Fault Tolerant Routing in Wireless Cellular Networks: An Approach for maximizing Call Acceptance and reducing Delay

PARAG KUMAR GUHA THAKURTA[1], SUJOY SETT[2], SPANDAN CHOWDHURY[3], SWAPAN BHATTACHARYA[4], SUBHANSU BANDYOPADHYAY[5]
NIT Durgapur, India [1,2,3]
Jadavpur University, India [4]
Brainware University, India [5]
[1]parag.nitdgp@gmail.com, [2]sujoysett@gmail.com, [3]spandie990@gmail.com, [4]bswapan2000@yahoo.co.in, [5]subhansu@gmail.com

Abstract In this paper, a fault tolerant routing scheme in wireless cellular networks is proposed for improving call acceptance and simultaneously reducing delay, despite the existence of Base Station fault. The local routing tables on the neighbours are created to identify the preference of any neighbouring base station for forwarding data. Using an Ant Colony Optimization (ACO) based approach, faulty regions in the network are detected and call requests are efficiently routed around such regions, thus avoiding congestion. It determines the optimal routes in the network with reduced delay. The experimental results confirm the theoretical findings including the way to obtain convergence for the proposed approach and its effectiveness over existing approaches.

Keywords: routing, fault tolerance, wireless cellular network, call acceptance, delay, Ant Colony Optimization

(Received May 18th, 2016 / Accepted December 15th, 2016)

1 Introduction

Wireless cellular networks divide the geographical terrain into smaller regions called cells [10] which are usually assumed to be hexagonal for analytical and experimental purposes. Each of the cells is serviced by a base station (BS) and these BSs are interconnected by wireless media for establishing reliable communication between subscribers. It follows a packet forwarding [14] technique along an optimal route in terms of successive BSs accounting as the number of hops. The path length is such networks are defined no in terms of physical distance, but the number of in-between BSs, which is also termed as hop length of the route. The performance of a BS is influenced by various facts such as circuitry failures in BS due to power disruption, software dysfunction etc. The failure in normal functioning of BS under such inhibitory conditions is known as BS fault. Call acceptance seeks major attention during this faulty scenario. Additionally, reduction in subsequent delay under such a scenario becomes an important factor guiding the routing method.

In practice, it is not entirely possible to avoid faulty situation for handling faults. However, the procedure of local fault notification in network may result in termination of call connections and subsequently the service is suspended until the faulty BS is restored. Again, the fault removal techniques are often time consuming due to necessity of external intervention. The fault tolerant mechanism provides the ability of a system to respond gracefully during an unexpected hardware or software malfunctioning. Thus a message passing communication is fault tolerant if failures in some BSs do not prevent other BSs from communicating. Here, the adaptability afforded by multiple paths is the key to fault tolerance when a BS is truly unusable. The dynamic recovery [17] of faulty BSs often characterizes strong network architecture. In cellular networks, data initiates along any path towards its destination and then, route around faulty BSs as these are encountered. If such communication reaches a faulty BS or a cluster of the same, it would introduce congestion and detour. Thus tolerant systems avoid forwarding communication to such faulty regions.

To this date, several conventional and evolutionary computing approaches are utilized towards solving the fault tolerant routing. The evolutionary algorithms (EAs) [3] have gained popularity over the conventional approaches due to significant reduction in computation cost for larger networks. Furthermore, modern EAs are currently inspired by biological processes of nature [8]. Genetic algorithm (GA) [15] is one of such EA approaches which requires knowledge of entire network and the initial guess. However, this information is not always available in packet forwarding network. Ant colony optimization (ACO) [7] is another bio inspired methodology that finds application in instantaneous data forwarding. It is based on the concept that established solutions to a problem get reinforced by utilizing the foraging [8] behaviour of ants with a scope of finding an alternate optimal solution against the existing one. Here, ants deposit a chemical substrate called pheromone in their trail, which draws other ants to follow the same route. The deposited pheromone evaporates with time and the shorter path gets reinforced faster which ultimately leads to gradual deletion of the longer paths. Thus the use of ACO is beneficial over other approaches as it directly maps the mobile agents to ants transmitting packets towards solving routing problems in networks. In addition, the agents are capable to make instantaneous decision for adaptive routing procedure that simulates ant behaviour to obtain improved network service.

In this paper, an efficient fault tolerant routing scheme in wireless cellular networks is proposed for improving call acceptance and reducing delay in spite of existence of BS fault. The BS in each cell of the network is assumed as a node. The work concentrates on creating local routing table for the neighbours of the current node with respect to destination nodes. This information is used to identify the preference of any neighbour for forwarding data. Using an ACO based approach, the faulty regions in the network are detected and call requests are efficiently routed around such regions. Successful routing is used to determine the call acceptance for the system. The proposed approach also determines the optimal routes in the network, which in turn reduces delay by avoiding congestion in faulty regions. The experimental results confirm the theoretical findings as well as demonstrate its effectiveness over existing approaches.

Section 2 introduces existing approaches for completeness of the current work. Section 3 presents the system model. Next, the proposed approach is discussed in Section 4. The experimental results are shown in Section 5, which is followed by the conclusion and future research direction in Section 6.

