Developing QoS Aware DSR for JiST/SWANS Simulator Using Cross-Layer Communication

Sharwari S. Solapure and P.J. Kulkarni

Abstract In MANET there is a need of a QoS path for real-time traffic. QoS is considered in terms of queueing and MAC delay. Dynamic Source Routing (DSR), the most popular routing protocol is proposed here for QoS routing. The work is only related to the DSR of the JiST/SWAN Simulator. This chapter includes the implementation steps for QSADSR (Quality of Service Aware DSR) for the JiST/SWANS simulator and also the results of it. The results show that QSADSR works best for end-to-end delay but is not fine for throughput. However, to increase the throughput in the same environment, new techniques are mentioned here.

Keywords DSR · QoS · MANET · QSADSR

1 Introduction

In a mobile ad hoc network (MANET) each node acts as an ordinary node and also as a router. Each is accompanied by a routing protocol and maintains route information. This information is used to find a routing path between nodes.

Traditional routing protocols are not straightforwardly useful to ad hoc networks. Routing protocols for ad hoc networks are of three types: on demand, table driven, and hybrid protocols. There are different routing protocols, which are uniform according to the IETF (Internet Engineering Task Force) [1, 2]. In this chapter we focus on dynamic source routing (DSR) [3–5], an on demand and most popular protocol for MANET. The DSR implementation in the simulator JiST/SWANS does not have QoS aware features. A few modifications with the default DSR protocol define a new protocol, Quality of Service Aware DSR

S.S. Solapure (✉) · P.J. Kulkarni
Department of Computer Science and Engineering, Walchand College of Engineering, Sangli 416415, Maharashtra, India
e-mail: solapure_ss@yahoo.co.in

P.J. Kulkarni
e-mail: pjk_walchand@rediffmail.com

N.R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, DOI 10.1007/978-981-10-0287-8_2

(QSADSR), which is discussed here. The chapter presents results of QSADSR with respect to end-to-end delay, packet delivery ratio, and throughput. The chapter is organized as follows: Sect. 2 presents the DSR details. Section 3 presents the basic idea of cross-layer design. Sections 4 and 6 present bandwidth calculation and modification steps with DSR. Section 5 gives details of the JiST/SWANS simulator and Sect. 7 shows modification in the simulator and results. The last section concludes the chapter.

2 DSR Details

The DSR protocol [1, 3] is a routing protocol for MANET. It has functionalities for finding a path using multihop mobile ad hoc networks. The DSR protocol operation is divided into the following phases.

2.1 Route Discovery

In this phase the protocol intention is to find a path between two mobile nodes. To accomplish this, the source node generates a route request that is flooded over the network. This flooding is done until the request reaches the destination node. Every intermediate node inserts its IP address in the header of the route request after receiving this request, and rebroadcasts it.

2.2 Route Reply

In this phase the protocol at the intended destination side adds routing information to its cache. It also adds its IP address at the end of the route request header. After that it sends a reply, along with the established path in the header. The destination does it for all the route requests. This gives the number of available routes for the same group of two nodes.

2.3 Route Maintenance

This phase protocol maintains the route. It is done by an intermediate node in response to link failures. The node detects link failure and initiates a packet to inform the source node as a route error.

2.4 Route Cache

This phase protocol enhances the performance by restricting new route requests for each packet. It uses a cache, which saves all the established routes for the future.

3 Cross-Layer Design

In the layer model, each layer picks up data from the neighbor layers, runs its own algorithm, and gives services to them. This is done via service access points. In the cross-layer design it is not the same. Here it shares data between all layers if required.

Numbers of techniques are proposed in the literature [6–8]. They have enhanced the performance of ad hoc network services. One technique is the cross-layer communication between media access control (MAC) and the network layer, which is shown in Fig. 1.

This technique takes information from a number of layers and uses it in the routing algorithm. It enhances the performance of packet forwarding. For this new route metrics are used such as MAC delay, queueing delay, and so on.

