Research Journal of Applied Sciences, Engineering and Technology 6(4): 622-630, 2013
DOI:10.19026/rjaset.6.4172
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: August 29, 2012 Accepted: October 22, 2012 Published: June 20, 2013

Research Article

More Improvement by Helping Ant to Fault-Tolerant Heuristic Routing Algorithm in Mesh Networks

[1]Alireza Soleimany and [2]Somayeh Azmoodeh
[1]Department of Computer Engineering, Meshkin Shahr Branch, Islaminc Azad University, Meshkin Shahr, Iran
[2]Department of Computer Engineering, Amirkabir University, Tehran, Iran

Abstract: Routing with fault-tolerant mechanisms has a crucial effect on the fast exchange of information in variety of networks including mesh networks. This study attempts to choose an optimal path in terms of fault tolerance to transmit messages from source to destination while taking into account faulty nodes in such mesh networks. In this study, we take advantage of ant colony optimization algorithm to propose Adaptive Heuristic Routing algorithms to this problem. We use color pheromone ants to overcome problem of fail-recover behavior of network components. The proposed method is compared with fault-tolerant routing algorithm in mesh networks using the balanced ring. Simulation results depict that this method reacted quickly in terms of network faults, meanwhile in each time step the data can choose the optimal path to reach their destination. In this study, we improve performance of the proposed method using update ants to inform other nodes about the discovered shortest path. Simulation results show that the proposed method dramaticcaly increase efficiency of routing mechanism in mesh networks.

Keywords: Ant colony, computer networks, fault-tolerant, mesh networks, network routing, update ant

INTRODUCTION

The direct interconnection networks (Duato *et al*., 2003; Dally, 2000;Narasimha *et al*., 2005) have been extensively studied as the critical components in many fields, such as multiprocessor systems, I/O interconnect and supercomputers. Recently, direct interconnection networks have also become popular architectures for switching fabrics in the core routers. Mesh and torus networks are typical families of direct interconnection networks. Besides their use in terabit routers, these networks have also been used in supercomputers or parallel computers (Dally, 2000; Narasimha *et al*., 2005) such as SGIOrigin 2000, iPSC860, CrayT3D,CrayT3E and BlueGene/L. Recently, it has also been used for I/O interconnect in infiniBand architecture (Pfister, 2001).

One of the main concerns in computer networks is associated with the correct Routing function (Chien and Kim, 1992; R.V. Boppana and Chalasani, 1995; Seong-Pyo and Taisook Han, 1998; Wu, 2003; S. Chalasani and Boppana, 1997; Zhou and Lau, 2001). As the size of network increases, the probability of fault among the nodes and connecting links increases too. Therefore, it is essential to design and implement a fault-tolerant routing algorithm in order to deliver messages to their destinations in the presence of Faulty Nodes. Zhou and Lau (2001) and Jipeng and Francis (2004) presented a fault-tolerant routing in mesh networks in which data constitute two routes in moving from source to destination, clock wise and vice versa and after comparison the shortest path was chosen as final route. This method is not satisfactory when there are more than one region. In Boppana and Chalasani (1995) and Seong-Pyo and Taisook Han (1998), it was introduced a fault tolerance routing method by using wormhole method in which after occurring fault some virtual channels are used for routing. This method is suitable for fault tolerant routing with several fault regions. But, the main problem in this case is the additional routes which must be traveled in getting to achieve a fault chain that was not good. Zhang and Chiu (Pao-Hwa and Sheng-De, 1999; Huei-Huang and Ge-Ming, 2002) examined an optimized fault tolerance routing in mesh networks with convex faults using two virtual channels in multiple fault region. Su and Wang also introduced another method in mesh networks for fault tolerance routing that after forming fault region, regions of large faults are divided into parts with small fault region in order to prevent nodes deactivation (Pao-Hwa and Sheng-De, 1999).

Chen and Chiu expressed a fault tolerant routing method in mesh network based on nodes labeling technique (Kuo-Hsuan and Ge-Ming, 1998). Huaxi and Zhang introduced a fault tolerant routing in mesh

