Performance of the Ad-hoc On-Demand Distance Vector Routing Protocol

Tuulia Kullberg
Helsinki University of Technology
`tuulia.kullberg@hut.fi`

Abstract

The amount of wireless communication devices has increased dramatically over the last few years. This has created new kinds of requirements to the technology as the growing number of users want to be able to communicate with each other anywhere and anytime without having to rely on any existing infrastructure or centralized access point. Ad-hoc network is composed of a collection of mobile nodes co-operating together to form a such network. Every node in ad hoc network acts both as a host and a router, which eliminates the need for existing infrastructure. Ad-hoc On-demand Distance Vector Routing protocol (AODV) is one of the developed protocols that enable routing with continuously changing topologies. AODV is reactive which means that it builds routes only when they are first needed. It uses extensive flooding of messages when discovering routes but tries to increase the overall bandwidth available by minimizing the use of any periodic advertisements. The increasing popularity of these on-the-fly networks has arisen the question about the efficiency and accuracy of the routing protocols used. This paper presents the performance and scalability of the AODV protocol both in small and large networks.

KEYWORDS: Ad-hoc On-Demand Distance Vector, packet delivery fraction, delay, control message overhead, expanding ring search, query localization, local repair

1 Introduction

The wireless communication has experienced considerable changes during the last few years as the communication devices have gained remarkable improvements both in convenience and performance. The technology has enabled longer battery powers and bigger memory capacities while the number of applications targeted at wireless communication has risen. Wires no longer tie users at their desks and the need for on-the-fly networks has grown.

The idea of forming a network without any existing infrastructure originates already from DARPA (Defence Advanced Research Projects Agency) packet radio network's days [1]. Today's available and licence-free mobile communication devices has increased the interest in the subject to a higher level than ever before and ad hoc networks have become a widely studied subject among the computer scientists.

The basic idea in ad hoc networks is that every node must participate not only as a host, but also as a router and forward packets to their destinations. As the topology goes through continuous changes, the task of completing the routing efficiently and correctly is demanding. Many different routing protocols based on different assumptions and intuitions have been proposed and implemented. Ad-hoc On-Demand Distance Vector routing protocol (AODV) is one of them. It establishes routes when they are first needed, from where the name *on demand*. AODV enables the dynamic route discovery and maintenance between the nodes using symmetric links and three kinds of message types.

As the number of nodes participating in forming ad hoc networks has risen, the performance and scalability of the proposed routing protocols have become a widely studied subject. This paper studies the efficiency and correctness of AODV in terms of node mobility and the amount of packet transmissions.

The remainder of this paper has the following structure. Section 2 introduces the AODV protocol in short. Section 3 concentrates on the performance of AODV in small networks. Large networks and scalability are studied in Section 4. Section 5 presents additional observations regarding the performance and scalability. Finally section 6 concludes the paper.

2 The Ad-Hoc On-Demand Distance Vector Algorithm

According to Lee, Royer and Perkins [2] AODV [8, 1, 5, 3] is currently the leading and mostly used routing protocol in ad hoc networks. It does not maintain any routing information about routes before they are truly needed and can thus be called a *pure on-demand route acquisition* system.

Like all ad hoc routing protocols, AODV has been designed to be used in networks, where all the participating nodes are considered to be friendly and trust each other. No security features were designed to the original AODV because of this. However, as the number of nodes in ad hoc networks increases, it becomes impossible for the participants to know each other well enough to be able to trust. This is the reason an extension called Secure Ad hoc On-Demand Distance Vector (SAODV) [10, 7] has been developed. In short, it provides integrity, authentication and non-repudiation mechanisms for route discovery processes but securing the actual data transmissions must be handled by other means. Quality of Service is neither largely supported in AODV. It has not a guaranteed packet delivery as packets are not retransmitted if they are lost during the transmissions. However, extensions [9] to the routing tables can be made to ensure that the experienced delays are under some predefined value *t* and the bandwith stays over the minimum requirements.

AODV has two main objectives. First, it enables route establishment between a source and destination node by initiating a *route discovery process*. Second, it maintains the active routes, which means finding alternative routes in a case of a link failure and deleting routes when they are no longer desired. In a highly mobile network this is a demanding task to be performed efficiently and accurately.

The next sections will introduce the principles of route discovery in Section 2.1 and route maintenance in section 2.2.

2.1 Route discovery

Before a source node is able to send data to the destination node, it has to have access to a valid route information in its routing table. Whenever the needed information is unknown, a *Route Discovery* process has to be initiated. It consists of two phases: broadcasting the query and waiting for the reply. The actual route setup is done during these two phases.

Both of the operations will next be covered in more detail. First is examined the use of Route Request and Route Reply messages as part of the route establishment, after which the dynamic route setup processes are studied.

2.1.1 Route Request and Route Reply

Every node using AODV is assumed to maintain two kinds of sequence numbers: *a node sequence number* and *a broadcast id*. Both of these are incremented monotonically and meant to ensure fresh and loop-free routes. The increment for the node's sequence number is done right before a node initiates route discovery or replies to a route query. Broadcast id is also incremented for every new route discovery.

To initiate a route discovery process, a source node broadcasts a Route Request (RREQ) message to all of its neighbours. The RREQ includes IP addresses for source and destination nodes, source node's sequence number, the last known sequence number for the destination, broadcast id and a hop counter. Broadcast id together with the source node's IP address uniquely identifies every RREQ sent. They are used by nodes to recognize the duplicates of the same RREQ, which upon reception are silently discarded. Hop count tells the distance between a source and destination node in hops and is used by the source node to select the shortest route among all the received route suggestions with same destination sequence number.

