International Journal of Computer Science and its Applications [ISSN 2250 - 3765]

Efficient On-demand Routing Protocol for Ad-hoc Network

Vinod Mahor
Department of IT
Samrat Ashok Technological institute
Vidisha, M.P., India.
{*vinodengg.mt@rediffmail.com*}

Sanjaydeep Singh Lodhi and Mahendra Pratap Singh Dohare
Department of C.S.E. (Software System)
Samrat Ashok Technological institute
Vidisha, M.P., India.
{sanjayeng.mt@rediffmail.com and mpsingh.mtech@gmail.com}

Abstract-

A mobile ad hoc network (MANET) is a wireless network that does not rely on any fixed infrastructure (i.e., routing facilities, such as wired networks and access points), and whose nodes must coordinate among themselves to determine connectivity and routing. This paper addresses the process of designing a routing protocol for an ad-hoc network. There have been many proposed algorithms that solve the routing problem in a mobile ad-hoc network. It is a difficult task to compare the performance of these protocols qualitatively as there are many parameters that affect network performance. Various simulation packages for networks of this type exist. One such package is the Network Simulator. It is a discrete time event simulator that can be used to model wired and wireless networks. This paper presents the simulations results that compare recently proposed routing algorithms. From this comparison study it is shown that on-demand algorithms perform best in a mobile ad-hoc environment and also describes the design of a novel on-demand routing algorithm. The on-demand algorithms proposed thus far use a blind flooding technique during the route discovery process. This method is inefficient and creates excessive routing overhead. The routing protocol proposed in the work implements a query localization technique that significantly reduces the network traffic. Our simulation results show that such a scheme makes the on-demand routing algorithm more efficient and scalable than existing ones.

Keywords- Wireless Network, MANET, AODV.

Introduction

Mobile ad-hoc technology has attracted the attention of the communications field since the development of the Mobile Packet Radio Networks in research projects initiated by the US military in the 1970 and 1980s. The MANET is an autonomous network of mobile computers that are connected via wireless links. There is no pre-existing infrastructure and thus each node in the network may act as a host or as a router (an intermediate node) to allow connectivity between other source and destination hosts in the network. The term ad-hoc implies that the network is formed in an impromptu manner to meet an immediate and specific goal. Since the nodes in the network are mobile, the network topology can be configured in an arbitrary manner and can change dynamically. An ad-hoc network can operate in an isolated fashion or it can be connected to the wider internet via gateways. Due to the mobility of the nodes in a MANET, the network topology may be connected in any arbitrary manner and may change dynamically. Such a topology is randomly changing and is unpredictable. There has been advancement in the development of wireless modem technology which offers significantly higher data rates than in the past. However, the capacity of the wireless links is still significantly lower than the links in the wired environment. In addition, the radio communication in the wireless environment has to account for other issues such as multiple access, channel fading, noise and interference. This leads to a marked decrease in the realized throughput when compared to the radio's maximum transmission rate. The mobile stations in an ad-hoc network rely on batteries or other exhaustible sources for power. There have been leaps in the development of battery technology for mobile computing. However, for ad-hoc network systems, the efficient use of this most important resource to a node is vital to its operation. The envisaged applications of MANETs could use platforms other than laptops and notebook PCs such as Personal Digital Assistants (PDAs), and even smaller embedded devices. These devices will not offer the same amount of processing power and memory resources. Thus these resources available to a node have to be taken into account in the design of ad-hoc network systems. There are two technologies that are being considered for the wireless interface in ad-hoc networks: Bluetooth and IEEE 802.11 standards. The production of Bluetooth is as a result a collaborative effort of several communications companies (Ericsson, IBM, Intel, Nokia and Toshiba). Their aim was to produce a solution that would give mobile devices access to a wireless channel for communication purposes. The standard is ideal for small devices with short range low power radio links. It operates in the unlicensed 2.4 GHz band using Frequency Hopping Spread Spectrum (FHSS). Low power Bluetooth devices are the norm today and provide a range of approximately 15 m, although higher powered devices with ranges of 150 m are available. The limitation of the standard is in terms of the speed of operation with only 2 Mbps data rates being offered. The mobile hosts in an ad-hoc network are most likely powered by exhaustive resources such as battery power. It is therefore essential that any application running the ad-hoc network architecture uses some form of power control during transmissions. The transmit power not only has an impact on the battery life of a host, it also affects the range in terms of hops that a host's transmissions achieve. A higher transmission power would increase the range and make routing simple but it would also negatively affect the traffic carrying capacity of the channel with the increased congestion. Thus intelligent power control in an ad-hoc network application is important consideration. One of the major challenges in the research into ad-hoc networks is the security of connections between hosts in the network. With free-space radio transmission in the wireless environment it is fairly easy for a malicious host to eavesdrop on a communication session. This could lead to unauthorized access, information theft, interference, jamming and service degradation. Due to the multi-hop nature of ad-hoc transmissions, it is a very difficult task to even detect such intrusion. The field of security for ad-hoc networks is at a very premature stage and this issue has to be thoroughly studied before ad-hoc network systems can be practically deployed in real world applications [1], [3] and [5].