Corresponding Author: Alireza Soleimany, Department of Computer Engineering, Meshkin Shahr Branch, Islaminc Azad University, Meshkin Shahr, Iran

networks using balanced rings (Huei *et al*., 2007). Linder and Harden extend the concept of virtual channel to multiple virtual interconnection networks that provide adaptability and fault-tolerance (Linder and Harden, 1991). Messages routing in a virtual network are constrained to travel the virtual networks in a pre-defined order. The main disadvantage of their method is that the number of virtual channels per physical channel required is large. Chien and Kim presented (Chien and Kim, 1992) a partially adaptive algorithm for mesh networks, which requires only three virtual channels per physical channel. However, their methods have to disable some nodes when faults are located on the boundaries of the meshes. Boppana and Chalasani solved this problem in Boppana and Chalasani (1995) and Chalasani and Boppana (1997) which uses four virtual channels. Kim and Taisook Han (Seong-Pyo and Taisook, 1998) proposed a fault-tolerant wormhole routing algorithm with four virtual channels in meshes with overlapping fault regions. Dally and Aoki (Dally and Aoki, 1993) present two adaptive routing algorithms for direct networks. Both algorithms, based on the concept of dimension reversal, are deadlock-free by eliminating cycles in the resource dependency graph. Faults are not expanded to be rectangular faults. However, the number of faults tolerated depends on the number of virtual channels used, in the static algorithm and depends on the locations of the faults, in the dynamic algorithm. Wu (Wu, 2003) proposed a fault-tolerant extended X–Y routing protocol, which is based on the odd–even turn model and does not use any virtual channels in 2D meshes. They use extended faulty blocks (disjointed rectangles), which consist of connected disabled and faulty nodes. Su and Shin presented fault-tolerant routing algorithms, which decompose the network into two virtual interconnection networks, VIN1 and VIN20VIN1supports deadlock-free routing and VIN2 support fully adaptive routing. VIN2 trades its adaptability for fault-tolerance when the requested channels in VIN1 are not connected to safe nodes. For mesh networks, faulty blocks are expanded to disconnected rectangular blocks, which are composed of unsafe and faulty nodes. Two rectangular blocks, B1 and B2, are disconnected, if for each node x1 ∈ B1 and x2 ∈ B2, the distance between x1 and x2 in at least one-dimension is no smaller than three. Expanding faulty blocks to be disconnected rectangular faults may cause even more nodes disabled than just expanding faulty blocks to be rectangular faults. Although only two virtual channels per physical channel are needed in their method, deadlocks can also occur under some fault situation (Su and Shin, 1996).

Chalasani and Boppana proposed algorithms that can tolerate some special convex faults, such as L, T or +, with four virtual channels in meshes. In Zhou and Lau (2001) and Jipeng and Francis (2004), Zhou and Lau proposed a fault-tolerant routing algorithm with three virtual channels that can tolerate convex faults. As for torus, it is easy to apply the above algorithms by simply doubling the number of virtual channels. Hence, the number of the algorithms for torus is relatively less than those for mesh. However, virtual channels may not be fully utilized in this way.

The fault-tolerant routing algorithm specifies the best route between source and destination. A good fault-tolerant routing algorithm must be free from lock-live and starvation. Lock-live may act as a depended cycle among those nodes that request resources. Therefore, any ahead movement will not be accomplished. By "Lock-live", it is meant that the pockets are moving without any useful movement toward destination node. Starvation occurs when a pocket demands a channel in the buffer but that channel is allocated frequently by other packets.

In Dally and Aoki (1993), we developed an efficient routing method based on ant colony system. This study aims at finding an effective algorithm to perform a fault-tolerant routing in Mesh networks. To be precise, Intelligent agent, is an existent in which can perceive its surrounding environment by sensors and also can affect its environment by effectors. In this study five types of ants have been used to fault-tolerant routing of Mesh networks. To improve performance of routing mechanism based on ant colony with colored oheromone, we introduce a new type of ant called update ant. Indeed, update ant update new discovered shortest paths in neighbours.

Ant colony with helping ants and colorful pheromone: Most of the multi agent-based algorithms take their inspiration from ants' behavior. Real ants are able to find shortest path between their nest and food source by following pheromone trail of other ants. For example Schonderwoerd *et al*. (1997) implemented ant-like agents for routing (Schoonderwoerd *et al.*, 1997). In his algorithm, each source node s sends an ant toward destination d at regular intervals, where d is selected in a random scheme. When ants reach node i, select their next hop n to their destination according to routing table of node i, then update node i's routing table. They increase the probability of choosing n as a next hop (increasing the pheromone) while selection probability of other neighbors is decreased for destination (Di Caro and Dorigo, 1997) also introduced a new algorithm based on ant behavior (Ant Net) (Di Caro and Dorigo, 1997). In their method two types of ants exist, forward and backward ants. Similar to the Schonderwoerd's algorithm, each node sends a forward ant to different destinations at regular intervals. But in this algorithm the forward ant does not update the routing tables of nodes that it visits, its only duty is to find a path to destination d and to collect information. When a forward ant arrives at its destination d, it generates a backward ant and dies. The backward ant then goes back in the same path as the forward ant that created it and updates the routing tables for intermediate and destination nodes.

