Asian Journal of Information Technology 15 (14): 2434-2446, 2016
ISSN: 1682-3915

Urgency Aware Packet Scheduling in AODV (UaPS-AODV) Protocol for Enhanced QoS under Delay Sensitive Applications in Manet

[1]S. Mohanasundaram and [2]T. Ravichandran
[1]Department of Information Technology, Institute of Road and Transport Technology, Erode, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering, S.N.S. College of Technology, Coimbatore, Tamil Nadu, India

Abstract: The establishment of Quality of Service (QoS) guarantees as required by applications through conventional routing protocols is a challenging endeavor in MANET due to its dynamic topology, lack of load balancing capabilities, limited power supply and a deficiency of unified authority. Hence, it is essential to have an efficient routing algorithm to satisfy a set of QoS constraints in spite of network dynamics. In this research, we afford a new improvement in AODV routing protocol to deliver QoS guarantees for hard real time applications in MANET. Conventional AODV routing protocol uses priority-only-based packet scheduling to arrange the packets in its queue. Existing scheduling algorithms in AODV are not efficient enough to provide QoS for hard real time packets in uncertain and dynamic environments because these approaches assume that the network is deterministic and predefined decisions will be statically implemented during scheduling. To tackle this problem, we present an interval number theory to explore the network dynamics and an argency aware architecture for packet scheduling to enhance the service demands. According to our architecture, we develop a novel Urgency aware Packet Scheduling (UaPS) algorithm and priority based buffer management policy for real time packets. We perform extensive experiments in order to examine the efficiency of the UaPS-AODV in the NS-2.34 (Network Simulator) and compare with basic AODV and Delay aware AODV (AODV-D) protocols. Our results reveal that the UaPS-AODV outperforms those existing algorithms in terms of performance metrics.

Key words: AODV, packet scheduling, priority, quality of service, routing, real time applications, urgency

INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a crew of arbitrarily moving wireless nodes which do not necessitate any centralized infrastructure support and as a fixed access point or a pre-existent base station to make the communication among hops. The self-organizing nodes are capable of communicating with each other directly based on one-hop communication or multiple-hop communication via intermediate nodes (i.e., relay nodes). Hence, each node acts as a wireless transceiver, router or gateway of data packets. The rapid implementation and the ever-present access capacity of MANET make them crucial in the computing environment in many military and civil applications.

Quality of service is an essential feature of MANET. It is a set of user defined guarantees such as maximum bandwidth, minimum delay, minimum jitter and minimum packet loss rate to be assured by the network. Modern network applications require strict QoS limitations that need to be guaranteed under uncertain and dynamic network environments. On the other hand so far, Internet Protocol (IP) only offers services without guarantees (i.e., best-effort services) and it does not deliver any fail-safe QoS for real time applications. The key to QoS provisioning system in MANET is to accomplish more deterministic performance of the network, therefore the data supported by the system can be supplied effectively and scarce network resources can be exploited efficiently. In MANET, the existing mobile terminals are permitted to depart from the network and new devices are permitted to connect the network arbitrarily. So, the communication link and the network topology change with the nodes' mobility. Most of the service providers fail to consider dynamics of the network. Therefore, establishing the QoS assurance as required by applications via conventional protocols is more challenging and a difficult endeavor in MANET.

Corresponding Author: S. Mohanasundaram, Department of Information Technology, Institute of Road and Transport Technology, Erode, Tamil Nadu, India

Table 1: QoS requirements of network applications

Applications	Delay	Jitter	Bandwidth (Bits/sec)	Packet loss rate
Voice over IP	<100 m sec	<100 m sec	8K-80 K	<1%
Video conference	<150 m sec	<100 m sec	80K-2M	<0.01%
Video broadcasting	<150 m sec	<100 m sec	28.8K-60M	<0.0001-0.001%
Audio broadcasting	<150 m sec	<100 m sec	60-80 K	<0.1%
Internet relay chat	<200 m sec	N/A	<1 K	Zero
Telnet	<250 m sec	N/A	<1 K	Zero
Telemetry	<250 m sec	<100 m sec	2K-52M	Zero
Web browsing	<400 m sec	N/A	<30.5 K	Zero
FTP	2-5 sec	N/A	<10 K	Zero
E-mail	2-5 sec	N/A	<10 K	Zero

Table 1 shows some of the widespread real time applications and their required service qualities. The uprising of video, voice and data applications open innovative directions and challenges to the current and future generation wireless networks where static behavior of the network is not at all conceivable to operate.

Almost all protocols operate at network layer are developed to reduce the network congestion or to select the shortest route for supplying packets (Lin and Liu, 1999). For example, traditional protocols such as Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA) (Lim and atta, 2012) and Ad hoc On demand Distance Vector (AODV) are established without considering the requirement of service demands explicitly. Several researches have been done in this domain and numerous routing protocols have appeared in recent years to support QoS in MANET. The routing layer protocols can fall in three categories: table-driven or proactive routing protocols, on-demand or reactive routing protocols and hybrid (both proactive and reactive) routing protocols (Chand and Soni, 2012).

The mobile terminals that exploit proactive protocols constantly discover the routes to neighboring nodes and update routing information continuously. This approach has the advantage of comparatively less route procurement latency which is essential for some real time applications. However, these protocols incur some additional overheads to update routing statistics. Destination Sequenced Distance Vector (DSDV) and Optimized Link State Routing (OLSR) are examples of proactive routing protocols (Giagko and Wilson, 2010).

Contrarily, the route acquisition process in reactive protocols is performed between a source and a final destination only on demand. Unlike proactive routing protocol, this provides a reduced overhead and consumes less bandwidth but they suffer from a significant latency for searching the paths. Examples of reactive protocols are DSR and AODV (Chand and Soni, 2012). The potential benefits of proactive protocols and reactive protocols are consolidated in hybrid routing protocols. It decreases the control overheads of table-driven protocols and equally reduces the delay in the route discovery process as on-demand protocols. The well-known example of this adaptive protocol is Zone Routing Protocol (ZRP) (Ramamoorthy and Karthikeyani, 2014).

Overview of AODV: The fundamental objective of our research is to enhance the QoS of the real time applications in MANET. For this purpose we are focusing over AODV routing protocol that can support diverse real time applications. Farkas and coauthors examine and validate four popular Ad hoc routing protocols using some QoS extensions such as priority queueing, broken link detection, timeouts and rate control policies. Based on their experimental results, they demonstrate that the AODV outdoes all other protocols including DSR, OLSR and DSDV. Therefore, we select AODV as the optimal routing protocol of choice for our research.