The objective of this research work is to simulate and compare prominent ad-hoc routing schemes and determine which one best suited the PCS network. This study show that on-demand algorithms are more suited to highly mobile ad-hoc networks than proactive algorithms. The contribution in this work is a new on-demand protocol that shows significantly better performance than other proposed ad-hoc routing schemes.

Background

The commonly used routing protocols in the wired networks are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). RIP is a distance vector protocol while OSPF is based on the link-state routing philosophy. The two protocols, although quite efficient for routing data in the wired networks of the Internet, are entirely unsuited for application in the mobile ad-hoc networks. The dynamic nature of MANETs causes random and unpredictable changes in the routes in the network. The slow update rate of the wired protocols diminishes their ability to converge to a steady state for finding routes in the ever-changing topology. As mentioned in the previous chapter, the bandwidth available in the links of the wireless ad-hoc network is significantly less than that in the wired environment. The routing overhead incurred by the distance vector and link state protocols in terms of protocol control messaging thus becomes much of a factor in the ad-hoc network environment. Finally, the computationally expensive operations of the traditional wired protocols would be highly taxing on the scarce CPU, memory and battery power resources of the mobile nodes in an ad-hoc network.

Classification of routing protocols

The task of routing involves making forwarding decisions for data packets depending on the routing state of the network. The routing protocol thus has a two-fold operation. The first is to collect information about the state of the network and secondly to use this information to create routes through which data packets are forwarded.

There are many protocols that have been proposed to solve the routing problem in ad-hoc networks. With the wide variety available there are different criteria that can be used to classify these protocols. One of the major criteria used is based on the idea of "when" a routing protocol collects information regarding the state of the network. There are protocols that constantly maintain the paths to all destinations and thus learn of the network topology before a forwarding request is made. These protocols are grouped into the ***proactive*** routing schemes. The second group of protocols in this classification, called ***reactive*** routing schemes, only becomes active after there is a request for a route. In most cases this type of protocol does not have a route to the destination before such a request is made as they purge routes that have not been used recently. The source node, in such cases, has to discover a route to the destination. Once a route is found it can be used for routing data traffic. The protocols execute route maintenance procedures in the event of a route to a desired destination breaking. There are different advantages and disadvantages with each type of routing scheme and so some protocol designs attempt to incorporate more than one philosophy. These protocols, termed ***hybrid*** routing schemes, use both proactive and reactive actions in their operation.

In addition to these broad criteria, routing protocols can be further classified according to the type of addressing used. Protocols that use *flat* addressing maintain an architecture where all the nodes in the network are on the same level. In *hierarchical* addressing the network is aggregated to form groups. This type of addressing is particularly appropriate in large networks where it is essential to reduce the control messaging overhead in the network [2], [4].

Destination Sequenced Distance Vector (DSDV)

The Destination Sequenced Distance Vector (DSDV) routing algorithm is the modification of the classic Distributed Bellman-Ford (DBF) algorithm. In a MANET any node in the network may be required to act as router and so each node maintains a routing table that lists all the nodes in the network of which it is aware. Each entry in the table contains the destination and the next hop addresses as well as the cost (in terms of hops) to get to the destination. The reason DSDV is an improvement on the original wired network protocol is that it avoids DBF's tendency to create routing loops. Each entry in the routing table and a protocol message update is marked with a *sequence number*. This number is maintained by the destination node of a route entry and is increased whenever the node publishes its routing information. The sequence number value is used by all other nodes in the network to determine the "freshness" of the information contained in a route update for the destination. Since the value is sequentially incremented, a higher sequence number implies that the routing information is newer.

In order to maintain routing information consistency in the network each router shares its routing table with its neighbours by means of routing updates. These updates are done both in a periodic and triggered fashion. The designers of the protocol proposed this method with the aim of alleviating the potentially large amount of network traffic that will be induced by the routing updates. In a periodic update which occurs at predetermined regular intervals, a node broadcasts its entire routing table in a packet termed a *full dump*. *Incremental* routing update packets are used when triggered significant topological change. The change could be either due to node mobility or link breakages to next hop neighbours. The *incremental* update packets only contain those entries which have changed since the last periodic update. The triggered updates with the smaller packet sizes result in the reduced overhead incurred by the protocol. A route table update entry contains the destination address of a node, the cost to reach it and the highest known sequence number for the destination. When a node receives an entry for a particular destination with a higher sequence number its old entry is replaced with the newer route. In the case where a node has to choose between two entries with the same sequence number, it selects the path with the least cost. An intermediate node that detects a broken route to a destination assigns an *infinity* value to the route's path cost, increments the entry destination sequence number and immediately broadcasts the information as an update. Using this technique critical network topology information such as link breakages is disseminated quickly across the network [6] and [7].

