IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 155-158

# PERFORMANCE REVIEW OF REACTIVE PROTOCOLS USED IN WIRELESS ADHOC NETWORKS

**Gopindra Kumar[1], K.K. Tripathi[2] and Neeraj Tyagi[3]**

[1]RKGIT, Ghaziabad (UP), *gopindrakm@rediffmail.com*

[2]Department of electronics Engineering, RKGIT, Ghaziabad (UP)

[3]Department of Computer Science, MNNIT, Allahabad (UP)

ABSTRACT

Different types of wireless adhoc networks can be designed by the number of nodes participating in the network, the mobility pattern of nodes, the traffic demands of nodes etc.

***Keywords***: DSR (dynamic source routing), AODV (adhoc on demand distance vector), TORA (Temporally-Ordered Routing Algorithm), QRY (query), UPD(update), CLR (clear), and OPT (optimization )

## 1. INTRODUCTION

Generally, there are three types of routing protocols used in adhoc wireless networks, each of them has distinct features and suitable for certain types of network environment. The first one is called table-driven or proactive routing. The second one is called on-demand or reactive routing. One of the distinctive features of this type of routing protocol is that nodes participate in routing only when necessary. The third type is called hybrid, which combines the features from both table-driven and on-demand routing protocols.

## 2. DYNAMIC SOURCE ROUTING (DSR)

The Dynamic Source Routing (DSR) [3] protocol is a distance-vector routing protocol for wireless adhoc networks. When a node generates a packet to a certain destination and it does not have a known route to that destination, this node starts a route discovery procedure. Therefore, DSR is a reactive protocol. One advantage of DSR is that no periodic routing packets are required. DSR also has the capability to handle unidirectional links. Since DSR discovers routes on-demand, it may have poor performance in terms of control overhead in networks with high mobility and heavy traffic loads. Scalability is said to be another disadvantage of DSR [1], because DSR relies on blind broadcasts to discover routes.

There are two main operations in DSR (*i*) route discovery (*ii*) route maintenance.

Figure 1 shows a simple example for DSR. Routers *A*, *B*, and *C* form a MANET. Routers *A* and *C* are disconnected, while both of them connect to router *B*. Assume that at the beginning, the route caches that memorize previous routes in the routers are empty. When Router *A* wants to send a packet to Router *C*, it broadcasts a route request to start the corresponding route discovery procedure. Router *B* receives the request since it is within the radio range of *A*. Router *C* is the destination in the request and *B* does not have a route entry to *C* in its cache at this time. Hence, Router *B* appends its own ID to the list of intermediate router IDs in the request and rebroadcasts it. When *C* receives the broadcast route request message originated by *B*, it determines that the destination ID matches its own ID. Thus, the route from *A* to *C* is found. To help the initiator and all intermediate routers construct proper routing entries, Router *C* sends a reply back to *A* using source routing if links are bi-directional. This procedure is feasible because all intermediate routers are in the ID list of the corresponding route request. Intermediate routers construct proper routing tables when they receive the reply originated from *C*. Thus, a route from *A* to *C* is built.