4 Delay Measurement

There are many metrics that define the quality of a link. But it is not possible to use all metrics for route selection as it increases end-to-end delay. The routing metrics used here for QoS routing are link delay and queueing delay. Both delays are used as one metric for selecting the path. To implement this strategy one existing protocol is selected, DSR. It has some special characteristics that improve the efficiency of routing in MANET. The sum of both delays represents each link status and congestion at each node. This is calculated as follows.

4.1 Link Delay Measurement

Bandwidth is calculated by considering two neighboring nodes' link layer capacity. The IEEE 802.11 [9] standard for carrier sense multiple access with collision avoidance (CSMA/CA) is considered for this work. The packet broadcast stages are shown in Fig. 2.

The approximate available bandwidth for transmitting a packet of size S as described in Fig. 2 is defined as

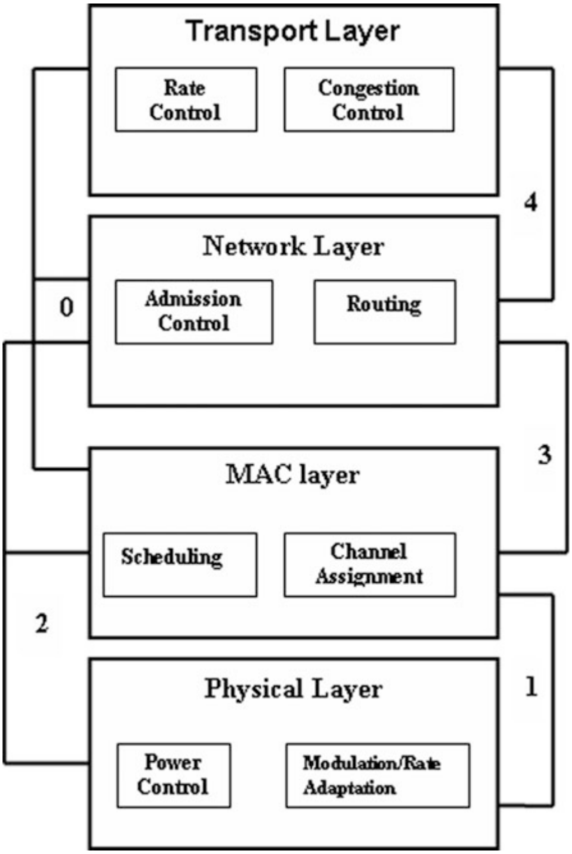

Fig. 1 Cross-layer framework and interaction among layers

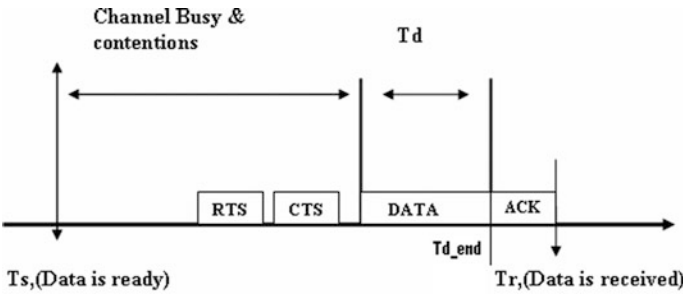

Fig. 2 Packet transmission stages in 802.11 MAC

$$\text{Bandwidth} = S/(T_r - T_s) \tag{1}$$

where S = Packet size and T_r, T_s in seconds.

