributed networks made up of small sensing devices equipped with processors, memory, and short-range wireless communication. They differ from traditional computer networks in that they have resource constraints, unbalanced mixture traffic, data redundancy, network dynamics, and energy balance. Work within wireless sensor networks (WSNs) Quality of Service (QoS) has been isolated and specific either on certain functional layers or application scenarios. However, the area of sensor network quality of service (QoS) remains largely open. In this paper, we examine and discuss the requirements, challenges of handling of QoS traffic and open research issues on QoS management in WSN and the various approaches for obtaining the QoS in WSN.

1. INTRODUCTION
Wireless sensor networks are a new class of distributed systems that are an integral part of the physical space they inhabit. Unlike most computers, which work primarily with data created by humans, sensor networks reason about the state of the world that embodies them. This bridge to the physical world has captured the attention and imagination of many researchers, encompassing a broad spectrum of ideas, from environmental protection to military applications. Recent advances in wireless communications and electronics have enabled the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate in short distances. These sensor nodes consist of sensing, data processing, and communicating components and sensor networks represent a significant improvement over traditional sensors. A sensor network is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. Although many protocols and algorithms have been proposed for traditional wireless ad hoc networks, they are not well suited to the unique features and application requirements of wireless sensor networks (WSNs). WSNs fundamentally differ from traditional wireless networks because WSNs devices have limited

1Assistant Professor, Department of Computer Science and Engineering, Kumaraguru College of Technology, Anna University, Coimbatore, Tamil Nadu, India. Email: vanithajayaprakash@yahoo.com
2Student, II M.E (CSE), Department of Computer Science and Engineering, Kumaraguru College of Technology, Anna University, Coimbatore, Tamil Nadu, India. Email: Palanivelrajan.d@gmail.com
capabilities, tight resource capacities, and are typically deployed in high densities with unpredictable distribution in dynamic and often harsh environments. All physical characteristics add to the complexity of determining whether certain Quality of Service (QoS) requirements can actually be met for WSNs’ application. To meet application level QoS requirements in actual WSNs deployments, many research have been focused on WSNs’ QoS. But in WSNs QoS has been isolated and specific either on certain functional layers or application scenarios. Although such research is beneficial to the specific cases that it investigates, it may not be considered all the elements that affect the required QoS performance. This will become even more apparent as we analysis WSN capabilities in higher levels of sophistication, and the interaction complexity among a WSN component functionalities increases.

Energy-aware network management will ensure a desired level of performance for the data transfer while extending the life of the network. Energy constraints combined with a typical deployment of large number of sensor nodes have necessitated energy-awareness at most layers of networking protocol stack including network and link layers. Current research on routing in wireless sensor networks mostly focused on protocols that are energy aware to maximize the lifetime of the network, are scalable to accommodate a large number of sensor nodes, and are tolerant to sensor damage and battery exhaustion. Energy-aware routing can optimize the transmission energy, while collision avoidance and minimization of energy consumed by the receiver can be achieved via energy-efficient medium access control (MAC) mechanisms. Since such energy consideration has dominated most of the research in sensor networks, the concepts of latency, throughput and delay jitter were not primary concerns in most of the published work on sensor networks. However, the increasing interest in real-time applications along with the introduction of imaging and video sensors has posed additional challenges. Such performance metrics are usually referred to as quality of service (QoS) of the communication network. Energy-aware QoS routing in sensor networks will ensure guaranteed bandwidth (or delay) through the duration of a connection as well as providing the use of the most energy efficient path. In this paper, we analyze the requirements of QoS, the system architecture design issues for sensor networks, challenges of supporting QoS in traffic at the network and also about the various algorithms for obtain QoS in WSN.

2. QoS REQUIREMENTS FOR WSN

In this section we outline WSN QoS requirements for several layers which we refer to the OSI 7-Layers as in traditional networks. For each layer, we give the definition of QoS requirements.

2.1 Application Layer

The QoS requirements in the application layer are specified by users. We define the following QoS requirements for WSNs applications: System Lifetime, Response Time, Data novelty, Detection Probability Data Reliability and Data Resolution. WSNs are often required to sustain its functionalities for a certain time period. System Lifetime is defined as the time from system deployment up to the time when it cannot satisfy users’ requirements. In on-demand WSNs applications, Response Time refers to the latency from the time that a user sends a query to the time that the user receives the response. Data novelty refers to the latency from the time an event is detected by a sensor to the time the data about the event arrive at storage sensors or sink points. In addition to data transmission, WSNs also needs to
monitor phenomena in the real world and generate sensing data. Detection Probability refers to the probability that a real world phenomenon can be detected and reported to a user. Two requirements on data quality, Data reliability and Data Resolution refer to the degree that the reported data corresponds to real world phenomena and the sampling rate in the spatial/temporal scale, respectively. Data reliability describes the accuracy of the data and Data Resolution imposes temporal/spatial granularity on the data.

2.2 Transport Layer
The QoS requirements for the transport layer: Reliability, Bandwidth, Latency and Cost. For convenience of discussion, differentiate the concept of packets and unique packets are differentiated as defined by the “collective” concept introduced in from a sensor’s point of view, unique packets refer to the packets containing data that are not correlated with the already received data. All of the QoS requirements within the transport layer use the collective concept, which means only unique packets are counted as received by the destination. Reliability refers to the percentage of unique packets successfully received from all sending sources in reference to those that were actually transmitted. Bandwidth refers to the number of unique packets received per unit time from all sending sources. Latency refers to the shortest total delay at the intermediate nodes/channels in transmitting a unique packet from all sending sources to a destination, which includes propagation delay, queuing delay, routing delay, etc. Cost is defined as the number of transmissions to retrieve a unique packet from all sending sources.

2.3 Network Layer
The following QoS requirements of the network layer: Path Latency, Routing Maintenance, Congestion Probability, Routing Robustness and Energy Efficiency. Path Latency refers to the average number of hops between all source destination pairs in the network. Routing Maintenance refers to the energy consumption rate to maintain routes between all source destination pairs. Congestion Probability is the probability that the traffic load on any path exceeds the bottleneck capacity of all the links on the path. Routing Robustness is defined as the maximal probability of packet loss that routing must sustain. Energy Efficiency measures the amount of energy consumed to transmit a data packet along a path.

2.4 Connectivity Maintenance Layer
The QoS requirements for the connectivity maintenance layer: Network Diameter, Network Capacity, Average Path Cost, Connectivity Robustness, and Connectivity Maintenance. Network Diameter is defined as the maximal transmission latency between two sensors in the formed network topology. Network Capacity is defined as the number of packets that can be transmitted concurrently in the network. Average Path Cost is defined as the average amount of energy consumed to transmit one packet between all source destination pairs. Connectivity Robustness is defined as the maximal allowed number of failed sensors/links that the network connectivity must sustain. Connectivity Maintenance measures the energy consumption rate to maintain a connected network topology.

2.5 Coverage Maintenance Layer
The QoS requirements for the coverage maintenance layer: Coverage Percentage, Coverage Reliability, Coverage Robustness and Coverage Maintenance. Coverage Percentage refers to the percentage of area monitored by at least one sensor. Coverage Reliability refers to the minimal allowed sensing probability. Coverage Robustness is defined as the minimal number
of sensors monitoring the same location. Coverage Maintenance measures the number of messages exchanged to provide and maintain network coverage.

2.6 MAC Layer
The QoS requirements for the MAC layer: Communication Range, Throughput, Transmission Reliability and Energy Efficiency. Communication Range refers to the maximal distance of one-hop data transmission. Throughput refers to the maximal number of data frames that can be transmitted successfully by the MAC layer per unit time. Transmission Reliability refers to the percentage of successfully transmitted frames. Energy Efficiency measures the amount of energy consumed to successfully transmit one frame within one-hop.

2.7 Physical Layer
The physical layer describes wireless sensor capabilities, which encompass wireless unit capabilities, processor capabilities and sensing unit capabilities. Specifically, wireless unit capabilities refer to Channel Speed, Coding and RF Power. Processor capabilities are Location capabilities, Timing capabilities, Processing Speed and Computation Power. Sensing unit capabilities include Measurement Accuracy, Sensing Range and Sensing Power. A sensor’s physical capabilities impose resource constraints on the QoS requirements of other layers. For wireless unit capabilities, Channel Speed impacts Throughput in the MAC layer; Coding impacts Throughput and Transmission Reliability in the MAC layer; RF Power impacts Communication Range, Transmission Reliability and Energy Efficiency in the MAC layer. For processor capabilities, Location and Timing capabilities impact Location Accuracy and Timing Accuracy respectively; Processing Speed determines Processing Latency in the data management layer; Computation Power impacts Computation Cost, Data Abstraction and Data Accuracy in the data management layer, as well as Energy Consumption in the location/time service layer. For sensing unit capabilities, Measurement Accuracy impacts Coverage Reliability and Coverage Robustness in the coverage maintenance layer; Sensing Range affects Coverage Percentage in the coverage maintenance layer; Sensing Power affects all the requirements in the coverage maintenance layer.

3. Design Issues
Depending on the application, design goals/constraints have been considered for sensor networks. Since the performance of a routing and MAC protocols are closely related. A summary of design issues is given in Table 1.

3.1 Network Dynamics
There are three main components in a sensor network. These are the sensor nodes, sink and monitored events. Aside from the very few setups that utilize mobile sensors, most of the network architectures assume that sensor nodes are stationary. On the other hand, supporting the mobility of sinks or cluster-heads (gateways) is sometimes deemed necessary. Routing messages from or to moving nodes is more challenging since route stability becomes an important optimization factor, in addition to energy, bandwidth etc. The sensed event can be either dynamic or static depending on the application. Monitoring static events allows the network to work in a reactive mode, simply generating traffic when reporting. Dynamic events in most applications require periodic reporting and consequently generate significant traffic to be routed to the sink.

