Proceedings of the 10th WSEAS International Conference on COMMUNICATIONS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp266-271)

Performance Evaluation of AODV & DSR for Wireless Sensor Networks

RIZWAN AHMED KHAN, SHOAB A KHAN
Department of Computer Engineering
National University of Sciences & Technology
College of E&ME, Peshawar Road, Rawalpindi
PAKISTAN

Abstract: - This paper evaluates the performance of two prominent on-demand routing protocols, which are Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector Routing (AODV) for wireless sensor networks. Results obtained from simulations demonstrates that even though DSR and AODV share a similar on-demand behavior, the differences in the protocol mechanics can lead to significant performance differentials. The performance differentials are analyzed using varying network load and network size. Tiny Operating System (TinyOS) is used as a platform for simulating AODV & DSR under various conditions. TinyOS is free software and is designed specially for wireless sensor networks. Wireless Sensor Networks (WSNs) are characterized by multi-hop wireless connectivity, frequently changing network topology and the need for efficient dynamic routing protocols.

Key-Words: - Wireless Sensor Networks, Ad Hoc Routing Protocols, AODV, DSR, Tiny OS, multi-hop wireless links

1 Introduction

The emerging field of wireless sensor networks integrates sensing, computation and communication into a single tiny device. Through advanced mesh networking protocols, these devices form a sea of connectivity that extends the reach of cyberspace out into the physical world [1]. While the capabilities of any single device are minimal, the composition of hundreds of devices offers radical new technological possibilities.

Sensor networks of the future are envisioned as thousands or more of inexpensive wireless nodes. Operating unattended [2], each of these sensors will be equipped with some computational power and sensing ability (e.g., sonar, radar, seismic, etc.).

The power of wireless sensor networks lies in the ability to deploy large numbers of tiny nodes that assemble and configure themselves. Usage scenarios for these devices range from real-time tracking, to monitoring of environmental conditions, to Military applications, to monitoring of the health of structures or equipment.

Generally, when people consider wireless devices they think of items such as cell phones, personal digital assistants, or laptops with 802.11. These items costs hundreds of dollars, target specialized applications, and rely on the pre-deployment of extensive infrastructure support. In contrast, wireless sensor networks use small, low-cost embedded devices for a wide range of applications and do not rely on any pre-existing infrastructure.

Sensor networks are very different from conventional computer networks [3]. First, because sensors have a limited supply of energy, energy-conserving forms of communication and computation are essential to wireless sensor networks. Second, since sensors have limited computing power, they may not be able to run sophisticated network protocols. Third, since the bandwidth of wireless links connecting sensor nodes is often limited, inter-sensor communication is further constrained.

In WSNs, mobile nodes communicate with each other using multi-hop wireless links [4]. There is no stationary infrastructure; for instance, there are no base stations. Each node in the network also acts as a router, forwarding data packets for other nodes. A central challenge in the design of WSN is the development of dynamic routing protocols that can efficiently find routes between two communicating nodes.

The goal of this paper is to carry out a systematic performance study of two dynamic routing protocols for WSN, the Dynamic Source Routing protocol (DSR)[5] and the Ad Hoc On-Demand Distance Vector protocol (AODV)[6] for wireless sensor networks using Tiny Operating System. The motivation is that a better understanding of the relative merits will serve as a cornerstone for development of more effective routing protocols for WSNs.

TinyOS [7] is probably the earliest operating system that directly targets the specific applications and limitations of sensor devices. TinyOS uses a special description language for composing a system of smaller components [8] which are statically linked with the kernel to a complete image of the system.

DSR and AODV both have shown some limitations [9] but they also share an interesting common characteristic, they both initiate routing activities on an "on demand" basis. This reactive nature of these protocols is a significant departure from more traditional proactive protocols that find routes between all source-destination pairs regardless of the use or need of such routes. The key motivation behind the design of on-demand protocols is the reduction of the routing load. High routing load usually has a significant performance impact in low bandwidth wireless links.

Including this section, this paper has five sections. Section2 highlights sensor's simulation model. Section3 describes how protocols mechanisms are simulated and enlists simulation model parameters. Section4 discusses results which are obtained from various cases. Finally in section5 analysis on the basis of results is given.

2 Simulation

The goal of simulation is to simulate and closely model the sensor network scenario. The broad outline of any sensor network can be represented by high-level representation as in Fig.1. The sensor model can be represented by the sensor node model and the Power model.