The same estimation gives an idea to calculate link delay/MAC delay. As illustrated in Fig. 2, the MAC delay is defined as the interval from when the data are ready to when the data packet is received. This delay is used as a routing metric for interference awareness. If there is a lot of interference, MAC delay is high due to the contention of the channel, and it gives an idea about interference of the link. It is another way of defining the available bandwidth of the link. It is defined with Eq. (2):

$$\text{Link delay} = T_{\text{d\_end}} - T_s\,(\text{sec}) \tag{2}$$

4.2 *Queueing Delay Measurement*

The queueing delay T_Q in the node buffer of the transmit node is used as the routing metric. The queueing delay is a measurement of congestion at each node. The routing algorithm handles the load by avoiding nodes having a buffer full of packets. This delay is calculated at the network layer. It is the time interval between the instance when the packet enters the queue and when it leaves the queue. It is expressed as

$$\begin{aligned}\text{Queuing Delay} = {} & T(\text{packet leaves the queue}) \\ & - T(\text{when packet enters the queue})(\text{sec})\end{aligned} \tag{3}$$

The delay value for the routing metric is the sum of both Eqs. (2) and (3) and it is simply defined as

$$\text{Delay} = \text{Link Delay} + \text{Queueing Delay}\,(\text{sec}) \tag{4}$$

From this point onwards, the term Delay is used to denote the sum of both delays.

5 JiST/SWANS Simulator [10]

The JiST [10] system, which stands for Java in simulation time, is a new Java-based discrete-event simulation engine with a number of novel and unique design features.

The simulator design using SWANS is organized as self-regulating working. It can be used to form a complete wireless network or sensor network simulation. There are many advantages to this simulator; that is why it is selected for the QoS work.

5.1 Modification with DSR for QSADSR

The design of the quality of service aware DSR [7, 11–13] protocol is mainly concerned with route discovery, route reply, and cache. Some modifications are required for DSR source code of the JiST/SWANS simulator to make it QoS SSSaware. Few modifications are made in the source code of Mac802_11.java, NetIP.java code for calculation MAC, and queueing delay. Extensive modifications are made in the source code of RouteDsr.java and RouteDsrMsg.java of JiST/SWANS simulators to make it QSADSR. It has been carried out mainly with the following phases of DSR.

A. Route discovery It modifies the contents of the original DSR header with a Delay value. The value is computed as described in Eq. (4). Now every node receives a new route request with the Delay value and inserts it in the header of the route request.

B. Route reply In QSADSR the destination node does not respond instantly to the first route request, although it waits for a predefined time to receive new route requests in order to choose the top one by the minimum Delay value. After wait time, the destination node selects only one route and replies to the source nodes.

C. Route cache The modifications at the cache are used to select routes according to the Delay in each cached route.

6 Result Analysis

Although the DSR protocol and DSR with QoS have similarities, they diverge in the mechanism of certain requirements. This can be analyzed using various network load and mobility conditions. The metrics used to evaluate the DSR and the QSADSR routing protocols performances are as follows.

Average end-to-end delay It is the time taken by the packet to travel from the application layer of the source to the application layer of the destination. It is related to the packets that are successfully transmitted from the source to the destination.

$$\text{Avg End}-\text{End Delay} = \frac{\text{Sum of all packets end}-\text{end delay}}{\text{Total number of received packets}} \tag{5}$$

Packet delivery ratio (PDR) It is the ratio of data packets received at destinations and the packets generated at the sources.

$$\text{PDR} = \frac{\text{Total number of received packets}}{\text{Total number of sent packets}} \tag{6}$$

Throughput It is the ratio of the total data that reach the receiver and the time taken by the receiver to receive them. Throughput is expressed in bytes or bits per second (byte/sec or bit/sec).

Goodput/application level throughput It is the application level throughput, that is, the number of useful bits per unit of time forwarded by the network from a certain source address to a certain destination, excluding protocol overhead and retransmitted data packets.

It can be represented mathematically as in Eq. (7),

$$\text{Throughput} = \frac{\text{Number of received packets} * \text{packet size} * 8}{\text{Time required to receive the packets}} \tag{7}$$

From this point onwards, the term throughput is used to denote application level Throughput/Goodput.

