ISSN No. 0976-5697
Volume 2, No. 1, Jan-Feb 2011
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

New Dynamic Routing Protocol for Wireless Adhoc Network

Pankaj Sharma* and Nikhil Patearia and Anil Rai
Department of C.S.E
Samrat Ashok Technological Institute
Vidisha, M.P., India,
pankaj.sati2010@gmail.com and nikhil_sati29@rediffmail.com and onlyanilrai@gmail.com

Abstract:: An ad hoc network is a spontaneous network that can be established with no fixed infrastructure. This means that all its nodes behave as routers and take part in its discovery and maintenance of routes to other nodes in the network. Realizing such a network presents very significant challenges, especially at the protocol and software level. Major steps forward are required in the field of communications protocol, data processing, and application support. Although sensor nodes will be equipped with a power supply (battery) and embedded processor that makes them autonomous and self-aware, their functionality and capabilities will be very limited. The resource limitations of Wireless Sensor Networks (WSN), especially in terms of energy, require novel and collaborative approach for the wireless communication. Therefore, collaboration between nodes is essential to deliver smart services in a ubiquitous setting. Current research in this area generally assumes a rather static network, leading to strong performance degradation in a dynamic environment. In this thesis we investigate new algorithms for routing in dynamic wireless environment and evaluate their feasibility through experimentation. The algorithms will be key for building self-organizing and collaborative networks that show emergent behavior and can operate in a challenging environment where nodes move, fail and energy is a scarce resource. This thesis has major contributions to the routing in dynamic wireless networks. Firstly, a combination between a new multipath on-Demand Routing protocol and a data-splitting scheme which results in an efficient solution for high reliability and low traffic. Also a data-centric approach based on cost estimation is designed to disseminate aggregated data from data source to destination with high efficiency.

Keywords: Wireless Network, AODV, Dynamic Routing Protocol.

I. INTRODUCTION

A wireless network is a computer network that uses wireless radio as its carrier or physical layer. It enables users to access information and services electronically, regardless of their geographic position. Wireless networks can be classified in two types:
Infrastructure-based networks and infrastructure less (ad hoc)networks. An infrastructure-based network consists of a network with fixed base stations which are connected by wires. The mobile hosts communicate with the base stations by wireless links, which enables them to move geographically within the communication radius of the base station. When the mobile host moves out of range of the connecting base stations, it connects with new base station for communication. In contrast to infrastructure-based networks, ad hoc networks are self-configuring networks which consist of mobile hosts only connected by wireless links. In an ad hoc network, all nodes are free to move randomly and organize themselves arbitrarily. Minimal configuration and quick deployment make ad hoc networks very suitable for emergency situations like natural or human-induced disasters, meetings or conventions in which persons wish to quickly share information [1]. A wireless sensor network, a special kind of ad hoc network, consists of a number of sensors spread across a geographical area. Each sensor has wireless communication capability and sufficient intelligence for signal processing and networking of data. However, sensor nodes are constrained in energy supply and bandwidth. Such constraints, combined with a typical deployment of a large number of sensor nodes have posed many challenges to the design and management of sensor networks. These challenges necessitate energy-awareness at all layers of the networking protocol stack. The issues related to physical and link layers are generally common for all types of sensor applications. Therefore, the research on these areas has been focused on system-level power awareness such as dynamic voltage scaling, radio communication hardware, low duty cycle issues, system partitioning and energy aware MAC protocols. At the network layer, the main aim is to find methods for energy efficient route setup and reliable relaying of data from the sensor nodes to the sink so that the lifetime of the network is maximized.

Wireless ad-hoc networks are autonomous systems of mobile nodes that form a network in the absence of any centralized support. This is a new type of network and might be able to provide services at places where otherwise no communication is possible. The absence of a fixed infrastructure poses several types of challenges for the routing of this type of network. Routing protocols for ad hoc networks can be divided into two categories: table-driven (pro-active) and on-demand routing (reactive), based on when and how the routes are discovered [2] and [5].

