Practical Enhanced Topology Discovery Algorithm for MANET

Huasen FU
CIO Science and Technology & Information Center
Tobacco Company of Yichang City
Yichang City Hubei province CHINA
18908605288@189.cn

***Abstract*—In order to effectively manage the communication nodes in Mobile Ad Hoc Network (MANET), it is important to know the topology of such networks. Traditional Ad-hoc On-demand Distance Vector (AODV) protocol can not get the MANET topology due to the lack of the topology discovery mechanism. Aiming at this problem, an enhanced topology discovery algorithm based on AODV (AODV-TD) is proposed. AODV-TD can not only obtain the topology of dynamic network by topology discovery request, but also realize route discovery among nodes. Finally, the proposed algorithm is implemented on a hardware platform using CC1100, and the experiment results show that, even the network changes dynamically, AODV-TD can still discover the network topology rapidly and accurately, so that contribute to reducing the route discovery time and the RREQ forwarding packets.**

Keywords – MANET, enhanced AODV, CC1100, dynamic topology discovery

I. INTRODUCTION

MANET is more and more widely used in commerce and industry applications such as intelligent home, data acquisition in industrial sites, indoor location and others. The topology of MANET changes very frequently, so it is important to get the topology to achieve network management [1], i.e. location, management [2], diagnosis [3] and providing information of routings and nodes [4]. In [5-6], one or more mobile agents based topology discovery protocol roams constantly in the entire network to collect topology information. However, mobile agent is hardly implemented in some applications such as complex remote MANETs or large networks. In [7], a topology discovery algorithm (TopDisc) is presented in which a monitoring node requires the topology of the network by a topology discovery request (TDREQ). When a node receives a packet, it forwards the request immediately but waits for its children nodes to respond before sending its own topology discovery response (TDREP). On receiving responses from its children, it aggregates the data and sends it to its own parent. The monitoring node can get topology from these responses.

TopDisc forms a Tree of Clusters rooted at the monitoring node, which can accurately obtain the topology, but the lack of route discovery makes the algorithm hard to achieve communication between ordinary nodes. Moreover, once node receives a TDREQ, it will ignore other TDREQs. If the topology changed, the monitoring node cannot start a new topology discovery. In order to realize route discovery and repeatable topology discovery of dynamic network, this paper presents an improved TopDisc based on AODV. In AODV-TD, the recency of TDREQs can be indicated by broadcast number (BN). The monitoring node periodically broadcasts TDREQ to get current topology, the receiving node will handle and broadcast new TDREQs, and drop outdated TDREQs.

AODV routing protocol is one of the representative reactive routing protocols in MANET [8]. Currently, many research institutions have achieved some typical MANET routing protocols such as AODV-UU, DSR-UU, Kernel-AODV, AODV-UIUC, AODV-UCSB and others. Most of the mentioned implementations are based on Linux Testbed or FPGA. References [9-10] give improvements of AODV-UU and present implementation of proposed protocol on Linux Testbed. Reference [11] develops route repair scheme of AODV protocol, and the improved algorithm is verified on hardware architecture based Virtex-IV. Reference [12] gives a detail design of AODV protocol on FPGA.

These implementations based on Linux PC or FPGA are usually high-cost. And the lack of mobility is hard to meet the requirements of dynamic variable network. So it is necessary to realize AODV protocol on low-cost and low-power embedded hardware platform. In this paper, we focus on how to implement AODV-TD on hardware platform based on single chip. The rest paper is organized as follows. Section 2 presents the detail design of AODV-TD, the hardware design based on CC1100 is given in section 3. Section 4 gives the program design of AODV-TD. Section 5 presents the experiment environments and results. Finally a conclusion is drawn in section 6.