Ad-hoc On-demand Distance Vector (AODV)

AODV is another example of an on-demand route acquisition system where a route between two hosts in an ad-hoc network is only created when they wish to communicate. The protocol is similar to DSR in the route acquisition and route maintenance mechanisms. However the two protocols differ in that AODV stores the route information in a distributed fashion at each node on the route while DSR includes the route information in the header packet of each data packet that is transmitted. AODV maintains loop free routes at all times using sequence numbers. This mechanism is imported from the DSDV routing algorithm. Each node in the network maintains its own monotonically increasing sequence number, which is incremented whenever the node generates and sends a route request packet. The sequence number is used as a form of logical time-stamping and ensures that the most recent route is selected in the route discovery procedure.

AODV is a pure on demand algorithm and uses the route request/route reply cycle to discover routes to new destinations. The three main message types used by the algorithm are route requests (RREQ), route reply (RREP) and route errors (RERR). The protocol comes into action whenever a new route is needed to a destination. AODV utilizes an enhanced version of the traditional route table to store and maintain routes to destination nodes. These routes, however, are cached only as long as they are being actively used. Thus in most cases routes to destinations are not known prior to a route being requested. The protocol initiates a *route discovery process* by generating and transmitting a RREQ packet. Each route request packet is uniquely identified by the source IP address and a broadcast ID. The packet is broadcast to the source's neighbouring nodes. A node receiving the route request first checks to determine if it has recently processed a RREQ with the same source IP and broadcast ID. If a match is found the RREQ is silently discarded. If on the other hand the request is new to the node, it records a *reverse route* entry to the source node in its route table (or activates an old one). If the node is not the destination node or an intermediate node with a current route to the destination, it broadcasts the route request packet to its neighbours. This process continues until a node is reached which meets the two conditions. In this manner the RREQ packet is disseminated using a network wide flood until a route is found (refer to Figure 1). The reverse path setup at each intermediate node is shown in Figure 2. The destination node D does not accept the route request packet from node 7 since it has already received a request with the same details from node 5.

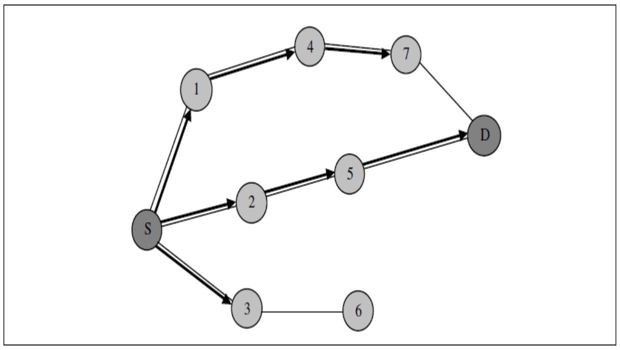

Figure 1: Network flood of route request packets in AODV

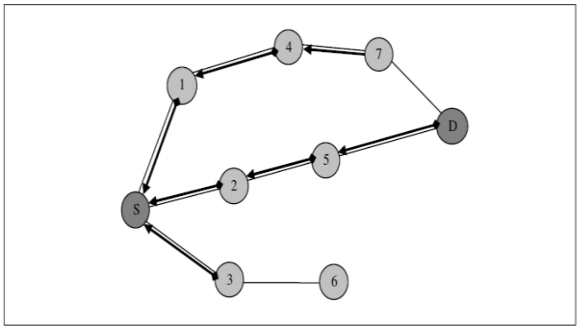

Figure 2: Reverse Path setup in AODV

The processing of the RREQ is different depending on whether the node is the destination node itself or an intermediate node with a current, active route to the destination. The decision on how current or "fresh" a route is, is determined by the value of the sequence number associated with the route entry. If the destination sequence number of the entry at the intermediate node is higher than that contained in the RREQ, the route is considered to be fresher. The intermediate node is permitted to reply to the route if such a condition is met. The destination node simply replies to the route request by generating and transmitting a route reply (RREP) packet. As seen in Figure 2, by the time the RREQ arrives at a node that can provide a route to the destination (or the destination itself), a reverse route is established to the source of the route request. The route reply that is transmitted travels along this route to get back to the source node. Each node through which the RREP packet hops sets up a forward pointer to the node from which the packet was received. The forward path set up can be seen in Figure 3. The hop count field in the RREP is incremented by each intermediate node that processes the message. When the reply reaches the source node, the hop count value presents the distance in terms of hop count between the source and the destination. Once the route reply arrives the route discovery process is terminated and the source can begin to send the packets that were queued for the destination.