The above AntNet has received considerable attention by various researchers. For example, B. Baran and R. Sosa improved AntNet by proposing an intelligent initialization of routing tables, an intelligent update after network resource failures and a noisy decision making against undesirable networks ‘‘freezing’’ their routing probabilities in dynamic environment (Baran and Sosa, 2001). Later, in 2002, Kassabalidis and El-Sharkawi, M.A. showed that for large networks good routing solutions can be achieved by combined use of network clustering, autonomous systems and ant colony (Kassabalidis *et al*., 2002). In other research, AntNet is used in routing for ad hoc wireless networks. For example, Marwaha, S., *et al*. introduced a novel method for routing based on AntNet and AODV (Adhoc On Distance Vector routing) (Marwaha *et al*., 2001). AntNet is also used in QoS routing by Subing and Zemin (2000) among others.

As several of the above research have pointed out, while strong in regards to distributed routing, Standard AntNet still has a weakness in terms of speed of convergence and response to network changes/failures. While the gradual process of pheromone aggregation and evaporation allows AntNet to reach its globally optimal performance with only distributed information, the inability of nodes to directly share knowledge among each other may cause the algorithm to respond too slowly to the rapid network changes. In other words, AntNet’s strength in a totally indirect and distributed information sharing may be its own fallacy as well. This is while, in addition to aggregation and evaporation, natural ant colonies do have a method of pheromone propagation as a method of sharing information among neighboring nodes.

Authors in Soltani *et al*. (2004) and Akbarzadeh *et al*. (2005) introduced a new type of helping ants to increase cooperation among neighboring nodes, thereby reducing AntNet algorithm’s convergence This concept of helping ants, from a naturalistic perspective, is inspired by the fact that insect coordination via pheromones relies on at least three dynamics: aggregation to successive deposits by different ants, evaporation of pheromone to discard obsolete information and propagation of pheromone from one location to other nearby locations to share information. ACO uses aggregation and evaporation, but does not account for pheromone propagation. In the proposed algorithm, this natural process of pheromone propagation is considered by introducing a novel type of helping ants.

In Alireza and Shahriyar (2011), a novel method for adaptive routing using AntNet was proposed. It efficiently routes data packets to destination event if there are faulty nodes. In this study we propose Helping Ant to achieve more improvements by ACO.

THE PROPOSED METHODS

The proposed algorithm in this study is based on the idea of colored ants including two ants in red and blue colors. Ants are capable of finding the shortest path between a food source and the nest without having any kind of visual information. However, in mesh networks with many nodes, faults can occur frequently and dynamically and nodes may be recovered during the lifetime of network. Hence, recovery mechanism needs to be designed to recover from faults. In this study, we propose to take advantage of forward ants aiming at investigating new paths. Besides, we propose to use new type of ant called Helping Ants or Update Ants to inform neighbors about finding new shortest path to destination node.

The proposed method use colored pheromone to quickly adapt with topology changes. Our customized ACO contains five different types of ants including:

Blue Forward Ant (BFANT): It discovers all routes including new established route to find the best one. It follows blue and red tracks to destination. Indeed, it enforces the blue pheromone value of path. In first, it selects the links with highest value of red pheromone. Then, if there isn’t any link with red pheromone, it chooses the links with highest value of blue pheromone

Discover Forward Ant (DFANT): It tries to discover only new established routes, which have minimum value of blue pheromone. Its aim is to probe recovered links.

Blue Backward Ant (BBANT): It is a response to BFANT, when it received by destination node.

Red Backward Ant (RBA): When a destination receives a Red Forward Ant, it sends Red Backward Ant to highlight the discovered route. This ant marks the path to direct to RFANTs.

Helping Ant (HA): It is sent to neighbors to inform about new discovered path.

Figure 1 shows handle function of blue forwarding ant. There are three situations when node receives a BFANT. All three situations are presented in Fig. 1. Blue ants are able to balance traffices over multiple available paths, which have approximately same value of pheromone(same type of pheromone).otherwise, convergance is done very slowly. To speed up this convergance especially when a new shortest path is discovered, we propose Dants. BFAnt are launched by source periodically to maintain network connectivity while Dants are sent may be launched by every node, which its neighbour node has been recovered. Indeed, it be sent to explore new available paths. So, we have to

```
------------------------------------------------------------------------------------
Source Node
------------------------------------------------------------------------------------
     Launch periodically BFANT to adaptively route packet to destination node
------------------------------------------------------------------------------------
Intermediate node
------------------------------------------------------------------------------------
If (there is at least one path with Red Pheromone) then
        {
               Compute forwarding probability according to values of Red Pheromoness
               Forward fant in according to forwarding probabilities;
               Update value of Blue Pheromone;
        }
        Else If (there is at least one path with Blue Pheromone) then
        {
               Compute forwarding probability according to values of Blue Pheromone
               Forward fant in according to forwarding probabilities;
               Update value of Blue Pheromone;
                                                        }
        Else
        {
                              Randomly select one path;
                              Update Blue Pheromone and forward fant;
        }
     ------------------------------------------------------------------------------------
Destination node
------------------------------------------------------------------------------------
                  Create a Blue BANT and backward to source;
------------------------------------------------------------------------------------
```