3.2 Node Deployment
Another consideration is the topological deployment of nodes. This is application dependent and affects the performance of the routing protocol. The deployment is
either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data is routed through pre-determined paths. In addition, collision among the transmissions of the different nodes can be minimized through the pre-scheduling of medium access. However in self-organizing systems, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner. In that infrastructure, the position of the sink or the cluster-head is also crucial in terms of energy efficiency and performance. When the distribution of nodes is not uniform, optimal clustering becomes a pressing issue to enable energy efficient network operation.

3.3 Node Communications
During the creation of an infrastructure, the process of setting up the routes is greatly influenced by energy considerations. Since the transmission power of a wireless radio is proportional to distance squared or even higher order in the presence of obstacles, multi-hop routing will consume less energy than direct communication. However, multi-hop routing introduces significant overhead for topology management and medium access control. Direct routing would perform well enough if all the nodes were very close to the sink. Most of the time sensors are scattered randomly over an area of interest and multi-hop routing becomes unavoidable. Arbitrating medium access in this case becomes cumbersome.

3.4 Data Delivery Models
Depending on the application of the sensor network, the data delivery model to the sink can be continuous, event-driven, query-driven and hybrid. In the continuous delivery model, each sensor sends data periodically. In event-driven and query-driven models, the transmission of data is triggered when an event occurs or a query is generated by the sink. Some networks apply a hybrid model using a combination of continuous, event-driven and query-driven data delivery. The routing and MAC protocols are highly influenced by the data delivery model, especially with regard to the minimization of energy consumption and route stability. For instance, it has been concluded that for a habitat monitoring application where data is continuously transmitted to the sink, a hierarchical routing protocol is the most efficient alternative. This is due to the fact that such an application generates significant redundant data that can be aggregated on.

<table>
<thead>
<tr>
<th>Design Issue</th>
<th>Primary Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Dynamics</td>
<td>mobility of node, target, and sink</td>
</tr>
<tr>
<td>Node Deployment</td>
<td>Deterministic or ad Hoc</td>
</tr>
<tr>
<td>Node Communications</td>
<td>Single-hop or multi-hop</td>
</tr>
<tr>
<td>Data Delivery Models</td>
<td>continuous, event-driven, query-driven, or hybrid</td>
</tr>
<tr>
<td>Node Capabilities</td>
<td>multi- or single function; homogeneous or heterogeneous capabilities</td>
</tr>
<tr>
<td>Data Aggregation/Fusion</td>
<td>in-network (partially or fully) or out-of-network</td>
</tr>
</tbody>
</table>

In addition, in continuous data delivery model time-based medium access can achieve significant energy saving since it will enable turning off sensors’ radio receivers CSMA medium access arbitration is a good fit for event-based data delivery models since the data is generated sporadically.

3.5 Node Capabilities
In a sensor network, different functionalities can be associated with the sensor nodes. In early work on sensor networks, all sensor nodes are assumed to be homogenous, having equal capacity in terms route to the sink, thus reducing traffic and saving energy.
of computation, communication and power. However, depending on the application a node can be dedicated to a particular special function such as relaying, sensing and aggregation since engaging the three functionalities at the same time on a node might quickly drain the energy of that node. Some of the hierarchical protocols proposed in the literature designate a cluster-head different from the normal sensors. While some networks have picked cluster-heads from the deployed sensors, in other applications a cluster-head is more powerful than the sensor nodes in terms of energy, bandwidth and memory. In such cases the burden of transmission to the sink and aggregations are handled by the cluster-head.

3.6 Data Aggregation/Fusion
Since sensor nodes might generate significant redundant data, in some applications similar packets from multiple nodes can be aggregated so that the number of transmissions would be reduced. Data aggregation is the combination of data from different sources by using functions such as suppression (eliminating duplicates), min, max and average. Some of these functions can be performed either partially or fully in each sensor node, by allowing sensor nodes to conduct in-network data reduction. Recognizing that computation would be less energy consuming than communication, substantial energy savings can be obtained through data aggregation. This technique has been used to achieve energy efficiency and traffic optimization in a number of routing protocols. In some network architectures, all aggregation functions are assigned to more powerful and specialized nodes. Data aggregation is also feasible through signal processing techniques. In that case, it is referred as data fusion where a node is capable of producing a more accurate signal by reducing the noise and using some techniques such as beams forming to combine the signals. Data aggregation makes medium access control complex since redundant packets will be eliminated and such elimination will require instantaneous medium access arbitration. In such case, only CSMA and CDMA-based MAC protocols are typically applicable leading to an increase in energy consumption.

4. QoS Challenges In Sensor Networks
While sensor networks inherit most of the QoS issues from the general wireless networks. The following is an outline of design considerations for handling QoS traffic in wireless sensor networks:

4.1 Bandwidth Limitation
A typical issue for general wireless networks is securing the bandwidth needed for achieving the required QoS. Bandwidth limitation is going to be a more pressing issue for wireless sensor networks. Traffic in sensor networks can be burst with a mixture of real-time and non-real-time traffic. Dedicating available bandwidth solely to QoS traffic will not be acceptable. A trade-off in image/video quality may be necessary to accommodate non-real-time traffic. In addition, simultaneously using multiple independent routes will be sometime needed to split the traffic and allow for meeting the QoS requirements. Setting up independent routes for the same flow can be very complex and challenging in sensor networks due energy constraints, limited computational resources and potential increase in collisions among the transmission of sensors.

4.2 Removal of Redundancy
The sensor networks are characterized with high redundancy in the generated data. For unconstrained traffic, elimination of redundant data messages is somewhat easy since simple aggregation functions would suffice. However, conducting data aggregation for QoS traffic is much more complex. Comparison of images and
QoS in Wireless Sensor Networks – Survey

video streams is not computationally trivial and can consume significant energy resources. A combination of system and sensor level rules would be necessary to make aggregation of QoS data computationally feasible. For example, data aggregation of imaging data can be selectively performed for traffic generated by sensors pointing to same direction since the images may be very similar. Another factor of consideration is the amount of QoS traffic at a particular moment. For low traffic it may be more efficient to cease data aggregation since the overhead would become dominant. Despite the complexity of data aggregation of imaging and video data, it can be very rewarding from a network performance point-of-view given the size of the data and the frequency of the transmission.

Energy and delay trade-off: Since the transmission power of radio is proportional to the distance squared or even higher order in noisy environments or in the non-flat terrain, the use of multi-hop routing is almost a standard in wireless sensor networks. Although the increase in the number of hops dramatically reduces the energy consumed for data collection, the accumulative packet delay magnifies. Since packet queuing delay dominates its propagation delay, the increase in the number of hops can, not only slow down packet delivery but also complicate the analysis and the handling of delay-constrained traffic. Therefore, it is expected that QoS routing of sensor data would have to sacrifice energy efficiency to meet delivery requirements. In addition, redundant routing of data may be unavoidable to cope with the typical high error rate in wireless communication, further complicating the trade-off between energy consumption and delay of packet delivery.

4.3 Buffer Size Limitation

Sensor nodes are usually constrained in processing and storage capabilities. Multi-hop routing relies on intermediate relaying nodes for storing incoming packets for forwarding to the next hop. While a small buffer size can conceivably suffice, buffering of multiple packets has some advantages in wireless sensor networks. First, the transition of the radio circuitry between transmission and reception modes consumes considerable energy and thus it is advantageous to receive many packets prior to forwarding them. In addition, data aggregation and fusion involves multiple packets. Multi-hop routing of QoS data would typically require long sessions and buffering of even larger data, especially when the delay jitter is of interest. The buffer size limitation will increase the delay variation that packets incur while traveling on different routes and even on the same route. Such an issue will complicate medium access scheduling and make it difficult to meet QoS requirements.

4.4 Support Of Multiple Traffic Types

Inclusion of heterogeneous set of sensors raises multiple technical issues related to data routing. For instance, some applications might require a diverse mixture of sensors for monitoring temperature, pressure and humidity of the surrounding environment, detecting motion via acoustic signatures and capturing the image or video tracking of moving objects. These special sensors are either deployed independently or the functionality can be included on the normal sensors to be used on demand. Reading generated from these sensors can be at different rates, subject to diverse quality of service constraints and following multiple data delivery models, as explained earlier. Therefore, such a heterogeneous environment makes data routing more challenging.

5. Survey On QoS In WSN

Different research works have been done for QoS in wireless sensor networks:
5.1 Application-Specific QoS Control
In this paper, authors [3] define the optimal number of power-up nodes in the focused area as the QoS target. In order to make the optimal number of nodes to power up in the focused area, a modified Gur Game strategy is given. In the strategy, the base station receives the QoS feedback and gives the dynamic domination information including the area shape information and the dynamic gradient parameters. This results show that the strategy can effectively control the number of power-up nodes and meet the requirement of QoS.

5.2 QoS Reliability Of Hierarchical Clustered WSN
In this paper authors [4] discuss about the problem of reliability modeling and analysis of hierarchical clustered wireless sensor networks. They propose reliability measures that integrate the conventional connectivity-based network with the sensing coverage measure indicating the Quality of Service (QoS) of the WSN and they propose a progressive approach for evaluating such coverage-oriented QoS reliability.

5.3 A QoS-Based Adaptive Clustering Algorithm
The hierarchical routing algorithms for wireless sensor networks mainly focus on distributing energy load among all the nodes and pay little attention to Quality of Services (QoS) support in WSN. In this paper, author [7] developed a algorithm, QAC (QoS-based Adaptive Clustering algorithm), which not only concerns energy consumption but also can improve the reliability and the steadiness of wireless sensor networks by establishing a dual cluster-head model. QAC proposes a local-centralized mechanism for electing cluster-head and suggests a parameter to measure the QoS support in hierarchical applications of WSNs. This model can increase the reliability and the steadiness of wireless sensor network by distributing evenly the communication load and the load of data fusion among cluster-heads. The dual cluster-head model can also improve the survival ability of wireless sensor networks and makes the network fitting in with the fierce changes of environment.