2 Related Work

Various strategies for fault diagnosis have been studied till date for mobile networking systems. A survey of the fault detection techniques based on the nature of the tests and characteristics of nodes in the network is established by Mahapatro and Khilar [2]. To improve the computation time for routing in presence of faults, several relevant approaches have been discussed by Treaster [13]. A proposal for a grid system to utilize fault occurrences of nodes for evaluating the quality of routing, using occurrences of failure and request acceptance period, has been put forward by Khanli et al. [12]. Virtual path [4] based routing algorithms have been designed for fault restoration in grid systems using the concept of fault rings. Khanbary and Vidyarthi [11] have proposed the design of fault-tolerant route management algorithms under hostile circumstances in cellular networks.

AntMesh [6] is a routing algorithm is concerned with finding paths that lead to high throughput. Three types of control packets, or smart ants, are produced periodically directed towards the destinations. These ants are responsible for source to destination path discovery, routing table updation through path retracement from destination to source, and collection of link quality information to populate the link estimation table respectively.Two other protocols, periodic event-driven and query-based protocol (PEQ) and its clustered variation CPEQ discussed in [1], draw attention to fault-tolerant and low-latency methods towards routing in the networks. PEQ is a routing algorithm that targets minimization of data transmission using hop level for simple processing techniques at each node. PEQ involves routing in networks within three basic steps. It initiates with the destination node constructing a hop tree that serves as a packet transmission scheme to the network, followed by subscription propagation in the network. Finally, packet delivery occurs from source to destination using the optimality in terms of speed and cost. CPEQ is essentially a variation of PEQ that aims to reduce network traffic and data latency even more than what is obtained through PEQ. It also facilitates a uniform distribution of energy among network nodes. The main idea behind cluster-based PEQ (or CPEQ) is the transmission of data to destination by accumulating data from various neighbouring source nodes. The CPEQ algorithm can be broken into five segments: preliminary configuration; assortment of aggregators; clusters configuration; data communication to the aggregator; and data transmission to the sink. Recently, another fault tolerant routing mechanism in [9] has been inspired by the concept of these two protocols.

Man et al. [19] discussed an evolutionary preventive fault-tolerant routing to find the working path by user constraints. A concise and optimal fault-tolerant routing algorithm through GA is introduced by Pries et al. [16]. Furthermore, the foraging behaviour of ants has also been utilized to solve routing problems in networks. An ACO based adaptive routing algorithm with reinforcement learning was first developed by AntNet [5].

In short, most of the authors resolved the routing problem in context of fault diagnosis by optimizing the network performance. However, the overall performance can be further optimized by performing reliability based routing. To obtain the solution under such circumstances, computation cost in exhaustive search grows unboundedly for larger networks. Evolutionary approaches used in such route optimality have attracted attention for its capability of reducing computation cost to a large extent. Among other EAs, ACO based approaches induce autonomous processing for each node that search for optimal forwards within a faulty network without unnecessary computation burden. In addition, such an approach in routing under faulty circumstances can be realized as a simple procedure that simulates ant behaviour for obtaining improved call acceptance and reduced delay.

3 System Model

The hexagonal cellular layout in wireless networks is viewed as a coordinate system. The BS in each cell is assumed to be a node with coordinates (x, y) in slant axes. The six closest possible neighbours of a node (x, y) are {(x, y-1), (x-1, y+1), (x-1, y), (x, y+1), (x+1, y-1), (x+1, y)}. A 5x5 Cartesian grid consisting of 25 cells is shown in Figure 1 where the layout initiates with a central node (0, 0) and grows outwardly in concentric circles. Neighbours are clustered in groups according to their geographical distance from the central node. Based on this clustering, the coordinates of neighbours are obtained as shown in Table 1. However, to exclude out-of-grid neighbours, a restriction is enforced upon the coordinates. In the hex-grid as shown in Figure 1, ∀ neighbor node $N(x_n, y_n)$ corresponding to reference node $R(x_r, y_r)$, it needs to be ensured that $\left|\left((x_n + y_n) - (x_r + y_r)\right)\right| < C$, where C denotes a constant. For example, the neighbour $(-1, -2)$ for node $(0, -2)$ in Figure 1 is out of range considering $C = 3$. In addition, the number of cells within transmission range of each node grows in a progressive series. To prune out the extra cells that are unreachable being out of range, a summation (Σ) upon x and y coordinates is used. Here, the nodes (-2,-2), (-2,-1), (-1,-2), (2,1), (1,2), and (2,2) in Figure 1 are considered as unreachable with respect to node (0,0).

The nodes within the transmission range considered for routing are classified into different categories according to their functioning capabilities. The faulty nodes are highlighted in Figure 2 using the black dots. These nodes have no active contribution in routing. Further, the concept of unsafe nodes and unsafe zones in the layout are introduced with respect to such faulty nodes. Unsafe nodes shown with grey dots in Figure 2 are determined by the existence of certain percentage of faulty or unsafe neighbours in the network. The faulty nodes along with the unsafe nodes form specific areas known as unsafe zones. It is important to mention that any communication, unless destined for any node lying within the unsafe zones, generally tend to move in a path around such unsafe zones. In this context, the term safe ring is also introduced which refers to the cluster of safe nodes around unsafe zones as illustrated with the arrows in Figure 2. In addition, the communication in this ring tends to propagate in a single direction.

