DETECTING NODE FAILURES AND REBUILDING FAILED NODES

[1] Adithya Laxmi.A , [2] Harini.M, [3] Indhumathi.M
[1,2,3] Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
[1] 13bc001@skcet.ac.in

Abstract

Node failure detection in mobile wireless networks is very challenging because of the dynamically changing network topology, the network may not be always connected, and the resources are limited. In this paper, we use localized monitoring, location estimation and node collaboration to detect failed node and to rebuild them if failed. Compared to approaches that use centralized monitoring and localized monitoring, our approach has high packet delivery ratio, low routing overhead, high throughput ratio, low average end-to-end delay ratio for packet transfer.

I.INTRODUCTION

Wireless networks have been used for many mission critical applications, including search and rescue, environment monitoring, disaster relief, and military operations. Many types of wireless communication systems exist, but a distinguishing attribute of a wireless network is that communication takes place between computer devices. These devices include personal digital assistants (PDAs), laptops, personal computers (PCs), servers, and printers. Mobile Ad-hoc network is a set of wireless devices called wireless nodes, which dynamically connect and transfer information. Wireless nodes can be personal computers (desktops/laptops) with wireless LAN cards, Personal Digital Assistants (PDA), or other types of wireless or mobile communication devices. Mobile Ad-hoc networks has various potential applications. Some typical examples include emergency search-rescue operations, meeting events, conferences, and battlefield communication between moving vehicles and/or soldiers. With the abilities to meet the new demand of mobile computation, the Mobile Ad-hoc networks has a very bright future.

Wireless Ad-hoc network have many advantages like low cost of deployment, fast deployment and dynamic Configuration. Ad-hoc networks can be deployed on the fly; hence no expensive infrastructure such as copper wires or data cables is required. Ad-hoc network configuration can change dynamically over time. When compared to configurability of LANs, it is very easy to change the network topology of a wireless network.

In Mobile Ad-hoc networks, a wireless node can be the source, the destination, or an intermediate node of data transmission. When a wireless node plays the role of intermediate node, it serves as a router that can receive and forward data packets to its neighbor closer to the destination node. Nodes in such networks are vulnerable to failures due to hardware defects, battery drainage, hackers or a harsh environment. Node failure detection is important for keeping tabs on the network. Due to the nature of an Ad-hoc network, wireless nodes tend to keep moving instead of staying still. Therefore the network topology changes from time to time. Therefore, techniques that are designed for static networks are not applicable.

Existing studies reveal that there are many approaches adopted for node failure detection. One of the approach is based on centralized monitoring. It requires that each node send periodic “heartbeat” messages to a central monitor, which uses the lack of heartbeat messages from a node (after a certain timeout) as an indicator of node failure. This approach assumes that there always exists a path from a node to the central monitor, and hence is only applicable to networks with persistent connectivity. In addition, since a node can be multiple hops away from the central monitor, this approach can lead to a large amount of network-wide traffic, in conflict with the constrained resources in mobile wireless networks.

Another approach is based on localized monitoring, where nodes broadcast heartbeat messages to their one-hop neighbors and nodes in a neighborhood monitor each other through heartbeat messages. Localized monitoring only generates localized traffic and has been used successfully for node failure detection in static networks. However, when being applied to mobile networks, this approach suffers from inherent ambiguities — when a node A stops hearing heartbeat messages from another node B , A cannot conclude that B has failed because the lack of heartbeat messages might be caused by node B having moved out of range instead of node failure.

Another approach is based on probabilistic approach that judiciously combines localized monitoring, location estimation and node collaboration to detect node failures in mobile wireless networks. Specifically, we propose two schemes. In the first scheme, when a node A cannot hear from a neighboring node B, it uses its own information about B and binary feedback from its neighbors to decide whether B has failed or not. In the second scheme, A gathers information from its neighbors, and uses the information jointly to make the decision. The first scheme incurs lower communication overhead than the second scheme. On the other hand, the second scheme fully utilizes information from the neighbors and can achieve better performance in failure detection and false positive rates.