The conventional AODV is a source-initiated protocol which determines possible routes purely on an "as required basis". AODV protocol works in two main phases: route discovery and route maintenance. It defines five different control messages for route discovery and maintenance: Route Request message (RREQ), Route Reply message (RREP), Route Error message (RERR), Route Reply Acknowledgment (RREP-ACK) message and HELLO message (Krco and Dupcinov, 2003). Whenever data packets need to be transferred from a source to sought destination, the source starts its route discovery phase by flooding RREQ packages to its immediate adjacent nodes. These intermediate nodes further propagate the request message until it reaches the final destination. Every intermediate node either responds to the RREQ by directing the RREP message backward to the source of RREQ or relays the RREQ to its nearby nodes after incrementing the hop count. AODV exploits sequence numbers for all packets to decide whether the routing information is "sufficiently current" and to ensure loop-free routing (Bose *et al*., 2001).

As a source node issues RREQ messages to other nodes, a reverse route is automatically generated. While, the route reply packet transverses backward to the source, each single node which lies on the route generates a forward pointer to the source of RREP package, updates its route-timeout information and registers the latest destination sequence number for the sought destination (Syarif and Sari, 2011). The established viable route is preserved as long as it is needed by the source node. The RERR message is created to inform other nodes about the route failure. It specifies those destinations which are no longer accessible through the failed link. To facilitate this reporting technique, every node maintains a "precursor list", comprising the Internet Protocol address (IP address) for all its nearby nodes that are used as a next hop for each destination (Krco and Dupcinov, 2003). Optionally, the source node of the RREQ message forwards RREP-ACK package to acknowledge the route reply message. HELLO packets can be utilized periodically to assure symmetric links and to identify the link breakages.

Motivation of our research: Spurred by the increasing interest in an endowment of QoS through routing in Ad hoc networking, a number of enhancements have been evolved for AODV protocol. In original AODV, fixed priority scheduling is used to arrange the packets in a queue and the drop tail policy is used to manage its buffer space. It discards packets from the rear of the queue when the buffer is overloaded. The AODV protocol always gives higher priority to its routing control packets over data packets and they are maintained in separate queues. Nevertheless, due to the dynamic behavior of real time packets and network condition, the static predefined scheduling algorithms are no longer effective in real time operations. Regrettably, the majority of algorithms do not consider the dynamics of the network due to node migration which will create a vast disparity between actual execution and an assessed performance. To address this issue, we investigate how to explore the network dynamics, how to regulate its effects on scheduling efficiency and how to meet the QoS demands without forfeiting the system performance. We propose an urgency aware packet scheduling with priority based buffer management architecture to provide an enhanced QoS for delay sensitive real time applications. Here, the urgency of the packet is defined as the amount of time left to accomplish its communication on time. The proposed approach in this study considers the inherent uncertainties associated with the network layout and hop count involved in the routing process. The major contributions of this research that can handle the dynamics of network behavior are as follows:

- We use interval number theory (Sengupta and Pal, 2009) to model the abrupt fluctuations in the topology of the network
- We introduce an urgency aware packet scheduling algorithm with priority based buffer management policy for highly delay sensitive real time tasks
- We embed our proposed architecture in AODV protocol to integrate the potential benefits of both AODV and UaPS algorithm
- We rigorously provide experimental verifications of the proposed UaPS-AODV protocol based on arbitrarily generated packets

Survey of qos routing protocols: The increasing growth of multimedia applications in mobile traffic has resulted in a shift of research interests towards better degrees of QoS rather than the best effort service. Designing of QoS routing algorithm for MANET is much more difficult and a challenging task because it has to deal with critical circumstances such as dynamic topology frequent link failures and energy constraints. In the past years, considerable efforts have been devoted to compact these challenges which led to the emergence of several routing strategies which have been suggested, aiming to augment different QoS parameters but no particular protocol affords an overall solution. The purpose of this section is to present an overview of the existing enhancements in original AODV to provide improved QoS.

Sedrati *et al.* (2011) propose a modified variant to conventional AODV (M-AODV) protocol to enhance the QoS in MANET which exploits an adaptive multipath concept for providing improved packet delivery ratio. When there is a link failure, the source node selects alternative routes for packet propagation. Hence, M-AODV can significantly lessen the packet drop ratio by means of alternative paths for all source-destination pairs against link failure or node failure.

Another extension to AODV protocol is proposed by Boshoff and Helberg, called Delay Aware AODV-Multi-path (DAAM) (Boshoff and Helberg, 2008). The proposed DAAM algorithm selects the route based on packet latency rather than the number of hops. Multiple back up paths along with the latency for every route are gathered in the routing table. If there is a path failure, the algorithm determines a new alternate path to the final destination before a new route acquisition process is started. Hence, DAAM offers a considerable reduction in the packet latency, jitter and routing overhead.

Tabatabaei and Tabatabaei (2012) develop a novel QoS routing algorithm, namely QOS AODV (QAODV) which considers parameters including the speed of the nodes, bandwidth, the RADIO-RX-SENSITIVITY, the RADIO-ANTENNAGAIN, battery power and PROPAGATION-LIMIT along with the hop count used by conventional AODV protocol. The proposed algorithm calculates the reliability of a viable path by combining all the seven metrics with weighing factors. Simulation results revealed that the proposed algorithm enhances performance measures such as PDR, end-to-end latency and the fault tolerance of the network considerably.

Yi *et al.* (2013) investigate the effect of jitter in route discovery procedure of AODV. They develop a modi?ed jitter distribution, called window jitter which considers the quality of communication links to estimate the jitter before retransmission of RREQ by intermediate nodes. So control messages are propagated quicker over optimal routes. Experimental results indicate that the utilization of window jitter improves the performance of the route acquisition process of AODV and overcomes the disadvantages recognized for “naive” jitter.

Veerayya and coauthors propose a stability-based, QoS-capable AODV (SQ-AODV) protocol which uses a cooperative cross-layer approach between physical, network and application layers. SQ-AODV exploits the state information about the residual energy of the node for path discovery and maintenance. Simulation results demonstrate that the improved route stability provided by SQ-AODV leads to better QoS in terms of delay and PDR as compared to original AODV.

Sharma *et al.* (2012) propose weight hop based packet scheduling technique for AODV in which the intermediate node initiates the packet scheduling and handles its buffer memory based on the data transfer rate. In this technique, data packets with fewer hop count to reach their destinations are granted higher weight. If there is a loss of link connectivity, the relay node buffers the data packets and recovers the failure route. This modified AODV protocol tries to transfer the data packet through alternative paths instead of discarding it.