II. BACKGROUND

This related work first introduces the concept and development of multipath routing. Then we give a more detailed explanation of Dynamic Source Routing, which is used in the simulation later for comparison. Finally, we make a distinction between disjoint and braided multipath routing.

A. Multipath Routing

Multiple paths can be useful in improving the effective bandwidth of communication pairs, responding to congestion and bursty traffic, and increasing delivery reliability. It has been studied in several different contexts. Traditional circuit switched telephone networks used a type of multipath routing called alternate path routing to decrease the call blocking probability and increase overall network utilization. In alternate path routing, the shortest path between phone exchanges is typically one hop across the backbone network; the network core consists of a fully connected set of switches. When the shortest path for a particular source destination pair becomes unavailable, rather than blocking a connection, an alternate path, which is typically two hops, is used. Multipath routing has also been addressed in data networks which are intended to support connection-oriented service with QoS. For example, the PNNI signalling protocol has been used in Asynchronous Transfer Mode (ATM) networks to set up multiple paths between a source node and a destination node. The primary path is used until it either fails or becomes over-utilized, and then alternate paths are tried. Alternate or multipath routing has typically lent itself for use in connection oriented networks. However, in packet-oriented networks, like the Internet, multipath routing could be used to alleviate congestion by routing packets from highly utilized links to links which are less highly utilized. The drawback of this approach is that the cost of storing extra routes at each router usually precludes the use of multipath routing. However, multipath routing techniques have been proposed for Open Shortest Path First (OSPF), a widely used Internet routing protocol [3].

B. Split Multipath Routing (SMR)

which is an on demand multipath source routing protocol and similar to DSR. Unlike DSR, intermediate nodes do not keep a route cache, and therefore, do not reply to a route request. This is to allow the destination to receive all the routes so that it can select the maximally disjoint paths. Maximally disjoint paths have as few links or nodes in common as possible. Duplicate route requests are not necessarily discarded. Instead, intermediate nodes forward route requests that are received through a different incoming link, and whose hop count is not larger than the previously received route requests. The proposed route selection algorithm only selects two routes. However, the algorithm can be extended to select more than two routes. In the algorithm, the destination sends an route reply for the first route request it receives, which represents the shortest delay path. The destination then waits to receive more route requests. From the received route requests, the path that is maximally disjoint from the shortest delay path is selected. If more than one maximally disjoint path exists, the shortest hop path is selected. If more than one shortest hop path exists, the path whose route request was received first is selected. The destination then sends a route reply for the selected route request [2].

Temporally Ordered Routing Algorithm provides loop free multiple alternate paths for mobile wireless network by maintaining a destination oriented" directed acyclic graph (DAG) from the source. It rapidly adapts to topological changes, and has the ability to detect network partitions and erase all invalid routes within a finite time. Another candidate for multipath routing for WSN is *Directed Diffusion*, which features data centric dissemination and in network data aggregation. It can realize robust multipath delivery, empirically adapt to a small subset of network paths, and achieve significant energy savings when intermediate nodes aggregate responses to queries. Based on directed diffusion, a novel braided multipath routing scheme, which results in several partially disjoint paths. Results show it is a viable alternative for energy efficient recovery from failures in WSN [3] and [10].

An extension to the AODV protocol for computing multiple loop-free and link disjoint paths. Loop-freedom is guaranteed by using a notion of *advertised hop-count*. Link disjointness of multiple paths is achieved by using a particular property of flooding. To keep track of multiple routes, the routing entries for each destination contain a list of the next-hops along with the corresponding hop counts. All the next hops have the same sequence number. For each destination, a node maintains the advertised hop count, which is defined as the maximum hop count for all the paths. This is the hop count used for sending route advertisements of the destination. Each duplicate route advertisement received by a node defines an alternate path to the destination. To ensure loop freedom, a node only accepts an alternate path to the destination if it has a lower hop count than the advertised hop count for that destination. Because the maximum hop count is used, the advertised hop count therefore does not change for the same sequence number. When a route advertisement is received for a destination with a greater sequence number, the next-hop list and advertised hop count are reinitialized [4].