5.4 An Efficient QoS Management in WSN
In this paper authors [6] emphasize the need for repositioning of the Consolidating and Advancing node, (CAN: Principal Node) to improve network lifetime in terms of energy and other QoS parameters such as latency and throughput. The address issues related to its repositioning such as the time and position of relocation and the control of its movement without causing negative impact on the performance of the WSN. Mobility factor is exploited to support QoS requirements. In this paper a scheme called Energy Conserving Relocation (ECR) to pursue relocation of CAN to a safe location on demand is presented. ECR performs relocation based on the minimum energy concept of sensor nodes. The concept of repositioning adds a new dimension to the existing Heterogeneous Wireless Sensor Network. It is observed through simulation that lifetime of the network, average energy consumption and QoS parameters are much better when compared to earlier algorithms.

5.5 An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks
Recent advances in wireless sensor networks have led to many new routing protocols specifically designed for sensor networks. Almost all of these routing protocols considered energy efficiency as the ultimate objective in order to maximize the whole network lifetime. In this paper, they propose an energy-aware QoS routing protocol for sensor networks which can also run efficiently with best-effort traffic. The protocol finds a least-cost, delay-constrained path for real-time data in terms of link cost that captures nodes’ energy reserve, transmission energy,
error rate and other communication parameters. Moreover, the throughput for non-real-time data is maximized by adjusting the service rate for both real-time and non-real-time data at the sensor nodes. Simulation results have demonstrated the effectiveness of our approach for different metrics.

6. CONCLUSION
Recent QoS studies in sensor networks focus on only QoS domain, either timeliness or reliability. They are also limited in differentiating services for traffics with different levels of timeliness and reliability requirements. Here we have analyzed the technical issues for supporting QoS constrained traffic in wireless sensor networks. In the QoS algorithm we discuss about the dual cluster-head model of QAC can improve the survival ability of WSNs, and it has the feature of load balance. This feature helps QAC to balance energy consumption in all nodes and to avoid congestion at cluster-heads by distributing evenly nodes in all clusters, which means that it can distribute evenly the communication load and the data fusion load among all cluster-heads. In a word, QAC has a balanced utilization of resources. QAC also suggests that designer can use standard deviation of number of nodes in a cluster to estimate the robustness while designing WSNs. Finally, QAC is quite simple and is suitable for heterogenous applications with large number of nodes that are densely deployed.

REFERENCE

Author's Biography

V. Vanitha is working as Asst Prof at Kumaraguru College of Technology Anna University, Coimbatore, Tamilnadu, INDIA. She has received B.E, M.E, Degree in Computer Science and Engineering and currently pursuing Ph.D at Anna University. Her main research interest includes Wireless Ad-hoc Networks.

D. Palanivel Rajan worked as Lecturer at Karpagam College of Engineering Anna University, Coimbatore, Tamilnadu, INDIA. He has received B.E, Degree in Computer Science and Engineering and currently pursuing M.E, at Kumaraguru College of Technology Anna University, Coimbatore, Tamilnadu, INDIA. His interest includes Networks.
An Agent - Based Technique For Network - Wide File Search In Heterogeneous Peer - To - Peer Systems

T.Amudha1, Manu Rajan Nair2

ABSTRACT

Heterogeneity is a major characteristic of all modern computer networks, as a consequence of the proliferation of different types of Operating Systems (OS) acting as nodes in such an environment. This presents unique problems to common activities such as locating resources on a network. Agent computing presents us with several solutions when combined with the Peer-to-peer networking to these resource discovery problems. We approach this issue by presenting the solution to a seemingly innocuous problem, i.e., locating a particular file or files that match a given regular expression on the available network.

Keywords: Agents, peer-to-peer system, multi-agent system, heterogeneous network

1. INTRODUCTION

Search problem Scenario

To define this problem, the problem space is defined, here referred to as its “Search Space”. The search space when defined with respect to a single node in a network is called the “Local Search Space” (LSS), which contains all searchable files and folders on that node for a normal user. The “Global Search Space” (GSS) is defined as all files and folders on that network that are search-able by a normal user on any node on the network. The LSS and GSS may vary from node to node depending on the user on that node. Further the GSS is an extremely dynamic environment with various nodes connecting/disconnecting from the network and the files/folders on the nodes varying, as the files/folders are added/removed/changed. Further there are limitations on the files/folders accessible from a given node due to security restrictions imposed on the network. The inherent dynamism and unpredictability of the problem space demands a solution that may be well based on Agent computing and Peer-to-peer networking.

The standard file searching solution available on most OS is simplistic and assumes that the user has knowledge about locating the file and provides a “search term” that is reliable enough, further it is assumed that the user is aware about the limitations imposed by the network. Simply stated, the user knows what search term will help him to locate a particular file and that the file is located in an accessible location. This general simplification applied to the problem may be effective for searching the LSS but may fail in the larger context of the GSS. This “Goal-directed” behavior presents us with the opportunity to select a solution that best reflects the problem, in this case “Agent Computing”.

Agent Computing Paradigm

Agents can be described as a small program running on a node tasked with a specific purpose. The agent is singularly responsible for finishing the task, with several...
other agents with other assigned tasks executing simultaneously on that node. The agent is independent, self-regulating and goal-oriented; it may communicate with other agents on the node or network. Functional autonomy being one of the defining features of agent paradigm, social-communication is another. Agent autonomy implies that the entity can observe its environment and act upon it in a fashion to further the goal assigned to it, without explicit directives from a user. Here the user is expected to only set the goal to the agent; the agent by it has to figure out ways to reach that goal. Agents may be Mobile or Static, intelligent or dumb.

Peer-to-peer and Agent Based Solutions

Peer-to-peer (P2P) systems are normally characterized by decentralized control, large scale and extreme dynamism of their operating environment. Such P2P networks may be of several types, such as self-regulating centralized node tracking based systems with limited intelligence used in P2P file-sharing tools such as Gnutella and BitTorrent networks, or intelligent distributed node tracking based systems such as Anthill.

All existing models based on P2P Agents can be broadly classified along traditional lines as Static or Mobile and further as intelligent or dumb, but all of them still use the central-tracking model for locating and tracking the peers and the publish/subscribe model for exchanging data. Taken together these provide several advantages such as lower communication overheads in locating resources or services and exchanging data. But such systems compromise on critical Agent-computing features such as Agent autonomy by stipulating that an Agent should be part of a peer network only after it subscribes to another published agent on that network. Furthermore such agents need to be user-pre-configured or pre-programmed to handle interaction between peers thereby increasing maintenance requirements for the whole system. Several solutions have been implemented to solve the above problems, one of them is the usage of a Client-Server based peer tracking system, and another method is the use of mobile agents.

2. METHODOLOGY

In the context of the Search Problem, an agent should display goal-directed behavior; hence only intelligent, static agents are considered for the solution. With respect to the Search problem, generally we assume that a static agent is available on each node with access to the LSS on that node; the GSS is made up of all LSS accessible to all the agents on each of the respective nodes. The network is assumed to allow some form of datagram communication among the agents.

In this Agent computing based solution, the Initializer agent is set the goal of locating matches to the search term from the GSS. The Initializer agent communicates this goal to other Search agents in the network using datagram broadcasting, the other agents on receiving the broadcast can “choose” to execute the search request, and in doing so act as “Search Executers” or “Executers”.

The user is expected to pass a “search term” which may be a file name that the user needs to locate or a regular expression to a particular set of files to the Search Agent on that node defined as the “search manager” or “Initializer”. The goal of the Initializer is to return a list of matching locations on the GSS to the user in the smallest possible time. To define a performance metric, we use the First Positive Response Time (FPRT) defined as the time required by an agent to return the first matching location on the GSS to the user in the smallest possible time. To define a performance metric, we use the First Positive Response Time (FPRT) defined as the time required by an agent to return the first matching location on the GSS to the user in the smallest possible time. To define a performance metric, we use the First Positive Response Time (FPRT) defined as the time required by an agent to return the first matching location on the GSS to the user in the smallest possible time.
the number of nodes in the GSS. The FPRT is essentially the sum of all the LSS FPRTs with respect to the Initializer node. This method ignores the parallelizability of the problem; the Initializer node has to search the LSS on each node serially.

3. Specific Goals
The specific goals of the system are

- **To develop a File search application** that
  - Allows searching for specific files/folders on the entire network
  - Allows searching for files/folders matching a pattern on the entire network
  - Display results for a given search from the entire network.

- **Robustness**
  - System is not adversely affected by network communication
  - System is not adversely affected by individual node breakdown.

- **Efficiency**
  - System gets maximum positive responses in minimum time
  - Concise communication (very small communication overhead)

- **Reliability**
  - Results are always correct
  - No redundancy

4. Features of the Approach
It is obvious that any solution to the Search Problem in a network must have the following features:

1. **Low maintenance overhead**
   There must be minimum pre-requisites to setup and install the agents.

2. **Low Communication overhead**
   Inter-agent communication must take up minimum bandwidth.

3. **Best-case FPRT**
   Search results must be returned to the Initializer with minimum delay.

With the above aims, an Agent-based peer-to-peer application was developed in JAVA. This uses datagram/UDP protocol for inter-agent communication (ensuring minimum communication overhead), using features of the JAVA environment to ensure low maintenance costs and implemented machine learning based intelligence on the agents to optimize searching to improve the FPRT. Datagram based communication is the basis for most modern Agent and P2P based systems, as this allows for minimum bandwidth usage on a network. Since in the proposed system, all communication between agents is in the form of single-line text messages, UDP-protocol based messaging system gives the best solution. The platform independent JAVA environment provides the application with a uniform application environment protecting the agent from variations in OS implementations.