Othmen *et al.* (2014) suggest a Power and Delay aware Multipath Routing Protocol (PDMRP) which is not only to discover more consistent routes based on the remaining lifetime of battery but also to discover multipath that guarantee QoS requirements in terms of delay and bandwidth .The simulation results reveal that the PDMRP considerably outdoes the Stable Path Routing Protocol based on Power Awareness (SPR) and the Modified Ad hoc On-Demand Distance Vector (MAODV) routing protocols based on throughput, latency and packet drop rate (Othmen *et al.*, 2014). Macharla *et al.* (2008) develop an on demand delay based QoS routing protocol (AODV-D) to guarantee that delay does not surpass a max threshold value. In addition to minimum hops, AODV-D will consider the channel contention information and number of packets buffered in the interface queue (Ifq). From simulation studies, they demonstrate that AODV-D outperforms original AODV under average mobility and traffic load.

MATERIALS AND METHODS

Modeling of proposed system architecture: In this study, we first discuss the interval number theory (Sengupta and Pal, 2009) to define the network dynamics and then we design a scheduling architecture for real time packets. According to this proposed design, we provide our packet scheduling algorithms.

Interval number theory: Generally, the interval number theory is applied to define uncertain information and decision making variables (Sengupta and Pal, 2009). The interval number theory and its associated arithmetic operations which will be useful in the scheduling process are detailed as follows.

Definition 1: Consider R is the set of all real numbers and u^-, $u^+ \in R$. An interval number is designated as: $\bar{u} = [u^-, u^+] = \{a|u^- \leq a \leq u^+, a \in R\}$ where u^- is lower limit and u^+ is the upper limit of the $\bar{u}$. If $u^- = u^+ = a$ then $\bar{u} = [a, a] = a$ where ‘a’ is a real number. Let I(R) be the set of all interval numbers. The predicted upper and lower limits of an uncertain number are used to describe the rate of uncertainty (Sengupta and Pal, 2009).

Definition 2: (Sengupta and Pal, 2009) If $\bar{u} = [u^-, u^+]$, $\bar{u} = [u^-, u^+] \in I(R)$ then performing arithmetic operations such as addition $\oplus$ and subtraction $\bar{u}$ of two interval numbers are stated as follows:

$$\bar{u} \oplus \bar{u}\left[u^- + u^-, u^+ + u^+\right] \quad (1)$$

$$\bar{u}\bar{u}\left[u^- - u^-, u^+ - u^+\right] \quad (2)$$

Implication 1: Given a set of interval numbers: $u_1 = [\bar{u}_1^-, u_1^+]$, $u_2 = [u_2^-, u_2^+]$, $u_3 = [u_3^-, u_3^+]$, ..., $u_n = [u_n^-, u_n^+]$ then the summation of these interval numbers is:

$$\sum_{i=1}^{n}(\bar{u}_i) = \sum_{i=1}^{n}(\bar{u}_i, u_i^+) = \left[\sum_{i=1}^{n}(\bar{u}_i), \sum_{i=1}^{n}(u_i^+)\right] \quad (3)$$

From implication 1, we can understand that the interval of $\sum_{i=1}^{n}(\bar{u}_i)$ is the summation of the interval of every interval number. Hence, the rate of uncertainty rises as their amount of interval numbers rises.

Definition 3: If $\bar{u}$ = $[u^-, u^+]$, $\bar{u}$ = $[u^-, u^+] \in I(R)$ then the multiplication ⊗ and division ∅ operations are stated as follows (Sengupta and Pal, 2009):

$$\bar{u}\otimes = \begin{bmatrix} \min\left(u^-u^-, u^-u^+, u^+u^-, u^+u^+\right), \\ \max\left(u^-u^-, u^-u^+, u^+u^-, u^+u^+\right) \end{bmatrix} \quad (4)$$

$$\bar{u}\oslash\bar{u} = \left[u^-, u^+\right] X \left[1/u^-, 1/u^+\right], \text{ provided } 0\left[u^-, u^+\right] \quad (5)$$

By exploiting interval number theory, we describe the dynamic parameters of MANET which will be detailed in the following sections.

A motivational example: Before discussing our approach in detail, we first present a running example to demonstrate the usefulness of our algorithm. We assume a MANET of n communicating mobile nodes. In our research, we emphasize on independent packets, represented as $\Gamma = (r_1, r_2, r_3, ..., r_n)$. For any packet $r_i \in \Gamma$, it can be characterized by four-parameter tuple $r_i = (P_i, TTL_i, r_i, a_i)$ where P_i, TTL_i, r_i and a_i represents the priority (in our simulation P_i =1-4), time to live, release time and arrival time of packet r_i. The time to live specifies the lifetime of the packet within which every packet should be delivered at the destination node or else destroyed. Assume 9 packets r_1-r_9 from four different sources A, B, C and D to be transferred through the intermediate node E to various destinations as shown in Fig. 1. The attributes of incoming packets are given in the Table 2. Assume there is no valid route from the node E to F initially. Now, E buffers the incoming packets and begins its route discovery process. After discovering the route, E propagates scheduled packets from its queue along the viable route towards the sought destination.

Traffic arriving at any node in the network is mixed with real time and best effort packets. The packet classifier or policy database in network devices (e.g., firewalls, routers and intrusion detection systems) are used to classify received packets into one of the four categories and prioritizing them as shown in Fig. 2. Conventionally, there are three types of packet classification methods are available in practice.

In port based traffic classification, network devices use the port numbers in the User Datagram Protocol (UDP) header of source and destination nodes to recognize the type of packets (Cai *et al.*, 2010). Recognizing applications through well-known port numbers is no longer trustworthy because modern applications use unpredictable dynamic port numbers.