Another extension to AODV (AODVM) for finding reliablerouting paths. Intermediate nodes are not allowed to send a route reply directly to the source. Also, duplicate route request packets are not discarded by intermediate nodes. Instead, all received request packets are recorded in a route request table at the intermediate nodes. The destination sends a route reply for all the received route request packets. An intermediate node forwards a received route reply packet to the neighbor in the route request table that is along the shortest path to the source. To ensure that nodes do not participate in more than one route, whenever a node overhears one of its neighbors broadcasting a route reply packet, it deletes that neighbor from its route request table. Because a node cannot participate in more than one route, the discovered routes must be node-disjoint [5].

C. Dynamic Source Routing

We compare the performance of our protocol with Dynamic Source Routing, which is a simple and efficient on-demand routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR involves the following phases:

a. ***Route Request*:** the source floods the network with messages, trying to find in this way the destination. The messages increase in length by each hop they travel. If more than one route request message reaches a node, only the first one is processed and the others are discarded.

b. ***Route Reply:*** if the destination receives a route request message from the source it will reply with a message containing the path used to reach the source. In the case of bi-directional links, this path is simply reversed. The reply messages have constant length between the source and the destination. Still, their initial length depends on the number of hops between the source and the destination.

c. ***Route Maintenance :*** after the source has received a path to the destination, it sends the data packet on it. Each node is responsible for ensuring that the message travels to the next hop (this can be done for example by passive acknowledgement). If a node detects that a link is broken, it sends this information back on the path to the source. A new path has to be constructed or another cached path can be used. The length of the messages involved in this phase is dependent on the number of hops between them [1] and [2].

D. Disjoint And Braided Multipath

Out of many possible designs for multipath routing protocols, two distinct mechanisms exist: disjoint and braided.

a. ***Disjoint multipath*** routing tries to construct alternate paths which are node disjoint with the primary path, and with each other. Thus they are unaffected by failure on the primary path. But those alternate paths could potentially have much longer latency than the primary path and therefore consume significantly more energy than that on the primary path.

b. ***Braided multipath*** routing relaxes the requirement for node disjointness, which means alternate paths in a braid may partially overlay with the primary path and so are not completely node disjoint [3].

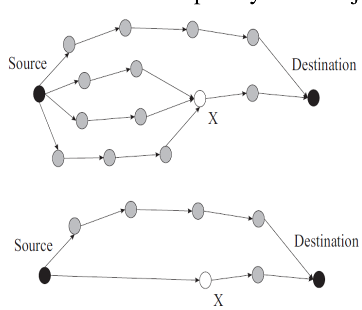

Figure.1 Braided multipath case

III. PROPOSED TECHNIQUE

Multipath Routing allows the establishment of multiple disjoint paths between source and destination, which provides an easy mechanism to increase the likelihood of reliable data delivery by sending multiple copies of data along different paths. Based on Dynamic Source Routing (DSR), we designed a new multipath routing algorithm Multipath on-Demand Routing algorithm (further referred to as MDR). The algorithm provides several paths from the source to the destination. A data splitting algorithm as presented to safely route data while keeping the amount of traffic low. The two phases of the algorithm are described below.

The MDR algorithm has two phases:

→***Route Request* -** when the source wants to find a destination it floods the network with a short message announcing this. The message contains the source ID, the destination ID and the ID of the request. Thus, the length of the message remains constant during the route request.