II. ALGORITHM DESIGN

A. *Detail design of network discovery*

In AODV-TD, the recency of route is indicated by destination sequence number (DSN), and the recency of packet is indicated by broadcast number (BN). BN can avoid the node repeatedly processing the same packets, and DSN ensures loop freedom. Before a message is sent out, the node increases its BN and adds new BN into this message. The receiving node can identify the recency of message by BN. The improved TopDisc based on AODV is as follows:

i. The monitoring node periodically increases BN and broadcasts a new TDREQ to initiate topology discovery procedure in network.
ii. If the receiving node has not received a TDREQ with the same BN, the receiving node will create a reverse route to monitoring node if the receiving node has no route to monitoring node, or update the reverse route when the DSN of TDREQ is greater than the one in local routing table, then replace precursor node field with the precursor node address in TDREQ and

broadcast new TDREQ to nodes within its transmission range.

iii. If the receiving node is a terminal node, it generates a TDREP and adds local node address into routing-link field in TDREP, then unicasts this TDREP to monitoring node along the reverse route created by TDREQ. At each intermediate of routing-link, a reverse route to terminal node is created or updated when a TDREP is received. The procedures of routing creating or updating are the same with TDREQ. The receiving node adds local MAC address into routing-link field in TDREP, and then unicasts it to monitoring node.

iv. When the TDREP message arrives to a monitoring node, the routing-link can be achieved from routing-link field in TDREP. After receiving all TDREPs, monitoring node can get the network topology from these routing-links.

B. *Definition of precursor node (PN) and terminal node*

Terminal node is determined by precursor node field in TDREQ. A node will be considered as terminal node if it cannot receive a TDREQ which precursor node field is match with local address in a period of time (TDREQ_TIMEOUT) after the node broadcasts this TDREQ.

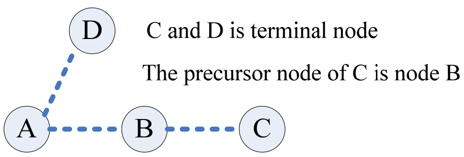


Figure 1. Definition of terminal node and precursor node

As shown in figure 1, node A is a monitoring node. If node C receives a TDREQ from node B, node B will be defined as the precursor node of C. After receiving TDREQ from node B, node C will broadcast a TDREQ which precursor node field is B, and node B will receive this packet, so node B isn't a terminal node. Though node C can receive a TDREQ packet from node D, the precursor node of this packet is A, so node C is a terminal node.

III. Hardware Platform Design

The hardware platform of node consists of 8 bits microprocessor STC90C516AD and wireless communication chip CC1100. The TC90C516AD is developed with 8051core, has 10-bit A/D converters, 3 timers and 39 I/O interfaces. It can operate at maximum 40 MHz, and its 4352 Byte RAM can provide adequate dynamic storage space for AODV-TD protocol. The CC1100 is a low-cost sub-1 GHz transceiver designed for low-power wireless applications. The circuit is mainly intended for the ISM (Industrial, Scientific and Medical) and SRD (Short Range Device) frequency bands at 315, 433, 868, and 915 MHz The modem supports various modulation formats and has a configurable data up to 500kbaud. The communication distance is about 100 to 500m at 10dbm wireless power.

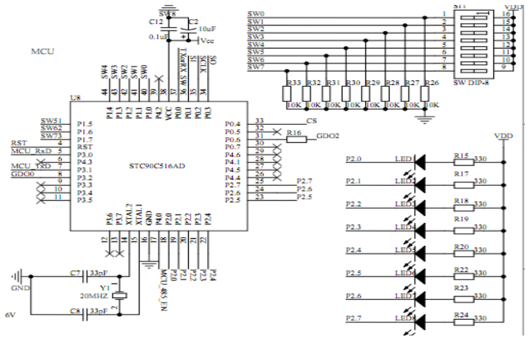


Figure 2. Design of hardware platform

This design adopts simulation SPI bus to complete the communication between STC90C516AD and CC1100. The simulation SPI bus (P01-P04) is connected to SPI interface (MISO, MOSI, CLK, CSN) of CC1100, and the GDO0 pin connected to external interrupt pin (P32) is utilized to detect the state change of CC1100 FIFO. In order to set MAC address for the node, an 8-pin toggle switch connected to STC90C516AD (P10-P17) is applied to support for a MANET of 254 nodes.

Nodes communicate with each other through CC1100. When the CC1100 receives a packet from neighborhood node, the change of its GDO0 pin state interrupts the processor to turn to the interrupt service routine to read data from CC1100 FIFO.

The program operates on STC90C516AD to realize AODV-TD protocol. Detailed design is given in section 4.

IV. Implementation of AODV-TD

The implementation of AODV-TD based on single chip consists of four layers: the physical (PHY) layer, the medium access control (MAC) layer, the network (NWK) layer and the application layer (APL). Each layer can use the services provided by its lower layer, and supplies services for its upper layer.