Fig. 1: Handling function of Blue FANT

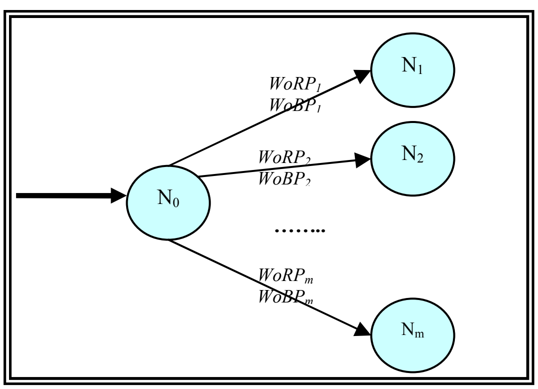

Fig. 2: Different value of Pheromone and making a decision

propose an efficient mechanism to direct ant without considering weight of pheromone. In the proposed method, we give more priority to red colored pheromone to blue colored pheromone. As mentioned previously, the main purpose of BAnts is to forward data packet via available topology. Dants add more available routes to current topology.

If some links of one node has blue and red pheromone, forwarding probabilities have to be computed by formulas 1 and 2. Forwarding probability for a link is a positive value between 0 and 1, which it determines probability of selection. There are different links from one node to other nodes, so BFANT have to make a decision. Figure 2 Shows process of making Decision. WoBP (Weight of Blue Pheromone) and WoRP (Weight of Red Pheromone) are pheromone updated by different ants to adapt routing mechanism with changes of networks. The propose method, FT-HRA, uses different values of pheromone to forward ants toward destination:

$$SWoBP = WoBP_1 + WoBP_2 + ... + WoBP_{n-1} + WoBP_n = \sum_{i=1}^{n} WoBP_i \quad (1)$$

$$SWoRP = WoRP_1 + WoRP_2 + ... + WoRP_{n-1} + WoRP_n = \sum_{i=1}^{n} WoRP_i \quad (2)$$

Formulas 1 and 2 compute sum of red and blue pheromones. These pheromones are used to determine forwarding probability. Formula 3% three different states in distributing frowarding ants. N_i is number of FANTs sending via link$_i$:

$$\begin{cases} N_i = N(WoRP_i / SWoRP) & SWoRP \neq 0 \\ N_i = N(WoBP_i / SWoBP) & SWoRP = 0 \\ N \ \ randomly \ \ distributed & SWoRP = 0 \ \ and \ \ SWoBP = 0 \end{cases} \quad (3)$$

The main purpose of Blue Forward Ants is to strengthen the discovered shortest paths while the

```
-----------------------------------------------------------------------------------------------------
Intermediate node
-----------------------------------------------------------------------------------------------------
    If(there is new recovered neighbour node)
            Launch Dants to explore new shortest path via new links
            All Dants record current delay to destination node
Else
{
    if( CDTC <DTD_DANT)
    if( there is at least one link has no pheromone)
    {
            Forward all DANTs equally to the links
        Update values of Red Pheromone
    }
    else if(there is at least one link with only blue pheromone)
    {
        For each link, which has only blue pheromone
            {
        Compute forwarding probability according to reverse values of Blue Pheromoness;
        Forward fant in according to forwarding probabilities;
        Update value of Red Pheromone;
                    }
    }
    else
    {
        Compute forwarding probability according to reverse values of Red Pheromoness;
        Forward fant in according to forwarding probabilities;
        Update value of Red Pheromone;
    }
}
-----------------------------------------------------------------------------------------------------
Destination node
-----------------------------------------------------------------------------------------------------
    Create a Red BANT and backward to source with determined weight of pheromone;
```

Fig. 3: Handling function of Discover FANT

important purpose of Discover ant is to find new shortest path. So , Dant does not follow BFANTs. Figure 3 presents handle function of DANT. Dant records current node's delay to destination node. They use it to compare with their round trip time. If their delay is greater that the recorded delay, they are killed by destination node.