The RREQ is forwarded until the destination node or node with a fresh enough route in its routing table is found. The routing information is fresh enough when the sequence number the intermediate node has for the destination is greater that or equal to the destination sequence number contained in the RREQ packet. Whenever a node is allowed to answer the RREQ, it replies by unicasting a Route Reply (RREP) message containing the route to the destination backwards the route RREQ had traversed. This is why AODV only supports symmetric links and with every RREQ reception a node must check that the packet was received over a bi-directional link.

If the RREP is generated by intermediate node, the destination node is not usually informed about the new route suggestion. This can, however, be forced by setting a so called *g*-flag in the RREQ at which time the replying node unicasts also a gratuitous RREP containing the route information to the destination.

One RREQ is usually replied by many nodes which results in multiple RREPs. From among these the source node then selects the route with the highest destination sequence number. If it receives multiple answers with same sequence number the shortest route is selected.

2.1.2 Route Setup

The reverse route from the node towards the source node is established whenever a node receives a RREQ originated by the source node. Upon reception, it updates its routing table entry to point to the neighbour sending the message, even though it has no knowledge whether the selected route will traverse through it or not. It also updates the sequence number of the source node it has to correspond to the one contained in the RREQ message. Each one of the added entries contains the information about the neighbour node, the query information and an expiration timer. This expiration timer causes the route entry to be removed from the table in a case that the final route does not traverse through the node.

The forward routes on the other hand are created in a similar fashion after the RREQ is answered and the RREP is unicasted along the just established. When receiving the RREP, the node updates its routing tables to contain this new entry information about the neighbour node sending the message and the sequence number for the destination. It also resets the timer for the reverse route as it is now closer to be selected. However, it still has no knowledge whether the source node will select this route or not and an expiration timer is needed for also this new entry. If the route is selected, the timer is reset for every packet delivery the node performs.

If the established routes are not used within a specific time interval, they are removed from table entry and thus become unusable.

2.2 Route maintenance

After the route between the source node and the destination node has successfully been established the data delivery can continue until the topology of the network changes such that it affects the active route. This so called *link failure* can happen when either the source node, destination node or some of the nodes along the route moves or encounters an error. AODV is conservative in a way that when a route encounters a failure and deletion is needed, all the nodes using the failed link on its route to any destination are notified.

If the link failure results from the movement of the source node, it is up to the source to reinitiate a new route discovery process to be able to continue communication. However, if the moving node is the destination node or one of the intermediate nodes, a special kind of Route Reply message called Route Error (RERR) is used to inform other nodes along the route and the source node of the route failure. The failure is always detected by an upstream node, who is thus responsible for initiating the RERR message. After receiving a RERR message the source node can reinitiate the route discovery process if it so desires.

To learn about their local connectivity nodes can use *hello* messages, which are periodically broadcasted to all nodes within the transmission range. Time To Live (TTL) field is set to 1 to limit the spreading of the packet to direct neighbour nodes. However, hello messages should only be used by nodes on the active route, which enables active link failure detections right after the error has encountered. Other nodes should use link layer detection mechanisms to reduce the amount of control overhead hello messages produce.

	[6]	[12]	[11]
#Nodes	50 (100)	50	100
Area (m*m)	1500*300 (2200*600)	1500*300	750*750
Speed (m/s)	0-20	0-20	0-10
Pause time (s)	0-900	0-900	0
Sim.time (s)	900 (500)	900	100

Table 1: Parameters used in AODV simulations [6, 12, 11]

3 Performance

Many simulations concerning the performance of the original AODV protocol have been made since the idea and the first implementations were developed. Different kinds of tests have been run to show that the protocol is both accurate and quick in forming and maintaining the routes. AODV has been compared to other ad hoc protocols like *Dynamic Source Routing* (DSR) [6] or *Temporally-Ordered Routing Algorithm* (TORA) [12] to find out the performance differences caused by different approaches in algorithms. The idea behind the many comparison experiments is that better knowledge of relative merits between the compared protocols serves as a cornerstone for development of more efficient routing protocols in the future. In 2000 Royer and Perkins [4] extended the simulations to cover the effects of transmission range in performance.

The main performance metrics evaluated during the simulations have been as follows.

- *Packet delivery fraction*

  Ratio between the data packets delivered to the destinations (i.e. throughput) and those originated by the source nodes. Packet delivery fraction affects the maximum throughput that the network can support.

- *End-to-end delay*

  The time it takes for a data packet to traverse from the source node to the destination node. End-to-end delay evaluates the ability of the protocol to use the network resources efficiently.

- *Control message overhead*

  The number of routing packets transmitted. Can be normalized against the number of all sent packets. Evaluates the efficiency and scalability of the routing protocol.

It is worth noticing that these metrics aren't totally independent of each other. As the network size increases, longer routes must be established and maintained, which increases the probability of a packet drop. This means less sample packets for the end-to-end delay evaluation. The existing samples will even be biased in favour of shorter route lengths, which causes a decrease in measured end-to-end delay.

The simulations are run by varying both mobility and number of sources as well as the offered load. The offered load influences the protocol's performance in a way that the higher the number of packets to send is, the higher the network congestion and packet interference gets, both of which decrease the measured packet delivery fraction and throughput. However, it should be remembered that they are more related to the network's properties than those of the routing protocol.

All of the simulations used the same simulation model based on *ns-2*, which is a discrete event simulator developed by the University of California at Berkeley and the VINT project. The nodes in the network move according to a *random waypoint*-model, which means that each node starts its journey from a random location towards a randomly chosen destination and moves there with a randomly chosen speed. The node then stays static for the defined pause time, after which it selects a new destination at random. The used parameters are listed in Table 1. The 100 node model was used only for the network loading simulations. The area, in which the nodes are able to move, was chosen to be a rectangle instead of a square to force the use of longer routes between the nodes.