A. *Implementation of PHY layer*

TABLE I. FUNCTIONS DESIGN OF PHY LAYER

Function	**Note**
U8 RFInit (void)	Initialize RF chip.
U8 ChannelDection (void)	Detect the radio signal intensity of channel
U8 ChannelSet (U8 channel,U8 baud,U8 frequency)	Set the channel frequency, baud and carrier channel
U8 RFSendPacket (U8 *txBuffer,U8 size)	Send data through CC1100
U8 *RFReceivePacket (void)	Read data from CC1100 FIFO

The PHY layer is the unique layer where data is physically transmitted to other nodes. All of the messages created by high layers must be transmitted through the PHY layer, and to be actually sent out in this layer. The PHY layer provides data service and management service for the MAC layer, including the initialization of CC1100, the set of channel frequency and carrier frequency, the sending and receiving data packets and the channel detection based on CSMA-CA.

B. Implementation of MAC layer

The MAC layer adopts CSMA/CA to coordinate nodes that share a common channel to achieve reliable and efficient transmission. Current channel status can be acquired from the PHY layer through ChannelDection function. If the channel is busy, the processor waits a random period of time (between MIN*DT and MAX* DT) and tries again, repeating this process until the channel is idle or the maximum retry count (RETRYCOUNT) is run out.

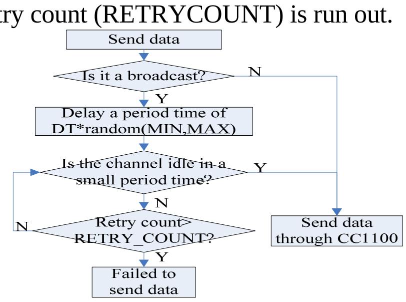


Figure 3. Collision detection before send data through CC1100

C. Implementation of network layer

i. Data structure of AODV-TD

```
typedef struct
{
 UINT8  Flag;           /* Flag of route table*/
 UINT8  DestNode;       /* Destination*/
 UINT8  DestNodeBid;    /* Destination BID*/
 UINT8  DSN;            /* DSN  */
 UINT8  NextNode;       /* Next node to Dest*/
 UINT8  Hops;           /* Hops to Destination*/
 UINT8  RouteLinkFlag;  /* Flag of route link*/
 UINT8  RL[MAX_LINK];   /* Route link*/
 UINT8  TimerFlag;      /* Flag of Timer */
 UINT8  TimerCount;     /* Timer count of route*/
}rt_table;
```

Figure 4. Data structure of route table

```
struct top_disc_request
{
 UINT8 Type;            /* Type of Message*/
 UINT8 ProtocolType;    /* Type of Protocol*/
 UINT8 SourceNode;      /* Source Node*/
 UINT8 BN;              /* Broadcast Number*/
 UINT8 PN;              /* Precursor Node*/
 UINT8 Hops;            /* Hops to Source*/
 UINT8 TTL;             /* Time to Live*/
};
struct top_disc_reply
{
 UINT8 Type;            /* Type of Message*/
 UINT8 ProtocolType;    /* Type of Protocol*/
 UINT8 SourceNode;      /* Source Node*/
 UINT8 BN;              /* Broadcast Number*/
 UINT8 Hops;            /* Hops to Source*/
 UINT8 Link[MAX_LINK];/* Route Link*/
};
```

Figure 5. Definitions of TDREQ and TDREP

```
struct route_request
{
 UINT8 Type;           /* Type of Message */
 UINT8 ProtocolType;   /* Type of Protocol*/
 UINT8 SourceNode;     /* Source Node     */
 UINT8 SSN;            /* SSN             */
 UINT8 BN;             /* Broadcast Number*/
 UINT8 DestNode;       /* Dest Node       */
 UINT8 DSN;            /* DSN             */
 UINT8 Hops;           /* Hops to Source  */
 UINT8 TTL;            /* Time to Live    */
};
struct route_reply
{
 UINT8 Type;           /* Type of Message */
 UINT8 ProtocolType;   /* Type of Protocol*/
 UINT8 SourceNode;     /* Source Node     */
 UINT8 SSN;            /* SSN             */
 UINT8 BN;             /* Broadcast Number*/
 UINT8 DestNode;       /* Dest Node       */
 UINT8 DSN;            /* DSN             */
 UINT8 Hops;           /* Hops to Source  */
};
```