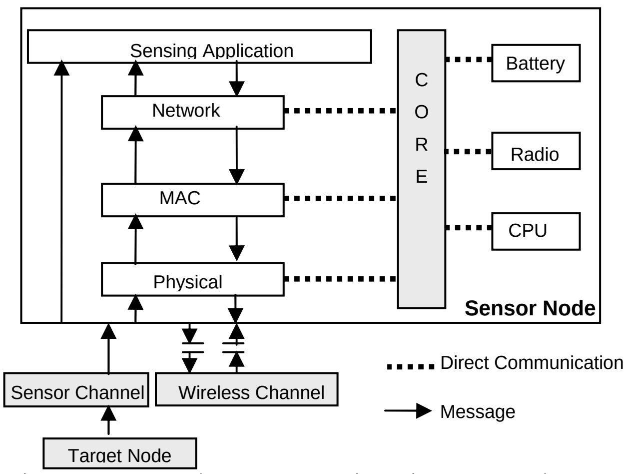

Fig.1 Sensor node representations in a network

3 Performance Evaluation

3.1 Performance Metrics

Numerous simulations were run both with AODV and DSR to compare performance metrics of both versions of the protocol. The performance metrics under consideration are:

- **Mean end-to-end packet latency/delay:** End-to-end packet latency is defined as the time elapsed from the moment a packet is generated by the data agent at the sending node, to the time the packet is received at the corresponding agent at the receiving node.
- **Packet delivery ratio / Success rate ratio:** Packet delivery ratio is the ratio of total number of data packets that were delivered successfully to intended destinations to the total number of data packets generated [10]. Packets may not be delivered to the destination mainly because of one of the following reasons: packet collisions, routing loop and queue drop.

3.2 Simulation Model

PARAMETERS	VALUES
Number of Sources	20,30,......100
Shape of traffic	CBR (Constant Bit Rate)
Nodes movement	Immobile
Messages Broadcast through	Flooding
Simulation area	2000m x 2000m
TxPower (transmission power of the antenna)	0.0280
RXThresh (lower bound on the receive power of any packet that can be successfully received)	3.652e-10
Propagation Model	Free Space

Table 1 Simulation Parameters

The free space propagation model assumes the ideal propagation condition that there is only one clear line-of-sight path between the transmitter and receiver. H. T. Friis presented equation1 to calculate the received signal power in free space at distance from the transmitter

$$Pr = (Pt * Gt * Gr * \lambda 2) / (4\pi)2 * d2 * L2$$

Equation.1 received signal power in free space

Where:

Pt is the transmitted signal power
Pr is the received signal power
G t, G r are the antenna gains of the transmitter and the receiver respectively.
L is the system loss

3.3 Implemented Algorithm Explanation

This section explains how AODV is simulated; DSR algorithm is simulated in the same manner except it has a slight difference in maintaining routing information.
In AODV, each node maintains two separate counters:

1. Sequence Number, a monotonically increasing counter used to maintain freshness information about the reverse route to the source.

2. Broadcast-ID, which is incremented whenever the source issues a new Route Request (RREQ) message.

Each node also maintains information about its reachable neighbors with bi-directional connectivity. Whenever a node (router) receives a request to send a message, it checks its routing table to see if a route exists. Each routing table entry consists of the following fields:

a) Destination address
b) Next hop address
c) Destination sequence number
d) Hop count

3.3.1 AODV Route Discovery Algorithm

When a node needs to determine a route to a destination node, it floods the network with a Route Request (RREQ) message as shown in Fig.2. If a route exists, the originating node sends data packet to destination. Otherwise, it saves the message in a message queue, and then it initiates a route request to determine a route. When RREQ message reaches to the destination (destination node) it replies with RREP (Route Reply) message, so that path can be determined/established by source node and communication can take place.