Each DFANT records delay time from source node to destination node as DTD (Delay to Destination node). It cumulates delays of traversed link as CDTC (Cummulative Delay to Current node). One DFANT is forwarded by one node when CDTC is less than DTD_{DANT}. Value of CDTC is computed by formula 3. CDTC cumulates delay of nodes and links from source node to destination node:

$$CDTC = \sum_{i=0}^{n} (nodeDelay + linkDelay) \quad (4)$$

In mesh networks, ant colony algorithm must be able to discover new paths with least cost from source to destination, However, in ordinary ant colony algorithm, there are forward ants, so called *FANT*, which pass through the routes that have the strongest acid. Such ants themselves are not able to discover new and shortest paths in mesh environments with high rate of fault. As a result, for this purpose new type of ants called DFANT is introduced.

The proposed algorithm is designed in a way that can produce new coloured ants, so called *DFANT*. *DFANTs* can explore and discover new and unknown paths. In each intermediate node, reverse value of red and blue pheromones are aggregated. Formula 4 and 5 presets how reverse values are aggregated. SoRBW is Sum of Reverse Blue Weight and SoRRW is Sum of Reverse Red Weight.

$$SoRBW = \sum_{i=1}^{n} 1/WoBP_i \quad \forall i \varepsilon n \quad if(WoRP_i = 0 \quad and \quad WoBP_i \neq 0) \quad (5)$$

$$SoRRW = \sum_{i=1}^{n} 1/WoRP_i \quad \forall i \varepsilon n \quad if(WoRP_i \neq 0) \quad (6)$$

forwarding probabilities are computed according to SoRBW and SoRRW values for each link. in handling of DFANT, FT-HRA initially exclude links traversed by DFANT due to be able to discover new routes(formulas 4 and6). Then, if there is not at least one link without red pheromone, it distribute DANTs according reverse value of red pheromone (formulas 6 and 8).

$$N_i = N * ((1/WoBP_i)/SoRBW) \quad (7)$$

$$N_i = N * ((1/WoRP_i)/SoRRW) \quad (8)$$

Handling of Helping Ant: The main purpose of Helping Ants is to inform neighbour nodes about the found shortest path. It launched just a new path is found. Each helping ant carries information about value of different pheromones. Indeed, Helping Ant or Update Ant is launched just one Red Backward Ant is received. Helping Ant informs neighbour nodes about new shortest path and its Red Weight. One neighbour node receiving HANT update weight of red phermone based on formula (9):

$$WoRP_{Neighbour} = \alpha * WoRP + \alpha * WoRP_{Neighbour} + (1-\alpha) * WoRP_{node} \quad (9)$$

EXPERIMENTS RESULTS

CBR traffics are transmitted from node 0 to other nodes where there are several good alternative paths for packet routing. AntNet can show its ability to find alternative routes better when traffic rate is higher than link bandwidth. Therefore, we consider our network to be under similar conditions, i.e. packet size is 512 bytes and mean packet inter-arrival time is changing to study its effect on the proposed method. The simulation runs for 300 seconds with the initial 20 seconds for priming the network. Immediately after the first 50 s of simulation is past, all link randomly start to broken and after about 10 sec are recovered. This experiment compares the behavior of the three algorithms in according packet delivery ration, end to end delay and throughput in a mesh networks with fault rate of 0.1 and 0.05.

In simulation study, we compare two ant based method in two different topology 8×8 and 16×16. Results of these experiments are summarized in Table 1and 2.

In the first simulation, three methods are run in a network with fault rate of 0.05. simulation results depict that ant-based methods has dominant improvement to B-ring method. Indeed, FT-HRA and EFT-HRA are able to overcome faults of nodes and detect new shortest path. Helping ants in EFT-HRA propose more improvement due to their informing to neighbour nodes. In two scenarios, packet delivery ratio, end to end delay and throughput are measured based on different input traffic. Input traffic increases from 0.1 mega bit per second to 0.5 megabit per second. Figure 4 presents end to end delay for different methods. EFT-HRA has the lowest delay due to efficient mechanisms.

Figure 5 shows throughput of different methods. Throughput is an important issue showing performance of routing mechanism. The proposed method discovers the best path for different services and update the best path with update Ants. Figure 6 shows another issue, packet delivery ratio. The proposed methods is able to detour the failed links and nodes.