In addition to the parameters presented in Table 1 the number of sending nodes was varied between 10, 20, 30 and 40. The sending nodes are also expected to send packets at a constant rate of 4 packets/sec, except for 40 sources, which send packets at a rate of 3 packets/sec. This decision was made to decrease the network congestion to enable a meaningful comparison between the results.

Many of the existing performance studies have been run to so called *long-lived* connections, where the connections after the establishment phase usually last throughout the simulations. To examine whether the *short-lived* small transfers behave in a similar fashion, Wang and Helmy [11] decided to run a series of simulations, where the route discovery process would become the dominant factor. These tests had only a single packet or a short stream of packets to be delivered between the source and destination nodes.

The following sections will take a closer look at the results gained from the simulations and analyse them. Section 3.1 concentrates on the original AODV simulations run with long-lived connections, while section 3.2 presents results gained for small transfers. Section 3.3 studies the effect of transmission range.

3.1 Effect of long-lived connections

This paper studies the effect of long-lived connections by comparing two AODV simulations. Samir et al. [6] studied the original AODV with periodical *hello*-messages, while Broch et al. [12] implemented AODV which uses only link layer feedback to gather knowledge about the neighbourhood connectivity in all situations. This implementation will be called AODV-LL in this document. AODV-LL saves the overhead of the periodic hello-messages, but when using it, the failures will be found on-demand and cannot be detected before the link is first needed, which can result in more data packets being lost.

Next sections will present the performance results and analysis first with varying mobility and number of sources and then with varying offered load.

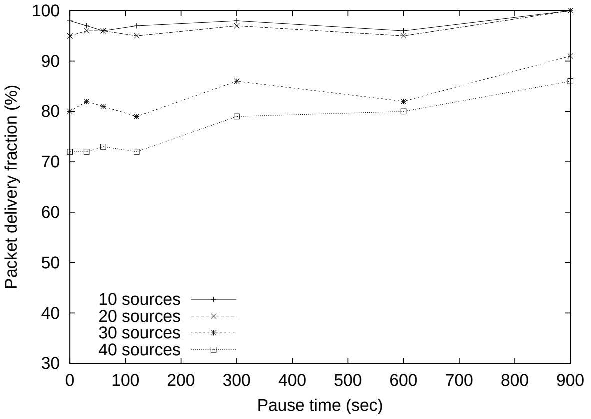

Figure 1: Packet delivery fraction for the original AODV [6].

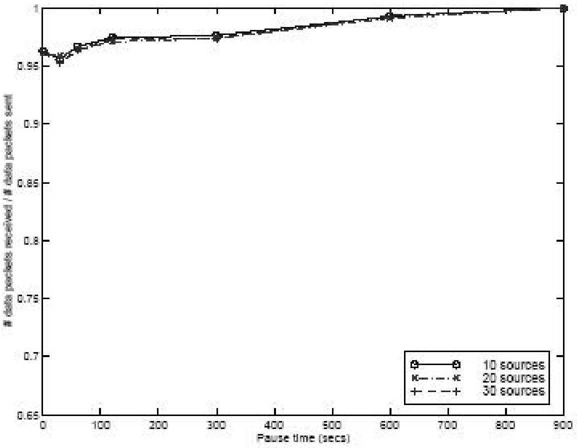

Figure 2: Packet delivery fraction for the AODV-LL [12].

3.1.1 Varying mobility and number of sources

A. Packet delivery fraction

Figure 1 presents the packet delivery fraction for the original AODV. From there it can be seen that the number of packets dropped along the path is quite similar for 10 and 20 sending nodes staying above 95% for all pause times but decreases 20-30% for 30 and 40 sending nodes being only approximately 80% and 72% at worst. As can be expected, the ratio is worse for a continuously changing network (i.e. small pause times) than for the static path conditions, because the number of link failures grows along with the mobility. However, it is interesting to notice that even with static topology conditions, 30 and 40 sending nodes do not achieve 100% packet delivery but only 85-90%. This clearly shows the impact of the network congestion and packet interference as the load on the network increases.

These results are partly similar to the results gained for AODV-LL, which are presented in Figure 2. It shows similar behaviour with 10 and 20 sending nodes staying over 95% for all pause times, but has no decrease in performance with 30 sending nodes like Figure 1 has. This could indicate that as the network congestion increases, the use of link layer feedback instead of periodical hello-messages could provide a better performance as it saves in control overhead.

B. End-To-End Delay

Next will be considered the measured delays for the original AODV. Figure 3 shows that for 10 and 20 nodes the delay is around 0.2 seconds with higher mobility rates and decreases to almost 0 seconds with static topology conditions. For 30 nodes the delay gets more than twice as large being almost 0.5 sec for high mobility and surprisingly increasing to over 0.8 seconds when the mobility is decreased. The same pattern can be seen for 40 nodes with delay about 0.65 seconds with high mobility and increasing to almost 1.3 seconds with longer pause times. This surprising increase with low mobility is due to the fact that AODV does not support any kind of load balancing, which would distribute the traffic more evenly within the network. With relatively large number of sending and receiving nodes (60 and 80% in this simulation) under static conditions the lack of balancing accumulates data packets to use the same routes, which causes high level of network congestion and lower packet delivery fraction. With higher mobility the routes get automatically more evenly distributed and the effect of the congestion decreases. On the other hand, higher mobility increases delay, as more link failures happen and route discoveries have to be reinitiated and completed before the data delivery can continue.