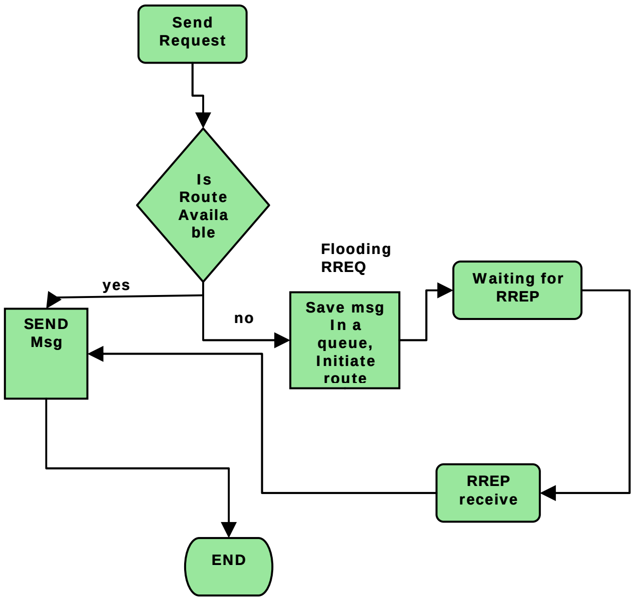

Fig.2 Propagation of RREQ

As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. Since an intermediate node could have many reverse routes, it always picks the route with the smallest hop count.

3.3.2 Link Monitoring and Route Maintenance

Each node periodically sends HELLO messages to its precursors. A node decides to send a HELLO message to a given precursor only if no message has been sent to that precursor recently.

Correspondingly, each node expects to periodically receive messages (not limited to HELLO messages) from each of its outgoing nodes. If a node has received no messages from some outgoing nodes for an extended period of time, then that node is presumed to be no longer reachable.

Whenever a node determines one of its next hops to be unreachable, it removes all affected route entries and generates a Route Error (RERR) message. This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables and in turn forward the RERR to their precursors, and so on, to prevent RERR message loops.

The Fig.3 displays a flowchart which summarizes the action of an AODV node when processing an incoming message.

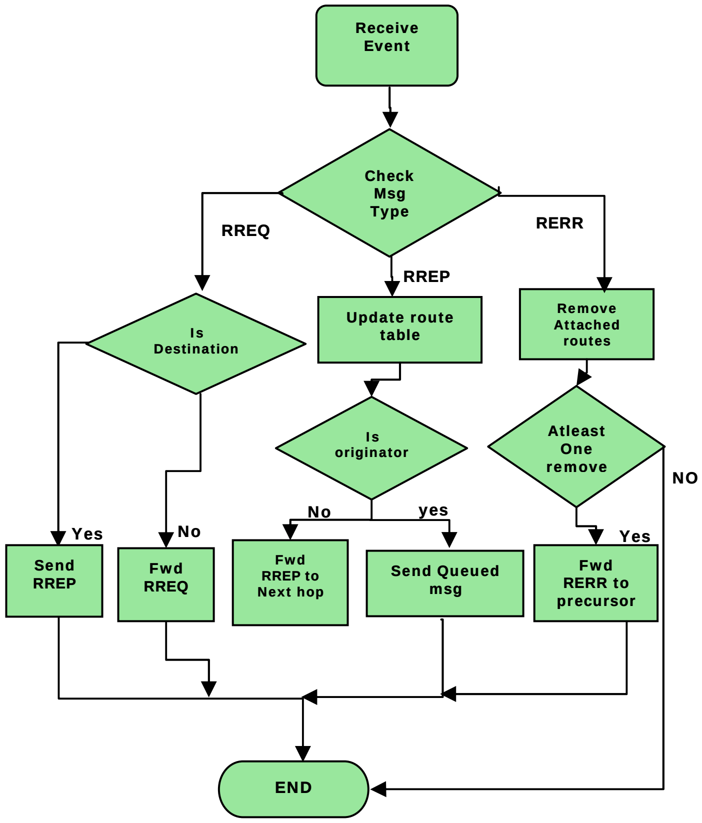

Fig.3 Flowchart for an AODV node when processing an incoming message

3.3.3 Packet Storage

The flow chart in Fig.4 explains how packets are serviced inside a node, before node does any thing it has to store a packet in a queue. The nodes in simulation stores packets in FIFO manner.