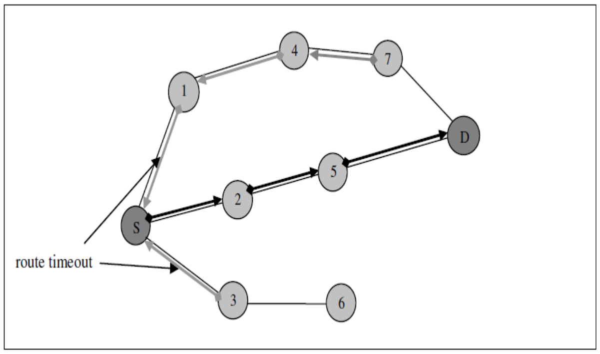

Figure 3: Forward Path setup in AODV

Once a route has been discovered, it is maintained as long as it is needed by the source node. If the network topology does not change there is little action on the part of the protocol. However route breakage, due to node mobility or link layer failure, results in execution of route maintenance procedures. There is an option in the protocol enabling the intermediate node to repair the route locally in a process termed *local repair*. If this option is not used or is not suitable, the intermediate node sends out a route error (RERR) message to the affected source node. One of the data structures in an entry in the protocols route table is a list of neighbours that use the current node as the next hop to get to the destination. This is known as the *precursor list*. Thus when an intermediate node detects a route breakage to a destination, it transmits the RERR to all its upstream neighbours in its precursor list. The message is unicast if the list contains only one neighbour, otherwise it is broadcast. When a neighbouring node receives the RERR, it marks the route to the destination as unreachable and transmits a route error to the nodes in its precursor list. After the reception of a RERR message the source node re-initiates a route discovery if the route is still required [8] and [9].

Associativity Based Routing (ABR)

The Associativity Based Routing1 algorithm was one of the first ad-hoc routing algorithms to consider a routing metric other than the smallest hop count. It defined a new metric called the *degree of association stability*. This associativity is a measure of a nodes connectivity relationship with its neighbours over time and space. Each node in the network periodically transmits a beacon to its neighbours signifying its presence. A node caches an entry for each neighbour which records the number of beacons received. This information is stored in a variable termed 'associativity tick' that is incremented each time a beacon is received. A high associativity tick value for a neighbouring node implies a low state of mobility for that node. A stable link with a neighbour provides an ideal opportunity to select the node for routing purposes. The protocol introduces other Quality of Service (QoS) parameters such as load, signal strength and battery life in addition to the associativity ticks to determine the degree of routing stability. The routes determined using this metric is expected to be long-lived routes. These routes however are not necessarily the shortest in terms of hop count between the source and destination. The protocol breaks the traditional paradigm, which holds that the shortest path is ideal. Thus, although a longer path is sometimes chosen, with the high degree of stability, the route will be maintained with lower probability of having to execute route recoveries [7] and [10].

Proposed Technique

The proposed technique presents the effect of adding query localization to the route discovery process of an on-demand algorithm. The aim is to make the flooding technique more efficient. The protocol presented in this chapter also introduces a load metric, along with the hop count, as a decision criterion for route selection. The protocol performs load checking with the aim of balancing the traffic load in the network. Flooding is a robust method of getting the route request packet to every possible node in the connected component network. However, it is unnecessary for the route request to reach every possible node, especially those intermediate nodes not in the path of the source and destination. In a large and highly mobile network considerable routing overhead is incurred by the flooding method. This reduces the advantage, in terms of protocol routing overhead, which on-demand algorithms have over table driven ones. If the flooding could be made more efficient it would lower the routing overhead incurred. There would be further benefits such as reducing network congestion with fewer route request packets being transmitted in the network. Route request packets are generally broadcast packets and can have an adverse effect on data transmission over the wireless channel due to the broadcast storm problem. The effect of this problem would be reduced with a more efficient flooding method. One way to make the flooding of the route request packets more efficient is to intelligently reduce the region in the network where the packet is flooded. Query localization in ad-hoc networks has been examined in different forms. As mentioned in the introductory chapter of this dissertation, each host in the Positional Communication System (PCS) network is enabled with a Global Positioning System (GPS) module that provides the location information of each host. If a router (which could be any node in the ad-hoc network) has prior knowledge of the destination's location information, it could use this information to aid the query localization process. The method proposed in this work is based on ideas presented in the Location Aided Routing (LAR) algorithm. LAR uses location information to separate the network space into different regions for each communication session and the transmission of packets differ depending on the region in the network. What is proposed herein is that the location information be used to determine the proximity of an intermediate router to the destination node. Once this is determined, if an intermediate node is closer to the destination node than the node that passed it the route request packet, it forwards the packet to its neighbours. The packet is dropped if the intermediate node is found to be further away. The aim of the query localization is to bring the route request packet physically closer to the destination node with each hop and hence prevent it from traversing to unnecessary parts of the network.