No simulations were run to measure the delay for the AODV-LL.

C. Control message overhead

Figure 4 presents the control message overhead measured for the original AODV, where the number of control packets is normalized against sent data packets. A fairly stable normalized control message overhead would be a desirable property when considering the performance as it would indicate that the actual control overhead increases linearly with the number of sending nodes. Unfortunately AODV seems to have a control message overhead more like exponential than linear. This high increase occurs because each of the route discoveries in AODV typically propagate to every node in the network.

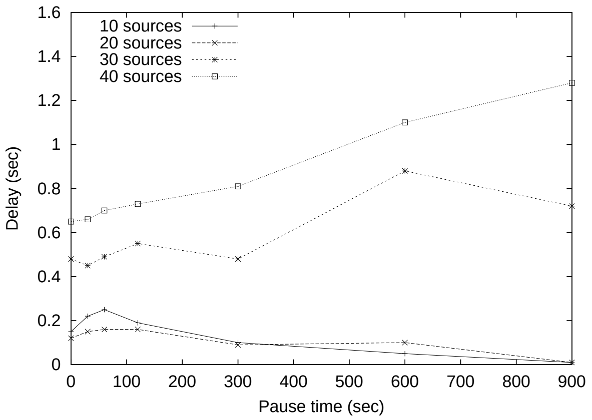

Figure 3: Average data packet delays for the original AODV [6].

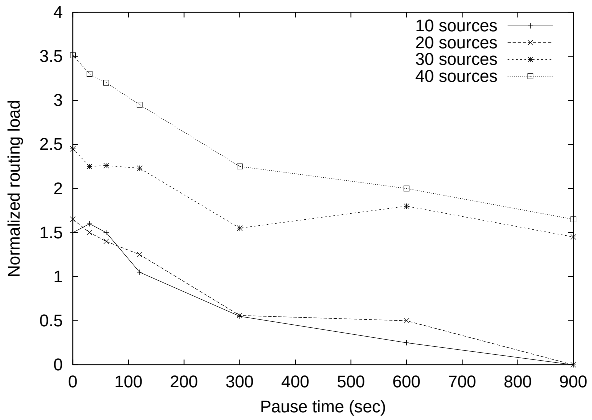

Figure 4: Normalized routing loads for the original AODV [6].

From Figure 4 it can be seen that as mobility increases,so does the number of needed routing packets. The results are quite similar for 10 and 20 sending nodes, which both send 1.5 times more routing packets than actual data packets when trying to keep up with the continuously changing topology. As mobility and thus the number of link failures decreases, the number of sent data packets pretty fast exceeds the number of sent routing packets.

The scheme for 30 and 40 nodes shows much worse results with the normalized routing loads of 2.5 and 3.5 with high mobility and even with static conditions 1.5 times more routing packets than data packets are delivered.

The comparison simulations also clearly show that AODV produces a huge amount of control overhead compared to DSR, which manages the topology changes with only one third of the control packets AODV needs. Interesting was also the high amount of RREQs as about 90% of the all control messages was found to be caused by RREQ forwards.

Figure 5 shows the control message overhead measured for AODV-LL. The results are presented in number of sent routing packets, which makes the comparison with the previous results harder as the total number of data packets delivered remains unknown. Still, the same exponential growth along with the increased mobility can be seen. The only difference between the shapes of the figures is with 30 sending nodes as AODL-LL implementation approaches zero with higher pause times just like 10 and 20 sending nodes, while the original implementation does not. The amount of control overhead also increases as the number of source nodes increases, which is quite understandable as there are more nodes establishing and maintaining routes.

3.1.2 Varying offered load

When simulating the effect of loading the network Samir et al. wanted to make the situation more challenging and chose to use highest mobility, i.e. continuously moving nodes, and a network with 100 nodes. The number of sending nodes was chosen to be either 10 or 40. Within these experiments the offered load, which is the combined data sending rate of all the source nodes, was slowly increased until the throughput saturation was achieved.

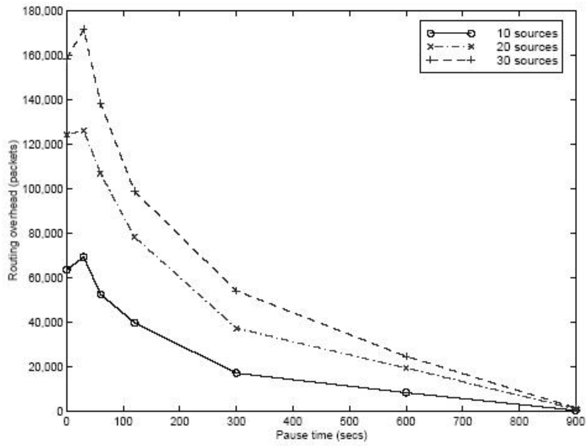

Figure 5: Routing overhead for the AODV-LL [12].

A. Packet delivery fraction

The packet delivery fraction was not measured during the loading simulations as it was replaced throughput measurements. Both of these, however, reflect the number of packets delivered to the destination and measure the network capacity.

As can be seen from Figure 6, with 10 sending nodes AODV responds quite well to the increasing data rate as the throughput increases quite linearly with the offered load until the saturation point around 700 Kbits/sec is reached. This indicates that the packet delivery fraction stays high until the saturation point. With 40 sending nodes, however, the saturation starts much earlier, already around 300 Kbit/sec, which can be seen in Figure 7. After the saturation point the throughput stays at the same level even though the offered load is increased. This indicates a dramatic decrease in packet delivery fraction after the saturation point.