Table 2: Packets arrival at node E in an interval of $[a_1, a_3]$

Packet	Priority	TTL (msec)	Packet arrival time	Packet release time
r_1	P_1	200	a_1	r_1
r_2	P_4	7000	a_2	r_2
r_3	P_2	300	a_1	r_3
r_4	P_1	150	a_3	r_4
r_5	P_3	2000	a_1	r_5
r_6	P_1	150	a_2	r_6
r_7	P_1	150	a_1	r_7
r_8	P_3	5000	a_1	r_8
r_9	P_4	8000	a_2	r_9

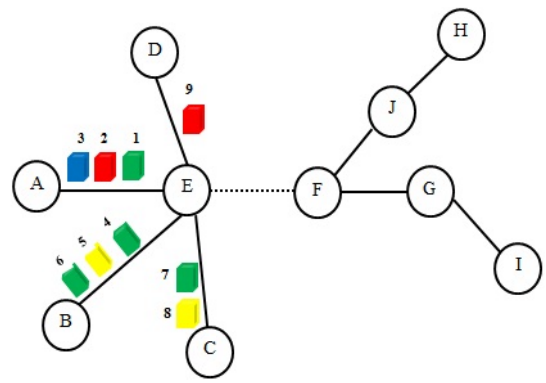

Fig. 1: Flow of packets through network

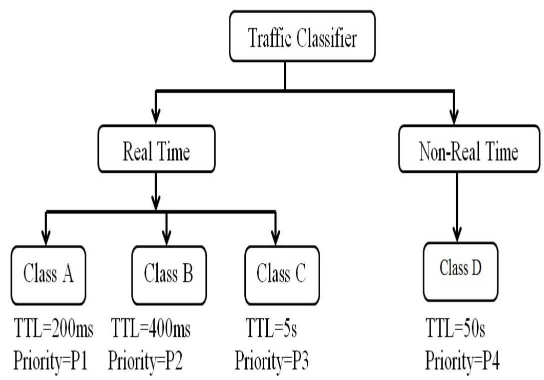

Fig. 2: Packets classification

In IP based taxonomy, classifiers look at the explicit labelling of the Type Of Service (TOS) field in the IP version 4 (IPv4) header or the traffic class field in the IP version 6 (IPv6) header (Bhatti and Crowcroft, 2000). When a packet arrives at a router, its header information is compared to a set of rules or packet filters. Each filter contains one or more fields and their value, a priority and a packet handling action to be triggered if matched. A packet is considered matching a rule only if it satisfies all the fields within that filter (Sun *et al.*, 2011).

Besides the fixed labelling approaches using network layer or transport layer header information matching, an approach such as Resource Reservation Protocol (RSVP) for dynamic classification can be exploited.

In our experiments, we use an IP based packet classification method. Packets from typical delay sensitive applications such as Voice over IP, Video Conference, Video Broadcasting, Audio Broadcasting and Internet Relay Chat having delay ≤200 m sec are marked as Class A (High priority) packets. All data packets from Telnet, Telemetry, Web browsing and AODV control packets having delay in the range of 200-400 m sec are marked as Class B (Medium priority) packets. Packets from applications such as File Transfer Protocol (FTP) and e-mail having delay in seconds are marked as Class C (Low priority) packets. All other non-real time packets which have no specific requirements are marked as Class D (best effort) packets.

As mentioned, packet classification is achieved through setting and identifying the three most significant bits (IP precedence bits) of TOS octet in an IP datagram packet. For example with Cisco VoIP router, programmers use voice dial peers to categorize the VoIP packets and set the precedence bits. The following configuration program illustrates how to set the IP precedence bits:

Classification and marking of VoIP Packets:

- Dial-peer voice 100 viop
- Destination-pattern 100
- Session target ipv4:10.10.10.02
- Ip precedence 3

Using the above set-up program, any VoIP packets that matches the specified dial-peer voice 100 VoIP instruction will be marked with IP precedence as 3 (i.e., 011 in binary) and are considered as the highest priority packets. The value of precedence bits for applications used in our simulation is shown in Table 3. Every hop in the network is able to identify the type of packet and its priority.

The current packet scheduling algorithms in AODV sort the packets based on their priority only and just ignore other factors such as time to live and urgency which can significantly affect the quality of real time tasks. At node E, the AODV protocol schedule the arriving packets given in Table 2 in its priority queue as shown in Fig. 3. The same priority packets are sorted based on their time of arrival (first in first out basis).

Consider the urgency (the amount of time leftover to reach the intended destinations) of packets r_1 and r_4 are 120 and 45 m sec, respectively. The most urgent packet r_4 is waiting in a queue for a longer time than the non-urgent packet r_1 as shown in Fig. 3. To tackle these issues, an urgency aware routing protocol and an intelligent buffer management policy to schedule the packets are needed.

Table 3: Value of IP precedence bits for various classes of traffic

Type of Service	Priority	IP precedence bits
Class A	1	011
Class B	2	010
Class C	3	001
Class D	4	000

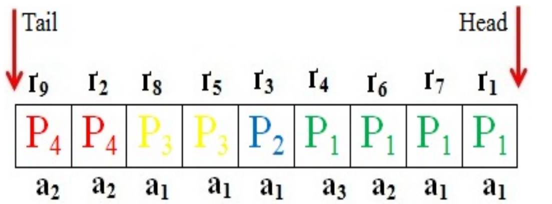

Fig. 3: Sheduling of packets in table 1 based on priority only based algorithm

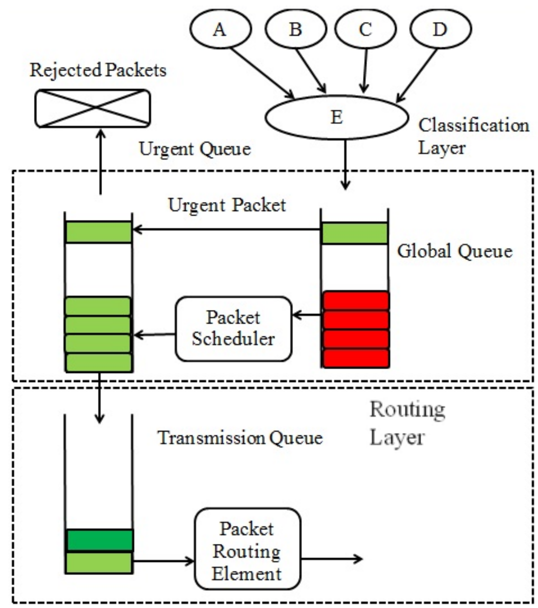

Fig. 4: Urgency aware Packets Sheduling (UaPS) architecture

System architecture and scheduling model: The scheduling architecture developed in the proposed research is shown in Fig. 4. This architecture is forked into three layers: Packet classification layer, scheduling layer and routing layer. As discussed earlier, the packet classifier in the network apparatus categorizes received packets into distinct classes and assigns a priority such that each class may be provided different services.