Description of the proposed protocol

The protocol proposed in this dissertation, called Tactical AODV (TAODV), is a modification of the Ad-hoc On-demand Distance Vector (AODV) routing protocol. Although the proposed improvements mentioned in the last section could be applied to most on-demand algorithms, AODV was chosen. This is because it was shown that AODV has the best performance under PCS appropriate network conditions when compared to other protocols in simulation comparisons done herein. TAODV is similar to AODV in that the routing information for each route to a destination is maintained in a distributed fashion in the routing tables of the nodes in the network. The protocol only creates routes to destination nodes when such routes are requested by the generation of data packets for the destination. Routes are only maintained as long as they are being actively used. There is a timeout period for each route, and if a route is not used in that period it is considered to be inactive and is purged. If a source node does not have a route to the destination, it initiates a route discovery. The data packets for the destination host are transmitted once a route is found. If the route is broken during the communication session between the source and destination node, it is repaired before further transmission can continue.

Route localization

The route localization used in TAODV is an optimization of the flooding technique used by on-demand algorithms. If available, the location information of the destination node is used to determine if an intermediate node (acting as a possible router) should rebroadcast a route request packet. It will only rebroadcast the packet if it is deemed to be closer to the destination than the node from which it received the route request packet. This method aims to prevent route request packets from traversing to unnecessary sections of the network i.e. going to nodes that are not in vicinity of the path between the source and destination pair. Preventing route request packets from reaching such areas will result in a reduced protocol routing overhead.

The dissemination of a node’s location data occurs in an on-demand manner. There is no periodic transmission of the location data, it is thus not necessary to change the basic routing mechanism of the protocol to accommodate the route localization algorithm. Other nodes in the network will only know of a nodes whereabouts if they have communicated with it, or acted as a router for any of its routes. The location information of a source and destination node is piggy-backed with each route request and route reply packet respectively. Refer to the appropriate fields of the route request (RREQ) and route reply (RREP) structures. The route localization is implemented as follows. When the route request is generated for the destination node, the source node inspects its location cache to see if it has a location entry for the destination. This is likely if it has either communicated with the destination previously or acted as router for it. If the location entry is found, the positional information of the destination (its *x* and *y* coordinates) is used to calculate the distance to it using equation (1). There is no account of height in this distance measure as currently NS-2 only supports a flat two dimensional grid. This could however be an optimization in the implementation of the proposed routing protocol in the real world test-bed.

$$D_{sd} = \sqrt{(\Delta x)^2 - (\Delta y)^2} + \lambda ...... Eq^n (1)$$

D_{sd} represents the distance from the source to the destination node.

Δx and Δy is the difference between the x and y coordinates of the source and destination nodes respectively.

l is a factor that takes into account the approximation of the distance measure and is given by equation (2).

$$\lambda = v \text{ X } (t_c - t_d) ....\text{Equation (2)}$$

t_c is the current time.

t_d *is* the timestamp of the location information. This holds the value of the time instance the destination node published its positional information. *v* is the specified maximum speed that a node can move.

The application for this protocol is the Positional Communication System which is aimed at foot soldiers in a battlefield situation. The maximum speed used in the design of the protocol is the maximum practical speed that a soldier can move in such an environment (the maximum speed used in the simulations was 8m/s). The measure given by equation (2) is the worst case scenario in terms of the distance to the destination node. It is a measure that is imilar to the *expected zone* concept proposed in the Location Aided Routing (LAR) protocol.

The distance calculated by the source node (D_{sd} in Figure 4) is included in the route request packet broadcast to its neighbours. The timestamp of the location information (t_{sd}) that was used to calculate D_{sd} is also one of the fields in the packet. When a node between the source and destination receives the packet, its first action is to query its location cache for an entry for the destination node. If an entry is found, the timestamp of the entry (t_{id}) is compared with the location information timestamp in the route request packet (t_{sd}). If tid is newer than t_{sd}, it implies that the intermediate node’s location information for the destination is more current.

