Asian Journal of Information Technology 1 (1): CCCC, 2008
ISSN: 1682-3915

A Quantum Based Algorithm for Computer Network Routing

Ahmed T. Sadik and Rana F. Ghani
Department of Computer Science, University of Technology, Baghdad

Abstract: The aim of this study, is to design and implement a quantum algorithm for network routing by exploiting the massive parallelism existing in the quantum environment and to deal with the demands of continuous growing of the internet. This algorithm is compared according to the number of iterations and time complexity with Dijkstra's algorithm and Genetic algorithm and the result shows that the quantum approach is better in finding the optimal path with less time complexity.

Key words: Quantum computing, grover's algorithm, dijkstra's algorithm, genetic algorithm, network routing

INTRODUCTION

The basic technique of the Internet, that binds these thousands of component networks into a single Internet, is addressing and routing. In the address part, each network uses a unique set of addresses drawn from a single global address space. Each connected device has a unique address that it uses to label its network interface. Each portion of information (packet) generated by these devices has a source and destination address. The source address references the local interface address and logically, the destination address is the corresponding interface address of the intended recipient. As it is being passed within the network from router to router, the router can identify this intended recipient. But within a network identity is only half of the solution. The network must be able to know location, or, where the packet is to be directed. The task of associating location with identity, or in other words maintaining routing information within a network is undertaken by routing protocols which include routing algorithms that determine the best path to send a packet from its source to destination (Huston, 2004).

QUANTUM COMPUTATION

In the 20th century, great achievements were made within the field of computation. The researches in the field of quantum mechanics made it possible to create transistors and microchips thereby opening the door for the high speed personal computers. However, even the building of such computers is in its first steps, the scientist have realized that when such a computer is being made possible the operations are still classical (i.e. the implementation techniques of information theory becomes quantum mechanical, while the theory itself remains classical).

The difference between classical computation and quantum computation is that the classical computer system is always is one of defined mutually exclusive set of states, while a quantum system can exist in what is known as superposition state. Hence, a quantum computer can process different input in parallel and produces a superposition of outputs (Faraj, 1999).

Quantum computer is a proposed mean of using quantum mechanical effects and phenomenon to achieve efficient computation and realize a new mode of information processing. When the computational model is implemented in a physical device, it must be able to adapt different internal states and provide means to perform the necessary transformation (Braunstein, 1995; Butler and Hartel, 1999; Omar, 1996). Quantum computer is quite similar to that of a classical computer, both of which have registers. The register in quantum computer initially is set to some initial state. As the quantum computation progresses, the register will be transformed via a quantum mechanical transformation. The register may then be observed to produce the output. The difference of the quantum computer is that this machine relays on the quantum phenomenon such as interference an entanglement in the computation process, the output of any process is produced through the observation of the register. This cycle will be repeated several times before the computation is complete (McCubbin, 2000).

The power of quantum computers comes from the possibility of employing a superposition of state that allows a massive parallel data processing within one piece of quantum hardware. As a result quantum computers can efficiently solve some problems which are believed to be intractable on any classical computer (Chi and Kim, 1997).

Corresponding Author: Ahmed T. Sadik, Department of Computer Science, University of Technology, Baghdad

A classical computer operates on strings of 0's and 1's, such as 1110010101011000, converting them into other strings. Each position in such a string is called a bit and contains either a 0 or a 1. The only thing a step in a classical computation can do to the state of a bit is to leave it alone, or flip it. A quantum computation can do much more to a quantum bit because quantum bits can have a much broader variety of possible states (Mermin, 2000).

QUANTUM BITS AND QUANTUM REGISTER

In a quantum computer, the fundamental unit of information is represented as the state of quantum subsystems, so the electric potential can be replaced by some quantum state: the quantum bit (qubit for short). Just as a bit has a state 0 or 1, a qubit also has a state $|0\rangle$ or $|1\rangle$. This is called the Dirac notation and it is the standard notation for states in quantum mechanics. The difference between bits and qubits is that a qubit $|\Psi\rangle$ can also be in a linear combination of states $|0\rangle$ and $|1\rangle$:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

This state is called a superposition of the states $|0\rangle$ and $|1\rangle$ with amplitudes α and β (α and β may be complex numbers). Thus, the state $|\Psi\rangle$ is a vector in a two-dimensional complex vector space, where the states $|0\rangle$ and $|1\rangle$ form an orthonormal basis, called the computational basis.

To achieve any computation we need an operator to change a state to another one that will lead to make a progress in the computation process. In quantum computers these operators represent an interaction with the quantum state.