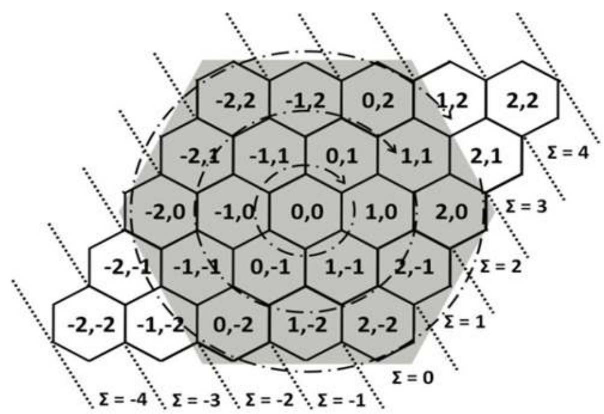


Figure 1: Coordinate Representation of the System Layout

Table 1: Clusters of Neighbours with Coordinates.

Neighbour cluster according to proximity	**Member BSs Relative to central BS represented by (x, y)**
1	(x,y-1); (x-1,y+1); (x-1,y); (x,y+1); (x+1,y-1); (x+1,y)
2	(x-1,y+2); (x+1,y+1); (x+2,y-1); (x+1,y-2); (x-1,y-1); (x-2,y+1)
3	(x-2,y-2); (x,y+2); (x+2,y); (x+2,y-2); (x,y-2); (x-2,y)
…	…

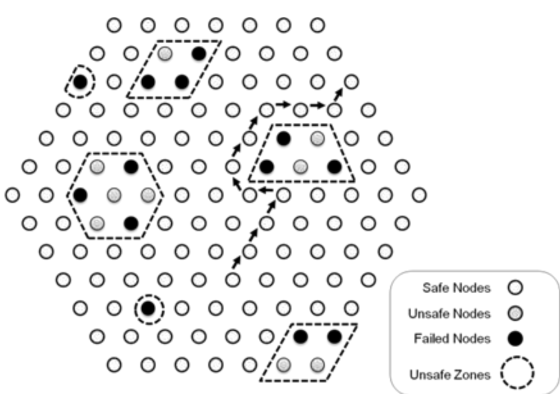


Figure 2: Various types of nodes

Further, all nodes maintain a local routing table as shown in Figure 3 which includes an indication of faulty neighbours or unsafe zones in the immediate neighbourhood. It keeps the record on reliability of each of the immediate neighbour as forwarders with respect to the destination in network. The local routing table is characterized by the reliability score (R) and minimum declared hop count(HC). The parameter R is the confidence of a node upon its neighbour in terms of safety. A high confidence value indicates a superior fault-tolerant path. HC is declared by a node on its own. Inability to sustain the declared HC while communicating or failure in routing reduces the value R of node in routing table of the neighbours.

		Neighbours								
		$N_1\ (x_{n_1}, y_{n_1})$		$N_2\ (x_{n_2}, y_{n_2})$		$N_3\ (x_{n_3}, y_{n_3})$		…	$N_k\ (x_{n_k}, y_{n_k})$	
		Declared Hop Count (HC)	Reliability Score (R)	Declared Hop Count (HC)	Reliability Score (R)	Declared Hop Count (HC)	Reliability Score (R)	…	Declared Hop Count (HC)	Reliability Score (R)
Destination	$D_1\ (x_{d_1}, y_{d_1})$	3	0.2	1	0.8	4	0.6	…	2	0.8
	$D_2\ (x_{d_2}, y_{d_2})$	2	0.9	2	0.1	2	0.2	…	4	0.5
	…	…	…	…	…	…	…	…	…	…
	$D_m\ (x_{d_m}, y_{d_m})$	5	0.3	6	0.8	3	0.6	…	4	0.8

Figure 3: Instance of Local Routing Table

In this work, a fault tolerant routing scheme in wireless cellular networks is proposed in such a way that the call acceptance in the network is improved as well as the delay in terms of number of hops for routing a request is reduced in spite of existence of faults. To formulate such routing problem obeying the decision parameters stated earlier, the nodes of the network are encoded into four functional states which are highlighted and briefly explained with Figure 4.