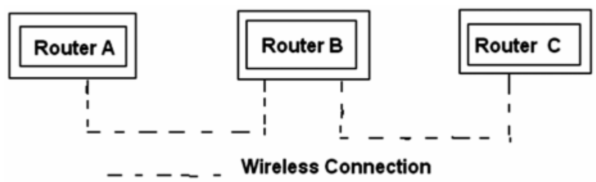


**Figure 1: Example of DSR and ADOV Routing Protocols**

### 2.1 Ad hoc On-Demand Distance Vector Routing (AODV)

The Ad hoc On-demand Distance Vector (AODV) routing protocol [4] is a reactive MANET routing protocol. Similar to DSR, AODV broadcasts a route request to discover a route in a reactive mode. The difference is that in AODV, a field of the number of hops is used in the route record, instead of a list of intermediate router addresses. Each intermediate router sets up a temporary reverse link in the process of a route discovery. This link points to the router that forwarded the request. Hence, the reply message can find its way back to the initiator when a route is discovered. When intermediate routers receive the reply, they can also set up corresponding forward routing entries. To prevent old routing information being used as a reply to the latest request, a destination sequence number is used in the route discovery packet and the route reply packet. A higher sequence number implies a more recent route request. Route maintenance in AODV is similar to that in DSR [2]. One advantage of AODV is that AODV is loop-free due to the destination sequence numbers associated with routes. The algorithm avoids the Bellman-Ford "count to infinity" problem [4]. Therefore, it offers quick convergence when the ad hoc network topology changes which, typically, occurs when a node moves in the network [4]. Similar to DSR, poor scalability is a disadvantage of AODV [1].We use the example topology shown in Figure 1 to illustrate the discovery procedure of AODV. Note that Routers *A* and *C* are disconnected from each other while both of them connect to *B*. When Router *A* starts a route discovery to *C*, a route request is broadcast. The request packet contains the requested destination sequence number, which is 1 greater than the one currently kept at *A*. For example, assume that the destination sequence number for *C* at *A* is 0 × 00000000, then the destination sequence number in the route discovery packet is 0 × 00000001. The intermediate routers reply to the source if they know the route to that destination with the same or higher destination sequence number. It is assumed that *B* does not have a record for a route to *C*. Therefore, *B* first sets up a temporary link pointing back to *A*. In the second step, it increases the number of hops by 1 and rebroadcasts the request. When *C* receives that request, it creates a new destination sequence number. *A* route reply with that new sequence number is sent by *C*. The initiator and all intermediate routers build routing entries associated with this new sequence number when they receive the reply. The number of hop values can be used to find a shorter path if a router receives two replies with the same destination sequence number.

### 2.2 Temporally-Ordered Routing Algorithm (TORA)

The Temporally-Ordered Routing Algorithm (TORA) is "an adaptive route protocol [5]. TORA is a distributed algorithm so that routers only need to maintain knowledge about their neighbors. TORA also maintains states on a predestination basis like other distance-vector algorithms. It uses a mix of reactive and proactive routing. Sources initiate route requests in a reactive mode. At the same time, selected destinations may start proactive operations to build traditional routing tables. Usually, routes to these destinations may be consistently or frequently required, such as routes to gateways or servers. TORA supports multiple path routing. It is said that TORA minimizes the communication overhead associated with adapting to network topology changes [5]. The reason is that TORA keeps multiple paths and it does not need to discover a new route when the network topology changes unless all routes in the local route cache fail. Hence, the trade off is that since multiple paths are used, routes may not always be the shortest ones.

Routers *A* and *C* have higher heights than the other routers. The destination router has the lowest height among all routers. The link between Routers *A* and *B* is a downstream link for *A* and is an upstream link for *B*. Note that Router *C* has a higher height than *B* although *C* is one hop away from the destination. This is because the height assignment algorithm used in TORA does not always favor the shortest path.

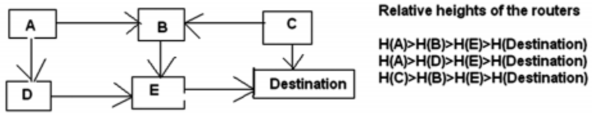


**Figure 2: TORA Routing Protocol Example**

TORA is a complex algorithm compared with DSR. It has four operations: (*i*) creating routes,(*ii*) maintaining routes, (*iii*) erasing routes, and (*iv*) optimizing routes. The creating routes operation is responsible for selecting proper heights for routers and forming a directed sequence of links leading to the destination in a previously undirected network. The maintaining routes procedure is the operation that responds to network topology changes. The operation of erasing routes is used to set routers' heights to NULL and set links to undirected. TORA uses the optimizing routes function to adjust the heights of routers to improve routing. Four packets are used to perform these operations: query (QRY), update (UPD), clear (CLR), and optimization (OPT) [5]. We use the example shown in Figure 1.0 to describe the procedure for creating routes in the reactive mode. Note that Routers *A* and *C* are disconnected from each other but both are connected to B. Assume that Router *A* wants to send a packet to *C*. TORA initiates the procedure by sending a QRY packet. Router *B* rebroadcasts the request since *B* is not the destination and it does not know a route to *C*. Router *B* sets a route-requested flag so that it can discard any further QRY packets received for the same destination before it knows how to get to *C*. When *C* gets the QRY packet, it replies with an UPD packet. This UPD packet identifies the relevant destination and the height of the source of the UPD packet. To reduce redundant replies to a given route request, Router *C* maintains the time at which the UPD packet was last sent and the time at which links to neighboring routers became active. Because of the preset route-requested flag, when Router *B* receives the UPD packet, it updates its local routing table by adding an entry with *C* as the destination. It also sends a new UPD to its known neighbors. The new UPD packet contains the relevant destination and the height of *B*. In this way, all routers in the network choose proper heights according to the height of *C*. All associated links are assigned directions based on those heights.