The critical aim of the application is to achieve minimum FPRT for each agent on a given GSS; hence optimizing the search at both ends, i.e. the Initializer and executor becomes necessary. Since the OS and file-systems impose limits on the efficiency of search strategies, other methods are necessary to improve the FPRT. The executor and Initializer both implement a simple machine learning based intelligence, that allows the agent to optimize the “search term” based on previous experience for the Initializer and use optimized search strategy for the executor also based on learned knowledge. This introduces the concept of Agent “aging” whereby an agent when installed is an
“infant” with no knowledge of its LSS or GSS. As the agent participates in the search process either as Initializer or executor, it gains experience by relating search terms to search results for its LSS, hence increasing in “maturity”.

A mature agent executor, in theory can return a search result to an Initializer agent with an FPRT of nearly zero. Further we use the concept of “social communication” among the mature agents to share the knowledge gained with “younger” agents on its GSS. This allows a GSS to mature overtime and lowers the FPRT for all searches which in turn leads to lower bandwidth usage and maintenance costs. The system consists of a static intelligent Initializer agent; a static intelligent executor agent and a UDP based communication protocol.

The executor shows certain qualities of mobile agents as the various agents exchange experience with each other. Hence an agent based P2P JAVA application allows the design of a robust and efficient network-wide search solution which can then be extended to cover other functionality such as network resource management and monitoring.

5. OPERATIONAL PARAMETERS OF THE SYSTEM

The environment in which the system operates can be divided into two levels, Files and Folders on the local node and Files and Folders on all other accessible nodes. Local file searches are limited only by node efficiency, but for searching other nodes, efficiency varies greatly depending on various unpredictable factors such as network traffic, network security limitations and communication reliability.

As the agent matures with every search that it participates in, the search accuracy increases with increasing maturity. Consequently, even though at the beginning, search results may not reflect all the matches, it is more than compensated in later stages by the increased search efficiency that is far greater than traditional approaches to the system.

The solution is based on the platform independent JAVA framework, containing an Agent-based application that clearly follows the Agent paradigm. The solution consists of several similar agents installed on various nodes in a network; each node solution consists of a Search gateway and Search Agent. A user at any one of these nodes can use the gateway to search for files on all the nodes in the solution. A Search-Agent communications protocol is implemented on top of the UDP Datagram protocol system in order to achieve maximum efficiency in data communication. The various agents are completely autonomous, with each deciding for it how it goes about efficiently searching its own node. This acts like a distributed application, but it displays a communal intelligence by working towards a common aim.

6. AGENT DESIGN

The overall system organization is given in the following figure:

The main module is the top-most module and acts as the system initializer. The search agent module represents the actual agent that executes the search, on initializing it waits listening for other agents broadcasting search queries, on receiving a query it evaluates the best strategy for executing the search. The search gateway module acts as a gateway to the GSS of the solution, where all search terms are entered and answers sought. The gateway on the solution is completely decoupled from the underlying search agent so as to allow the agent to be truly independent. The communications module provides the Listener for both single and multi-casting; the queries are passed to the Search agent and results to
the Search gateway. The communications module processes the search information cache and derives generalizations for search patterns, when the agent is idle; it tries to share the learned knowledge with other agents. The Search Agent, Gateway and Intelligence agents act as the recipients, whereby messages with the search query tag are passed to the search agent, display result tags to search gateway for displaying and Intelligence module.

7. SEARCH STRATEGIES

The Search Agent class uses Native JAVA file I/O functions to search the file system or the Local Search Space LSS for matches to the given search term. It utilizes the JAVA regular expression model to ascertain matches with the search term. The LSS search strategies are of three types:

1. Hard Search
2. Directed Search
3. Directed Hard Search

In Hard Search, the entire LSS of the Node is searched in a Breadth first algorithm for locating matches, i.e., all directories from the top level onward are searched recursively going down the directory tree.

In Directed search, only the given directory is searched for matches. In Directed hard search only the specified levels starting from the given directory down are searched.

The selection of the strategy for locating a file depends on the result of evaluation conducted on the search term by the agent. The SearchAgent first tries to match the search term with any axioms in the Axiom database, on locating a match; it uses the directed or directed hard search strategy. If no matches are found, it resorts to the hard search strategy. If the axiom corresponds to an earlier successful search, then the directed search strategy is used to validate the location, else if the axiom corresponds to a search directive, then directed hard search is used to search the directories below it.

When a “Hit” is achieved, i.e., a positive match is located during search, the Search Agent or executor sends this information back to the Initializer agent using a unicast packet. This forms the heart of the implementation...
model used by this solution. The query from another agent is the trigger that initializes the search; each hit is then sent back to the Initializer for display.

The intelligent agent based solution for locating files on an enterprise-wide network is developed using JAVA. It leverages JAVA technologies to develop a datagram protocol implemented on top of JAVA UDP protocol communication. It also uses the XML file format as the backend for structured data storage, analysis and retrieval. The XML file format based backend is required as it allows the developer to define a database-like format for data storage and retrieval, separating the database logic from the program logic. This thin-database allows user to store concise amounts of data without redundancy. JAVA provides user with functions that allow the developer to easily add, remove or update information in an XML file.

The Search Strategy Database is physically an XML file containing structured data, each node is made up of an axiom, and each axiom is a valid association between a search term and a location on the LSS. Each axiom also contains a value describing its strength as an integer, greater than or equal to zero. All axioms have an initial value of 1, every time a positive validation of the axiom is done; the current value is multiplied by 2. Every negative validation decrements the value by 1. When an axiom value on validation is found to be zero, it is deleted from the database.

8. RESULTS FROM TESTING AND ANALYSIS OF GSS USING FPRT
The various agents together form a Global Search Space (GSS) that can be searched from any agent on that GSS. The agents allow search execution in a distributed manner, where each agent is tasked with searching its own LSS. The main aim of the system is the reduction of the time required to locate the minimum amount of matches, i.e., the solution is designed for maximum sensitivity and responsiveness. The peer-to-peer system allows the solution to act in a one-to-one communication model, where no other intermediates are involved. This allows the solution to provide an interactive search system where a user can directly be aware of the goings on in the GSS. This further allows the minimum overhead in communication within the GSS.

Intelligent or “smart” agents display an Artificial Intelligence (AI), which is used to increase the efficiency of the solution. Searching can be made more efficient by using smart agents, which can interpret the cache to derive axioms. This derivation of axioms allows the creation of a much smaller search database, which does not depend on the number of searches. This also implies that axioms can grow in strength with increasing number of searches conducted on the GSS. The agents are allowed to share this information with other willing agents, thereby allowing a new agent that becomes a part of the GSS to quickly build up its search database without having to wait to take part in search execution.

The problem space dealt with by the solution is extremely dynamic and unstable; it consists of various nodes, each with its own local terrain defining its LSS. The sum of all these LSS form the GSS, the solution acting as a system that allows the user to locate any file that belongs in the GSS from any node attached to the GSS. Since the structure of the LSS and the search term that is being used are a subjective criterion, i.e., they may vary from system to system and network-to-network, any performance metrics based on them may provide values that cannot be replicated at another location. Hence a new metrics based on the FPRTs suggested as an
An Agent-Based Technique For Network-Wide File Search In Heterogeneous Peer-To-Peer Systems

objective criterion for evaluating the behavior and performance of this solution.

The Search Gateway calculates the First Positive Response Time (FPRT) where the search is initialized. The FPRT for each agent on the GSS is calculated and displayed at the end of the search. Initially, the FPRT values for a GSS made up of infant agents are very high, as all of them have to resort to searching the entire system in order to locate a match. Once the search is completed and the information indexed by the respective agents, further searches using the same search term is lowered dramatically.

A test-bed is setup and the performance of the system is analyzed to determine performance parameters of the Global Search Space (GSS) under real life environments. The GSS consists of three nodes NODE1, NODE2 and NODE3. Each node is made up of three entities, the SearchAgent (SA), the SearchGateway (SG) and the IntelligenceAgent (IA). They are represented node-wise respectively as follows-

On NODE1 - SA1, SG1 and IA1, On NODE2 – SA2, SG2 and IA2 and On NODE3 – SA3, SG3 and IA3

The First Positive Response Time is determined at the Initializer and is calculated for each node on the GSS that has an executor. For a given GSS with “N” nodes, there will be a maximum of “N” FPRTs as well for each search. The FPRT thus represents the time taken in seconds for the first positive response from an executor node reaches the Initializer node and is calculated at the Initializer end of the GSS.

The Testing procedure is as follows, the Test Battery consists of five search terms S1, S2, S3, S4 and S5. Three searches “First-Search”, “Second-Search” and “Third-Search” are conducted for each search term, one each of the searches are initialized from each of the different nodes on the GSS.

S1 = “*.gif”
S2 = “*dll”
S3 = “*exe”
S4 = “*.tmp”
S5 = “*ico”

NODE1 – SYSTEM32 with ip 10.0.0.32, LSS size = 28411 files
NODE2 – SYSTEM30 with ip 10.0.0.30, LSS size = 42173 files
NODE3 – SYSTEM27 with ip 10.0.0.27, LSS size = 15149 files

First-Search is initialized from SG1 on NODE1
Second-Search is initialized from SG2 on NODE2
Third-Search is initialized from SG3 on NODE3

![FPRTs for search S1 from SG1](image1.png)

![FPRTs for search S1 from SG2](image2.png)
For a given search term S1 input from the SearchGateway SG1, the FPRTs calculated from the Initializer SG1 for each of the three SearchAgent SA1, SA2 and SA3 is 15, 15 and 45 respectively in seconds. The test is repeated with the same search term SG2 on the second node yields FPRTs of 30, 0 and 15 seconds respectively from SA1, SA2 and SA3 respectively. These results clearly indicates to two factors, one is the reduction in FPRTs for NODE2 and NODE3, which translates to faster results for the above search term on all searches on the GSS. The second factor is the inherent dynamism of the GSS, which has caused a rise in FPRT for Node1. It is further clear that there is an overall significant improvement in the searching speed on the GSS indicated by the drop in average FPRT for the search from 25 seconds to 15 seconds, an improvement of nearly 40%.