Another approach is based on failure detection service for wireless ad-hoc and sensor systems that is based on an adaptation of a gossip-style failure detection protocol and the heartbeat failure detector.

II. RELATED WORK

Many existing systems use Probe- and –Ack or heartbeat based techniques. Probe-and-ACK based techniques require a central monitor to send probe messages to other nodes. When a node does not reply within a timeout interval, the central monitor regards the node as failed. Heartbeat based techniques differ from probe-and-ACK based techniques in that they eliminate the probing phase to reduce the amount of messages. A common drawback of probe-and-ACK and heartbeat based techniques is that they are only applicable to networks that are connected. In addition, they lead to a large amount of network-wide monitoring traffic. In contrast, our approach only generates localized monitoring traffic and is applicable to both connected and disconnected networks. Localized monitoring however is not suitable for mobile networks since it does not consider that failure to hear from a node might be due to node mobility instead of node failure. Our approach takes account of node mobility.

Some of the existing systems use approaches like Binary Feedback Scheme and Non-Binary Feedback Scheme for the detection of node failures. In the first scheme, when a node A cannot hear from a neighbouring node B, it uses its own information about B and binary feedback from its neighbours to decide whether B has failed or not. In the second scheme, A gathers information from its neighbours, and uses the information jointly to make the decision. The first scheme incurs lower communication overhead than the second scheme. On the other hand, the second scheme fully utilizes information from the neighbours and can achieve better performance in failure detection and false positive rates.

III. FAILURE DETECTION APPROACH

In this section, we first use an example to illustrate our approach, and then present a core building block of our approach.

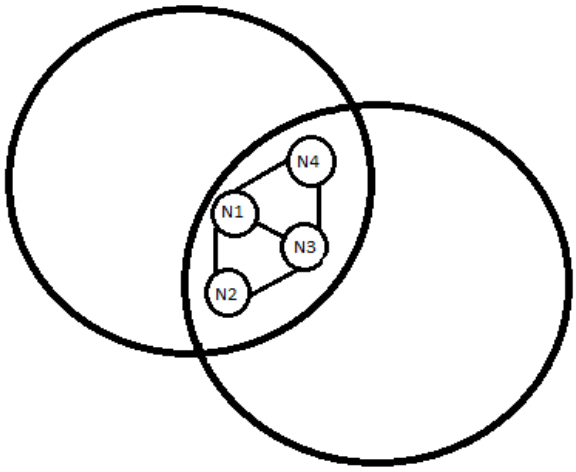

Fig 1(a) Time *t*

We use the example in Fig. 1 to illustrate our approach. In this example, for simplicity, we assume no packet losses and that each node has the same circular transmission range. At time t, all the

nodes are alive, and node N1 can hear heartbeat messages from N2 and N3 (see Fig. 1(a)). At time t+1 , node N2 fails and N3 moves out of N1 ’s transmission range (see Fig. 1(b)). By localized monitoring, N1 only knows that it can no longer hear from N2 and N3 , but does not know whether the lack of messages is due to node failure or node moving out of the transmission range.