To analyze the protocol for the number of nodes in a specific network size it is necessary to set up a simulator environment. This environment is set with the number of parameters such as network size, propagation model, transmission range, and the like. Also one general application is required, which should run the simulation and give us the information required for analyzing the protocols such as the following.

1. Packets send/receive time with respect to the sequence number.
2. Node positions in the network.
3. Log file that maintains each event of DSR, UDP, network layer.

Numbers of scenarios are generated to analyze the protocol. The scenarios differ in positions of nodes, speed of node, and so on. The parameters used for each scenario are given in Tables 1 and 2 while discussing the results.

Scenario 1: Analysis of Average End-to-End Delay for Mobility (Different Pause Times)

Simulator parameters used for scenarios 1 and 2 are as shown in Table 1.

The results in Fig. 3 show that QSASDR has less delay than DSR. The results in the Fig. 4 show that QSADSR gives a slight reduction in PDR value for high

Table 1 Simulator parameters for Scenarios 1 and 2

Parameter	Value	Parameter	Value
Nodes	120	MAC protocol	802.11 (without RTS)
Network size	1200 m × 1200 m	Transport protocol	UDP
Mobility	Random way mobility, min_speed = 5 m/s, max_speed = 1, 5, 10 m/s	Bandwidth	11 Mbps
Simulation time	200 s	Transmission range	672 m
No. of CBR conn.	4	Propagation path-loss	Free space
Packet size and data rate	512 byte, 4 Kbps	Packet sends per conn.	28

Table 2 Simulator parameters for Scenario 3

Parameter	Value	Parameter	Value
Nodes	120	MAC protocol	802.11(without RTS)
Network size	1200 m × 1200 m	Transport protocol	UDP
Mobility	Random way mobility, min_speed = 5 m/s, max_speed = 10 m/s	Bandwidth	11 Mbps
Simulation time	200 s	Transmission range	672 m
Number of CBR conn.	3	Propagation path-loss	Free space
Packet size and rate	1024 byte and **64, 72, 80 Kbps**	Packet send per conn.	1360

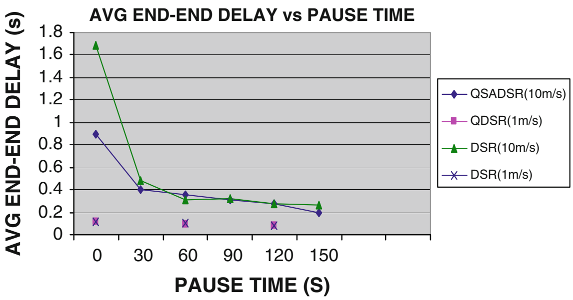

Fig. 3 Average end-to-end delay versus pause time for Scenario 1

Fig. 4 Packet delivery ratio versus pause for Scenario 1

mobility than does existing DSR. If simulation runs for 1 m/s mobility speed both work the same.

Scenario 2: Make Change in the Positions of Nodes with Changing Random Seed. Other parameters are the same as Scenario 1.

Analysis of average end-to-end delay versus pause time for both the scenarios in Figs. 3 and 5 shows that QSADSR is better for high mobility. But analysis of the packet delivery ratio versus pause time in Figs. 4 and 6 shows that for high mobility QSADSR gives a small reduction in PDR as compared to DSR. The main reason behind this is the routes are changing rapidly due to the high speed of the nodes.