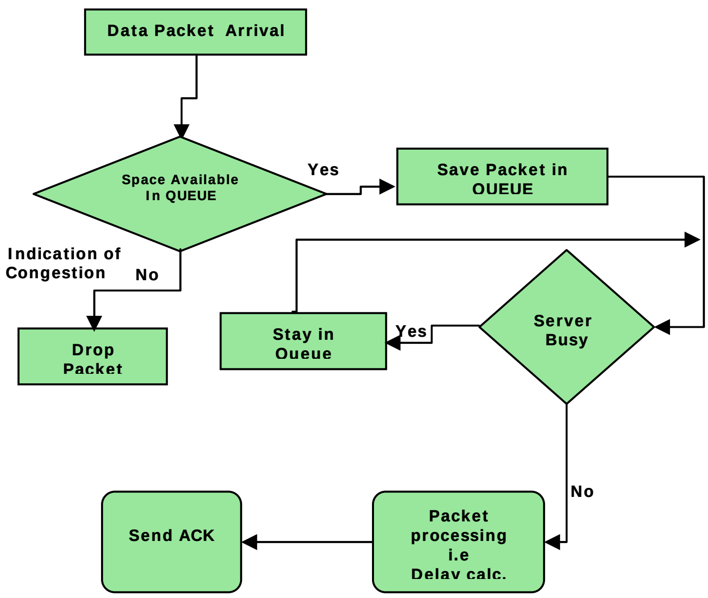

Fig.4 Flowchart of AODV node when storing Packet

4 Results & Conclusion

4.1 Comparison of Delay

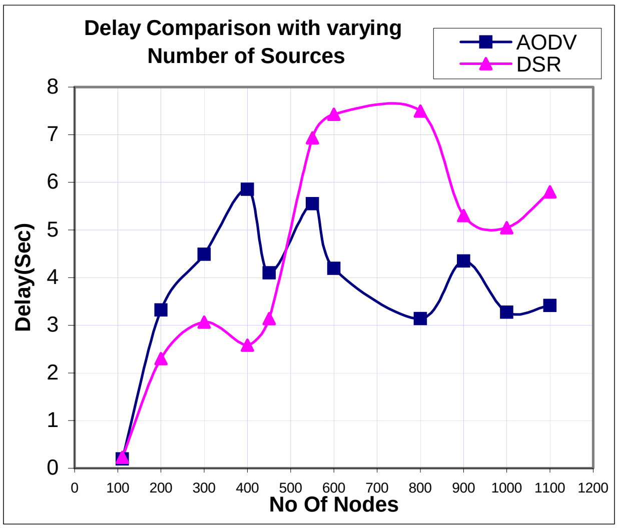

Fig.5 Comparison of Delay with Number of Sources = 10% of nodes

Simulations were run for varying number of sources with constant packet size and immobile nodes and the result is plotted in Fig.5. The result shows that the delay of DSR is slightly less than AODV for less number of sources but as the number of sources increases, delay also increases but the delay of AODV more or less remain the same.

According to above result, it can be said that AODV outperforms DSR for more number of sources or for more network traffic and DSR performs better for less number of nodes.

4.2 Delay Comparison with Varying Size of Packets

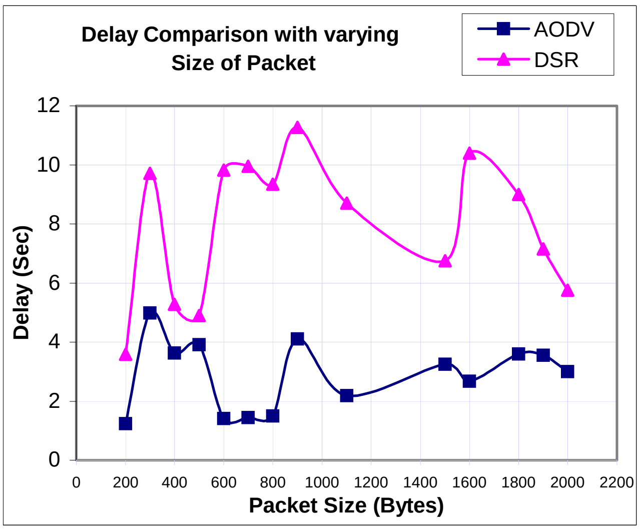

Fig.6 Delay Comparison with Varying Size of Packets

Simulations were run for varying size of packet with constant number of sources in order to compare delay whose result is plotted in Fig.6.

The result validates our first conclusion that AODV is a better choice for more number of sources or for more data. It also proves that delay of DSR is always greater than AODV. The delay curve of DSR doesn't have any continuous pattern as compared to AODV, who has some pattern in delay curve. But in all cases AODV's delay is less than DSR for varying packet size.