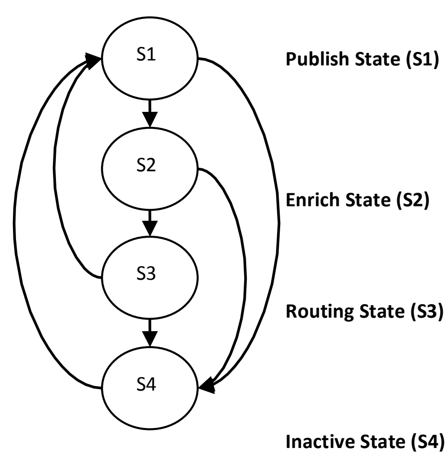


Figure 4: Functional States of a Node in the Network

Initially any node is in inactive state (S4) and whenever a node gets active, it first enters publish state (S1). Subsequently it prepares its initial local routing table, publishes it to immediate neighbours and moves into the enrich state (S2). Here it waits for a predefined interval to enrich its local routing table using information from neighbours. Next the node moves into routing state (S3). The local routing table also needs a periodic update which is initiated upon two conditions – fault of any immediate neighbour and update request from any neighbour. Such action can move a node back from S3 to S1. Any node upon encountering fault moves abruptly from any of its state into S4. The transitions between the various states shown in Figure 4 are interpreted in Table 2

So, the current work is emphasized with S3 as it addresses a proposal of fault tolerant routing scheme. In addition, the functionalities of S1 and S2 are used to execute the state S3.

Table 2: State Transition and its Interpretation

Transition	Interpretation
S4 → S1	*Any inactive node first activates itself, either for the first time, or due to recovery from a fault.*
S1 → S4	*The node becomes inactive due to any kind of fault.*
S1 → S2	*The node completes publishing its initial local routing table and starts waiting for response from neighbours.*
S2 → S4	*The node becomes inactive due to any kind of fault.*
S2 → S3	*The node enriches its own local routing table based on responses from neighbours and moves into the routing state.*
S3 → S4	*The node becomes inactive due to any kind of fault.*
S3 → S1	*The node needs to modify own local routing table due to any change of state of the immediate neighbours.*

4 Proposed Approach

This section concentrates on discussion upon publish and enrich mechanism followed by proposed ACO induced routing activity of the nodes towards obtaining an improved performance for the system.

4.1 Publish and Enrich Mechanism

The nodes initially consider each of its immediate neighbours as destination and populates the entry corresponding to the same neighbouring node as the single hop route. Accordingly, HC is set to 1 and corresponding R is abruptly set to 0.5. After populating the information locally, every node broadcasts it to all neighbouring nodes. Each node waits for a predefined interval to receive all the neighbours' information about the local routing table. This interval is decided upon by the span of the entire network. A higher value of the interval extends the span of the Enrich State (S2), but does not impact the routing activity. This interval can be in order of few milliseconds in a software simulated network, but in actual networks it should be realistic to match the communication delay of the physical devices. After that interval, it accumulates the information to record the minimum declared HC of a neighbor for a destination. If it seems superior to existing local information in terms of HC, the later is replaced with the newly obtained value. This is how the information about the entire network gets enriched at every node. The entire procedure is described by the following algorithm 1 which is performed at every node.

Algorithm 1: *publishing and enriching local routing table*

1. N_c = current node
2. IB_{N_c} = local routing table
3. S_g = set of neighboring nodes of N_c
4. For every node $N_g \in S_g$
5. Populate IB_{N_c} with $\{HC_{N_g,N_g} = 1; R_{N_g,N_g} = 0.5\}$
6. End for

7. Repeat steps 8 – 19 for several predefined iterations
8. Publish IB_{N_c} to every node $N_g \in S_g$
9. Repeat steps 10 – 16 for every IB_{N_g} received
10. For every destination node $N_d \in$ Network
11. For every node $N_g \in S_g$
12. Compare IB_{N_g} and IB_{N_c}
13. Populate IB_{N_c} with best value of $\{HC_{N_d,N_g}; R_{N_d,N_g}\}$
14. If N_g is unsafe, reduce R_{N_d,N_g} by factor 0.5
15. End for
16. End for
17. For every node $N_g \in S_g$ for which IB_{N_g} not received
18. Assume N_g is faulty, reduce R_{N_d,N_g} to 0 for all N_d
19. End for

4.2 Routing Mechanism

The proposed routing mechanism is accomplished by ACO based approach. The state S3 is initiated for every node by preparing the enriched local routing table. Here, the node maintains a priority based local buffer for handling incoming requests. Every request is thus forwarded to any of the neighbours by consulting the local routing table. Both the declared HC and R of the neighbouring nodes are considered while forwarding any communication to that node. The parameter HC inversely denotes the desirability of a neighbour node as the selected one to forward communication, and is derived from a priori knowledge. The parameter R plays the same role as the pheromone in ACO algorithms, influencing the attractiveness of a popular trail. Two constants λ_1 and λ_2 are used to regulate the relative emphasis of these two deciding factors of routing. The selection of the next node is achieved by roulette wheel logic [6] upon the cumulative desirability of each neighbouring node. This mechanism of forwarding requests to next hop is thus described by algorithm 2.