B. End-To-End Delay

The average delays for both schemes are presented in Figure 8. For 10 nodes scheme, the delay slightly increases along with the increasing load. With 40 nodes, however, the delay is 1.5 times higher even with very light loads and grows

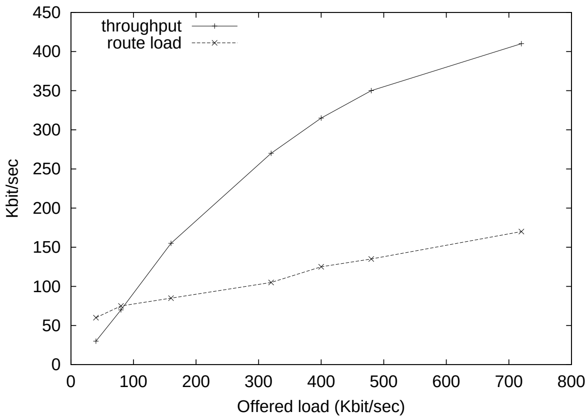

Figure 6: Throughput and routing load for 10 sources [6].

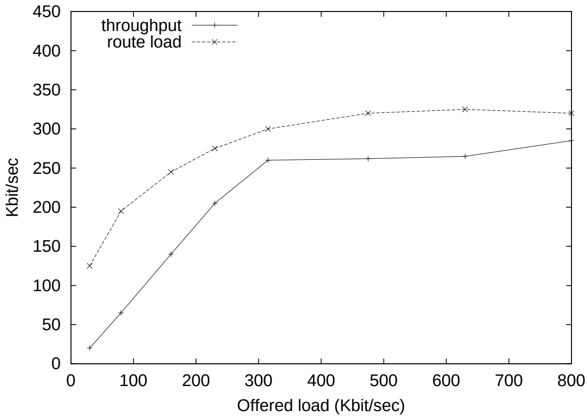

Figure 7: Throughput and routing load for 40 sources [6].

even further as offered load is increased. A considerable increase in delay is experienced after the saturation point is reached.

C. Control Message Overhead

When considering the control message overhead for both of these schemes, the tendency seems to be that the bigger the offered load is, the larger is the number of the control messages sent, as was to be expected. The interesting observation is that for 40 sending nodes, the control overhead is actually higher than the throughput. The situation is, however, understandable as the number of sending nodes is four times bigger, which produces about four times as much more control overhead when using an on-demand routing protocol.

3.2 Effect of Small Transfers

The previous studies have evaluated the performance of AODV with long-lived connections. Wang et al. run the simulations with small transfers consisting of a single packet or a short stream of packets only. Unlike previous simulations, they considered also the cache hit ratio as a part of the performance analysis. Cache hit ratio means the number of queries that can be satisfied with the cache's information to all replied queries. In these simulations there was no predefined number of sending or receiving nodes and the simulations were done by varying the total number of packets transferred between the random peers. They studied the performance using four different experiment scenarios, which will all be covered and analysed next.

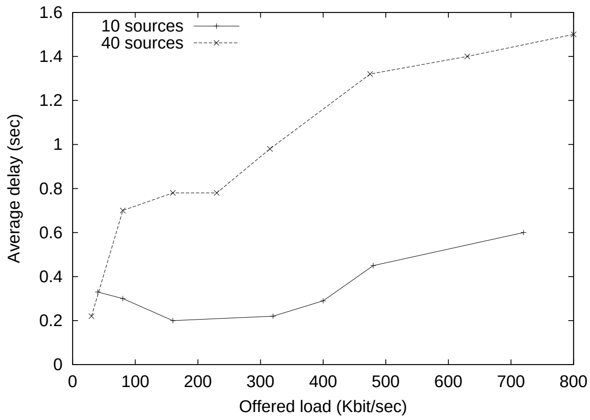

Figure 8: Delay for 10 and 40 sources [6].

The first scenario varied the load on the network. As opposed to the situation with the long-lived connections presented in Figures 6 and 7, the packet delivery fraction was found to drop already at the beginning of the simulation. With very low loads the delivery ratio was around 80%, but soon experienced a large decrease to 50%, where it stayed for the rest of the loads. The same pattern was seen for the delay as it decreased from 1 second to half very quickly with the increasing load. With short-lived transfers the amount of overhead generated was found to be large as AODV broadcasts a new RREQ for every new destination, which on the other hand increases the previously described latency. The increase of the offered load also decreased the validity of the received RREPs and caches, which on its behalf lowered the experienced packet delivery fraction.

The second scenario grouped packets as 2, 4, or 10 packets per group, which then were sent back-to-back with 0.1 seconds interval. AODV was found to clearly benefit from this scenario. Once the first packets got a valid route, the rest of the group was able to use the same routes, which made the successful delivery almost guaranteed.

The third scenario was quite similar to the second one as it also used grouped packets. In addition to this it varied the sending interval between random, 1, 2 or 0.1 seconds. Again AODV benefited as could be expected based on the previous results.

The fourth scenario studied the effect of spatial correlation between the packets by limiting the number of either source or destination nodes or both to 10. As source nodes using AODV maintain per-destination cache entries, it is understandable that spatial correlation improves the performance of the AODV protocol with higher loads.

As a conclusion, the simulations showed that the performance is strongly correlated with cache efficiency, which tends to decrease when the number of nodes increases. This on the other hand affects the scalability properties, which will be studied in Section 4.