→ ***Route Reply* -** the destination will eventually receive one of the route request messages. It only knows that there exists a path. It is not interested in what the path is. The destination just returns a route reply to the neighbor from which it received the route request message. The message contains a supplementary field that indicates the number of hops it travelled so far. When this neighbor node receives the route reply, it increments the hop count of the reply message and then forwards the route reply to the neighbor from which it got the original route request.
This mechanism reduces the size of the messages considerably when compared to the original DSR. In fact we are moving the information stored inside the messages to the sensor nodes themselves. The sensor nodes are responsible to "remember" where the flooding message came from.
One can notice that there is no route maintenance. This approach will be discussed more in detail after the way the multiple paths are handled and the simulation results are presented.
The second groups of modifications involve the multiple paths management. In the original DSR, if the same route request message was received several times by a node, only the first one was considered and the rest were discarded. MDR considers all the messages and uses the whole information it can get out of them.

I. Pseudo code of the source node in MDR

For source node **S**
1. **IF S** has new packet to send and no route is known to the targeted destination
2. **THEN** forward route request message to all neighbor nodes of **S**; set route discovering timer;
3. **END IF**
4. **IF S** receives route reply from destination
5. **THEN IF** this is the first route reply
6. **THEN** set reply timer;
7. **END IF**
8. **IF** reply timer is not expired
9. **THEN** record the route and wait for more route reply
10. **ELSE** start transmission of the packet
11. **END IF**
12. **END IF**
13. **IF S** does not receive route reply from destination before route discovering timer expires

14. **THEN** restart route request (go to 1)
15. **END IF**

II. Pseudo code of the intermediate node in MDR

For intermediate node **i**
1. **IF** node **i** receives route request message from neighbor **j**
2. **THEN IF** same route request is not received before
3. **THEN** remember the S node, D node, ID of the request and the neighbor **j**; forward the route request with new *ack* and *lasthop*; put neighbor **j** in n-1 neighbor list
4. **END IF**
5. **IF** same route request is received from neighbor **j** before
6. **THEN** discard the route request
7. **ELSE IF** ack field in the route request is **i**
8. **THEN** put neighbor **j** in n+1 neighbor list
9. **ELSE** put neighbor **j** in n-1 neighbor list
10. **END IF**
11. **END IF**
12. **END IF**
13. **IF** node **i** receives route reply message from neighbor **j**
14. **THEN IF** node **i** is addressed by the route reply and this route
15. reply has not been forwarded by node **i** before
16. **THEN IF** n-1 neighbor list is not empty
17. **THEN** forward the route reply to the first neighbor in n-1 neighbor list
18. **ELSE IF** detours field in route request is larger than 0 and n+1 neighbor list is not empty
19. **THEN** forward the route reply to the first neighbor in n+1 neighbor list
20. **ELSE** discard the route reply
21. **END IF**
22. **END IF**
23. **ELSE IF** same route reply is received before
24. **THEN** remove neighbor **j** form both n-1 and n+1 neighbor list
25. **END IF**
26. **END IF**
27. **END IF**

III. Pseudo code of the destination node in MDR

For destination node **D**
1. **IF D** receives a route request addressed to it from neighbor **j**
2. **THEN IF** same route request is not received before
3. **THEN** remember node **S**, ID of the request and the neighbor **j**; set timer
4. **END IF**
5. **IF** timer is not expired
6. **THEN** send route reply to neighbor **j**
7. **END IF**
8. **END IF**

Route reply phase-

The Route Reply phase is the part of the algorithm in which several paths between the destination and the source are reported to the source (if they exist). The reply messages have fixed length. Because in the previous phase each node stored information about the neighbors that forwarded the route request message, the complete path between the source and the destination does not have to be stored inside the message.