Algorithm 2: ***forwarding requests to next hop***

1. $N_c =$ current node
2. $Q_c =$ current request at node N_c
3. $N_d =$ destination node for request Q_c
4. If $N_d == N_c$
5. Accept content of Q_c
6. $N_s =$ immediate sender node of Q_c
7. If N_s is faulty, broadcasts acknowledgement A_c to all neighbors
8. Else revert acknowledgement A_c to N_s
9. End if
10. End if
11. $S_g =$ set of neighboring nodes of N_c
12. For every node $N_g \in S_g$
13. If $R_{N_d,N_g} > 0$
14. $L_g =$ current length of buffer at N_g
15. $factor_1 = (HC_{N_d,N_g} * L_g)^{-\lambda_1}$
16. $factor_2 = (R_{N_d,N_g})^{\lambda_2}$
17. $P_g = factor_1 * factor_2$
18. End if
19. End for
20. Cumulative desirability, $P_{cuml} =$ cumulative sum (P_g)
21. $N_t =$ Selected node based on roulette wheel selection (P_{cuml})
22. Transmit Q_c to N_t

Another procedure is followed for returning the acknowledgement from the destination to the source for any successful routing. This is essential for increasing the popularity of a route in the network by updating the intermediate values of R. In case the acknowledgement packet is blocked by a faulty region, it broadcasts through the network. A node receiving the acknowledgement from any node other than the immediate forwarders interprets the situation as a fault and readjusts its local routing table with reduced values of R for the concerned nodes. Here, two parameters ρ_1 and ρ_2 as pheromone evaporation and deposit rates are used to adjust the preference of routes in the network. Here the nodes need not preserve the transmission history of any of the packets. The back-propagation of the acknowledgement packet servers the purpose of reliability score update, which is sufficient for later routing decision, The entire procedure for pheromone update is described by Algorithm 3.

Algorithm 3: ***pheromone update***

1. $N_c =$ current node
2. $A_c =$ Any acknowledgement received at N_c for request Q_c
3. If $Q_c \in$ request transmission history (N_c)
4. $L_c =$ Hop length of the successful request
5. $N_d =$ actual destination of the request
6. $N_g =$ immediate sender of the acknowledgement A_c
7. $R_{N_d,N_g} = \left((1-\rho_1) * HC_{N_d,N_g}\right) + (\rho_2/L_c)$
8. $N_s =$ immediate sender of Q_c
9. If N_s is faulty, broadcasts A_c to all neighbors
10. Else revert acknowledgement A_c to N_s
11. End if
12. Else
13. $R_{N_d,N_g} = R_{N_d,N_g} * \rho_2$
14. Broadcasts A_c to all neighbors
15. End if

Following the Algorithm 2 and Algorithm 3 described earlier, any existing optimal trail gets preference of selection, and still a level of randomness is retained to search for alternate optimal trails. Thus the proposed approach provides a uniformity of route utilization throughout the non-faulty regions in the network.

5 Experimental Results

The major system parameters are summarized in Table 3 which are used to obtain the results for various scenarios in order to assess the performance of proposed approach. We use Java version 1.7 and MATLAB 7.6 as software setup to conduct the simulation experiments.

Table 3: System Parameters with Values

Maximum Number of Nodes	1027
Neighbourhood Range	3 (18 nodes)
Relative Emphasis Factor (λ_1)	3
Relative Emphasis Factor (λ_2)	2
Pheromone Deposit Rate (ρ_2)	2
Pheromone Evaporation Rate (ρ_1)	0.5
Initial Pheromone	0.5
Request Rate (Event Rate)	2/source/unit time
Reliability Loss Factor	0.5
Simulation Period	1000 time units
Percentage of Fault	0.1 to 0.7

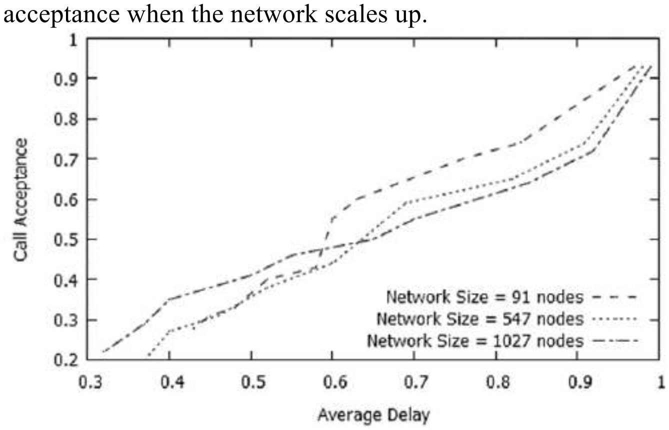
The decision parameters such as call acceptance and average delay in the proposed work are plotted for ascertaining the variation of one against the other. The efficiency of the algorithm is evaluated by finding the preferred solution in terms of efficient route from conflicting interests of the objectives. In experiment, it is observed that maximizing call acceptance and minimizing average request delay contradicts each other, and the proposed approach tries to strike a balance between these two. The factor λ_1 can be regulated to reduce the delay in completion of the requests. Higher values of λ_1 increase the possibility of the requests following a route, where the intermediate nodes have low values of HC to a destination, and low queue in the buffers. However, the factor R is ignored in such situations, which can lead to requests getting stuck around fault zones. This scenario is shown in Figure 5 (a) for varying number of nodes in the network, keeping the percentage of failure at minimum value (0.1) for the simulation. Similar variation is shown in Figure 5 (b) by varying the fault occurrence in the network for a fixed maximum network size with 1027 nodes. In both illustrations, the success of call request is shown as a percentage. Similarly, the average delay in delivery of requests is also assumed as a percentage of maximum allowable delay in the network, beyond which the request is treated as failed. Thus, an optimal balance between the parameters λ_1 and λ_2 is necessary to obtain satisfactory performance for the proposed approach.