3.3 Effect of Transmission Range

The available communication devices on the market offer a wide range of power levels, which affect the transmission power and thus connectivity. It it often thought that the larger the transmission range is, the better it is for data delivery as connectivity increases along the transmission range. It has, however, its drawbacks. With larger transmission ranges more nodes are within the transmission range of each other, which limits the effective bandwidth of all neighbours. With higher node density, the messages begin to suffer from channel access delays and increased number of collisions. This does not, however, necessarily mean lower throughput when considering unicasting as effective channel access schemes can be used. The experienced delay may even decrease as the number of hops between the source and destination nodes is smaller.

The simulations run for varying transmission ranges proved the claims. Bigger transmission ranges showed better packet delivery fractions as routes became shorter in terms of hops, which decreased the probability of a link failure. Improved accessibility also reduced the amount of control messages needed. As a drawback, the number of packet collisions was found to rapidly increase along with the number of neighbours. Moreover, the needless spreading of messages wastes battery power as all the nodes within the transmission range must handle the transmitted packets. A reduction on effective bandwidth was also measured.

4 Scalability

The number of wireless devices is experiencing an extensive growth and new multi-user applications are developed as a respond to this increase. This on the other hand creates a need for the formation of larger ad hoc networks with more participating nodes than before.

Traditionally the simulations run to the AODV have had only 100 nodes in maximum and the scalability of the AODV has remained unknown. While the extensive use of flooding does not deteriorate the performance in smaller networks, it may have a significant impact on larger networks. In large, highly mobile networks link failures become more common and create a situation where source nodes are bombed with the link failure messages and have barely time to initiate a route repair before the route suffers from another failure.

These were the reasons Lee et al. [2] decided to run some simulations with as much as 10,000 nodes. During the simulations the number of neighbours was decided to be kept approximately constant and the scalability was thus studied in terms of increasing the network area instead. Increased density would mean more congestive failures which are not closely related to routing protocol performance.

The scalability simultations used a *random waypoint* model just like the previous simulations. The nodes had a maximum speed of 10 m/s and a constant pause time of 30 seconds. Simulation time was chosen to be 300 seconds and packet sending rate 4 packets/sec.

Lee et al. also suggested some methods to improve the scalability properties of the original AODV. These suggestions are covered shortly next in section 4.1. After these the performance results and analysis are presented in section 4.2.

4.1 Improvements

The main purpose of the improvements suggested is to limit the flooding to a smaller area instead of the whole network. The first two methods aimed to accomplish this are called *Expanding Ring Search* and *Query Localization*, which both try to limit the flooding initiated by a source node. The third one prevents flooding with the event of a link failure and is called *Local Repair*. The principles of each are covered next.

A. Expanding Ring Search

The idea behind the expanding ring search is to limit the route discovery process to a certain area defined in advance and check whether the destination node or any node containing the route to the destination are located within that search area. If this is not the situation then the packet is flooded throughout the network like in the original AODV.

The flooding is limited by using the Time-To-Live value (TTL) in the RREQ packets. Every rebroadcast decrements the TTL value initiated by the source node and when it reaches zero, RREQ is no longer forwarded. When a source node wants to establish a route to the destination for the first time, it has no knowledge of how far away the destination node could be so it starts with a small TTL value, i.e. two hops. After broadcasting the RREQ packet the source node waits enough time to be able to receive RREP if such is sent. If it receives no reply it increments TTL value and broadcasts the RREQ again. The process continues until a route to the destination is found or a threshold value for TTL is reached, after which the RREQ is broadcasted to all the nodes in the network.

During the first route discovery process the source node gets the hop count to the destination. If later the route encounters a failure and the source node wants to reinitiate the route discovery process it can place the hop count on the TTL field and limit the flooding to this area. If the route is not found on a first attempt, TTL is again incremented until a threshold is reached, after which the RREQ flooding through entire network takes place.

The obvious drawback with this algorithm is the latency in finding the rout if the localized attempts are not successful.

B. Query Localization

Like Expanding Ring Search, Query Localization takes advantage of the previously known route to the destination when limiting the flooding of RREQ packets. It uses integer k to restrict the number of new nodes on the route needed. Query localization is really only used with route repairs, because when the route discovery is initiated for the first time, k is set to be the diameter of the network and the RREQ packets will be thus flooded through the network.

Whenever a route repair is needed, k is set to some small value, i.e. two, and RREQ including a counter and k is broadcasted. When a node that was not on the initial route receives the RREQ, it increments a counter before broadcasting the packet and checks that the counter does not exceed k, in which situation the RREQ would be discarded. The

Combination	Abbreviation
Original AODV	AODV
AODV and Expanding Ring Search	AODV-ERS
AODV and Query Localization	AODV-QL
AODV and Local Repair	AODV-LR
AODV, Expanding Ring Search and Local Repair	AODV-ERS-LR
AODV, Query Localization and Local Repair	AODV-QL-LR

Table 2: Used combinations and abbreviations [2].

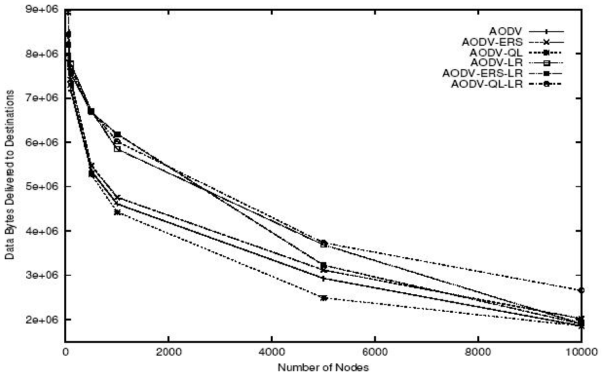

Figure 9: Throughput [2].