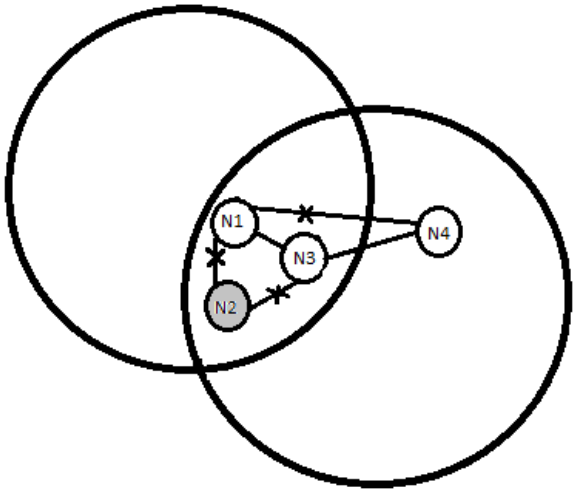

Fig 1(b) Time *t*+1

Location estimation is helpful to resolve this ambiguity: based on location estimation, N1 obtains the probability that N2 is within its transmission range, finds that the probability is high, and hence conjectures that the absence of messages from N2 is likely due to N2 ’s failure; similarly, N1 obtains the probability that N3 is within its transmission range, finds that the probability is low, and hence conjectures that the absence of messages from N3 is likely because N3 is out of the transmission range. The above decision can be improved through node collaboration. For instance, N1 can broadcast an inquiry about N2 to its one-hop neighbors at time t + 1, and use the response from N4 to either confirm or correct its conjecture about N2 . The above example indicates that it is important to systematically combine localized monitoring, location estimation and node collaboration, which is the fundamental of our approach.

IV. NODE FAILURE DETECTION

A probabilistic approach and a node failure detection scheme that combines localized monitoring, location estimation and node collaboration for mobile wireless networks is designed. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. Whenever a node fails, it is re-builded and the packet continues to flow through the same path. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic.

In MANET, a wireless node can be the source, the destination, or an intermediate node of data transmission. When a wireless node plays the role of intermediate node, it serves as a router that can receive and forward data packets to its neighbor closer to the destination node. We apply Binary scheme to detect and retrieve data in case of any node failure in a mobile ad hoc network.

When sending a message from source to destination, first the shortest path is found. Then based on the feedback sent by the binary schemes the node failure is detected if any. If the node failure is detected, the node is rebuilt and then the data is sent in the same path to the destination node.

In MANET, a wireless node can be the source, the destination, or an intermediate node of data transmission. When a wireless node plays the role of intermediate node, it serves as a router that can receive and forward data packets to its neighbor closer to the destination node. Due to the nature of an ad-hoc network, wireless nodes tend to keep moving rather than stay still. Therefore the network topology changes from time to time.

The network performance is improved by rebuilding the failed node and sending the packets in the same path. Thus the average end-to-end delay is reduced.

V. NETWORK DEPLOYMENT

Deployment, in the context of network administration, refers to the process of setting up a new computer or system to the point where it ready for productive work in a live environment. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route

traffic. In MANET, a wireless node can be the source, the destination, or an intermediate node of data transmission.

VI. DATA COMMUNICATION

Data communication refers to the transmission of the data packets between the nodes in a computer network. In MANET, a wireless node can be the source, the destination, or an intermediate node of data transmission. When a wireless node plays the role of intermediate node, it serves as a router that can receive and forward data packets to its neighbor closer to the destination node. When sending a message from source to destination, first the shortest path is found.

VII. PROBABLISTIC BINARY

There are two probabilistic approaches, namely, Binary Feedback Scheme and Non-Binary Feedback Scheme. Our approach uses Binary Feedback Scheme. Consider a node, A, no longer hears from another node, B , at time t+1. In the binary feedback scheme, A calculates the conditional probability p that B has failed. Let $\theta \in (0,1)$ denote a pre-defined detection threshold. If p is larger than the threshold θ , then A has a high confidence that B has failed. To reduce the risk of false alarms, A broadcasts to its neighbourhood an inquiry message about B. In order to avoid multiple nodes broad cast inquiry messages about B, assume A starts a timer with a random timeout value, and only broadcasts a query message about B when the timer times out and A has not heard any query about B. In this case, only the node has the lowest random timeout value will broadcast a query message about B; the other nodes refrain from sending an inquiry about B . Suppose that A broadcasts a query message about B. Any neighbor, C, after receiving the inquiry, makes a binary response: it responds with a single bit 0 if it has heard from B at time t+1 ; it responds with a single bit 1 if its calculated failure probability for B is larger than θ ; otherwise, it keeps silent. Then A generates a failure alarm about B and sends it to the manager node unless it receives a 0.