Table 1: Average results of FT-HRA and EFT-HRA in a 8*8 Mesh Network

Average results	0-100s	100-200s	200-300s
FT-HRA			
Average ETE delay	231.46ms	201.292ms	242.417ms
Average jitter	283.53ms	263.374ms	183.2974ms
Packet delivery ratio	0.8264	0.8173	0.86374
# Ant packets	23465.8	46283.9	71827.37
EFT-HRA			
Average ETE delay	265.48ms	198.283ms	181.25ms
Average jitter	216.38ms	196.8ms	102.8ms
Packet delivery ratio	0.87293	0.8628	0.88928
# Ant packets	25938.4	33827.5	53829.6

Table 2: Average results of FT-HRA and EFT-HRA in a 16*16 Mesh Network

Average results	0-100s	100-200s	200-300s
FT-HRA			
Average ETE delay	562.39 ms	672.327 ms	632.38 ms
Average jitter	332.26 ms	437.82 ms	376.27 ms
Packet delivery ratio	0.8173	0.74278	0.78293
# Ant packets	134627.38	482791.02	781628.8
EFT-HRA			
Average ETE delay	437.38 ms	529.19 ms	386.92 ms
Average jitter	271.7 ms	318.8 ms	201.2 ms
Packet delivery ratio	0.844228	0.8136	0.8618
# Ant packets	178263.9	326479.1	539183.8

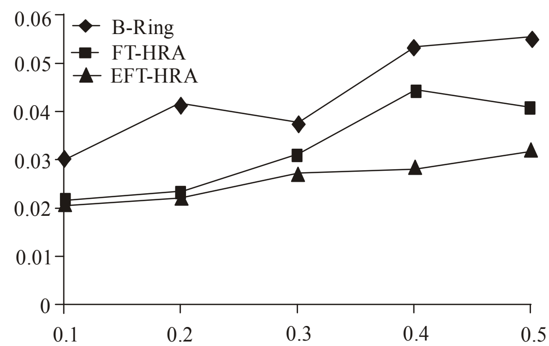

Fig. 4: ETE delay as function of load (Mbps) when fault rate is 0.05

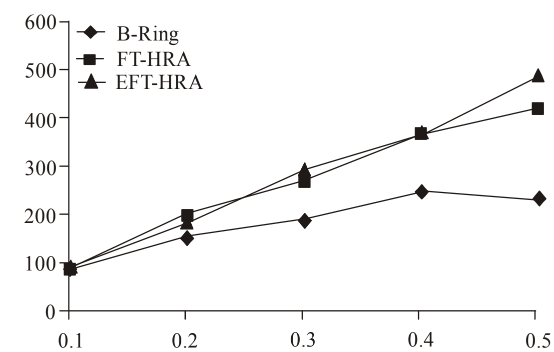

Fig. 5: Throughput as function of load (Mbps) when fault rate is 0.05

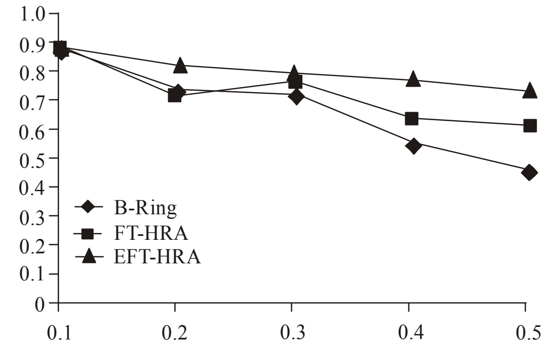

Fig. 6: Packet delivery ratio as function of load (Mbps) when fault rate is 0.05

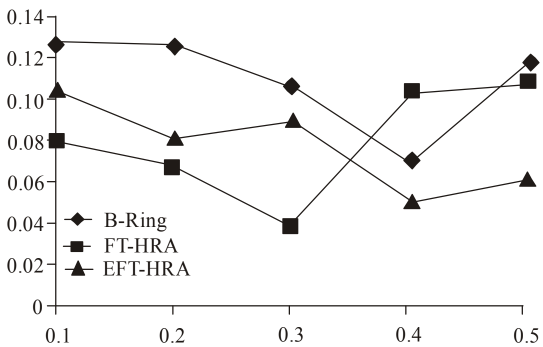

Fig. 7: ETE delay as function of load (Mbps) when fault rate is 0.1

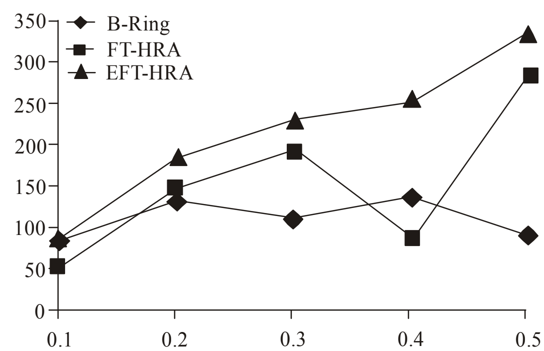

Fig. 8: Throughput as function of load (Mbps) when fault rate is 0.1

In the second scenario, we increase the fault rate, to study the performance of th proposed methods under different input loads. Figures 7, 8 and 9 present result of the different methods. when fault rate is high, B-ring's performance decreases substantially. In other hand, the proposed methods present more tolerancy.