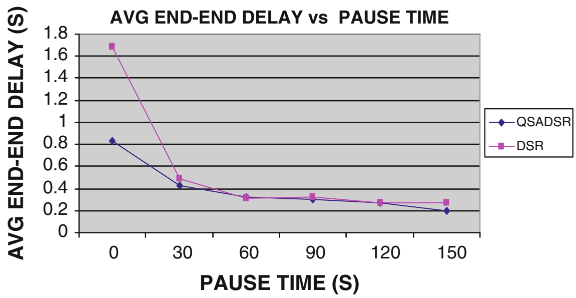

Fig. 5 Average end-to-end delay versus pause time for Scenario 2

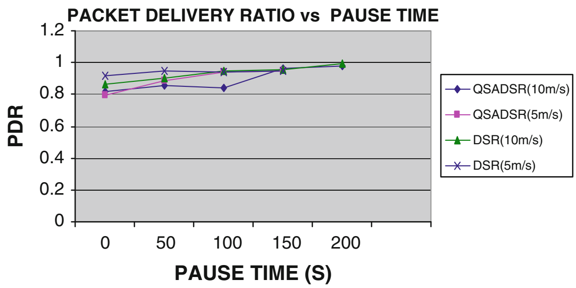

Fig. 6 Packet delivery ratio versus pause time for Scenario 2

Another reason is due to the delay calculation used in the route selection algorithm at the destination to select the best path. The performance for QSADSR for high mobility is enhanced using the following approaches.

1. Reducing $R_x - T_x$ turnaround time at the link layer
2. Increasing the count of retransmissions using passive acknowledgments
3. Increasing the count of retransmissions using network-level acknowledgments

The $R_x - T_x$ turnaround time is the time taken by a device to switch from the receiver state to the transmitter state. It is measured at the MAC/PHY interface. The number of acknowledgments increases the count of turnarounds. The count of retransmissions using passive or network-level acknowledgment is defined in the QSADSR algorithm.

High mobility means where the pause time varies between 0 and 60 s. The performance is enhanced by increasing the count of retransmissions with the DSR algorithm. The PDR value is now around 90 % but it is in the cost of average end-to-end delay. The increase in PDR will not give much increase in average end-to-end delay.

The parameters used for this new approach under Scenario 2 are:

1. $R_x - T_x$ turnaround time at link layer = 4 microsec.
2. The maximum number of times a packet will be retransmitted using network-level acknowledgments = 3.

6.1 New Approach Results

The results in Figs. 7 and 8 show that QSADSR gives more improvement in packet delivery ratio and throughput value for high mobility than existing DSR. Then the new approach is applied for QSADSR for high load condition, as most of the communications are in the range of 64 Kbps.

Scenario 3: Analysis of Average End-to-End Delay for Mobility (Different Pause Times)

Simulator parameters used for Scenario 3 are shown in Table 2.

From the entire results in Figs. 9, 10, 11 and 12 it can be stated that QSADSR works better for moderate speed and moderate flow rate. The pedestrian model is a model where all nodes move with the speed of 1 m/s and pause time of 1 s. The results taken for low mobility (pedestrian model) are shown in Figs. 13 and 14.