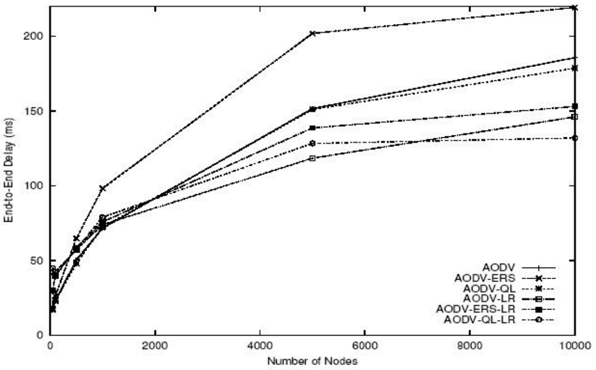

Figure 10: Delay for the varying number of nodes [2].

counter is reset to zero whenever a node that used to be on the initial route receives the RREQ. If no RREP is received, the source node broadcasts a new RREQ with incremented *k* until *k* reaches a threshold value and the RREQ is flooded across the network.

The drawback with this algorithm is also the latency in refinding routes.

C. Local Repair

Local repair is used with link failures to try to repair the point in error without notifying the source nodes and causing an interruption in data delivery. The upstream neighbour, which notices the failure, will be the node that initiates the local repair. It broadcasts a RREQ with a TTL field set to last known distance to the destination incremented by some small value. The sequence number for the destination is also incremented to prevent the formation of loops. If the RREQ reaches the destination, a substitute route is found and data delivery can continue. However, if a new route is not found on a first attempt, the source nodes using the point in failure are notified with RERR message.

Local repair has clear benefits when it is successful. First, it may result in fewer data packets being lost as the substitute route is found quicker. Secondly, the source node hasn't have to be disturbed. In large, highly mobile networks which have long routes link failures will happen so often that it would be impossible for the source node to keep up with the initiated repairs. However, if the route repair is unsuccessful, more data packets will be lost during the repair attempt and the source node has to be interrupted anyway. It has neither a way to know whether the destination has moved closer to the source node and can create longer routes than necessary.

4.2 Results and analysis

The tests were run to varying number of nodes and six different improvement combinations, which are listed in Table 2 along with their abbreviations.

A. Throughput

Figure 9 shows the number of successful data deliveries for the combinations presented in Table 2. It clearly shows that as the number of nodes in the network grows from 50 towards 10000, the overall throughput decreases dramatically. As the packet sending rate stays constant at the same time this indicates a reduced ability to deliver packets to their destinations. With larger networks the number of hops along the route from the source to the destination grows, which means that the routes become more prone to disconnections as there are more possible error points. This has a direct influence on throughput. The original AODV faces a reduction to a fourth from the original throughput. Not even the improvements made to the original AODV can produce considerably better results in sustaining the throughput level as the number of nodes increases. However, local repair seems to help with networks under 5000 nodes as the reduction is 25-30% smaller than with implementations without it. This is because the routes are repaired quicker and which results in less data packets being lost. Combined with query localization, it can produce two times better results than other combinations when the node number reaches 10000. Without local repair, however, query localization has the poorest throughput with all number of nodes.

Additional interesting notification made during the scalability simulations was that even though using local repair yields in best throughput, it tends to establish 50-300% longer routes than other combinations. This causes more route breaks and rediscoveries, but still creates the best throughput.

B. End-To-End Delay

The measured end-to-end delays are presented in Figure 10. It shows that again implementations using local re-

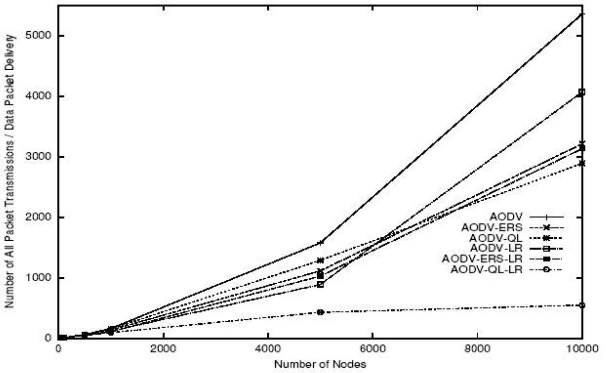

Figure 11: Normalized routing load for the varying number of nodes [2].

pair have gained best results, this time shorter delays. The local route rediscovery needs less time in searching and setting up a new route than protocols that use source initiated route rediscoveries. This reflects directly into the end-to-end latency as data packets have to be buffered during the route discovery phases, duration of which is added to the end-to-end latency.

Figure 10 shows that from the source initiated route repair schemes AODV-ERS faces the longest delays. This is because the new route may not be built on a first attempt and additional attempts take more time. This affects also AODV-ERS-LR as it has the longest delays among local repair protocols.

C. Control overhead

The normalized control message overhead which is used to evaluate protocol efficiency, is shown in figure 11.

When the considered network has only a small number of nodes, i.e. 50 or 100 nodes, the combinations have no huge differences in the amount of produced control messages compared to data packets. After 100 nodes, however, the differences become clear. The original AODV stands out from others as being the implementation with far most control message overhead, which grows nearly linearly with the number of nodes. AODV-QL-LR on the other hand can be distinguished as having clearly the smallest control message overhead, only about 10% of the amount AODV produces. More importantly, it stays nearly constant between 5000 and 10000 nodes, which is important when considering scalability. The remaining combinations are placed between these two having five to six times the control messages overhead AODV-QL-LR has and linear like growth.

Figure 11 clearly indicates a huge need for work in areas concerning the scalability of AODV. According to Lee et al., the normalized routing loads should be brought down as much three orders of magnitude to better the performance of the AODV in large networks and the future work will concentrate on trying to develop techniques for quicker route rediscovery.