The scheduling layer comprises of a Global Queue (GQ), an Urgent Queue (UQ) and a packet scheduler. The GQ consists of non-urgent waiting packets to be scheduled, whereas the UQ accommodates all the urgent packets that must be sent to the Transmission Queue (TQ) immediately. Packet scheduler is responsible to decide which packet is serviced next amongst the packets in urgent and global queues. The routing layer consists of a routing agent (AODV) and a Transmission Queue (TQ). The routing agent is responsible for routing the packets towards the intended destination. The TQ holds the servicing packet and utmost only one waiting packet is to be routed.

Problem formulation: Consider the current node E. Assume that parameters with subscript "e" represent that they are for node E. To simplify the investigation, the propagation delay is ignored since it is constant for a given route. Therefore, for a path consisting of n nodes (from source (s) to destination (d)), the following relation can be derived for the arrived packet at node e:

$$d_i = \sum_{n=s}^{e-1}(d_{in}) + (r_{ie} - a_{ie}) + \sum_{n=e+1}^{d}(d_{in}) \quad (6)$$

$$d_i = d_{ie\_before} + d_{ie\_at\,node\,e} + d_{ie\_after} \quad (7)$$

In the aforementioned equation d_i is the total delay experienced by the packet r_i to reach its destination d, $\sum_{n=s}^{e-1}(d_{in})$ is the delay that the packet r_i has already experienced before it arrives at intermediate node e which hinges on the number of hops in the route from the source to node e. The term (r_{ie}-a_{ie}) denotes the actual delay that the packet r_i experiences at node e where a_{ie} and r_{ie} denote arrival and release time of packet r_i. $\sum_{n=e+1}^{d}(d_{in})$ is the delay that the packet will experience after it leaves node e that depends on the remaining distance from node f to the required destination. Since the number of hop (h_i) in the path from e to destination is not predictable before scheduling, the parameters d_i and h_i are unpredictable. So the interval number is used to define these parameters (e.g., $d_i = [d_i^-, d_i^+]$ and $h_i = [h_i\text{-}, h_i^+]$):

$$d_{ie\_after} = \left[di^-, di^+\right] \otimes \left[hi^-, hi^+\right] \quad (8)$$

$$d_{ie_{after}} = [\alpha, \beta] \quad (9)$$

Where, $\alpha = \min\ (di^-hi^-, di^-hi^+, di^+hi^-, di^+hi^-)$ and $\beta = \max\ i^-(i^-hi^-, di^-hi^+, di^+hi^-, di^+hi^+)$. The delay experienced by the packet r_i in the remaining path from node e to destination can be estimated by setting proper values for the upper and lower bounds of d_i and h_i. The delay of packet r_i at node e depends upon the number of equal and smaller urgency packets already waiting in the queue. These parameters are also not predictable over time. So, the interval number is used to estimate the delay experienced in the current node:

$$d_{i-at\,node\,e} = \left[ri^-, ri^+\right] \theta \left[ai^-, ai^+\right] \quad (10)$$

$$d_{i-at\,node\,e} = \left[ri^- - ai^-, ri^+ - ai^+\right] \quad (11)$$

Every packet from real time applications has a level of urgency. This urgency is estimated from its TTL_i, packet release time from the source and current time as shown in Eq. 12 and 13:

$$d_{i\_before} = ct - r_{is} \quad (12)$$

Where, r_{is} of packet r_i is the release time from the source and ct is the current time. To identify the packet r_i with the lowest delay, we define the urgency of a packet, denoted by U_i to be:

$$U_i = TTL_i - d_i \quad (13)$$

Compared to the priority-only-based packet scheduling the proposed urgency aware scheduling algorithm can reveal the real significance of the packet in terms of its delay by considering its urgency. At every node, the packet with minimum urgency is selected for service first. If U_i<0, (i.e., negative), the packet r_i cannot be effectively served before its expiration time. A packet becomes urgent (or critical) when it's urgency:

$$U_i < U_{th} \quad (14)$$

In Eq. 14 U_{th} denotes a predefined threshold value. If threshold urgency is extremely small, very few packets are dispatched to the UQ directly and GQ becomes overloaded frequently. Alternatively if the threshold is very large, most of the incoming packets are dispatched to the UQ and will cause buffer overflow at UQ. In our simulation, we consider 200 m sec as threshold urgency. For example, from Table 1 the packets from applications having delay ≤200 m sec are considered as urgent packets initially (e.g., Voice over IP, video conference, video broadcasting, internet relay chat etc.). The non-urgent packets in GQ might become urgent over time.

Scheduling process: The summary of the scheduling procedure for the aforesaid design is discussed in this study. When a packet from non-urgent applications arrives at a particular node, it will be buffered into the global queue and all the packets in GQ are ordered in a non-descending order based on their urgency. Upon arrival of a critical packet or any packet in GQ turns out to be critical over time, then it will be transferred to UQ immediately. Subsequently, the packets in UQ will be transferred to TQ as servicing or waiting packet directly. Packets in GQ will be scheduled when a routing layer routes a packet successfully. On successful routing of the preceding packet, the waiting packet on the TQ is transferred immediately and the selection of a new waiting packet for the transmission queue from GQ will be executed. The elite characteristic of our proposed system is that most of the packets are waiting in the GQ instead of waiting on the TQ of AODV and utmost only one packet is permitted to wait on the TQ. The virtues of our proposed design are synopsized as follows.

This design prevents dissemination of network dynamics across the schedule. From implication 1, it is understood that as the count of waiting packets raises, the rate of uncertainty of waiting packets become greater which will create a considerable impact on the effectiveness of scheduling as well as routing. So, it is essential to regulate the number of packets waiting on TQ to prohibit the circulation of uncertainties.

It permits every packet waiting for routing on the transmission queue to start immediately if it's previous packet has been routed successfully, so the expected queueing delay for a newly arrived packet is eliminated.

This model allows concurrent execution of scheduling and routing processes. When AODV is routing a packet and the TQ is empty, the scheduler can concurrently accept another packet as a waiting packet. Hence, scheduling and routing processes are potentially overlapped to reduce delay and give a significant impact on QoS constraints of delay sensitive applications. It also can decrease the routing overheads of packet transfer among nodes.