9. CONCLUSION

The dynamic nature of the networked environment demands that any solution for locating network resources such as files and folders be able to inherently deal with the unpredictable nature of its operating environment. The peer-to-peer agent based system suggests a remedy in the form of independent agents located at each node with complete autonomy of action. 

This research work clearly proves that unlike the commonly used solutions for achieving improved search throughputs such as indexing and caching, this solution achieves even higher performance with much lower communication and processing overheads. The solution can demonstrate that by using the suitable paradigm to solve the problem, in this context, Intelligent Agents, a dramatic increase in solution efficiency can be achieved. A learning based artificial intelligence (AI) is used to increase the efficiency of the solution. The AI analyses the performance of the attached search agent and derives ways to make future searches more efficient.

REFERENCES

An Agent-Based Technique For Network-Wide File Search In Heterogeneous Peer-To-Peer Systems


Author’s Biography

Ms T Amudha received her B.Sc Degree in Physics, Masters Degree (MCA) in Computer Applications and M.Phil in Computer Science in 1995, 1999, and 2003 respectively, from Bharathidasan University, India. She has qualified UGC-NET for Lectureship in 2003. She has 10 years of academic experience and is currently serving as Lecturer, School of Computer Science and Engineering, Bharathiar University, India. She is currently pursuing her doctoral research at Bharathiar University in the area of Agent Based Computing. She is engaged in active research guiding MPhil research scholars and has successfully produced 8 MPhil so far. Her research interests include Object Technologies, Distributed Systems and Agent.
Based Computing. She has for his credit more than 15 publications in International/National Conferences/Journals. She is member of Computer Society of India (CSI), International Association of Engineers (IAENG), International Association of Computer Science and Information Technology (IACSIT).

Mr. Manu Rajan Nair received the B.Sc. Degree and M.Sc. Degree in Computer Science from University of Kerala, India. He pursued his MPhil in Computer Science under the guidance of Ms. T Amudha, during 2007-2008 in the School of Computer Science and Engineering, Bharathiar University, India. His research interests include agent technologies and Computer networks.
A New model for Evaluating Performance of Processor Memory Interconnections


ABSTRACT
Interconnection networks play a principal role in today’s parallel computers. In the existing model, performance evaluation of multistage interconnection networks is based on a uniform reference model and the assumption of independent requests. Existing model and analytical results have been widely adopted by numerous researchers as a basis to investigate various aspects of MIN’s. In this paper we study in detail the effects of the independence assumption on the accuracy of system performance and point out the factor which cause inaccuracy. A new queuing model is then proposed and is shown to be very accurate. Since only six states are needed, independent of the size of MIN’s, this new model is very efficient computationally.

Keywords: interconnection networks, multistage interconnection networks, performance evaluation, queuing theory, simulation.

1. INTRODUCTION
Interconnection Networks provide communication between different processing elements and system’s memory modules. Multistage Interconnection Networks (MIN’s) have been used in multiprocessor systems for supporting processor-memory and interprocessor communications as well as in communication systems as basic switching devices. The advantage of using MIN’s includes the low hardware complexity compared to that of crossbars, efficient distributed control schemes, partitionability, availability of multiple simultaneous paths, and ability to employ a variety of implementation techniques [19]. Depending on the clocking structure, switching mode and control strategy, MIN’s can have different operational characteristics giving rise to different system behaviors [11].

In this paper single-path MIN’s are only considered. In these MIN’s internal blocking at any switching element (SE) may occur, i.e., multiple requests from the inputs may need to utilize the same output ports of some SE’s or communication links and only one of them can occupy an output port or a link at a time such that the rest of the requests will be blocked and need to be reissued in the next cycle. In the existing model([1],[2],[4],[5]) a probabilistic approach is proposed to analyze the performance of MIN’s based on a uniform reference model and the assumption of independent requests, i.e., a blocked request is discarded and a new request, which is independent of the blocked one, will be generated in the next cycle to replace the previous blocked yet unserved request. Thus this independent request model can be called as Drop Model (DM) as the request of previous cycle is not considered for the same port in the current
cycle [6]. In a MIN, all requests from input ports must be issued at the beginning of a network cycle where a network cycle is defined as the time for a request to traverse the network and the time for data transmission. It is clear that this independent request assumption is not realistic because, in a real system, a blocked request will go to the same desired memory module in the next cycle. However, in the existing model ([1]-[4],[7] and [12]) compared to the results of simulation, the results are only slightly different if this independent request assumption is omitted. Thus, it is generally believed that the drop model (DM) not only simplifies the analysis but also accurately predicts the performance of a synchronous MIN.

Through extensive simulations it is found that the DM is indeed not accurate enough in many cases, but it has been widely adopted by many people without further justifications ([9],[10],[14]-[19],[21]) which motivates the research work of this paper. It has also been shown by Lee [6] that there is an as much as 30% difference between results using the existing model and those using dependent request models which are referred in this paper as Hold Model (HM) as the request of the previous cycle is also considered for current cycle to the same port. Of course, the HM is a much more realistic model. Yen, Patel, and Davidson [21] investigated, on crossbars, the accuracy of the DM (referred to as Model 1 in their paper) as well as two other proposed models (namely, Models 2-3) by comparing them with the simulation results of the HM. The other two models are briefly summarized as follows: Model 2 is a modified version of Model 1 by assuming that all the blocked requests will always be resubmitted as new independent requests, which is referred in this paper as modified drop model (MDM). By performing comparisons it is derived Model 3 is the most accurate model for the analysis of crossbars. However, since Model 3 is non-intuitive ([20],[21]) it is very difficult to extend this model to analyze cascaded crossbars such as MIN’s. MDM is slightly better than the DM in terms of maximum difference. Also, extending MDM to analyze a MIN will make the analysis more complicated. Thus, the DM is still widely used for the analysis of MIN’s, when crossbars are cascaded into a MIN.

In this paper we first investigate the existing model (DM), through extensive simulations, point out that the existing model (DM) is not accurate enough. After that, a new queuing model is proposed, which takes into account that a blocked request will be reissued and will still go to a previously determined destination, and shows a very high degree of accuracy. Moreover, since only six states are needed, independent of size of a MIN, this new model is very efficient computationally. Since the accuracy of MDM in [22] has only been analyzed on crossbars, in this paper we also developed a model by adopting the concept of MDM to the analysis of MIN’s and investigate its accuracy.

The remainder of the paper is organized as follows. In Section II, a brief review and investigation of the existing model is presented. In Section III we present a new model for analysis. The discussion of the performance of the system is presented in Section IV. In section V evaluation of the proposed model is given. Section VI gives the conclusions of this paper.

2. A REVIEW OF DROP MODEL

In order to understand the accuracy of the existing model (DM), a brief review of the existing model (DM) and the analytical results ([2],[4],[7] and 19) are presented here. The accuracy of its results is investigated in detail in the next section.
The drop model (DM) is based on the following assumptions
1) A \((P \times M)\) MIN consists of \(n\) stages of
switching elements (SE’s) where \(P\) is the total number of
inputs, \(M\) is the total number of outputs, and each SE is
an \((a \times b)\) crossbar. For example, a \((2 \times 2)\) crossbar is shown
in 1.
2) During each cycle a request will be generated at each
input port with probability \(r_0\), i.e. the request rate.
3) All the requests are uniformly distributed over all output
ports.
4) The requests which are blocked (not accepted) are
ignored and the requests issued in the next cycle are
independent of the requests previously blocked.

The fourth assumption is used to simplify the analysis
and of course, is not realistic. In a real system, a rejected
request must be reissued in the next cycle and its
destination must be the same as the blocked one.

Based on the assumptions stated above, in the existing
model an analysis on crossbars of size \(a \times b\) is performed.
Here \(a\) is input size, \(b\) is the output size. For an arbitrary
output, the probability that there is no request to the
output is \((1-r_0/b)^a\). Thus, the probability that there is a
request on an arbitrary output of the crossbar is

\[ r_{out} = 1 - (1-r_0/b)^a \] (1)

The analytical result of a crossbar is then used as the
basis for analyzing a \(\text{MIN}([13],[16])\). Since the output
rate of a stage is the input rate of the next stage, in the
existing model an analysis on the synchronous \(\text{MIN}\) it
can be shown that the probability that there is a request
on any particular output at the \((i+1)\)th stage of the network
satisfies the recurrence relation

\[ r_{out} = 1 - (1-r_0/b)^a \] (2)

Where \(r_0\) is the request rate, and \(M \times r_\text{out}\) is the throughput,
which is defined to be the average number of unit
messages delivered by the \(\text{MIN}\) in a unit time.

2.1. Investigations of Drop Model

Now we investigate the accuracy of the DM and point
out the factors that cause the inaccuracy. The new models
that can avoid those problems will be presented in the
next section.

To verify that DM is not accurate enough in many cases
and the independence assumption is indeed the major
reason for the inaccuracy, we implement two versions of
simulation for DM and HM. When a request is blocked,
it will stop traveling through the network. For the DM
the source node with a blocked request will generate a
new request with probability \(r_0\) and will generate another
destination. However, for the HM when a request is blocked, the source node will reissue the same request in the next cycle to the same destination. The results of the DM are always too conservative for a small value of \( r_0 \) and too optimistic for a large value of \( r_0 \). This observation can be understood through the following interpretations.