4.3 Success Rate/Throughput Comparison

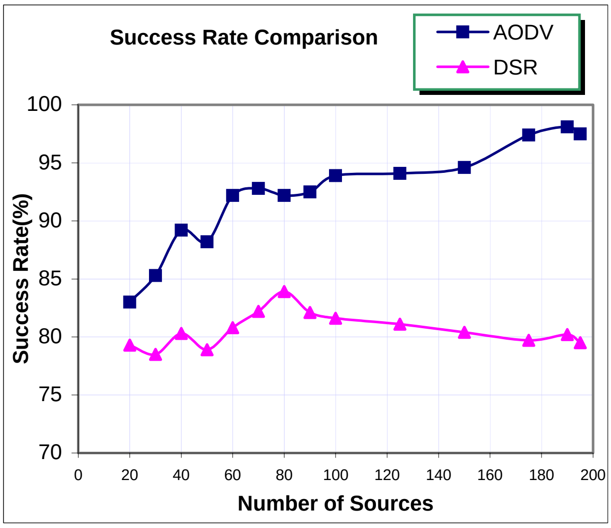

Fig.7 Success Rate Comparison

Third simulation was run to compare success rate of AODV and DSR protocols. Result proves that AODV exhibits higher throughput or success rate ratio than DSR.

Simulation was run for varying number of sources and constant packet size whose result is plotted in Fig.7. DSR success ratio is in the region of 80% and for AODV the success ratio is in the region of 85% - 90% for all cases.

5 Analysis

- **Packet Delivery Performance:** DSR loses about twice as many packets as AODV for higher traffic scenarios and DSR performance gets even worse with higher traffic.

- **Delay Performance**: For less number of sources DSR has lower delay than AODV. However DSR's delay performance worsens with large number of sources and gives about twice as much delay than AODV.

- **Throughput:** AODV has higher throughput than DSR.

- **Performance:** On the whole DSR has poor performance because of packet loss at high data rates. AODV outperforms DSR except under low load (i.e when number of sources is low).

DSR and AODV both use on-demand route discovery, but with different routing mechanics. In particular, DSR uses source routing and route caches, and does not depend on any periodic or timer-based activities. DSR exploits caching aggressively and maintains multiple routes per destination. AODV, on the other hand, uses routing tables, one route per destination, and destination sequence numbers, a mechanism to prevent loops and to determine freshness of routes. The general observation from the simulations is that for application-oriented metrics such as packet delivery fraction and delay AODV, outperforms DSR in more "stressful" situations (i.e., smaller number of nodes), with widening performance gaps with increasing stress (e.g., more load, higher mobility). DSR, however, consistently generates less routing load than AODV. The poor performances of DSR are mainly attributed to aggressive use of caching, and lack of any mechanism to expire stale routes or determine the freshness of routes when multiple choices are available. Aggressive caching, however, seems to help DSR at low loads and also keeps its routing load down.

The results discussed in this paper assume unlimited energy pool; therefore, this research work can be extended in simulation model with realistic energy consumption. Secondly, since these results did not incorporate obstacles, therefore, incorporating the same could lead to significant result variations.

References:

[1] Jason Lester Hill, PhD dissertation "System Architecture for Wireless Sensor Networks", http://www.jlhlabs.com/jhill_cs/jhill_thesis.pdf

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci. "Wireless Sensor Networks: A Survey". Computer Networks, 38(4):393-422, 2002.

[3] Praveen Namboori, Department of Computer Science, North Dakota State University, "Energy Efficient Protocols and Schemes for Wireless Sensor Networks", http://www.cs.ndsu.nodak.edu/~namboori/csci659.doc

[4] Ian F. Akyildiz, Xudong Wang, Weilin Wang "Wireless mesh networks: a survey", http://www.ece.gatech.edu/research/labs/bwn/mesh.pdf

[5] J.Broch, D.Johnson, and D. Maltz, The Dynamic Source Protocol for Mobile Ad hoc Networks, http://www.ietf.org/internetdrafts/draft-ieft-manet-dsr-03.txt, IETF Internet draft ,Oct.1999.

[6] Charles E. Perkings, Elizabeth M. Belding-Royer, Samir R.Das, AdHoc On-Demand Distance Vector (AODV) Routing, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-06.txt, IETF Internetdraft, July 2000

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In Proc. ASPLOS-IX, November 2000.

[8] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: A holistic approach to networked embedded systems. In Proc. SIGPLAN'03, 2003.

[9] Charles E. Perkings, Elizabeth M.Royer and Samir R.Das, Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks, IEEE Personal Communications, Feb 2001.

[10] Tarek- Masters Thesis. Modeling and simulation of a routing protocol for AD HOC networks combining queuing network analysis and ANT COLONY algorithm. http://miless.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-12937/Tarek-Thesis.pdf