VIII. PERFORMANCE EVALUATION

The performance of our system is evaluated using ns2 simulator. A network simulator is a software program that imitates the working of a computer network. In simulators, the computer network is typically modelled with devices, traffic etc and the performance is analysed. Typically, users can then customize the simulator to fulfill their specific analysis needs. Simulators typically come with support for the most popular protocols in use today, such as WLAN, Wi-Max, UDP, and TCP. Our approach uses UDP protocol for communication between nodes in the network.

Compared to approaches that use centralized monitoring and localized monitoring, our approach has high packet delivery ratio, low routing overhead, high throughput ratio, low average end-to-end delay ratio for packet transfer.

1) Average end-to-end delay

End-to-end delay or one-way delay (OWD) refers to the time taken for a packet to be transmitted across a network from source to destination. It is a common term in IP network monitoring, and differs from round-trip time (RTT).

The *ping* utility calculates the *RTT*, that is, the time to go and come back to a host. This does not assure that the go and back paths are the same in terms of congestion, number of hops, or *quality of service* (QoS). In order to avoid such problems, OWD concept comes into play. The most common method by which OWDs are calculated between two points A and B of an IP network is to first synchronize their clocks; A records a timestamp on the packet and sends it to B, which notes the receiving time and calculates the OWD as their difference.

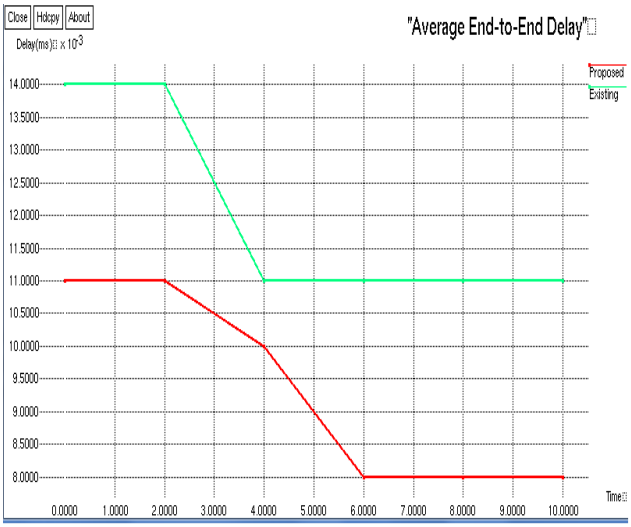

Fig 3a Average end-to-end delay

The transmitted packets need to be identified at source and destination in order to avoid packet loss or packet reordering. This method however suffers several limitations, such as requiring intensive cooperation between both parties, and the accuracy of the measured delay is subject to the synchronization precision.

Figure 3(a) plot end-to-end delay versus time. The graph shows comparison of average end-to-end delay rates between proposed and existing approaches. Observe that the end-to-end delay rates of our scheme are very close to the lower bound, indicating that our scheme achieves low end-to-end delay rates.

2) Packet delivery ratio

The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender, that is, Proportion of number of packets delivered against the number of packets sent.

Packet delivery ratio is defined as the ratio of data packets received by the destinations to those generated by the sources.

Mathematically, it can be defined as: PDR= S1÷ S2 Where, S1 is the sum of data packets received by the each destination and S2 is the sum of data packets generated by the each source.