Message description-

The route reply message contains two groups of fields. The first group of fields uniquely identifies the instance of the route reply, which consists of the following fields:
→*snodeID* the source node ID
→*dnodeID* the destination node ID
→*floodID* the flood message ID

The second group of fields changes when each intermediate node forwards the route reply, which consists of the following fields:
→ *lasthop* the ID of the node forwarding this message
→*nexthop* the ID of the node to which the message is forwarded
→*ack* the ID of the last hop
→*hops* the number of the hops the message travelled through
→*detours* the number of detours a message can take

The meaning of the field names is the same as in the previous phase. There are two new fields: the *nexthop fi*eld contains the ID of the node that has to receive this message. This information is provided by each node from their local data structure. The *hops fi*eld is incremented with each hop the message travels and represents the current path length. The *detours fi*eld specifies how many times the reply message is allowed to travel in an opposite direction (from source to destination).

Route Reply phase description-

When the first route reply message arrives at the source, this node stores the ID of the node that forwarded the message and the path length. It also sets up a timer to measure the interval that it will wait for other reply messages to come. When this timer expires it splits the original data message according to the number of paths, the maximum probability of failure and the length of the paths and forwards it. The paths can also be stored in a local cache (together with time information) for future usage (this feature is not implemented yet). A node that receives a route reply addressed to it will modify the second group of fields in the message according to the new parameters. Afterwards, it will forward the modified route reply to the first neighbor in the *n-1* neighbor list. If this list is empty and the detours field is not empty, it chooses the first neighbor in the *n+1* neighbor list and also decreases the *detour* variable by 1. A node that receives a route reply not addressed to it, searches its own data structure to find the entry corresponding to the first three fields. If such an entry is found, it removes the forwarding node from both *n-1* and *n+1* neighbor lists. A node that forwarded a message has to take care of two more things: first it sets a flag in his data structure saying that it will not forward any other message and second, it waits for the passive acknowledgement. If this does not arrive it assumes that the node, to which it sends the message is no longer there, is broken or it forwarded a message previously and it deletes it from his lists. It will try to resend the message to the next neighbor in the lists, until the lists become empty or the *detour fi*eld becomes 0.

The previous step of removing nodes from the list is needed to ensure that the source will receive only disjoint paths. If for various reasons, the paths from the destination to the source have to be known, each node that forwards a route reply message can append its ID to it. This way, the messages will grow in length, but this growth is controlled and involves only a subset of the nodes. After route request and route reply phase, the source obtains *n* multiple parallel paths to the destination node. When a link failure appears in the path, the intermediate node sends a route error message back to the source node. When the number of healthy paths is smaller than the estimated number of successfully paths, the source node will reinitiate a route request.

IV. RESULTS

The main parameters considered were the number of messages, the amount of traffic generated, the latency introduced and the connectivity of the network. An implementation of DSR with caching of the paths and the route maintenance enabled was also implemented for comparison. We have run both DSR and MDR for several network configurations.

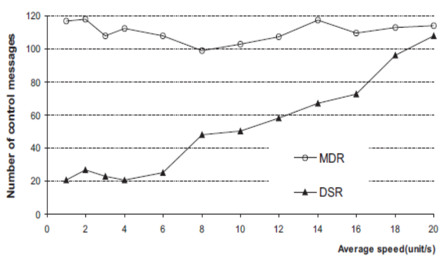

Figure.2 Comparison between MDR and DSR

The parameters were identical for both cases and also the generation of destinations. The DSR algorithm ad the caching of the paths and the route maintenance enabled. The results are presented in Figure which merely indicates the trends between the two compared the protocols. The figure shows that the number of overhead messages is higher for the MDR.

V. CONCLUSION AND FUTURE WORK

We have implemented this scheme and estimated the main characteristics. It greatly increases the reliability of packet delivery in wireless sensor networks, while keeping the total network traffic much lower than the traditional multipath routing. At the same time the latency of splitted multitpath routing is shorter than any retransmission scheme. An on-demand multipath routing algorithm offers the data source with several paths to any destination (if available). It is used in combination with a data splitting method based on odified Erasure Coding. By splitting the data across multiple paths, the traffic volume goes to much lower values compared with sending the same data across multiple paths. The trade-off is the reliability of the delivered packets. While this gives us a way to adjust the reliability while keeping the data traffic low. When using a lower value for *En* than the calculated one, the traffic increases but the percentage of failures decreases.