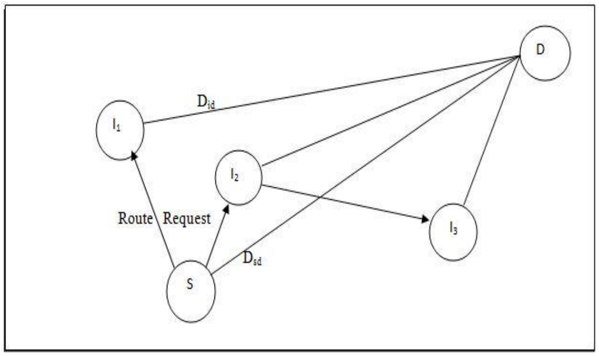

Figure 4: Rebroadcast decision in TAODV

The intermediate node then calculates the distance from the source to the destination (D_{sd}) using its information for the destination and the location information of the source from the route request packet. The distance field in the route request packet is updated accordingly. If t_{id} is not newer, the distance field remains unchanged. The intermediate node then calculates its distance to the destination node and compares this value with the source to destination distance. If D_{sd} is found to be larger than D_{id} (the intermediate node is closer to the destination than the node from which the route request arrived), the intermediate node broadcasts the route request packet. The distance metric to the destination is equated to D_{id}, narrowing the localization region with each hop. If this condition is not met the packet is dropped. During the localized route request process, if any node has no entry for the destination in its location cache, it does not execute the localization algorithm. The node instead broadcasts the route request as it is done in a blind flood. If the location aided route request fails to secure a route to the destination after a certain number of attempts, subsequent route requests are done using the blind flood method. If a route cannot be found using the network wide flood within a route request timeout period, it is assumed that destination is unreachable and the route discovery is terminated.

Load Checking

The load checking operation of the protocol is done at each of the intermediate nodes that process a route request. The length of the protocol queue at each node is taken as a measure of the load at a node. The protocol queue is a first-in-first-out (FIFO) queue in which packets that are awaiting routes are temporarily stored. Packets could be awaiting routes either due to route discoveries being attempted for unknown destinations or for broken routes to destinations that are being repaired. When an intermediate node receives a route request, the first decision that is made at the node is whether it will get involved in the route. The node inspects its protocol queue and if it is near to its capacity the intermediate node rejects the route request by dropping the packet. This will prevent already congested nodes from further overload. After the initial load check, if the intermediate node decides to act as a router for the source, it first creates a reverse route in its route table entry for the source node. This entry will be used to unicast reply packets back to the source node. The intermediate node then executes the load checking algorithm. The method used in this work is similar to one of the route selection procedures in the Dynamic Load Aware Routing (DLAR) protocol. A load variable in the route request packet counts the number of nodes between the source and destination that have their protocol queues loaded with packets above a certain threshold t. This variable is initialized to zero by the source node when it generates the packet. Prior to broadcasting the route request packet, an intermediate node inspects the length of its protocol queue. If the queue length is above the threshold value t, it increments the *load* variable in the packet. The variable keeps its previous hop value if the queue length is below the threshold value.

The load checking process is executed with each hop that the route request packet takes on route to the destination node. TAODV specifies that the route request should travel to the destination node and that no intermediate node is permitted to reply to the route request. Minimum hop count algorithms such as AODV and DSR specify that, if intermediate nodes have a cached route to the destination, they should send a route reply on behalf of the destination node. However, there are certain disadvantages to this methodology. Intermediate nodes replying to route requests generate a flood of route reply packets, which causes significant routing overhead. Route reply packets are unicast transmissions and use the RTS/CTS/DATA/ACK exchange of the 802.11 MAC protocol. Thus a route reply flood can create high level of congestion in the wireless channel. The least hop count method also results in certain routes in the network overlapping which creates congestion at certain nodes, creating bottlenecks. A further disadvantage of intermediate nodes replying on behalf of destination nodes was found from a real world implementation of the AODV protocol. It was noted that when an intermediate node sends a route reply to the source on behalf of the destination node, the route is unknown to the destination node. Since it does not receive the route request packet it does not learn of the route to the source node. It is possible that all route requests are replied to on behalf of the destination node, and consequently it will never learn of a route to the source node. This could result in poor performance if the source node wishes to establish a TCP connection with the destination. This discovery has led to the modification of the AODV protocol in the form of *gratuitous* route replies, which are sent to the destination node informing it of the route to the source. Thus, TAODV specifies that only the destination node responds to route requests by sending a route reply, eliminating the need to resolve such issues. This method, having the benefit of reducing routing overhead, is essential in the acquisition of the most up to date load information on route to the destination. The contents of the route reply (RREP) structure.

The heuristic used to select routes is a combination of the load information and the hop count. There are essentially three conditions that determine whether a route is better than another.

Case 1: A route is considered better if the *load* variable of a newly arrived route is lower than that of the previous route.

Case 2: The *load* variable is equal between the routes being compared. In such a situation the route with the lower hop count is deemed a better route.

Case 3: If the newly arrived route has a higher hop count than the previous route, it is only recognized as a route with a better metric, if the *load* variable is lower. In this case, although a longer route in terms of hop count is chosen, it is a route that is less loaded.