Figure 6. Definitions of RREQ and RREP

AODV-TD protocol is a network layer protocol. This layer is designed to achieve reliable data transmission among nodes, such as addressing and route discovery, routing maintenance, topology discovery.

ii. State machine of NWK layer

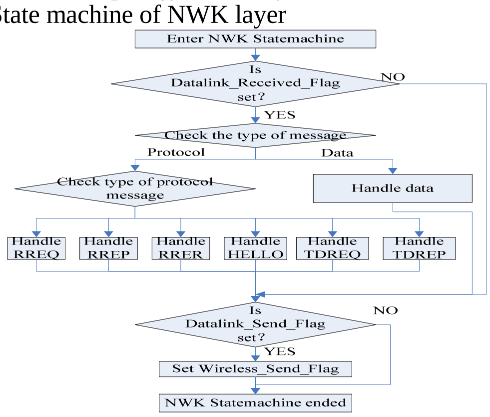


Figure 7. State machine of NWK layer

The state machine of NWK layer has two flags: Datalink_Received_Fla, Datalink_Send_Flag:

Datalink_Received_Flag will be set when the node receive a packet, and Datalink_ Send _Flag will be set when the node need to send data.

The process of TDREQ and TDREP are shown in figure 8 and figure 9. After broadcasting a new TDREQ, the node will start a timer (TD_TIMER) to check whether it is a terminal node. When the precursor node field of received TDREQ is equal to local address, TD_TIMER will be stop. If the node cannot receive such a TDREQ, the TD_TIMER will be over and this node will be judged as terminal node. Then a TDREP will be unicast to monitoring node.

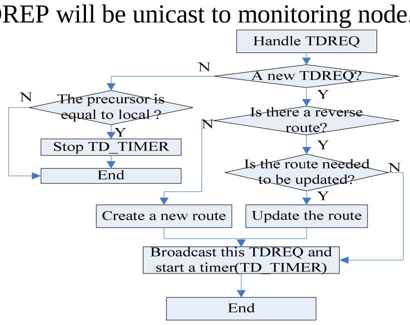


Figure 8. Handle process of TDREQ

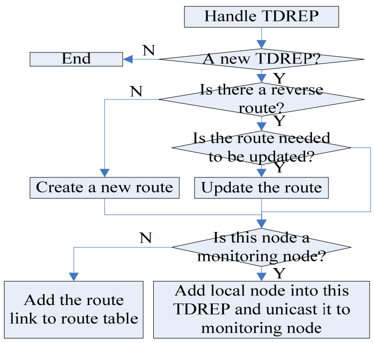


Figure 9. Handle process of TDREP

D. *Implementation of APL*

APL is the highest level of user-oriented that provides specific services for user application, such as send a command message to another node to turn off its light-emitting diode, or to get the network topology. Some NWK layer interfaces, such as GetTopology function and SendData function, can provide services for application software.

TABLE II. FUNCTIONS DESIGN OF APL LAYER

Function	Note
U8 SendCommand (U8 dest)	Send a command to dest node
U8 PrintTopology()	Print out network topology through uart
U8 HandleCommand (U8 *buffer)	Handle command message received from others

V. EXPERIMENTS AND EVALUATION

A. *Experiment environment setting*

The experiment of AODV-TD is carried on a small net consists of 8 nodes. In order to simulate multipath and multi-hop communication in experiment, each node can only communicate with the physically adjacent nodes by extra MAC address filtering. For example, as showed in figure 10, node 3 may receive messages from the other 7 nodes, but it will drop those data received from node 1 and node 5 for they are not physically adjacent to node 1. This filtration mechanism can help to create multi-hop route such as 1-2-3-4-5.

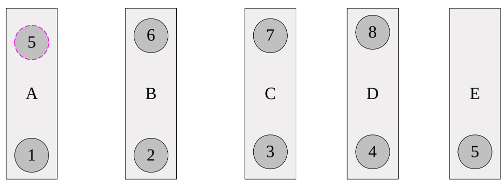


Figure 10. Experiment environments of 8 nodes

The baud rate of this implementation is 38.4kbps, and the max size of single protocol packet is 64 bytes. Under this condition, the max transfer time of a packet is: 64*8/38.4=13.3ms. So we set DT is 10ms.