Since most unitary transformations are linear operator, therefore these operators are applied simultaneously to all the base vectors of a quantum state, thus:

$$U\sum_{i=0}^{n} C_i\,|i\rangle = \sum_{i=0}^{n} C_i U\,|i\rangle \tag{2}$$

This unique feature of quantum computers is called quantum parallelism.

GROVER'S ALGORITHM

Suppose we have an unstructured database with N elements. Without loss of generality, suppose that the elements are numbers from 0 to N-1. The elements are not ordered. Classically, we would test each element at a time, until we hit the one searched for. This takes an average of N/2 attempts and N in the worst case, therefore the complexity is O (N). As we will see, using Quantum Mechanics only $O(\sqrt{N})$ trials are needed to achieve this task using Grover's Algorithm (Viamontes *et al*., 2005). For simplicity, assume that $N = 2^n$, for some integer n. Grover's algorithm has n qubits register. The first step is to create a superposition of all 2^n computational basis states $\{|0\rangle \ldots |2^n - 1\rangle\}$ of the register. The details of Grover's algorithm in (Viamontes *et al*., 2005).

BOYER, BASSARED, HOYER AND TAPP (BBHT) ALGORITHM

When the number of solutions is known in advance, we can use Grover algorithm to look for one of them. Without previous knowledge of the number of solutions t marked by the oracle we cannot use Grover algorithm. This impossibility arises because in the amplitude amplification process we cannot compute the number of iterations to be performed in order to maximize the coefficients of the solution. However, it is possible to use a quantum algorithm called BBHT for finding a solution in a set of items $\{T_i\}_{i=0\ldots N-1}$ given an oracle that recognizes a solution.

DÜRR-HOYER ALGORITHM

The Dürr-Hoyer algorithm is a quantum algorithm for finding the minimum within an unsorted table of N items. The core of the algorithm is a procedure which returns an index of items smaller than the item determined by a particular threshold index by using the BBHT algorithm. This procedure is iterated until the minimum is reached (Durr and Hoyer, 1999; Malossini *et al*., 2004).

THE GROVER'S BASED ROUTING ALGORITHM

In this research, Grover Algorithm and Dürr-Hoyer Algorithm are exploited to build a shortest path routing algorithm (here after we will call it Routing Variant Grover Algorithm RVGA). In RVGA we have modified Grover's Algorithm to overcome an obstacle that make this algorithm sometimes limited in solving wide spectrums of problems, one of these are the problems concerns the shortest path which is the basis of routing algorithms design (Rana and Ghany, 2006).

Initially the algorithm, which is based on the link state routing approach, should be provided with the source and destination nodes that the network needs to compute the shortest path between them, shortest path here in this algorithm is considered as the cost of information

transmission across the network or the metric length of the path between two nodes within the network. Also the algorithm should provide with a table of links within the nodes to define the connectivity of each node within the network. if the algorithm consists of N nodes N={1…N}, the link is defined as pair as follows:

$$l(m, n), \text{ where } m, n \in N.$$

In addition to the connectivity of the nodes, the algorithm should be provided with cost C_{mn} of each of these links, where:

$$C_{mn} \geq 0.$$

The cost of the path $p= (m_1,\ldots,m_k)$, is the sum of all costs of all the connections within this path

$$C(P) = \sum_{i=1}^{k-1} Cm_i m_{i+1} \quad (3)$$

The RVGA Algorithm

Input: S = Source node, D = Destination node, N = No. of nodes in the network.

Output: The Path P_D, the best path from S.

Initialization

$$\text{Link} = \begin{cases} 1 & \text{If m and n are directly connected} \\ 0 & \text{Otherwise} \end{cases}$$

$$\text{Cost} = \begin{cases} C_{mn} \geq 0 & \text{If m and n are directly connected} \\ \infty & \text{Otherwise} \end{cases}$$

M = no. of links in the network.

Register 1 $^{(t)}$ = A register to store the desired items resulted from time step (t-1), having three fields for each item (node name, node cost and path history).

Register 2 $^{(t)}$ = A register to store all the nodes connected to the node in Register1 at time step (t-1), having four fields for each item (node name, node cost, node amplitude and path history).

Initially Register 2 is in Grounded state $|0,\ldots..0\rangle_{\log(\text{count})}$.

Node database register = contains the cost and shortest path of each node at time (t).

Compute the rotation angle

$$\theta = \arccos\frac{\sqrt{M}}{N}$$

Compute the number of Grover's iterations $k = \pi - \theta/2\theta$.

Put in Register1, all the child nodes of S.

Repeat

For $p = p_1,\ldots,p_x$, (p_i: any nodes in regester1), compute:

Node(p_i) .cost=cost (S, p_i) and

Node(p_i) .history=[S P_i].

For $p = p_1,\ldots,p_x$, (p_i : any nodes in regester 1), generate the child nodes of p_i and put them in register 2.