In the first case when the request rate \( r_0 \) is small, and once a request is blocked, it should be reissued in the next cycle. Thus, the probability of issuing a request in the next cycle should be 1. However, the probability of issuing a request under the existing model (i.e., the DM) will still be \( r_0 \). Thus, the existing model will have much lower throughput than the realistic model. The inaccuracy will depend on the difference between the value of \( r_0 \) and the probability for a request to be blocked. A decrease of \( b/a \) will increase the difference between the DM and the HM when \( r_0 \) is small. An increase of \( n \) will increase the probability for a request to be blocked through the network.

In the second case when the request rate \( r_0 \) is high, there will be a considerable probability that many requests need the same path and only one of them will be chosen and all others will be blocked. In a real system once those requests are blocked, they are likely to use the same path in the following cycle. However, the DM assumes that all requests are uniformly distributed over all paths in each cycle; those requests that needed the same path and were blocked in the previous cycle will again be assumed to be uniformly distributed over all paths. When request rate \( r_0 \) is high, there will be a large number of requests that need the same path in a cycle. The difference between the DM and the realistic HM is not negligible. A realization of this optimistic condition is shown in Fig. 2.

![Figure 2: Realization Of The Optimistic Condition](image-url)

### 3. NEW QUEUING MODEL

We propose a new queuing model for analyzing crossbars\[8\] which utilizes the knowledge of the simulation results to adjust the dominant factors that cause the inaccuracy. From the discussion in the previous section it is clear that those factors which cause the inaccuracy in both the conservative and optimistic conditions of the DM need to be adjusted. Since the requests from all the input ports to all the output ports are assumed to be stochastically identical, we can then perform the analysis on the behavior of an arbitrary path. First, we perform analysis on a single crossbar. After that, we generalize it to the analysis of a MIN. The new queuing model for analyzing crossbars based on the understanding of the previous observation is then presented in Fig. 3. In the figure,

- state \( I_0 \) represents that, at the beginning of a network cycle, the source node is not issuing a request;
- state \( R_0 \) represents that the source node is issuing a request which can be a new request or the same request that was...
A New model for Evaluating Performance of Processor Memory Interconnections

Figure 3: State Transition Diagram Of The Proposed Model

...blocked in the previous cycle; and state $B_0$ represents that the source node is not joining the competition. Also, state $I_n$ represents that, at the end of a network cycle, the source node not issuing a request is staying idle; state $R_n$ represents that the requested path has been successfully established; and state $B_n$ represents that the request is blocked. Note that state $B_0$ is used to reduce the inaccuracy introduced in the optimistic condition. For example, in Fig. 2, all three requests will go to the same destination in the current cycle and two of them will be blocked again. Thus, when they are blocked, only one of the three requests will go to state $R_0$ and the rest of them will go to state $B_0$.

Utilizing the new queuing model, the problems which occur in the conservative condition of the DM can be improved by introducing a transition from state $B_0$ to state $R_n$ which ensures that the blocked request will be reissued in the next cycle. Also, the transition from state $B_n$ to state $B_0$ is used to adjust the overestimated input rate in the optimistic condition of the DM. The state transition probabilities can be derived through the following understanding:

1) $P_{[R_n|R_o]}$: which is the probability that, at the end the probability that there will be of a network cycle, a request is in State $R_n$ giving that the request was in State $R_o$ at the beginning of that network cycle. Let $m_{out}$ be the probability that there is a request on any particular output of the crossbar. Since we assume that the requests from different source nodes are stochastically independent, $m_{out}$ can be obtained by utilizing the formula in [18] as follows:

$$m_{out} = 1 - \left(1 - \frac{Pr[R_o]}{b}\right)^n$$  (3)

Since on the average there will be $(a \times Pr[R_j])$ requests at the beginning of a cycle and only $(b \times m_{out})$ requests can reach the output of the crossbar, it is clear that

$$Pr[R_n|R_o] = \frac{m_{out} \times b}{axPr[R_o]}$$  (4)

2) $P_{[R_0|B_n]}$: This transition is based on a concept, which ensures that the blocked request will be reissued in the next cycle. By setting $Pr[R_0|B_n] = 1$, though can effectively adjust the conservative condition, will result in an even more significant optimistic condition. We combine the effect of the conservative as well as the optimistic conditions by introducing the transition from state $B_0$ to states $R_0$ and $B_0$. The derivation of the transition probabilities will be shown later.

3) $P_{[B_0|B_n]}$: This transition probability is used to reduce the inaccuracy introduced in the optimistic condition of the DM, i.e., some of the blocked requests that needed the same path in the previous cycle will need the same path again. Only one of those requests will successfully go through the network and the rest of them will be blocked again. Since the DM assumes that all requests that needed the same path in the previous cycle will be assumed to be uniformly distributed over all paths in any cycle, those requests that needed the same path in the previous cycle will be uniformly distributed over all paths. A straightforward approach for solving this problem is to build a queuing model where each state represents the number of requests that need the same path. However, this approach will require a large number of states and it
is very difficult to derive the state transition probabilities between states. Thus, we solve this problem by utilizing the knowledge that not all the blocked requests will go to the state of reissuing a request; instead, some of them will go to the state which represents that it will be blocked again. Consider an arbitrary blocked request, \( i \) requests from the other \( (a-1) \) source nodes which need the same path is

\[
\sum_{i=1}^{a-1} \left( \frac{P[R_o]}{b} \right)^i \left( 1 - \frac{P[R_o]}{b} \right)^{a-i-1} \frac{1}{i}
\]

Since this request is blocked and there are \( i \) requests from other \( (a-1) \) source nodes in this cycle, one of the \( i \) requests will successfully establish the desired path and the other \( (i-1) \) requests will need the same path in the next cycle. In this new model, we allow only one of the \( i \) blocked requests to go to state \( R_o \) and the rest of them to state \( B_o \) at the beginning of the next cycle. Since all the requests are randomly selected, the probability that this request will not be blocked by the other \( i-1 \) requests and can go to state \( R_o \) is \( \frac{1}{i} \). Then the conditional probability is derived as follows:

\[
Pr[R_0|B_n] = \frac{\sum_{i=1}^{a-1} \left( \frac{P[R_o]}{b} \right)^i \left( 1 - \frac{P[R_o]}{b} \right)^{a-i-1} \frac{1}{i}}{1 - \left( 1 - \frac{P[R_o]}{b} \right)^{a-1}}
\]

and

\[
Pr[R_0|B_o] = \frac{\sum_{i=1}^{a-1} \left( \frac{P[R_o]}{b} \right)^i \left( 1 - \frac{P[R_o]}{b} \right)^{a-i-1} \frac{1}{i}}{1 - \left( 1 - \frac{P[R_o]}{b} \right)^{a-1}}
\]

The rest of the state transition probabilities can be easily obtained and are given as follows:

\[
Pr[B_o|B_o] = 1 - \frac{m_{tot} \times b}{axPr[R_o]}
\]

\[
Pr[R_o|B_o] = Pr[I_o|I_o] = 1
\]

\[
Pr[R_o|B_o] = Pr[R_o|I_o] = r_0
\]

\[
Pr[I_o|B_o] = Pr[I_o|I_o] = 1 - r_0
\]

3.1. Proposed Model for the MIN’s

The state transition diagram for a MIN can be obtained similarly by using six states at each stage and using the output rate of a stage as the input rate of the next stage. However, this approach will significantly increase the cost of analysis. Thus, for a MIN, we propose a model which uses the same number of states as that for a single crossbar independent of the system size. In the new model the meanings of all the states remain the same as they are in a crossbar but the state transition probabilities have to be modified. Let \( r_k \) be the probability that there is a request on any particular output at the \( k \)th stage. We can then have the following recurrence relation:

\[
m[k] = 1 - \left( 1 - \frac{m[k-1]}{b} \right)^3
\]

where \( m[0] \) is the adjusted initial request rate, i.e., \( m[0] = Pr[R_o] \), also

\[
Pr[R_o|Ro] = \frac{m[n] \times M}{N \times P[R_o]}
\]

We also propose to keep the state transition probabilities from \( B_o \) to \( B_o \) and from \( B_o \) to \( R_o \) to reduce the analysis cost. Those conditional probabilities, of course, are only an approximation of the model. A further partitioning of the states will produce a more accurate result. However, through extensive simulation verification, this model has been found to be accurate enough for the analysis of the system.

Deriving the state flow balance equations, we can then summarize the system of equations as follows:

\[
Pr[Bo] = Pr[B_o] \times Pr[Bo|B_o] \\
Pr[B_o] = Pr[Bo] + Pr[R_o] \times Pr[B_o|R_o] \\
Pr[R_o] = Pr[B_o] \times Pr[R_o|B_o] + (Pr[R_o] + Pr[I_o] \times r_0)
\]
A New Model for Evaluating Performance of Processor Memory Interconnections

\[ Pr[R_n] = Pr[R_0] + Pr[R_n | R_0] \]
\[ Pr[I_n] = Pr[I] \]

where

\[ Pr[R_n | R_0] = Pr[R_n | R_0] = r_0 \]
\[ Pr[I_n | R_0] = Pr[I_n | I_0] = 1 - r_0 \]

\[ Pr[B_n | R_0] = \frac{m[n] \times M}{N \times P_{[R_0]}} \]

\[ Pr[B_n | R_n] = 1 - \frac{m[n] \times M}{N \times P_{[R_0]}} \]

\[ Pr[I_0 | R_n] = \frac{N - i - 1}{i} \left( P_{[R_0]} \right)^{i-1} \left( 1 - P_{[R_0]} \right)^{N-1+i} \]

\[ Pr[I_0 | I_0] = \frac{N - i - 1}{i} \left( P_{[R_0]} \right)^{i-1} \left( 1 - P_{[R_0]} \right)^{N-1+i} \]

\[ Pr[R_0, B_n] = 1 - \left( \frac{P_{[R_0]}}{M} \right)^{N-1} \]

with two boundary conditions

\[ Pr[I] + Pr[B_0] + Pr[R_0] = 1, \text{for } i = 0, n. \]

These equations are useful in solving the balancing equations for crossbars and MIN’s numerically, so that we can derive performance measures from them.