The protocol is simulated with and without this case and it is observed from the results that the inclusion of the case in the heuristic improves its performance. It would be expected that the end-to-end delay performance would suffer due to longer routes being preferred. However, it was noted that the correlation between the end-to-end delay and the number of hops is usually small. This is due to delays caused by various buffering and queuing, and the time spent gaining access to the radio medium in a congested node is more significant than a less congested single hop.

Results

Packet Delivery Ratio:

The packet delivery ratio results for the two different network configurations are shown in the graphs that follow. The packet delivery ratio for both the networks in the highly mobile scenario is presented in Figure 5 and Figure 6. It can be observed from these two graphs that the protocol's performance decreases with increasing speed. The packet delivery ratio diminishes at high speeds as both the protocols drop packets when there is considerable topology change and links to next hops are consistently broken. The route repair process involves route re-discoveries, which if not achieved within a certain number of attempts causes further data packets to be dropped. The curves for the two protocols display a similar shape which is expected due to similarities in their operations. However, the proposed protocol's delivery ratio is consistently better than AODV in all scenarios.

TAODV's packet delivery ratio results show particular advantage over AODV in the 16 node network configuration. There is (at least) a 10% gap in the packet delivery ratios achieved by the two protocols with varying mobility speeds. The routing of network traffic by the shortest-path AODV algorithm creates congestion at certain nodes in the network. The protocol queues of the routing agents are of a limited capacity, and when overloaded, packets are dropped. Since TAODV avoids the creation of such congestion scenarios, the dropping of packets from overloaded protocol queues is significantly reduced. This observation was made from the study of the output *trace* files of the simulations. The number of packets dropped due to a node's interface queues being overloaded was determined and it was noticed that AODV dropped far more packets due to this than TAODV.

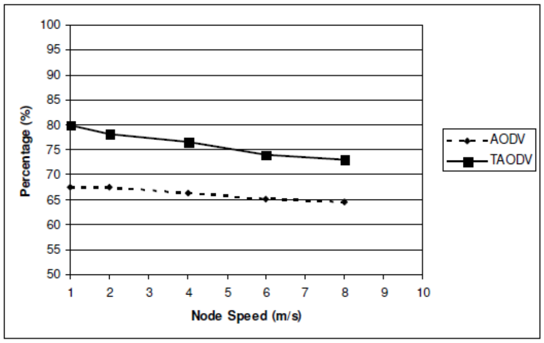

Figure 5: Packet delivery ratio for highly mobile network (pause time of 1 second)

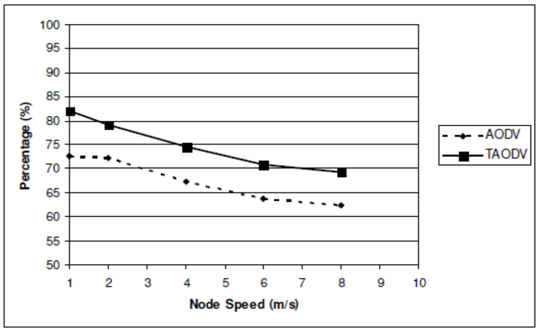

Figure 6: Packet delivery ratio for highly mobile network (pause time of 1 second)

Average end-to-end delay:

The average end-to-end packet delivery results show that TAODV delivers packets with less delay than AODV in all of the scenarios considered for the simulations in Figure 6 and Figure 7. The delay is significantly lower in the 16 node network configuration where there is a considerable network traffic load. In both the high mobility and the low mobility scenario sets, the delay shown by TAODV is at least 200ms lower. In the larger 50 node network the delay performance difference exceeds 100ms for the high mobility network scenario. This is similar to the results shown in figure below.

The quantitative value for the delay in the 50 node is considerably higher than the 16 node network, and this is expected since (on average) the routes in this network will have a higher number of hop counts.

The magnitude of the end-to-end delay for AODV is close to that presented for a similar network setup.

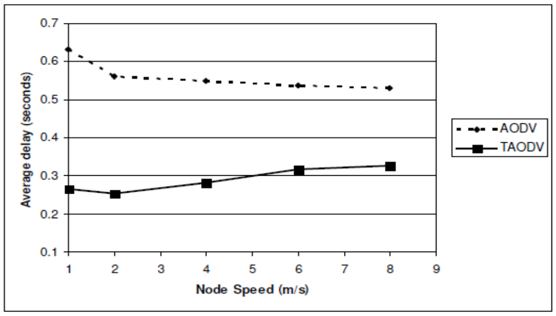

Figure 7: Delay performance for a highly mobile network (pause time of 1 second)