Here, an assignment variable A_i is introduced to represent the transferring packets to transmission queue. The value of A_i is 1 when packet r_i is transferred to TQ, otherwise, A_i equals 0:

$$A_i = \begin{cases} 1 & \text{if packet ri is transferred to TQ} \\ 0 & \text{otherwise} \end{cases} \qquad (15)$$

The releasing time of the packet from node E (r_{ie}) depends upon the number of equal and smaller urgency packets already waiting in the queues. These parameters are also not predictable before scheduling. So, the interval number is used to define the release time of packet r_i from the current node E.

$$r_{ie} = \left[d_{ie}^-, d_{ie}^+ \right] \oplus \left[a_i^-, a_i^+ \right] \qquad (16)$$

$$r_{ie} = \left[d_{ie}^- + a_i^-, d_{ie}^+ + a_i^+ \right] \qquad (17)$$

In addition, let r_{ie}^{act} is the actual release time of the packet r_i which can be any value in the interval of r_{ie}. For example if the predicted release time of the packet r_i is r_{ie} = [150, 250] ms before scheduling; the actual release time r_{ie}^{act} should be between 150 and 250 msec (e.g., 180 msec). Under the dynamic scheduling environments, the actual release time decides whether the timing constraint of the packet has been assured or not. Therefore, a variable X_i is defined to decide whether the timing constraint of packet r_i has been satisfied or not. If a packet r_i is transferred to TQ and its r_{ie}^{act} is less than its TTL_i then the timing constraint of that particular packet is assured as defined in Eq. 18:

$$Xi = \begin{cases} 1 & \text{if}\left(\text{actual release time} < \text{TTLi}\right)\text{and}\left(\text{Ai} = 1\right) \\ 0 & \text{otherwise} \end{cases} \qquad (18)$$

The most important optimization objective of our research is to enhance the success ratio of packet delivery as much as possible which can be denoted as:

$$\text{Maximize} \sum_{i=0}^{m} \frac{X_i}{m} \qquad (19)$$

where, m indicates the number of packets submitted to a particular node in a specified interval.

Algorithm design: In this study, we describe our novel urgency aware packet scheduling algorithm for dynamic ad hoc environments. The following events are considered as triggers: a new packet arrives; the packet just has been routed by the routing agent; the queues become overloaded and a new urgent packet comes or packets in the GQ turn out to be urgent. These occurrences happen arbitrarily and if any one of the events takes place, the scheduling process will be triggered automatically. We consider the following postulations to implement our UaPS-AODV protocol.

- All the data packets received at the queue are of equal size
- Every node can route only one packet at any instance
- When a packet cannot be serviced before its expiration time, it should be discarded
- The urgent packets will have superior rights. An urgent packet is permitted to swap non-urgent waiting packets as a new packet in UQ and the swapped non-urgent packet will be returned to GQ. Figure 5 describes an example of this property

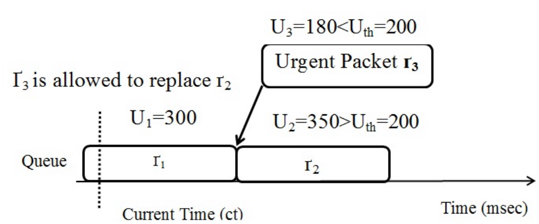

Fig. 5: The illustration of postolation 4

- Since, the scheduling and routing processes are overlapped, a packet transfer time from scheduler to TQ is negligible and has no impact on the releasing time of the packets
- The waiting packet on TQ is permitted to route immediately when the servicing packet on the routing agent has finished
- All mobile nodes in the network use this scheduling algorithm for routing of packets

In Fig. 5, consider the packet r_2 is waiting in the urgent queue. The waiting task r_2's urgency μ_2 is 350 m sec that is greater than the threshold urgency U_{th} = 200 m sec and the task r_3's urgency U_3 is 180 m sec that is $<\mu_{th}$ = 200 m sec. Now, r_3 is considered as an urgent packet as compared to r_2. According to Postulate 4, the urgent packet r_3 is permitted to swap the non-urgent packet r_2 and r_2 is returned to the GQ. Subsequently, AODV routes urgent packets earlier, increasing the probability of satisfying the QoS demands of applications. The UaPS executes the following procedures on the arrival a new packet as given in algorithm 1.

Algorithm 1: UaPS-When a new packet arrives:

For every new packet r_i do
If $U_i < U_{th}$ then
UQ←r_i
    Sort all the packets in UQ by their urgency U_i in an ascending order
else
    GQ←r_i
  sort all packets in the GQ by U_i in an ascending order
    If the waiting packet in TQ = = Null then
        If UQ! = Null then go to step 11
          UQ←get the packet at the head in GQ
          move packet r_i from UQ to TQ as the routing packet
        AODV←search Waiting Packet ()
        end if
      end if
    end if
  end for

On the arrival of a new packet r_i, the algorithm UaPS will determine whether this packet is urgent or not. If this packet is urgent, it will be buffered into the UQ directly and all the packets in UQ are ordered in the non-decreasing order (Lines: 3-4) by their urgency. Else, packet r_i will be buffered into the GQ (Line: 6) and then all the waiting packets in GQ are arranged in a non-decreasing order (Line: 7) by their urgency. Subsequently, the algorithm UaPS will move the packets from UQ to TQ as a routing packet (Line: 11). If the UQ is empty, the scheduler moves the minimum urgency waiting packet from GQ to TQ through UQ (Line: 10 and 11). After that the function SearchWaitingPacket() is called to search the next packet for routing (Line: 12) which is shown in algorithm 2.

Algorithm 2: Function search waiting packet()-on the reaction to routing of the preceding packet:

r_i←get the packet at the head in GQ
While r_i! = Null do
    estimate the U_i of packet r_i
     if U_i<0
        discard the packet
        if $0<U_i<U_{th}$
        move packet r_i from GQ to UQ
        sort all the packets in UQ by U_i in an ascending order
        else
       r_i←get the next packet in GQ
    end if
    TQ←get the packet at the head in UQ
end if
end while

In the aforesaid function search waiting packet (), shown in algorithm 2, the packet with minimum urgency will be treated first (Line:1 and 8) along with a condition (Line: 4) to satisfy the real time requirements. If a packet is expired (i.e., the urgency is negative) it will be discarded and this function will get the next packet from GQ (Line: 5 and 10). If the urgency of the packet is less than Uth, it is moved into UQ (Line: 6 and 7). After sorting of UQ in a non-decreasing order based on urgency (Line 8), the most urgent packet is moved into TQ as a waiting packet (Line: 12).

The UaPS algorithm uses priority based buffer management approach to avoid queue overflow. To prevent the drop of significant higher priority packets, we intelligently drop the lower priority packets as shown in algorithm 3.