### 4. Analysis Of Hold Model

The performance of MIN’s depends greatly on many different system parameters like the number of stages in the MIN and the size of SE’s, and workload parameters like request rate etc. The optimum design of a system with respect to various parameters is discussed in this section. There is a tradeoff between cost and performance of a system. Depending on applications, it is desirable to ensure that some measures will always meet certain performance criteria. Also, in [15] and [17], the authors pointed out that the bandwidth of a squared MIN is asymptotic independent of request rate. Thus, in this paper, all the performance measures will be taken into account for analysis. The designer, with all the information, can then design a system that best meets some desired performance criteria and is also cost-effective.

The performance measures used in the rest of this paper are described as follows:

1) **Throughput (\( \text{thr} \))** is defined as the average number of unit messages delivered by the MIN in a unit time. It can be shown that throughput

\[ \text{Throughput} = Pr[R_n] \times N. \]

Normalized throughput (\( \text{nor}_\text{thr} \)) is defined as throughput of each input/output port. Thus for a non-squared MIN, there are input normalized throughput \( \text{nor}_\text{thr}_\text{in} = Pr[R_n] \) and output normalized throughput \( \text{nor}_\text{thr}_\text{out} = Pr[R_n] \times N/M \).

2) **Transmission delay (\( T_d \))** is defined as the mean number of network cycles required to transfer a message. Since the probability that a request is accepted given that it is issued is \( Pr[R_n]/r_0 \) clear that

\[ \text{Transmission delay} = \frac{r_0}{Pr[R_n]}. \]

A system with larger number of stages may have a longer network cycle which is not necessarily proportional to number of stages of a MIN because a cycle contains not only the time to setup a path for a request but also the time for data transmission.

The probability that a request is accepted will be independent of the value of b/a of an SE. When \( r_0 \) is small, a system with a large number of input ports and a smaller number of output ports will be more cost-effective, which will have similar \( \text{nor}_\text{thr}_\text{in} \) but much higher \( \text{nor}_\text{thr}_\text{out} \) compared to a system with a large number of output ports.
ports. When \( r_0 \) is high, the performance of a MIN will become complicated. A system with a large number of output ports will have much higher nor\(_{\text{thr}}\) yet much lower nor\(_{\text{thr,_{out}}}\). As \( r_0 \) increases, \( T_d \) will increase significantly. It should be noted that when \( r_0 \) becomes large, since throughput will become asymptotically independent of \( r_0 \), transmission delay will increase linearly with the rate of \( r_0 \).

5. EVALUATION USING PROPOSED MODEL

The performance measures can be evaluated using equations (15), (16) which were derived in the previous section. The following program is useful for calculating the transmission delay and throughput etc. performance measures.

Program for Calculation of Normalized throughput and transmission delay of Multistage Interconnection Networks.

```c
#include<stdio.h>
#include<math.h>

void main()
{
float td, r, tp, rin;
int b, a, n;
clrscr();
printf("Enter memory, nthstage, nthstagerequest, processor, reqrate values'');
scanf("%d%d%f%d%f", &b, &n, &rin, &a, &r);
rint=1-(pow((1-rin/b),a));

rd=r/rint;
printf("Normalized throughput is \( r_{\text{in}} \)'' rd);
printf("Transmission delay is \( T_d \)'' td);
getch();}
```

### Table 1: Throughput of 2x2 MIN

<table>
<thead>
<tr>
<th>( r )</th>
<th>( p )</th>
<th>1.0</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4881</td>
<td>1.7191</td>
<td>1.9455</td>
<td>2.1684</td>
<td>2.3887</td>
<td>2.6070</td>
<td>2.8237</td>
</tr>
<tr>
<td>1.3484</td>
<td>1.5171</td>
<td>1.6836</td>
<td>1.8483</td>
<td>2.0121</td>
<td>2.1764</td>
<td>2.3347</td>
</tr>
<tr>
<td>1.2210</td>
<td>1.3299</td>
<td>1.4381</td>
<td>1.5455</td>
<td>1.6622</td>
<td>1.7589</td>
<td>1.8648</td>
</tr>
<tr>
<td>1.1051</td>
<td>1.1569</td>
<td>1.2098</td>
<td>1.2619</td>
<td>1.3139</td>
<td>1.3660</td>
<td>1.4179</td>
</tr>
</tbody>
</table>

### Table 2: Transmission delay of 2x2 MIN

<table>
<thead>
<tr>
<th>( r )</th>
<th>( p )</th>
<th>1.0</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6094</td>
<td>0.5165</td>
<td>0.4498</td>
<td>0.3992</td>
<td>0.3594</td>
<td>0.3271</td>
<td>0.3004</td>
</tr>
<tr>
<td>0.5376</td>
<td>0.4654</td>
<td>0.4112</td>
<td>0.3689</td>
<td>0.3349</td>
<td>0.3069</td>
<td>0.2833</td>
</tr>
<tr>
<td>0.4450</td>
<td>0.3955</td>
<td>0.3563</td>
<td>0.3246</td>
<td>0.2982</td>
<td>0.2760</td>
<td>0.2570</td>
</tr>
<tr>
<td>0.3276</td>
<td>0.3004</td>
<td>0.2782</td>
<td>0.2588</td>
<td>0.2407</td>
<td>0.2274</td>
<td>0.2145</td>
</tr>
<tr>
<td>0.1810</td>
<td>0.1729</td>
<td>0.1653</td>
<td>0.1585</td>
<td>0.1522</td>
<td>0.1464</td>
<td>0.1411</td>
</tr>
<tr>
<td>0.1810</td>
<td>0.1729</td>
<td>0.1653</td>
<td>0.1585</td>
<td>0.1522</td>
<td>0.1464</td>
<td>0.1411</td>
</tr>
<tr>
<td>0.1810</td>
<td>0.1729</td>
<td>0.1653</td>
<td>0.1585</td>
<td>0.1522</td>
<td>0.1464</td>
<td>0.1411</td>
</tr>
</tbody>
</table>
6. CONCLUSIONS

We have shown that the model for the performance evaluation of synchronous multistage interconnection networks in existing model is not accurate enough in many cases, by carefully pointing out the causes for inaccuracy. A new queuing model is then proposed which has been shown to be very accurate in performance analysis and very efficient in computation. The performance of MIN’s has been discussed in detail with respect to various system parameters, such as the number of stages in the MIN, the sizes of switching elements in the MIN and the request rate.

Further research can be done in extra stage MIN’s considering the hold model using on queuing model and transitions for evaluating the performance measures.

REFERENCES


A New model for Evaluating Performance of Processor Memory Interconnections

Author’s Biography

Dr. A. Subramanyam received his Ph.D. degree in Computer Science and Engineering from JNTU College of Engineering, Anantapur. He has obtained his B.E.(ECE) from University of Madras and M.Tech.(CSE) from Visveswaraiah Technological University. He is having 17 years experience in teaching. He is currently working as Professor & HOD in the Department of Computer Science and Information Technology of Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa dt. A.P. He has presented and published 7 technical papers in international and national conferences and 2 technical papers in international and national journals. He is guiding few Ph.D.s. His research areas of interest are parallel processing, network security and data warehousing.

Dr. E. V. Prasad, received his Ph.D. degree in Computer Science and Engineering from I.I.T., Roorke. He is having 29 years of experience in teaching. He joined in the JNTU university in the year 1979 and served in various positions. Now he is Professor in computer science and engineering department and also principal of JNTU college of engineering, Kakinada. He has published 41 technical papers in international and national journals and conferences. He has already guided few Ph.D.’s and guiding several Ph.D.s. His research areas of interest including parallel processing, image processing, computer networks, network security and data warehousing.

T. Harikrishna working as Assistant Professor in Computer science & Information Technology department in Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa dt. A.P. He is having 5 years of experience in teaching. He completed his B.Tech. in Information Technology from JNTU. He is also having M Tech degree in Information Technology.

K. Sreenivasulu presently working as Associate professor & HOD CSE in Madina Engineering College Kadapa. He is currently pursuing Ph.D from JNTU. He received B.E in Computer science from Bangalore University. He received his M.Tech in computer science from JNT University. He is having more than 11 years of experience in teaching. He has guided several graduate & post graduate students in their Academic projects.
Analysis of Thread Scheduling With Multiple Processors
Under A Markov Chain Model

D. Shukla1  Saurabh Jain2  and  Shweta Ojha3

1Department of Maths and Statistics, Dr. H.S. Gour University, Sagar (M.P.) – 470003. E-mail: diwakarshukla@rediffmail.com
2Department of Computer Science and Applications, Dr. H.S. Gour University, Sagar (M.P.)- 470003. E-mail : iamsaurabh_4@yahoo.co.in
3Department. of Computer Science and Applications, Dr. H.S. Gour University, Sagar (M.P.) - 470003. E-mail :meshweta1@gmail.com

ABSTRACT
This paper presents a Markov chain model based study in the environment of multi-level queue scheduling with the multiple processors, assuming the random movement of scheduler over various processes and queues. Each processor assumes random selection of threads from different queues. In particular, the discussion incorporates only three processors along with three queues and the procedure of thread scheduling is examined in light of Markov chain model. A simulation study is incorporated to support the findings.

Keywords : Process scheduling, Markov chain model, State of system, Process queue, Multi-level queue scheduling, Transition probability matrix, Central Processing Unit (CPU).

1. INTRODUCTION
The scheduling is a methodology of managing multiple queues of processes in order to minimize delay and to optimize performance of the system in the environment where queues of processes exist with servers. A scheduler is an OS module whose primary objective is to optimize the system performance according to the criteria set by the system designers. Scheduler refers to a set of policies and mechanism, built into the operating system, that governs the order in which work require to be done by the computer.