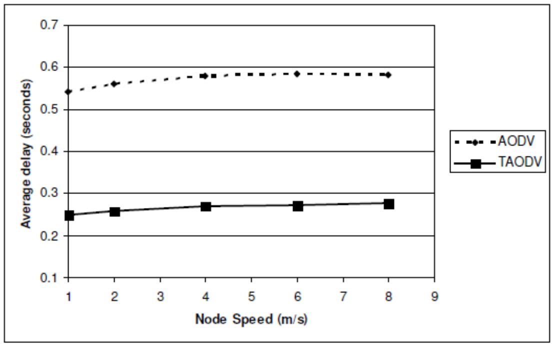

Figure 8: Delay performance for a stable network (pause time of 250 seconds)

The reduced delay is shown by TAODV is expected as the protocol prevents the creation of congested critical nodes in the network. Such congestion results in packets being queued at overburdened protocol queues. Since these queues are implemented in a first-in-first-out (FIFO) structure, such packets will remain queued for a considerable amount of time. It should be noted that TAODV's route selection procedure will choose a route with a higher hop count if the route load is lower. It would be normally expected that traversing through extra hops would increase delay. However the delay results presented in the graphs that follow show that delay due to queuing at nodes in a congested route is considerably higher than delay due to less loaded routes with higher hop counts. The load balancing executed by TAODV results in network traffic being shared amongst the nodes in the network. This has a marked effect in reducing the latencies in the delivery of data packets.

Conclusion and Future Work

The current wireless network structures offer only limited mobility and this paper has highlighted the reasons for ad-hoc networks being the next step towards truly ubiquitous computing and communications. Developing solutions for mobile ad-hoc networks is a significant challenge because of their unique characteristics such as the networks having dynamic topologies, and nodes in the network having limited resources. Bandwidth and energy are included in the list of exhaustible resources available to a node. Despite the challenges there have been a wide variety of applications envisaged for ad-hoc networks. The military applications such as sensor networks and tactical networks are the primary focus of this dissertation. In particular, the Positional Communication System is being developed for situational awareness in the modern battlefield.

One of the main obstacles in ad-hoc network technology is the routing problem. Due to the challenges previously mentioned the design of ad-hoc routing protocols has received a great deal of attention recently. There have been many proposed solutions to the routing problem. The dissertation presented a classification of the algorithms showing the different philosophies used in the design of ad-hoc routing protocols. The purpose of this research was to select or possibly develop a routing algorithm that would be suitable for its intended application, namely the PCS tactical network. Qualitative performance analysis is limited in indicating which ad-hoc routing philosophy and more specifically which routing algorithm is best suited for a general ad-hoc network application.

The future work in the development of the proposed algorithm includes improvements to both the query localization and the load checking algorithm. Currently the location information used for the query localization is disseminated in an on-demand manner. Further techniques, which could possibly make the dissemination process faster and more efficient, have to be investigated.

References

[1] G.Narsimha Dr. A.Venugopal Reddy Prof. S .S.V.N.Sarma, "The Effective Multicasting Routing Protocol in Wireless Mobile Adhoc Network", IEEE Proceedings of the Sixth International Conference on Networking (ICN'07).

[2] A.A. Bhorkar, M. Naghshvar, T. Javidi and B.D. Rao, "Exploring and Exploiting Routing Opportunities in Wireless Ad-hoc Networks", Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference.

[3] Jie Wu and Yunsheng Wang, “A Non-Replication Multicasting Scheme in Delay Tolerant Networks”, IEEE2010.

[4] G. LAVANYA, C.KUMAR and A. REX MACEDO AROKIARAJ, “SECURED BACKUP ROUTING PROTOCOL FOR AD HOC NETWORKS”, IEEE 2010 International Conference on Signal Acquisition and Processing.

[5] Sohail Jabbar, Abid Ali Minhas, Raja Adeel Akhtar, Muhammad Zubair Aziz, “REAR: Real-time Energy Aware Routing for Wireless Adhoc Micro Sensors Network”, 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing.

[6] N.Bhalaji, Dr.A.Shanmugam, “ASSOCIATION BETWEEN NODES TO COMBAT BLACKHOLE ATTACK IN DSR BASED MANET”, IEEE 2009.

[7] Rekha Patil, A. Damodaram, Rupam Das, “Cross Layer AODV with Position based Forwarding Routing for Mobile Adhoc Network”, IEEE.

[8] N.Sengottaiyan, Dr. Rm. Somasundaram, Dr. S.Arumugam, “An Modified approach for measuring TCP Performance in Wireless Adhoc Network”, 2010 International Conference on Advances in Recent Technologies in Communication and Computing IEEE.

[9] L. Zhang, H. Sun, Q.Sun, X.-Y.Hu, C.-J.Guo, “Improvement of routing protocols AODV in wireless Ad-hoc network”, World Congress on Software Engineering, IEEE 2009.

[10] N. M. Murad, “Low Energy Clustering Adaptation Protocol for an Adhoc Wireless Sensor Network”, IEEE 2009.