### 2.3 Optimized Link State Routing Protocol (OLSR)

The Optimized Link State Routing (OLSR) protocol [7] is a proactive link state routing protocol for MANETs. One key idea is to reduce control overhead by reducing the number of broadcasts as compared with pure flooding mechanisms. The basic concept to support this idea in OLSR is the use of multipoint relays (MPRs) [53, 54]. MPRs refer to selected routers that can forward broadcast messages during the flooding process. "The protocol is particularly suitable for large and dense networks" [7]. MPRs act as intermediate routers in route discovery procedures. Hence, the path discovered by OLSR may not be the shortest path. This is a potential disadvantage of OLSR. OLSR has three functions: Packet forwarding, neighbor sensing, and topology discovery. Packet forwarding and neighbor sensing mechanisms provide routers with information about neighbors and offer an optimized way to flood messages in the OLSR network using MPRs. The neighbor sensing operation allows routers to diffuse local information to the whole network. Topology discovery is used to determine the topology of the entire network and calculate routing tables. OLSR uses four message types: Hello message, Topology Control (TC) message, Multiple Interface Declaration (MID) message, and Host and Network Association (HNA) message. Hello messages are used for neighbor sensing. Topology declarations are based on TC messages. MID messages contain multiple interface addresses and perform the task of multiple interface declarations. Since hosts that have multiple interfaces connected with different subnets, HNA messages are used to declare host and associated network information. Extensions of message types may include power saving mode, multicast mode, etc. [7]

The example shown in Figure 3 presents the basic idea of the OLSR protocol.

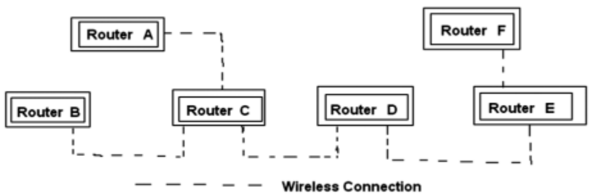


Figure 3: OLSR Routing Protocol

First, we discuss some concepts used in OLSR, namely one-hop neighbor set, two hop neighbor set, MPR set, and MPR selector (MPRS). The one-hop neighbor set is formed by all adjacent routers. For example, Router *C* forms the one-hop neighbor set of Router *A*. A two-hop neighbor set is the set of routers that are two hops away. Routers *B* and *D* form the two-hop neighbor set for Router *A*. The MPR set of a router is a subset of neighboring routers that are responsible for forwarding control messages sent by that router. The MPR set should be able to cover all the two-hop neighbors of that router.

For example, Router *D* is a neighboring node to Router *C*. It covers Router *C*'s two-hop neighbor, Router E. Therefore, Router D is the MPR set of Router C. Since the MPR set of a router is responsible for rebroadcasting messages sent by that router, the routing protocol is closer to optimal with a smaller MPR set. Qayyum, et al. [8] give a simple algorithm to select MPRs, together with an example. The MPR selector (MPRS) set of one router is formed by routers that select this router as one of their MPR routers.

**Table 1**
**Shows the MPR Set and MPR Selector Set for Each Router of Figure 3.**

| *Routers ID* | *MPR Set* | *MPR Selectors* |
|---|---|---|
| A | C | NULL |
| B | C | NULL |
| C | D | A, B, D |
| D | C, E | C, E |
| E | D | D, F |
| F | E | NULL |

## REFERENCE

[1] I.D. Aron and S.K.S. Gupta, "On the Scalability of on-demand Routing Protocols for Mobile ad hoc Networks: An Analytical Study", *in Journal of Interconnection Networks*, **2(1)**, pp. 5-29, 2001.

[2] A. Boukerche, "Performance Comparison and Analysis of ad hoc Routing Algorithms", *in Proc. of IEEE International Conference on Performance, Computing, and Communications*, pp. 171-178, 2001.

[3] D.B. Johnson, D.A. Maltz, Y.C. Hu, and J.G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", *Internet Engineering Task Force 208 (IETF) Draft, Febuary 2002*. Available at http://www.ietf.org/internet-drafts/draftietf-manet-dsr-07.txt.

[4] C.E. Perkins, E.M. Belding-Royer, and S.R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", *Internet Engineering Task Force (IETF) draft, November 2002*. Available at http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt.

[5] V. Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", *Internet Engineering Task Force (IETF) Draft*, July 2001.

[6] E. Gafni and D. Bertsekas, "Distributed Algorithms for Generating Loop-free Routes in Networks with Frequently Changing Topology", *in IEEE Trans. on Communication*, **29(1)**, pp. 11-18, January 1981.

[7] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol", *Internet Engineering Task Force (IETF) Draft*, March, 2002. Available at http://www.ietf.org/internet-drafts/draftietf-manet-olsr-06.txt.

[8] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks", *Research Report-3898*, INRIA, France, 2000.