Algorithm 3: UaPS-On overloading of GQ:

For every new packet r_i do
Find the priority of the incoming packet
Calculate the current length of queue (q_current)
if (q_current >= 95% of actual queue)
    drop the incoming packet r_i if (P_i = 4 or P_i = 3 or P_i = 2)
    else
    if (q_current >= 85% of actual queue)
    drop the incoming packet r_i if (P_i = 4 or P_i = 3)
    else
    if (q_current >= 75% of actual queue)
    drop the incoming packet r_i if (P_i = 4)
    else
    enqueue the packet
    end if
end if
end if
end for

If the current queue length is $\geq$75% of the total queue size; the incoming packet with priority P4 is dropped (Line 10). If the current queue length is $\geq$85% of the total queue size, the incoming packet with priority P3 or P4 is dropped (Line 7). If the current queue length is $\leq$95% of the total queue size, the incoming packet with priority P4 or P3 or P2 is dropped (Line 4). All of the dropped packets will be the packets with low priorities.

Whenever, there is an overflow in UQ, packets are returned to the GQ. The algorithm is used to avoid buffer overflow in UQ as shown below.

Algorithm 4: UaPS-on overloading of UQ:

```
for every new urgent packet ´i do
find the priority of the incoming packet
calculate the current length of queue (q_current)
if (q_current ≥95% of actual queue)
        return the incoming packet ri to GQ if (Pi = 4 or Pi = 3 or Pi = 2)
        else
if (q_current ≥85% of actual queue)
        return the incoming packet ri to GQ if (Pi = 4 or Pi = 3)
        else
        if (q_current ≥75% of actual queue)
        return the incoming packet ri to GQ if (Pi = 4)
        else
        enqueue the packet
        end if
end if
end if
end for
```

If the current queue length is ≥75% of the total queue size, the incoming packet with priority P4 is returned (Line 10) to GQ. If the current queue length is ≥85% of the total queue size, the incoming packet with priority P3 or P4 (Line 7) is returned to GQ. If the current queue length is 95% of the total queue size, the incoming packet with priority P4 or P3 or P2 (Line 4) is returned to GQ.

RESULTS AND DISCUSSION

Implementations: We implement UaPS-AODV protocol in NS-2.34 and quantitatively appraise its performance by comparing with basic AODV (Boshoff and Helberg, 2008) and AODV-D (Tabatabaei and Tabatabaei, 2010). To implement our proposed protocol, the buffer management scheme and scheduling algorithm of basic AODV are modified. The implementation of UaPS-AODV protocol in the network simulator can preserve a queue size of 64 packets during its path acquisition phase. We select the upper limit for the waiting time of each packet is 50 sec. If a packet waiting in the GQ for over 50 s then it will be discarded. The lower and upper limits of hop count are selected as 1 and 50, respectively. The minimum and max release time is considered as 1 m sec and 50 sec, respectively. In our experiment, we create four different sizes of the network by changing the number of mobile nodes to 20-50. We use three different mobility scenarios by changing the pause time to 5, 10 and 20 sec with velocity of min 2 m $\sec^{-1}$ and max 10 m $\sec^{-1}$. The packet inter-arrival time is considered as 20 m sec. All mobile terminals can transmit data with the length of 512 bytes and at a rate ranging from 5-20 packets per second. Other simulation parameters organized for our experiment are given in Table 4.

Table 4: NS-2.34 parameter setting

Parameters	Values
Transmission range	250 m
Area	1500×1500 m
Time	50 sec
Traffic type	60% real time+40% non-real time
Routing protocol	AODV, AODV-D and UbPS-AODV
Link layer type	LL
MAC type	802.11
Data payload	512 bytes
Interface queue	Drop tail/priqueue
Maximum packets in Ifq	50
Antenna model	Omni directional
Radio propagation model	Two ray ground
Channel type	Wireless channel

Performance metrics: To evaluate the performance of the UaPS-AODV against basic AODV and AODV-D, the following QoS metrics are utilized.

Average throughput: Average throughput is the number of bits arrived at the intended destination successfully in the given time:

$$\text{Average throughput} = \frac{\sum(\text{No. of bits recieved})}{\sum(\text{Tranmission time})} \tag{20}$$

The results are obtained from the simulation is as shown in the following Table 5. To validate the throughput performance of UaPS-AODV, a traffic environment with different real and non-real time packets is generated. For the network size of 20 nodes, the average throughput of AODV and AODV-D are 367.71 and 392.74, respectively. The average throughput of UaPS-AODV is 428.03 which is greater than AODV and AODV-D by 16.41 and 8.99%, respectively. For the network size of 30 nodes, the average throughput of AODV and AODV-D are 403.70 and 434.08, respectively. The average throughput of UaPS-AODV is 455.24 which is greater than AODV and AODV-D by 12.76 and 4.87%, respectively.

For 40 nodes, the average throughput of AODV and AODV-D are 665.92 and 692.73, respectively. The average throughput of UaPS-AODV is 716.01 which is greater than AODV and AODV-D by 7.52 and 3.36%, respectively. For 50 nodes, the average throughput of AODV and AODV-D are 589.02 and 615.61, respectively. The average throughput of UaPS-AODV is 629.71 which is greater than AODV and AODV-D by 6.91 and 2.29%, respectively.

From the above results, the following graph is plotted for all scenarios. With the help of Fig. 6, it is observed that the UaPS-AODV has better throughput performance as compared to other existing protocols. It is concluded that UaPS-AODV provides an average throughput 10.01% higher than AODV and 4.39% higher than AODV-D.

Table 5: Result on average throughput

	No. of nodes 20			No. of nodes 30			No. of nodes 40			No. of Nodes 50		
Protocol	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20
AODV	298.67	427.57	376.89	378.85	387.37	444.89	683.67	497.49	816.59	534.44	527.69	704.93
AODV-D	338.32	471.14	368.76	379.13	391.85	531.28	687.96	554.86	835.38	557.18	536.08	753.58
UaPS-AODV	386.94	502.47	394.69	401.58	407.89	556.27	694.19	586.24	867.59	580.37	542.85	765.92