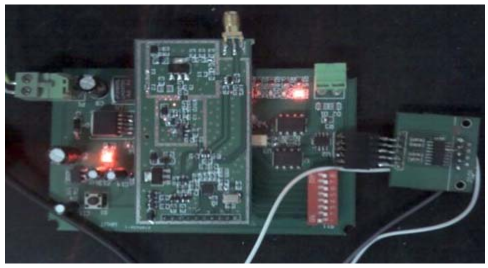


Figure 11. Hardware implementation based on CC1100

B. *Routing discovery*

This implementation of AODV-TD adopts single-path route mechanism. Turn on node 2, 3, 4 and 7, and start a routing discovery from node 2 to node 4.

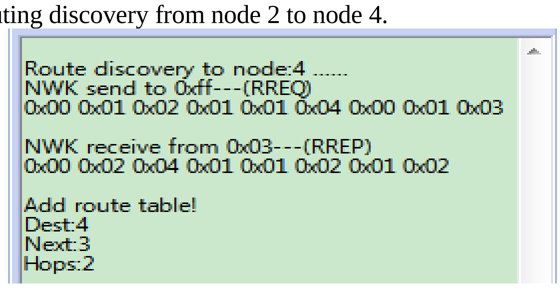
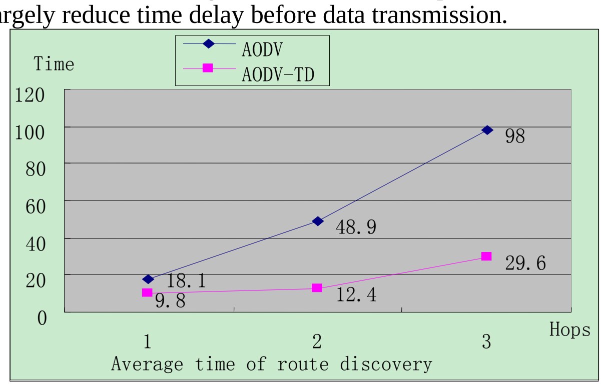


Figure 12. Routing discovery to node 4

Node 2 broadcasts a RREQ to find a route to node 4, and receives a RREP from node 4 through node 3. This route discovery creates a route 2-3-4.

Move node 5 to area A to format a 2*4 net. Each node will send RREQ to the other 7 nodes. In order to avoid the influence of old route, all routing tables will be cleared before a new discovery.

As shown in figure 13, the route discovery time and the RREQ forwarding numbers of AODV-TD are less than AODV. With the increase of hops, the advantage of AODV-TD becomes more obvious. The route discovery time of AODV-TD reduces about 70% compared to the conventional AODV algorithm at 3 hops. This is because the pre-established routing created by TDREQ and TDREP help to reduce the forwarding numbers of RREQ packet. This will largely reduce time delay before data transmission.



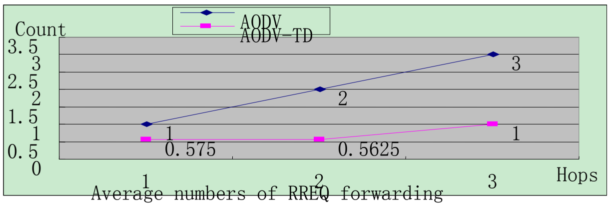


Figure 13. Performance of routing discovery

C. Topology discovery

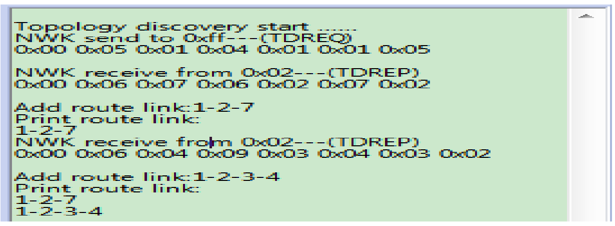


Figure 14. Result of topology discovery

Turn on node1, 2, 3, 4 and 7, and start a topology discovery in network. The process of topology discovery is shown in figure 14. The monitoring node receives two TDREPs from node 7 and node 4, and gets two route links from TDREPs: 1-2-7 and 1-2-3-4.