Nodes failing with rate of 0.05 and recovered after a period of time about 10 seconds. The proposed methods (FT-HRA and EFT-HRA) are able to detect new links just after being activated. This ability leads to higher improvement in some efficient indicators.

Throughput chart displays that ants deliver more efficiently data trffic to destination nodes. So,

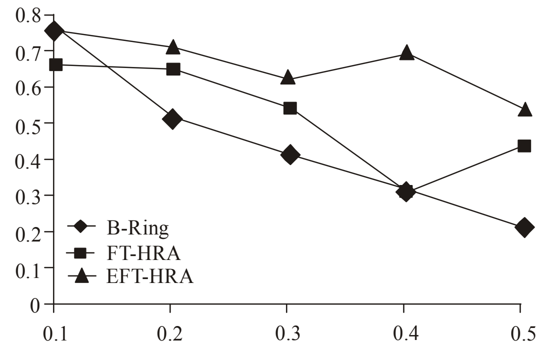

Fig. 9: Packet delivery ratio as function of load (Mbps) when fault rate is 0.1

throughput and packet delivery ratio in the proposed method are greater than B-ring method.

In the second scenario, fault rate is increased and performance of three methods is compared. However, higher fault rate results in less efficientcy in three method, EFT-HRA presents noticeable more tolerancy. In this scenario, traffic flows increase from 0.1 Mbps to 0.5.

To study the proposed methods, we simulate both of them in two different topology 8×8 and 16×16. Table 1 and 2 presents different parameters in three periods of time: 0-100,100-200 and 200-300. In these periods, methods of FT-HRA and EFT-HRA are compared.

In these simulations, number of different ants has been presented to show volume of overhead traffic. EFT-HRA has an extra type of control packet, so, its number of control packets is greater than FT-HRA. In Table 2, topology of mesh network is a 16*16 mesh. In this situation, ants could discover more possible paths while average length of path is greater than a 8*8 mesh.

To examine efficiency of the proposed algorithm, it has been compared experimentally in a network model using Ant Colony and *B-ring* algorithm for 20 minutes of time. To present performance of the proposed method , We have simulated B-ring algorithm, our proposed method with and without helping ants in a mesh network with different percent of faulty nodes.

Performance of the routing algorithms is evaluated in terms of two main metrics: End-To-End delay (E2E delay) and network throughput. The E2E delay is defined as the average time starting from the packet generation to the time packet reaches the destination. Also, we uses normalized throughput, which is equal to the number of packets that can be transmitted through the channel at the maximum load in a period of time.

CONCLUSION

This study has presented a fault-tolerant algorithm in mesh computer networks using an optimized ant colony algorithm. Ant colony algorithm is known one of the most powerful methods in terms of optimization.

In addition to FANT and BANT ants, another types of ant called DANT and UANT are presented to decrease delay, increase throughput and discover new shortest path in a faulty environment. Update ants aim to inform neighbour nodes about the discovered shortest path.

A good recommendation dealing with future consists of fault-tolerant routing in mesh computer networks by using ant colony optimized algorithm with cluster method in order to accelerate the routing operation. In cluster method the mesh region has been divided into four parts. To transfer the message from source to destination, first some algorithms are used to determine the region number of destination node then the message is sent.

REFERENCES

Akbarzadeh, M.R.T., A. Soltani and M. Naghibzadeh, 2005. A novel Ant Net-based algorithm with application to NSFNET. Proceedings of the 2005 International Conference on Modeling, Simulation and Optimization, Sharjah, UAE.

Alireza, S. and L. Shahriyar, 2011. A Fault-Tolerant Heuristic Routing Algorithm Using Ant Colonies in Mesh Networks. International Conference on Computer and Computer Intelligence (ICCCI), Paper 106, Retrieved from: Http:// Dx.Doi.Org/10.1115/1.859926

Boppana, R.V. and S. Chalasani, 1995. Fault-tolerant wormhole rout-ing algorithms for mesh networks. IEEE Trans. Comput., 44(7): 848-864.

Baran, B. and R. Sosa, 2001. Ant Net routing algorithm for data networks based on mobile agents: Inteligencia artificial. Artif. Intel. Rev. Iberoamericana, 12: 75-84.

Chalasani, S. and R.V. Boppana, 1997. Communication in multicom-puters with nonconvex faults. IEEE Trans. Comput., 46(5): 616-622.