A process and a thread differ to each other in terms of their execution priority. An application can be implemented as a number of threads that cooperate and execute concurrently in the same address space. On uni-processor, thread can be used as a program structuring aid and to overlap I/O with processing (when one thread is waiting for I/O, another thread may be executed of the same program). The switching cost for thread is less than the switching cost over process. The real advantage of thread appears in multi-processor systems where threads can be used to exploit the parallelism in applications and due to which significant interactions among threads occur.

A combination of thread management and process scheduling together can improve upon the performance of the system. Some popular multi-processor thread schedulings are:

(i) Load Sharing (ii) Gang (Group) Scheduling (iii) Dedicated Processor Assignment (iv) Dynamic Scheduling.

A multi-level queue scheduling algorithm partitions the ready queue into separate queues. Processes are permanently assigned to one queue, generally based on
some property of the process such as memory size, process priority or process type. Each queue has its own scheduling algorithm.

In this paper, a Markov chain model is used to examine the scheduler’s transition behavior among threads and multi-processors in the multi-level queue environment. Scheduler picks up threads from processes and randomly allocates to processors. The focus is on to analyze the transition probabilities in thread scheduling and to simulate the movement mechanism of the scheduler procedure under the assumed probabilistic environment in the form of a model.

**A. Motivation**

Naldi [4] has applied Markov chain model technique to interpret the flow of internet traffic among various network operators. Shukla et al. [9], [10] used the same to explain the pattern of information flow in Space Division switches and Knockout switches. Shukla and Jain [7], [8] utilized the Markov Chain model technique for the study of scheduler transition mechanism in the multilevel queue scheduling of an operating system. The fundamental basics of this technique are described in detail due to Medhi [3]. Deriving an idea from all these contributions, this paper also incorporates the use of Markov Chain model to study the scheduler transition behavior the thread scheduling procedure.

**B. A Review**


**2. Markov Chain Model**

Assume that there are three parallel processors; P1, P2, P3 and three queues Q1, Q2, Q3 in a processing system. The queue Qk has a process Rk each with three threads (k=1,2,3). Define a Markov chain \( X^{(n)} \) \( n \geq 0 \) where X is position of scheduler over states at the \( n^{th} \) jump (or transition). Each thread is a state \( (3+3+3=9) \) and processors Pk are additional three states. In all, to assume X jumps over 12 states \( (t_1, t_2, t_3, \ldots, t_9 \text{ for threads and } P_1, P_2, P_3 \text{ for processors} \) randomly among \( n \) transitions.

The transition diagram for the case of three queues in each \( R_k \) is given in fig. 2.0

![Figure 2: The Transition Diagram](image-url)
3. **Calculation Of Transition Probabilities**

Let priority of the queue $Q_i$ is $a_i$ and supports in multilevel queue scheduling such that each queue has a process with three threads. Suppose initial priorities of threads are $b_i$ so that $P[Y^{(0)}=t_i]=b_i$:

$$\sum_{i=1}^n b_i = 1, i=1,2,3...9.$$  

**Remark 3.0:** The state probabilities of processors (as per fig. 9.0) after the first transition:

$$p_i^{(0)}(X^{(0)}) = P_i^{(0)} = \sum_{i=1}^n p_i^{(0)}X^{(0)} = t_i = P_i^{(0)}X^{(0)} = t_i$$

$$= b_i t_{i10}^1 + b_i t_{i20}^2 + \ldots + b_i t_{i30}^3$$

$$= \sum_{j=1}^3 b_j t_{ij0}^j$$

**Remark 3.1:** The state probabilities of processors after the second transition:

$$p_i^{(1)}(X^{(1)}) = P_i^{(1)} = \sum_{j=1}^n p_i^{(1)}X^{(1)} = t_{ij} = P_i^{(1)}X^{(1)} = t_{ij}$$

$$= \sum_{j=1}^3 b_j t_{ijj}^j$$

**Remark 3.2:** Generalized probabilities of processors after transitions:

$$p_i^{(2)}(X^{(2)}) = P_i^{(2)} = \sum_{j=1}^3 b_j t_{ijj}^j$$

**Remark 3.3:** Generalized probabilities of thread state after the $n$ transitions:

$$p_i^{(n)}(X^{(n)}) = P_i^{(n)} = \sum_{j=1}^n b_j t_{ijj}^j$$

4. **Calculation Of Thread Probabilities**

**Remark 4.0:** State probabilities of processors after the first transition:

$$p_i^{(0)}(X^{(0)}) = P_i^{(0)} = \sum_{i=1}^n p_i^{(0)}X^{(0)} = t_i = P_i^{(0)}X^{(0)} = t_i$$

$$= b_i t_{i10} + b_i t_{i20} + \ldots + b_i t_{i30}$$

$$= \sum_{j=1}^3 b_j t_{ij0}^j$$

**Remark 4.1:** State probabilities of threads after the second transition:

$$p_i^{(1)}(X^{(1)}) = P_i^{(1)} = \sum_{j=1}^n p_i^{(1)}X^{(1)} = t_{ij} = P_i^{(1)}X^{(1)} = t_{ij}$$

$$= \sum_{j=1}^3 b_j t_{ijj}^j$$

**Remark 4.2:** State probabilities of processors after the third transition:

$$p_i^{(2)}(X^{(2)}) = P_i^{(2)} = \sum_{j=1}^3 b_j t_{ijj}^j$$

**Remark 4.3:** Generalized expressions for processors state after the $n$ transitions:

$$\sum_{i=1}^n a_i = 1 = \sum_{j=1}^n \left[ p_i^{(n)}(X^{(n)}) = P_i^{(n)} = \sum_{j=1}^n b_j t_{ijj}^j \right] = p_{i1}^{(n)}$$

**Remark 4.4:** Generalized expressions for thread state after the $n$ transitions:

$$p_i^{(n)}(X^{(n)}) = P_i^{(n)} = \sum_{j=1}^n b_j t_{ijj}^j$$
5. PROCESSOR AND THREAD INDICES

Define three kinds of processor and thread indices:

(a) Processor Index

\[ I_k = \frac{P(X(\omega) = P_k)}{\sum_{i=1}^{n} P(X(\omega) = P_i)} \]

(b) Processor-Thread Index

\[ I_k^j = \frac{P(X(\omega) = P_k, Y(\omega) = T_j)}{\sum_{i=1}^{n} P(X(\omega) = P_i, Y(\omega) = T_j)} \]

(c) Thread index

\[ I_k^j = \frac{P(X(\omega) = X_i, Y(\omega) = T_j)}{\sum_{i=1}^{n} P(X(\omega) = X_i, Y(\omega) = T_j)} \]

6. SIMULATION STUDY

Consider following three data sets on which graphs are obtained through simulation.

Data set I:

| Transition Probability Matrix For Data Set I |

Data set II:

| Transition Probability Matrix For Data Set II | Processor to Processor |

Figure 3: Transition Probability Matrix For Data Set I

Figure 6: Processor to Processor
7. CONCLUDING REMARKS

Over the increasing transitions, the state probabilities of processors are going down. The similar pattern of downward trend is found for all three data sets. When coming to the aspect of scheduler transition over processors under thread scheduling it seems, initially the processor may be assigned priorities for scheduler but with the increase of transitions this priority nullifies. While comparing state probabilities of threads and processors together, both constantly reduces over increasing n in all three data sets. But, the processors state probability remains high over the thread state probabilities. This indicates more and more involvement of scheduler towards processor in the thread scheduling algorithm.

REFERENCES


Author’s Biography

Dr. Diwakar Shukla, Professor, is a permanent faculty member in the Department of Mathematics and Statistics, Sagar University, Sagar, M.P. with over 19 years experience of teaching to U.G. and P.G. classes. He obtained M.Sc.(Stat.) Ph.D. degrees from Banaras Hindu University, Varanasi and served the Devi Ahilya University, Indore, for nearly nine years and obtained M.Tech (Computer Science) degree from there. He joined Sagar University, Sagar, MP, as a Reader in 1998. During Ph.D work at BHU., he was recipient of Junior Research fellowships of CSIR, New Delhi, through All India Fellowship Examination of 1983. Till now, he has published more than 50 research papers in national and international journals and participated in over 35 seminars/conferences at the national level. He is also the recipient of MPCOST Young Scientist Award, ISAS Young Scientist Medal, UGC Career Award and UGC Visiting Fellow to the Amerawati University, Maharashtra.

He worked as a Professor, for one year, to the Lucknow University, Lucknow, U.P., and visited abroad to Sydney (Australia) and Shanghai (China) for the conference participation. He has authored one book and supervised
seven Ph.D. theses in Statistics and Computer Science both; eight students are presently enrolled with him for their doctoral degree. He is a member of 10 learned bodies of Statistics and Computer Science at national level. The area of research he works for are Sampling Theory, Graph Theory, Stochastic Modeling, Computer Network Switching Systems and Operating Systems.

Mr Saurabh Jain has obtained M.C.A. degree from H.S. Gour University, Sagar, MP, in 2005. He is presently working as a Lecturer in the Department of Computer Science & Applications in the same University since 2007. He did his research in the field of CPU scheduling and Operating systems. He has authored and co-authored 6 research papers published in journals and conference proceedings. His current research interest is to analyze the scheduler’s performance and algorithms both under probability models.

Mrs. Shweta Ojha received her M.C.A. degree from H.S. Gour University, Sagar in 2005. She is presently working as a Lecturer in the Department of Computer Science & Applications in the same University since 2007. She is enrolled for the doctoral degree. Her research interest is to examine the scheduler’s performance under various algorithms in probabilistic environment. She has some research papers published in conferences proceedings.