Open node 2 to node 8 one by one to simulate change of topology, and broadcast TEREQ every 2 seconds in network test the performance of topology discovery.

Table 3 shows the success rate and average time of topology discovery. When there is only one routing link (the number of nodes is 2, 3, 4, 5), the success rate is relatively high. With the number of nodes increases (form two links), the success rate decreases. This is because back off mechanism failed when channel collision occurred among two adjacent logical nodes at two links. The message is dropped and downstream nodes can't receive this message. Experiment result also shows that average network time has less to do with multi-hop because the transmission of RREP is unicast along with a single link.

TABLE III. PERFORMANCE OF TOPOLOGY DISCOVERY

Number of node	Network times	Success times	Success rate	Average time
2	100	100	100%	356ms
3	100	100	100%	388ms
4	100	98	98%	465ms
5	100	96	96%	516ms
6	100	92	92%	520ms
7	100	89	89%	538ms
8	100	85	85%	540ms

VI. CONCLUSIONS

AODV-TD can not only obtain the topology of dynamic network by topology discovery request, but also realize route discovery among nodes. The experiment shows that AODV-TD can achieve route discovery in network, which make it is possible to transmit data in dynamic network. AODV-TD can also rapidly get the current topology when node moved in network, and the pre-established routing will help to reduce data transmission delay. And importantly, this implementation reflects the true characteristics of the wireless channel, such as packets collision and time delay. In the future work, we will improve the backoff scheme to reduce packets collision in data transmission. And more nodes will be used in network to test the performance of AODV-TD.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] Nassu, Bogdan T.; Nanya, Takashi; Duarte Jr., Elias P. Topology Discovery in Dynamic and Decentralized Networks with Mobile Agents and Swarm Intelligence. Proceedings of The 7th International Conference on Intelligent Systems Design and Applications, ISDA 2007, p 685-690, 2007

[3] Cisco Systems, Inc. Internetworking Technologies Hand-book, chapter 7: Network Management Basics. Cisco Press,4th edition, 2003.

[4] E. P. Duarte, Jr. and A. Weber. A Distributed Network Connectivity Algorithm. In Proceedings of the Sixth IEEE Inter-national Symposium on Autonomous Decentralized Systems (ISADS'2003), pages 285–292, Italy, 2003.

[5] K. A. Amin. Resource Efficient and Scalable Routing Using Intelligent Mobile Agents. Master's thesis, University of North Texas, May 2003.

[6] Jing, Wei; Wei, Guo; Jian, Su; Wei, Tang. Mobile agent based topology discovery in mobile ad hoc networks. Proceedings - 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009, 2009

[7] Ahmed, Adnan; Far, Behrouz. Mobile agent system for network topology discovery. Canadian Conference on Electrical and Computer Engineering, p 1376-1379, 2007

[8] Budhaditya Deb, Sudeept Bhatnagar and Badri Nath. A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management. Technical Report DCS-TR-441, Department of Computer Science, Rutgers University, May 2001.

[9] Motegi S; Horiuchi H. AODV-based multipath routing protocol for mobile ad hoc networks. IEICE Transactions on Communications, v E87-B, n 9, p 2477-2483, September 2004

[10] Yun Jangkyu, Lee Byunghwa, Baek Youngmi, Kim Junhyung, Han Jihun. An Implementation of AODV testbed with multi-metrics. 2011 8th International Conference on Information Technology: New Generations, ITNG 2011, p1072-1073, 2010

[11] Mubarik, Muhammad Asim, Khan, Shoab Ahmad, Hasan, Syed Ayad, Sarfraz Naveed. Implementation of Geocast Enhanced AODV-UU in Linux Testbed. 2009 8th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2009, p803-807, 2009

[12] Rathinam A, Natarajan V, Vanila S, Viswanath A, Guhan, M.S. An FPGA Implementation of Improved AODV Routing Protocol For Route Repair Scheme. 1st International Conference on Emerging Trends in Engineering and Technology, ICETET 2008, p 971-974, 2008

[13] Ramakrishnan M, Shanmugavel S. FPGA IMPLEMENTATION OF AODV ROUTING PROTOCOL IN MANET. 1st International Conference on Industrial and Information Systems, ICIIS 2006, p470-473, 2006