Chien, A.A. and J.H. Kim, 1992. Planar-adaptive routing: Low-cost adaptive networks for multiprocessors. 19th Annual International Proceeding of Computer Architecture, pp: 268-277.

Dally, W.J., 2000. Scalable Switching Fabrics for Internet Routers. White paper, Avici Systems Inc., Retrieved from: http:// www. avici. com/ technology/whitepapers/TSRfabric-Whitepaper.pdf.

Dally, W.J. and H. Aoki, 1993. Deadlock-free adaptive routing in multicomputer networks using virtual channels. IEEE T. Parallel Distribut. Syst., 4(4): 466-475.

Di Caro, G. and M. Dorigo, 1997. Mobile agent for adaptive routing. Technical Report IRIDIA.

Duato, J., S. Yalamanchili and L. Ni, 2003. Interconnection Networks, an Engineering Approach. Morgan-Kaufmann Press, San Francisco.

Huei-Huang, C. and C. Ge-Ming, 2002. An improved fault-tolerant routing algorithm in Meshes with convex faults. Parallel Comput., 28: 133-149.

Huei, H.G., Z. Jie, W. Kun, L. Zengji and K. Guochang, 2007. Enhanced fault-tolerant routing algorithms using a concept of ‘‘Balanced Ring”. J. Sys. Architec., 53: 902-912.

Jipeng, Z. and C.M.L. Francis, 2004. Multi-phase minimal fault-tolerant wormhole routing in meshes. Parallel Comput., 30(3): 423-442.

Kassabalidis, I., M.A. El-Sharkawi, R.J. Marks, P. Arabshahi and A.A. Gray, 2002. Adaptive-SDR: Adaptive swarm-based distributed routing. Proceedings of the 2002 International Joint Conference on Neural Networks, Honolulu, HI, 1: 51-354.

Kuo-Hsuan, C. and C. Ge-Ming, 1998. Fault-tolerant routing algorithm for meshes without using virtual channels. J. Info. Sci. Eng., 14: 765-783.

Linder, D.H. and J.C. Harden, 1991. An adaptive and fault-tolerant wormhole routing strategy for K-Aryn-Cubes. IEEE T. Comput., 40(1): 2-12.

Marwaha, S., C.T. Khong and D. Serinvasan, 2001. A Novel Routing Protocol Using Mobile Agents and Reactive Route Discovery for Ad Hoc Wireless Networks. Retrieved from: hhttp:// www. ece.nus. edu.sg/stfpage/eletck/papers/ShivaICON_conf8a51 .pdfi.

Narasimha, R. A., A.B. Matthias, C. Dong, A. Gara, M. Tasso, *et al*., 2005. Blue Gene, L/torus interconnection network, IBM. J. Res. Dev., 49 (2-3): 265-276.

Pao-Hwa, S. and W. Sheng-De, 1999. Fault-tolerant wormhole routing in two dimensional Mesh Networks with convex faults. Info. Sci., 121: 217-231.

Pfister, G., 2001. An Introduction to the Infiniband Architecture, High Performance Mass Storage and Parallel I/O. IEEE Press, New Jersey, USA.

Schoonderwoerd, R., O. Holland and J. Bruten, 1997. Ant-like agent for load balancing in telecommunications networks. Proceeding of the 1st International Conference on Autonomous Agents, Marina del Rey, CA, pp: 209-216.

Seong-Pyo, K. and H. Taisook, 1998. Fault-tolerant adaptive wormhole routing in 2D mesh. IEICE Trans. Inform. Syst. E81-D, (10): 1064-1072.

Su, C.C. and K.G. Shin, 1996. Adaptive fault-tolerant deadlock-free routing in meshes and hypercubes. IEEE T. Comput., 45(6): 666-682.

Subing, Z. and L. Zemin, 2000. A Qos routing algorithm based on an algorithm. Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN), Tampa, FL, pp: 574-578.

Soltani, A., M.R.T. Akbarzadeh and M. Naghibzadeh, 2004. Introducing helping ants in Ant Net and implementationon NSFNet. Proceedings of the 2004 Iranian Conference on Electrical Engineering, Mashhad, Iran, (In Farsi).

Wu, J., 2003. A fault-tolerant and deadlock-free routing in 2D meshes based on odd-even turn model. IEEE Trans. Com-put., 52(9): 1154-1169.

Zhou, J.P. and F.C.M. Lau, 2001. Adaptive fault-tolerant wormhole routing in 2D meshes. 15th Annual International Parallel and Distributed Processing Symposium, San Fran-Cisco, pp: 56-61.