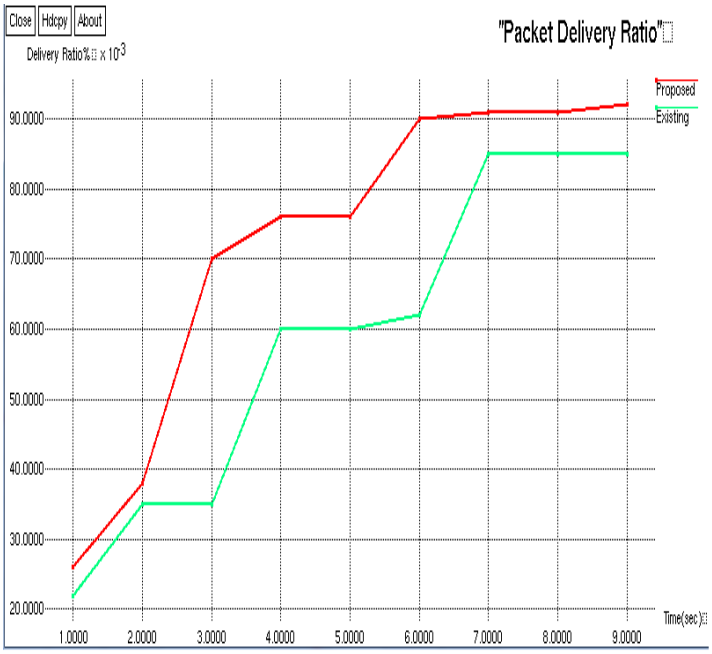

Fig 3b Packet delivery ratio

Figure 3(b) plot delivery ratio versus time. The graph shows comparison of packet delivery ratio between proposed and existing approaches. Observe that the packet delivery ratio of our scheme are very close to the upper bound, indicating that our scheme achieves high packet delivery ratio.

3) Routing overhead

To keep up-to-date information about network routes, routing algorithms generate small sized packets, called routing packets. One example of such packets is a HELLO packet, which is used to check whether the neighbour node is active. Note that routing packets do not carry any application content, like data packets do.

Both, routing and data packets have to share the same network bandwidth most of the times, and hence, routing packets are considered to be an overhead in the network. This overhead is called routing overhead. A good routing protocol should incur lesser routing overhead.

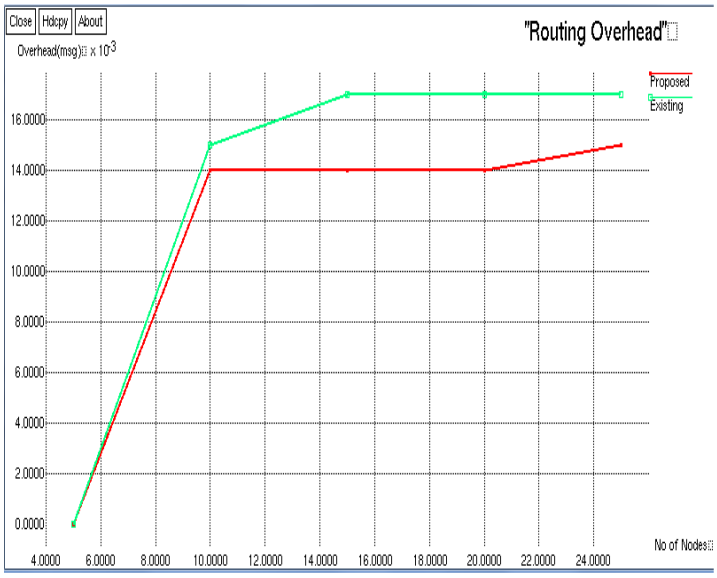

Fig 3c Routing overhead

Figure 3(c) plot routing overhead versus number of nodes. The graph shows comparison of routing overhead between proposed and existing approaches. Observe that the routing overhead of our scheme are very close to the lower bound, indicating that our scheme achieves low routing overhead.

4) Throughput ratio

Throughput ratio is defined as the total number of packets delivered over the total simulation time. Throughput is the number of messages successfully delivered per unit time. Throughput is controlled by available bandwidth, as well as the available signal-to-noise ratio and hardware limitations.