Our multipath routing with data splitting scheme provides an abstraction of a better transmission medium from the receiver side. In the future, a retransmission scheme could further handle the remaining error corrections. Future work will focus on the hybrid scheme of data splitting and retransmission similar which will ensure higher reliability in data dissemination.

VI. REFERENCES

[1] Kannan .T, M.Pushpavalli and Dr.A.M.Natarajan, “Fortification of QoS Routing in MANETs using Proactive Protocol”, IEEE2010.

[2] Yogesh Chaba, Yudhvir Singh and Manish Joon, ” Simulation based Performance Analysis of On-Demand Routing Protocols in MANETs,” IEEE 2010 Second International Conference on Computer Modeling and Simulation.

[3] A K Daniel, R Singh and Zubair Khan, “Position Based Multicast Routing Protocol for AD-hoc Wireless Network Using Backpressure Restoration”,IEEE2010

[4] L. Prema Rajeswari, R. Arockia Xavier Annie, A. Kannan, “ENHANCED INTRUSION DETECTION TECHNIQUES FOR MOBILE AD HOC NETWORKS” IEEE 2007 International Conference on Information and Communication Technology in Electrical Sciences.

[5] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications Magazine, 11(6):6{ 28, 2004.

[6] Amis, R. Prakash, D. Huynh, and T. Vuong. Max-min d-cluster formation in wireless ad hoc networks. In Proceedings of the Ninteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel-Aviv, Israel, volume 1, pages 32{41. IEEE Computer Society, 2000.

[7] G. E. Blomgren. Current status of lithium ion and lithium polymer secondary batteries. In Proceedings of IEEE Fifteenth Annual Battery Conference on Applications and Advances, Long Beach, USA, pages 16{23. IEEE Computer Society, 2000.

[8] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the fourth annual ACM/IEEE international conference on Mobile computing and networking, Dallas, USA, pages 85{97. ACM Press, 1998.

[9] B. H. Calhoun, D. C. Daly, N. Verma, D. Finchelstein, D. D. Wentzlo, A. Wang, S.-H. Cho, and A. P. Chandrakasan. Design considerations for ultra-low energy wireless microsensor nodes. IEEE Transactions on Comput-ers, 54(6):727{740, 2005.

[10] Jian Wu and Paul Havinga. Reliable cost-based data-centric routing protocol for wireless sensor networks. In Proceedings of 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Las Vegas, USA, pages 267 { 272. IEEE Computer Society, 2006.

VII. AUTHORS PROFILE

Mr. Pankaj Sharma presently pursuing M.Tech of the department Computer Science & Engineering at Samrat Ashok Technological Institute, Vidisha, M.P., India. The degree of B.E. secured in Information Technology from MITS, Gwalior. Research Interest includes Network Security and Data Mining. **Mobile:** +919179638494 **E-mail:** pankaj.sati2010@gmail.com.

Mr. Nikhil Patearia presently pursuing M.Tech of the department Computer Science & Engineering at Samrat Ashok Technological Institute, Vidisha, M.P., India. The degree of B.E. secured in Computer Science & Engineering at Truba Institute of Engineering & Information Technology Bhopal, in 2009. Research Interest includes Network Security, Data Mining and Artificial Intelligence. . **Mobile:** +91-9827632900, **E-mail:** nikhil_sati29@rediffmail.com.

Mr. Anil Rai presently pursuing M.Tech of the department Computer Science & Engineering at Samrat Ashok Technological Institute, Vidisha, M.P., India. The degree of B.E. secured in Information Technology from MITS, Gwalior. Research Interest includes Network Security and Data Mining. **E-mail:** onlyanilrai@gmail.com.