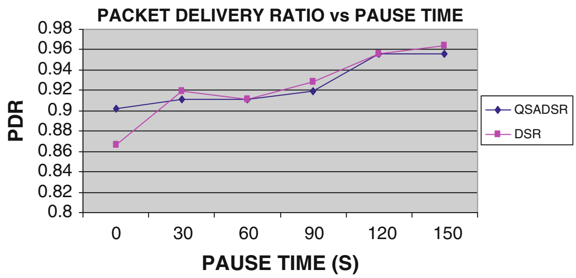

Fig. 7 Packet delivery ratio versus pause time for Scenario 2

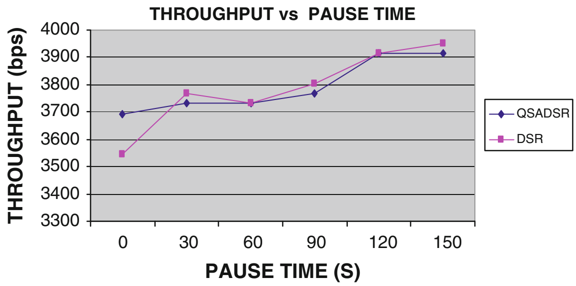

Fig. 8 Throughput versus pause time for Scenario 2

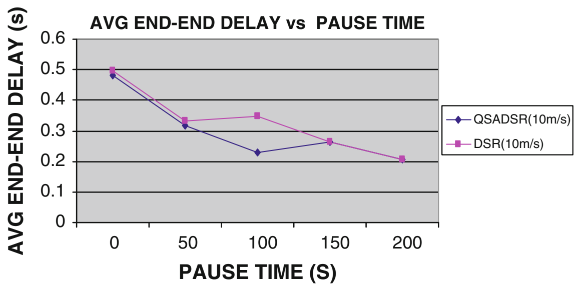

Fig. 9 Average end-to-end delay versus pause time for Scenario 3

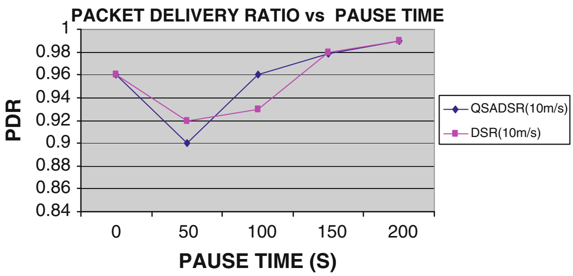

Fig. 10 Packet delivery ratio versus pause time for Scenario 3

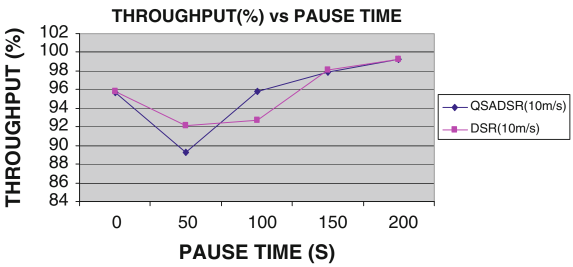

Fig. 11 Throughput (%) versus pause time for Scenario 3

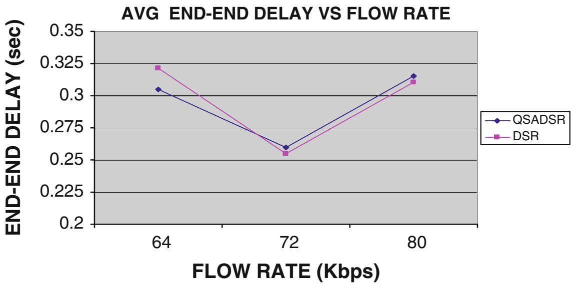

Fig. 12 Average end-to-end delay versus flow rate for Scenario 3

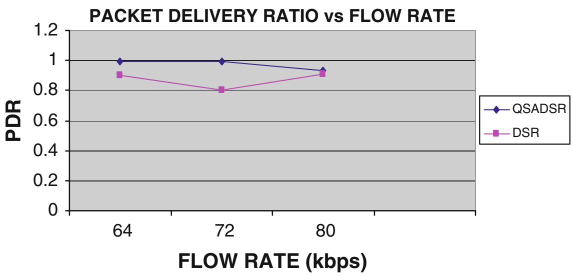

Fig. 13 Packet delivery ratio versus flow rate for Scenario 3

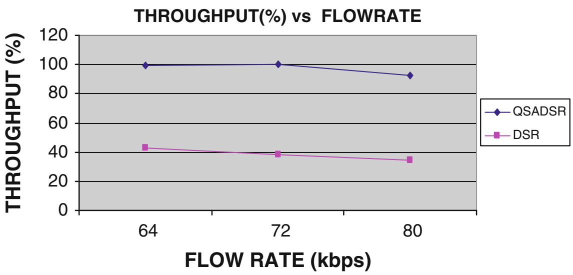

Fig. 14 Throughput (%) versus flow rate for Scenario 3

7 Conclusion and Future Work

In QSADSR, the basic DSR algorithm is modified to select a path by choosing delay as a routing metric. In spite of the additional computations, QSADSR gives a better result for average end-to-end delay than existing DSR. Comparison between QSADSR and DSR shows that QSADSR gives less packet delivery ratio for high mobility.