The concept of unsafe nodes depends upon the percentage of neighbours of a node which are faulty or unsafe. A low value of this percentage results in more number of unsafe zones in a network. The proposed routing approach results in higher delay while routing the requests under such conditions, since the routes preferably bypass all the unsafe zones, and become lengthy. On the contrary, a high value of the percentage restricts the formation of unsafe zones. Routing delay thus gets reduced, but requests often fail by getting congested at faulty nodes. In such cases, the balanced routing mechanism in the proposed approach is restricted. This phenomenon, and the optimal neighbour percentage for arriving at improved performance is illustrated in Figure 6 (a). Here the third parameter is the ratio between the call acceptance and the average delay which finds peak at the 0.5 value for faulty neighbour ratio for unsafe node.

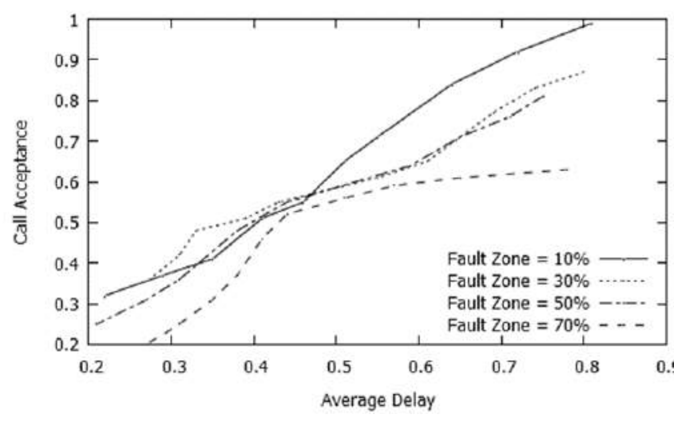
Assuming the threshold as 0.5, the unsafe zones are designed for varying numbers of faulty nodes in the network. Faults are randomly inserted into the network, and the duration and persistence of the faults are also set dynamically in the simulation. Subsequently, the call acceptance and average delay in the network are calculated based upon the varying percentages of faulty nodes. This is illustrated in the Figure 6 (b). A reduction in both parameters, call acceptance and delay, is observed with increase in the occurrences of random fault in the network. However, experimental studies show that the former leads to a steep fall when compared to the later.

AntMesh [6] is an algorithm which uses smart ants to perform data forwarding to solve a fault tolerant routing problem. The parameters of AntMesh algorithm are studied in terms of the routing load and end-to-end delay under fixed nodes in the network. The current approach is aligned with the AntMesh for a comparative evaluation as shown in Figure 7. Here the comparison highlights the improved performance, in terms of average network delay, obtained by the proposed approach. The request generation rate in this experiment denotes the requests originated per second.

For comparing the proposed approach with declared [9, 10] models, the number of nodes in the network are varied from 100-500, and number of source nodes are fixed at 14%. The occurrences of faults are random, as considered in the current work, though the extent of fault zones is limited to 10% of the network. It is observed in Figure 8 (a) that the average delay to accept call gets higher due to the larger number of hops as the network size increases. This makes sense that the re-routed path becomes longer as the number of faulty node increases. The amount of such delay is reduced by the proposed approach over these existing protocols. Figure 8 (b) shows that the proposed approach maintains a reasonable call acceptance compared with those PEQ and CPEQ protocols even at a high percentage of node failures. Therefore, the ACO based proposed approach outperforms over existing fault tolerant routing schemes [9, 10] in terms of both average delay and call acceptance when the network scales up.



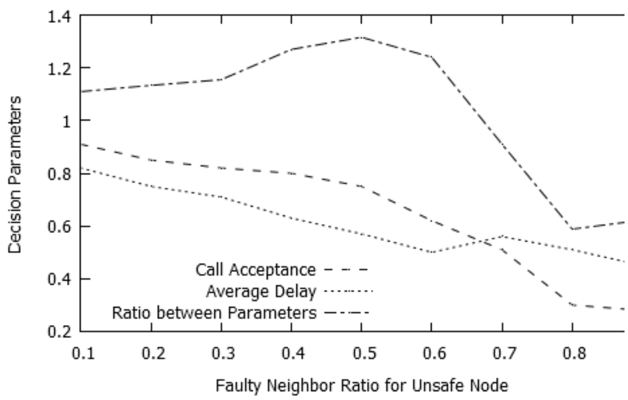
(a)



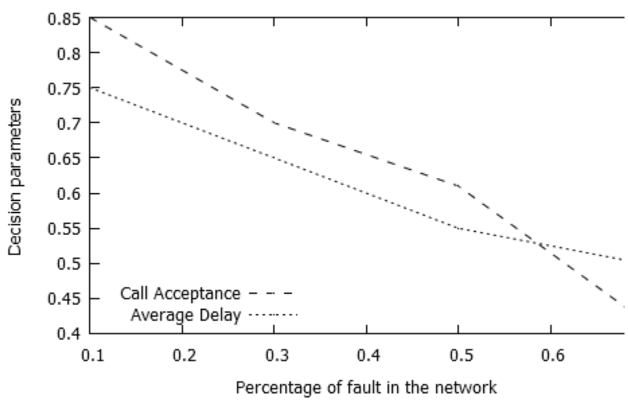
(b)

Figure 5: Call Acceptance vs. Average Delay (a) varying Network Size, (b) varying Fault Zone.