5 Additional Observations

The previous sections have studied the performance and scalability of the AODV protocol under different conditions. The simulations have brought up many important characteritics affecting the performance, most of which have already been analysed during the presentation of the results. However, there are still a few points worth consideration, which will be discussed next.

During the simulations, Samir et al. [6] found that the bigger number of hops along the routing path does not necessarily mean the longer end-to-end delays. As the nodes only reply to the first arriving RREQ it automatically favours the fastest route instead of the shortest route in terms of hops. Additionally, the longer delay often indicates a worse congestion along the route, so this kind of selection prefers the least congested route and balances the traffic in a very simple way, even though AODV does not officially support load balancing.

It was also noticed that AODV produces a lot of control overhead compared to DSR. The reason behind this is AODV’s extensive use of packet flooding, which is based on its ability to gather only a very limited amount of information during route discovery processes. Nodes in the network can only learn about a source node, when forwarding the RREQ packets and about a destination node when forwarding the RREP. All the routes to the intermediate nodes along the route remain unknown. The destination node is also allowed to answer only to the first arriving request and must ignore the rest, which forces the nodes on alternative routes to initiate a route discovery of their own if the route becomes needed. Likewise, the source has only one entry per destination in routing table, so no alternative routes are available upon link failure. Not only does every route discovery initiated produce more overhead, but also consume significant processing power at each node of the network as well as bandwidth.

Almost all the simulations defined the control overhead in terms of packets. However, if the control overhead is measured in bytes and set to include the source route headers, the control overhead of the AODV protocol begins to look much lighter compared to other protocols. AODV-LL [12] even becomes the protocol with least control overhead. Unfortunately, in reality this does not help because the costs to acquire the medium for packet transmission are much higher than the costs to add a few extra bytes to an existing packet.

Simulations was also found that the network capacity is poorly utilized when combining the 802.11 MAC protocol and an on-demand routing protocol like AODV. Samir et al. counted that the delivered throughput was at most as low as about 2-3% of the network capacity. No better combinations were nevertheless proposed.

6 Conclusion

The improved technology and the development of more powerfull communication devices has increased the interest in ad hoc networks to higher levels than ever before. It has become a widely studied subject among the computer scientists, which has resulted in many different performance studies. This paper has evaluated the performance and scalability results for the AODV routing protocol obtained by different research parties.

In small networks with only 50 participating nodes, AODV was found to perform efficiently and accurately. The packet delivery fraction was measured to be relatively high even in stressful, i.e. heavy load or high mobility, situations. However, after the saturation point was reached, the efficiency was rapidly reduced. The delays were found to be acceptable, but the control overhead was far larger than expected based on the throughput results. The reason behind this was found to be the extencive use of packet flooding with route discoveries as nearly 90% of the control messages were RREQ packets.

The impact of the network congestion was also clearly noticed in the results as a deteriorating factor. The use of congestion related metrics with route selection and removing the aged packets from the network could provide some help with the situation.

With networks as much as 10000 nodes different performance figures were observed. The simulations used three kinds of improvements called expanding ring search, query localization and local repair in addition to the original AODV implementation. In general the performance was found to significantly decrease as the number of nodes in the network grew, which indicates poor scalability. In more detail, the throughput was reduced to fourth from the original, delays experienced multiple increases and the control overhead grew exponentially.

The use of local repair was found to ease the situation a little in all aspects as in large and highly mobile networks the amount of link failures grows over the limits that source nodes can handle. Expanding ring search and query localization on the other hand reduced the amount of control overhead as they limited the number of nodes affected during route discoveries.

The interplay between the routing protocol and MAC layers was also found to have a remarkable part in performance. The poor capacity utilization with AODV and 802.11 MAC gives another reason for further studies.

References

[1] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[2] Sung-Ju Lee, Elizabeth M. Royer, and Charles E. Perkins Scalability Study of the Ad Hoc On-Demand Distance Vector Routing Protocol In *ACM/Wiley International Journal of Network Management*, vol. 13, no. 2, March 2003, pp. 97-114.

[3] Charles E. Perkins Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, IETF Network Working Group, July 1998.

[4] Elizabeth M. Royer and Charles E. Perkins. Transmission Range Effects on AODV Multicast Communication. In *ACM Mobile Networks and Applications* special issue on Multipoint Communication in Wireless Mobile Networks, 2000

[5] Elizabeth M. Royer and Charles E. Perkins. An Implementation Study of the AODV Routing Protocol. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, Chicago, IL, September 2000.

[6] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000, p. 3-12

[7] Manel Guerrero Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing". IETF Internet Draft, draft-guerrero-manet-saodv-00.txt, August 2001 (Work in Progress).

[8] Charles E. Perkins, Elizabeth M. Belding-Royer, and Ian Chakeres. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF Internet draft, draft-perkins-manet-aodvbis-00.txt, Oct 2003 (Work in Progress)

[9] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Quality of Service in Ad hoc On-Demand Distance Vector Routing. IETF Internet Draft, draft-ietf-manet-qos-00.txt, July 2000 (Work in Progress).

[10] Manel Querrero Zapata. Secure Ad hoc On-Demand Distance Vector Routing Mobile Computing and Communications Rewiev, Volume6, Number 3, 2002

[11] Shao-Cheng Wang and Ahmed Helmy (University of Southern California, US) Effects of Small Transfers and Traffic Patterns on Performance and Cache Efficacy of Ad Hoc Routing MobiCom 2003

[12] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols In *Proceedings of Mobicom '98*