This ratio is enhanced by using optimization at the network and link layer. This is carried out by increasing the count of retransmissions using passive/network-level acknowledgments and reducing the turnaround time period of the link layer. The results obtained under high load condition show few additional problems. As load increases, PDR and throughput decrease a little bit. Another reason for low throughput lies in the technique used to handle the contention period by CSMA at high speed. The packets get lost at high speed due to the sharing of the link by all the nodes.

Based on the full results, the analysis of QSADSR is now carried out for low mobility using the pedestrian model. The results are improved more than high mobility results. In the proposed work, QSADSR is tested for various speeds and load conditions. The results suggest that QSADSR works better for low mobility and medium load.

Communication in MANET is always needed for high speed and high load conditions. QSADSR can be used in the future for these conditions by undergoing modifications with novel techniques. There are many ways to improve its working. However, the main areas are the link layer and existing DSR overhead. It is a challenging task to improve QSADSR in all aspects such as average delay, PDR, and throughput. This is because of MANET's constraints and the limitations of DSR. It can be done by developing the new routing protocol. The new protocol should reflect advantages of all existing MANET protocols. It requires extensive research in this area.

References

1. Internet Engineering Task Force, Manet Working Group chapter. http://www.ietf.org/html.charters/manetcharter
2. ftp://isi.edu/in-notes/rfc2386.txt
3. Johnson, D.B., Maltz, D.A., Hu, Y.-C.: The dynamic source routing protocol for mobile ad-hoc networks (DSR) April 2003. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt
4. Othman, M.: A new distance based route maintenance strategy for dynamic source routing protocol. J. Comp. Sci. **4**(3), 172–180 (20080). ISSN 1549-3636
5. Johnson, D.B., Maltz, D.A., Broch, J.: DSR: the dynamic source routing protocol for multihop wireless ad-hoc networks, pp. 15213–3891. Computer Science Department Carnegie Mellon University Pittsburgh, PA
6. Alnajjar, F., Chen, Y.: SNR/RP aware routing algorithm: CROSS-LAYER design for MANETS. Int. J. Wireless Mobile Networks (IJWMN) **1**(2), (2009)

7. Kadri, B., Moussaoui, D., Feham, M., STIC Lab: A cross-layer design for QoS implementation in MANETs applied to DSR. Information and communication technologies: from theory to applications 2008. In: 3rd International Conference ICTTA 2008, pp. 81–85
8. Ho, W., Heung-no, Y., Timothy, L., Andersen, D.: A simple and effective cross layer networking system for mobile ad-hoc networks. Military Research Paper
9. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std 802.11, 1999 Edition, Sponsor LAN MAN Standards Committee of the IEEE Computer Society
10. http://JiST.ece.cornell.edu
11. AbuHajar, M., University of Northern Virginia: Quality of service model for dynamic source routing protocol. World Appl. Sci. J. **7**(4), 418–431. ISSN 1818-4952 IDOSI Publications (2009)
12. Zhang, Q., Zhang, Y.Q.: Cross-layer design for QoS support in multihop wireless networks. In: Proceedings of the IEEE, vol. 96, No. 1, January 2008 IEEE
13. Solapure, S., Kulkarni, P.J.: Development of a simulator module for QoS aware DSR Routing in MANET. In: International Conference on Emerging Trends in Computer Science, Communication and Information Technology, January 09–11 (CSCIT2010) PROC, pp. 323–326

http://www.springer.com/978-981-10-0286-1

Emerging Research in Computing, Information, Communication and Applications
ERCICA 2015, Volume 3
Shetty, N.R.; Prasad, N.H.; N., P. (Eds.)
2016, XX, 701 p. 380 illus., 269 illus. in color., Hardcover
ISBN: 978-981-10-0286-1