In an earlier work [20], Huang et. al. had proposed a formulation relating the iteration time of an ACO algorithm towards reaching the objective value with the pheromone rate, the chief controlling factor of the progress of the mechanism. This analysis is applicable in the present work as well for ensuring convergence, as well as deciding upon a defined execution time of the routing mechanism discussed in the previous section.



(a)



(b)

Figure 6: (a) Optimal Faulty Neighbour Ratio for Unsafe Nodes, (b) Variation of Call Acceptance and Average Delay with varying Fault in Network

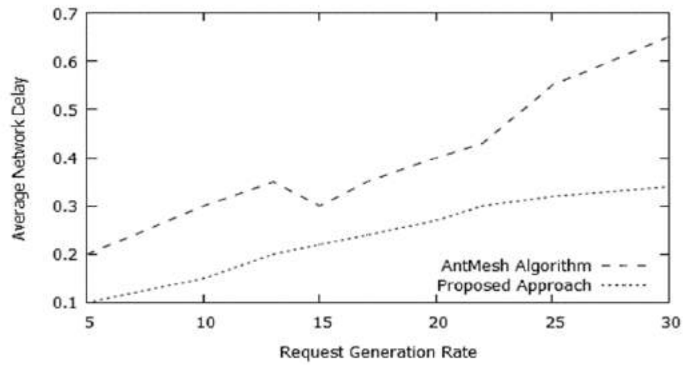
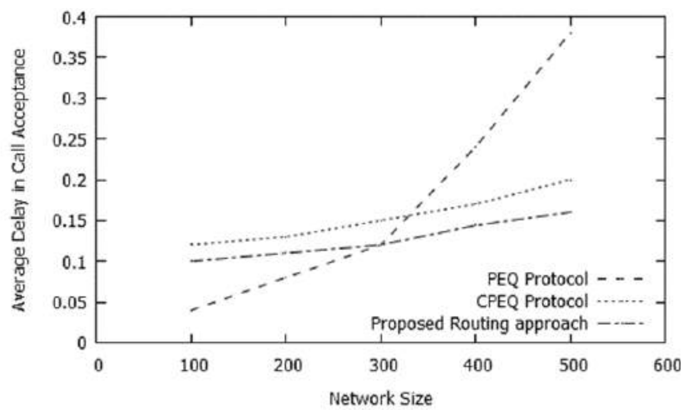
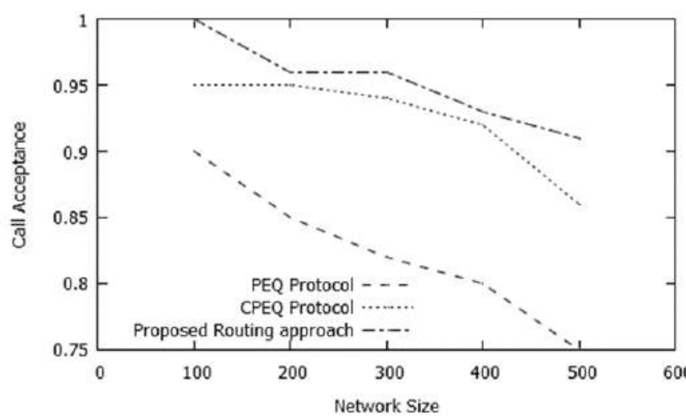


Figure 7: Comparison of the Proposed Approach with AntMesh Algorithm



(a)



(b)

Figure 8: Comparison of the proposed approach with existing schemes in terms of (a) Average Delay, (b) Call Acceptance

6 Conclusions

In this paper, an ACO based fault tolerant routing approach in wireless cellular networks is proposed. The collective intelligence of ants modelled in the proposed approach is essential in satisfying reliable service on call forwarding in a network still tolerating BS fault. Thus it increases network throughput in terms of call acceptance and simultaneously decreases average delay under such faulty scenario. This mechanism provides higher fault aware path diversity due to identification of potential fault zones and introduces re-routed data aside these zones. Following the proposed algorithms, any existing optimal trail gets preference of selection and still a level of randomness is retained to search for alternate one. Thus the proposed approach facilitates a uniformity of route utilization throughout the non-faulty regions in the network. The basic features of ACO such as initialization, path selection, pheromone deposition, confidence calculation, pheromone evaporation and reinforcement are exploited in the entire procedure of the proposed work. The performance of this comprehensive framework is compared with existing schemes. The proposed approach with guaranteed service quality in the opportunistic network paradigm is under consideration as a future scope.

References

[1] A. Boukerche, R. W. N. Pazzi and R. B. Araujo, ‘Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments’, Journal of Parallel and Distributed Computing, Vol. 66 No. 4, pp. 586-599, 2006.