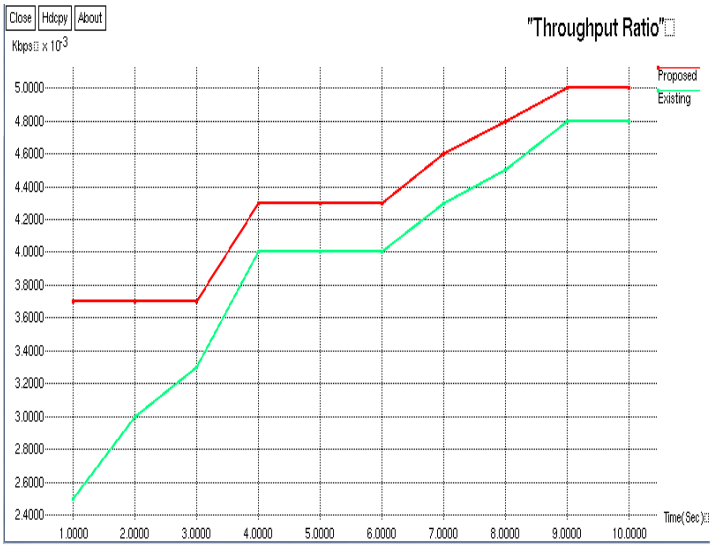

Fig 3d Throughput ratio

Figure 3(d) plot throughput ratio(kilo bytes per second) versus time. The graph shows comparison of throughput ratio between proposed and existing approaches. Observe that the throughput ratio of our scheme are very close to the upper bound, indicating that our scheme achieves high throughput ratio.

IX. CONCLUSION AND FUTURE WORK

In this paper, a probabilistic approach and designed two node failure detection schemes that combine localized monitoring, location estimation and node collaboration for mobile wireless networks. Extensive simulation results demonstrate that our schemes achieve high failure detection rates, low false positive rates, and low communication overhead. We further demonstrated the tradeoffs of the binary and non-binary feedback schemes.

As future work, we are planning to evaluate our schemes using real world mobility traces and in scenarios with irregular transmission ranges. Our approach relies on location estimation and the usage of heartbeat messages for nodes to monitor each other. Therefore, it does not work when location information is not available or there is communication blackout. Hence as a further enhancement we are planning to overcome the communication blackout.

REFERENCES

[1] http://ieeexplore.ieee.org
[2] http://en.m.wikipedia.org
[3] http://www.googleweblight.com
[4] http://ns2tutor.weebly.com
[5] http://installnam.blogspot.in
[6] http://www.pcquest.com
[7] http://evanjones.ca
[8] http://cygwin.com
[9] R. Badonnel, R. State, and O. Festor. Self-configurable fault monitoring in ad-hoc networks. Ad Hoc Networks, 6(3):458–473, May 2008.
[10] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and

tracking system. In Proc. of IEEE INFOCOM, 2000.

[11] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc., 2002.

[12] D. Ben Khedher, R. Glitho, and R. Dssouli. A Novel Overlay-Based Failure Detection Architecture for MANET Applications. In IEEE

[13] International Conference on Networks, pages 130–135, 2007.

[14] M. Elhadef and A. Boukerche. A Failure Detection Service for Large-Scale Dependable Wireless Ad-Hoc and Sensor Networks. In International Conference on Availability, Reliability and Security, pages 182–189, 2007.

[15] D. Liu and J. Payton. Adaptive Fault Detection Approaches for Dynamic Mobile Networks. In IEEE Consumer Communications and Networking Conference (CCNC), pages 735–739, 2011.

[16] C.-F. Hsin and M. Liu. A Distributed Monitoring Mechanism for Wireless Sensor Networks. In Proc. of ACM WiSe , December 2002.

[17] M. Natu and A. Sethi. Adaptive Fault Localization for Mobile, AdHoc Battlefield Networks. In Proc. of IEEE Milcom, Atlantic City, NJ, October 2005.

[18] N. Sridhar. Decentralized Local Failure Detection in Dynamic Distributed Systems. In IEEE Symposium on Reliable Distributed Systems (SRDS), pages 143–154, 2006.

[19] D.Chandra and G.S.Goldszmidt. OnScalableandEfficient Distributed Failure Detectors. In Proc. of ACM symposium on Principles of distributed computing (PODC), pages 170–179, 2001.

[20] A Binary Feedback Scheme for Congestion Avoidance in Computer Networks K. K. RAMAKRISHNAN and RAJ JAIN Digital Equipment Corporation.