For m=1,…,count, where count= no. of nodes in register 2, compute:

Node(X_m). cost= Node(P). cost +cost (P, X_m), where P is the parent node of X_m.

Node(X_m). history= Node(P). history $\cup X_m$

Apply the H transform: Node(X_m). amplitude =

$$\frac{1}{\sqrt{\text{count}}}$$

Apply the DH-RVGA Oracle Function to sign the desired nodes.

Apply the diffusion function:

$$|\psi_k\rangle = \sin\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|u\rangle + \cos\left[\left(\frac{2*k+1}{2}\right)*\theta\right]|i_0\rangle$$

Measure Register 2 to find the solutions $w = (w_1,\ldots.w_y)$.

Put $w=(w_1,\ldots.w_y)$, in register1

Until y = 0

Print P_D or the best path from S to all nodes in the network from the node database.

RESULTS AND DISCUSSION

Figure 1 and Table 1 shows a comparison in the number of iterations between the Dijkstra Algorithm and the proposed algorithms (RVGA), the figure shows the

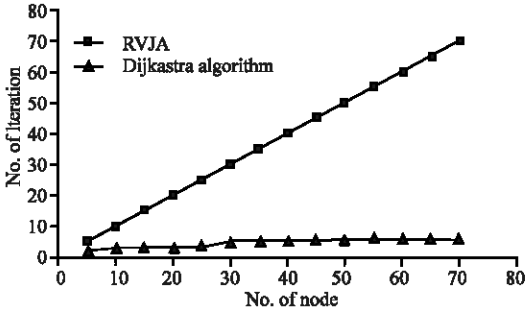

Fig. 1: A comparison in the number of iterations between the Genetic Algorithm and the proposed algorithms

Table 1: A comparison of the time complexity

Algorithm	Time complexity
Dijkstra Algorithm	$O(N^2)$, N: No. of nodes
RVGA and QQRA	$O(M * \sqrt{K})$ M: No. of iterations, K: no. of nodes Register2 at each of M value

difference in required iterations to get the optimal path between the Dijkstra algorithm and the proposed algorithms.

CONCLUSION

The following points are concluded from this research:

- It has been found it is not necessary to know the number of solutions or the number of points in the solution space that satisfy the conditions of solutions. It's possible to substitute that with feature related to the problem itself not the number of solutions.
- The number of iterations of these algorithms is found smaller than the Dijkstra algorithm and the genetic algorithm. Also the time complexity of each of these iterations is better than the time complexity of the Dijkstra's algorithm and the Genetic algorithm.
- When the RVGA is used, the whole routing table from a certain source to all destinations is computed in one query just as the Dijkstar algorithm. The genetic based routing algorithm requires N-1 (where N is the number of nodes in the network) runs of the algorithm from a certain source to all other nodes in the network. That is what makes the proposed algorithm fitter than the genetic algorithm for routing algorithms.

REFERENCES

Braunstein, S.L., 1995. Quantum Computation: A tutorial, http://www.users.cs.york.ac.uk.

Butler, M. and P. Hartel, 1999. Reasoning about Grover's Quantum Search Algorithm Using Probabilistic WP, ACM Transaction on Programming Language and Systems, 21: 417-429.

Chi, D.P. and J. Kim, 1997. Quantum Computation, Technical paper, Department of Mathematics. Seoul National University.

Durr, C. and P. Hoyer, 1999. A Quantum algorithm for Finding the Minimum, quanta-ph/9607014 V2.

Faraj, S.T., 1999. Quantum Cryptographic Key Distribution in Optical Communication Network, PhD. Thesis, Nahrain University, Iraq.

Huston, G., 2004. An Introduction to IP routing, Technical Report, www.potaroo.com.

Malossini, A., E. Blanzieri, E. Calarcot, 2004. QGA: A Quantum Genetic Algorithm, Technical Report, University of Trento, Department of Information and Communication Technology.

Mc Cubbin, C.B., 2000. Open QUACS, an Open Source Quantum Computation Simulator in Maple, MSc. Thesis, Faculty of Graduate School.

Mermin, N.D., 2000. Lecture Notes on Quantum Computation and Information Theory, Cornell University, Phys., pp: 481-681, CS 483.

Ömar, B., 1996. Simulation of Quantum Computers, MSc. thesis, Department of Theoretical Physics, Technical University of Vienna.

Rana, F. Ghany, 2006. Quantum Based Algorithms for Computer Network Routing, PhD Thesis, University of Technology, Iraq.

Viamontes, G.F., I.L. Markov and J.P. Hayes, 2005. Is Quantum Search Practical? IEEE. Comput. Sci. Eng., pp: 22-30.