[2] A. Mahapatro and P. M. Khilar, 'Fault Diagnosis in Wireless Sensor Networks: A Survey', IEEE Communications Surveys and Tutorials, Vol.15 No.4, pp. 2000-2026, 2013.

[3] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan and Q. Zhang, ‘Multiobjective evolutionary algorithms: A survey of the state of the art’, Journal on Swarm and Evolutionary Computation, Vol. 1 No.1, pp. 32-49, 2011.

[4] Chen, K. H. and Chiu, G. M. (1998), 'Fault-Tolerant Routing Algorithm for Meshes without Using Virtual Channels', Journal of Information Science and Engineering, Vol.14 No.4, pp. 765-783.

[5] Di Caro, G. and Dorigo, M. (1998), 'AntNet: Distributed stigmergetic control for communications networks', Journal of Artificial Intelligence Research, Vol.9, pp. 317–365.

[6] F. Bokhari and G. Zaruba, ‘On the Use of Smart Ants for Efficient Routing in Wireless Mesh Networks’, International Journal of Wireless & Mobile Networks, Vol. 4 No. 2, pp. 117-134, 2012.

[7] H. K. Hsin, E. J. Chang, C. A. Lin and A.Y.A. Wu, ‘Ant Colony Optimization-Based Fault-Aware Routing in Mesh-Based Network-on-Chip Systems’, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 33 No. 11, pp. 1693-1705, 2014.

[8] H. M. Dubey, B. K. Panigrahi and M. Pandit, ‘Bio-inspired optimisation for economic load dispatch: a review, International Journal of Bio-Inspired Computation’, Vol. 6 No. 1, pp. 7-21, 2014.

[9] J. R. Diaz, J. Lloret, J.M. Jimenez, S. Sendra and J. J. P. C. Rodrigues, ‘Fault Tolerant Mechanism for Multimedia Flows in Wireless Ad Hoc Networks Based on Fast Switching Paths’, Journal of Mathematical Problems in Engineering, pp. 1-12, 2014.

[10] K. B. Baltzis, Hexagonal vs Circular Cell Shape: A Comparative Analysis and Evaluation of the Two Popular Modeling Approximations, Cellular Networks - Positioning, Performance Analysis, Reliability, Dr. Agassi Melikov (Ed.), InTech, Sanghai, Chaina, pp. 103-122, 2011.

[11] Khanbary, L. M. O. and Vidyarthi, D. P. (2009), 'Reliability-Based Channel Allocation Using Genetic Algorithm in Mobile Computing', IEEE Transactions on Vehicular Technology, Vol.58 No.8, pp. 4248 - 4256.

[12] Khanli, L. M., Far M. E. and Rahmani, A. M. (2010), 'RFOH: A New Fault Tolerant Job Scheduler in Grid Computing', Proceedings of the 2nd International Conference on Computer Engineering and Applications, Bali Island, Indonesia, pp. 422-425.

[13] M. Treaster, 'A Survey of Fault-Tolerance and Fault-Recovery Techniques in Parallel Systems', ACM Computing Research Repository (CoRR), Vol.501002, pp. 1-11, 2005.

[14] P. K. Guha Thakurta, A New Priority Based Call Scheduling Technique in Mobile Networks, Journal of Advances in Computer Networks, Vol. 3 No. 1, , pp. 38-41, 2015.

[15] R. O. Oladele and J. S. Sadiku, ‘Genetic Algorithm Performance with Different Selection Methods in Solving Multi-Objective Network Design Problem’, International Journal of Computer Applications, Vol. 70 No. 12, pp. 5-9, 2013.

[16] R. Pries, D. Staehle, B. Staehle, and, P. Tran-Gia, 'On Optimization of Wireless Mesh Networks using Genetic Algorithms', International Journal on Advances in Internet Technology, Vol.3 No.1&2, pp. 13-28, 2010.

[17] S. B. Meskina, N. Doggaz and M. Khalgui, New solutions for fault detections and dynamic recoveries of flexible power smart grids, In 11th International Conference on Informatics in Control, Automation and Robotics, Austria, Vienna, pp. 370-377, 2014.

[18] S. Misra, S.K Dhurandher, M.S. Obaidat, K. Verma and P. Gupta, ‘Using Ant-Like Agents for Fault-Tolerant Routing in Mobile Ad-Hoc Networks’, In IEEE International Conference on Communications, Dresden, pp. 1-5, 2009.

[19] X. Man, X. W. Wang and M. Huang, 'An Evolutionary Multi-Agent Preventive Fault-Tolerant QoS Unicast Routing Mechanism', IEEE International Conference on Embedded and Ubiquitous Computing, Zhangjiajie, China, pp. 1254-1261, 2013.

[20] H. Huang, C. G. Wu, and Z. F. Hao, ‘A Pheromone-Rate-Based Analysis on the Convergence Time of ACO Algorithm’, In IEEE Transactions on Systems, Man, and Cybernetics —Part B: Cybernetics, Vol. 39, No. 4, August 2009, pp. 910-923.