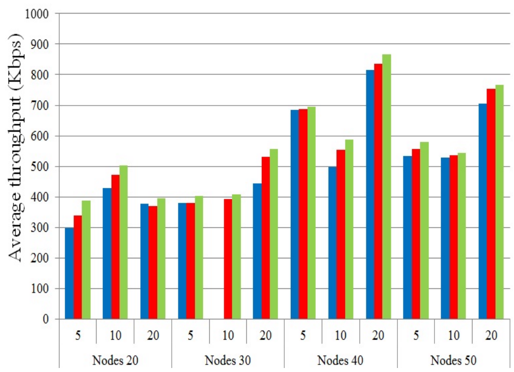

Fig. 6: Average throughput with variable pause time

End-to-end transmission Delay (ED) or latency: For hard real time applications, ED or latency is considered as a primary concern used to evaluate the performance. Latency of the packet is the mean time required to achieve its end-to-end transmission (from a source to the required destination). It comprises all the potential delays from the instant of the packet generation to the instant of its reception. The lesser value of ED reflects the enhanced performance of protocol. In this case queueing delay of the real time packet is reduced significantly which in turn reduces the overall transmission delay. ED is calculated using the following Eq. 21:

$$ED = \frac{\sum(\text{Arrival time} - \text{Release time})}{\sum \text{No.of connection}} \quad (21)$$

The obtained simulation results are given in Table 6. Latency of AODV for the network with 20 mobile terminals is 219.98, for AODV-D is 206.48 whereas the latency of UaPS-AODV is 181.67 which is 17.41% lesser than AODV and 12.01% lesser than AODV-D. For the network size of 30 nodes, the latency of AODV 246.29 for AODV-D is 215.95 whereas for UaPS-AODV, it is 199.63 which is 18.94 % lesser than AODV and 7.56 % lesser than AODV-D. For the network with 40 mobile nodes, the latency of AODV is 288.30, for AODV-D, it is 256.36 whereas for UaPS-AODV is 239.40 which is 16.96% lesser than AODV and 6.61% lesser than AODV-D. For the network with 50 nodes, the latency of AODV is 284.68, for AODV-D, it is 259.83 whereas for the UaPS-AODV, it is 241.61 which is 15.13 % lesser than AODV and 7.01 % lesser than AODV-D.

The proposed algorithm has significant improvement in latency by protecting urgent data from the competition of non-urgent packets. Since the scheduling algorithm takes into account the urgency of packets the performance of this scheme is better than AODV and AODV-D. Figure 7 indicates the latency comparison between the proposed scheme and existing protocols. The latency under varying pause time for each of protocol studied is given in Fig. 7. Hence, it is concluded that the latency of UaPS-AODV is 17.02% lesser than AODV and 8.12% lesser than AODV-D. This feature of the proposed approach makes our algorithm very much appropriate for delay sensitive multimedia applications.

Packet Delivery Ratio (PDR): The fraction of the number of packets reached at the destinations divided by the number of packets transmitted from a source node is called as PDR.

It is a significant parameter as it reflects the drop rate of the packets which will further influence the max bandwidth of the network. It estimates the loss rate and also defines both the accuracy and the effectiveness of network layer protocols. High PDR is always anticipated in any network. The following equation is used to calculate this ratio:

$$PDR = \frac{\text{Number of recieved packets}}{\text{Number of transmitted packets}} \quad (22)$$

The results shown in Table 7, illustrates that as the pause time and number of node increases, the PDR of all examined protocols decreases. This is owing to the fact that the higher node mobility and the larger size network bring more possibility of loss of connectivity. It is seen from Table 7, our UaPS-AODV can deliver >11.60%of packets as compared to AODV protocol and >2.43% of packets as compared to AODV-D in a network with 25 nodes. For the network with 30 nodes, the PDR of UaPS-AODV is improved by 8.83% as compared to AODV and 2.01% as compared to AODV-D. For the network with 40 nodes, the PDR of UaPS-AODV is improved by 10.22% as compared to AODV and 3.37% as compared to AODV-D. For the network with 50 nodes, the PDR of UaPS-AODV is improved by 11.75% as compared to AODV and 3.13% as compared to

Table 6: Result on end-to-end delay

	No of nodes 20			No of nodes 30			No of nodes 40			No of Nodes 50		
Protocol	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20
AODV	184.27	228.65	247.01	236.88	218.42	283.56	216.98	368.47	279.46	283.46	294.16	276.42
AODV-D	169.29	212.49	237.66	221.07	192.45	234.32	191.34	323.52	254.21	253.72	260.29	265.48
UaPS-AODV	142.11	192.22	210.69	198.30	179.91	220.68	176.48	304.86	236.87	229.56	246.53	248.74

Table 7: Result on Packet Delivery Ratio (PDR)

	No of nodes 20			No of nodes 30			No of nodes 40			No of Nodes 50		
Protocol	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20	Pause 5	Pause 10	Pause 20
AODV	88.12	87.23	87.01	87.68	86.69	86.22	86.89	85.37	85.46	86.25	85.29	84.16
AODV-D	96.42	95.26	94.16	93.15	92.87	91.99	92.33	91.36	91.11	93.25	92.23	91.58
UaPS-AODV	98.12	97.56	97.12	95.16	94.33	94.12	95.26	94.58	94.23	96.25	95.24	94.25

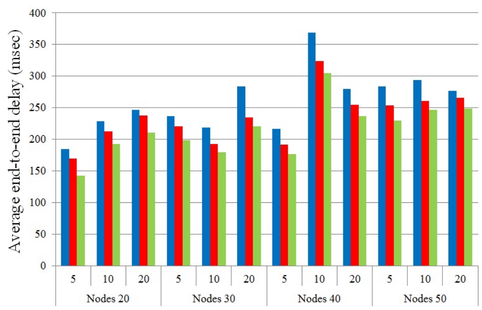

Fig. 7: Average end-to-end delay with variable pause time

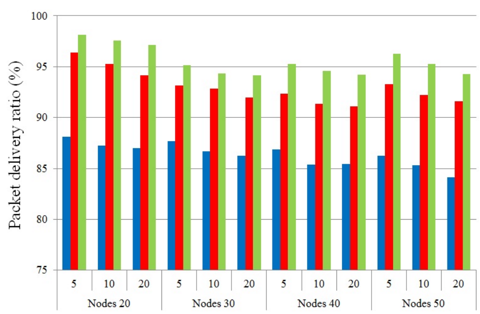

Fig. 8: Packet delivery ratio with variable pause time

AODV-D. It is concluded that the PDR of UaPS-AODV is 10.59% greater than AODV and 2.73% higher than AODV-D. This is because UaPS-AODV maximizes the ratio of packets routed before their deadlines as much as possible. The results in Fig. 8 compare the packet delivery ratio versus pause time for UaPS-AODV, AODV and AODV-D. It reveals that the PDR of UaPS-AODV is substantially better than that of AODV or AODV.

CONCLUSION

This study delves into QoS routing for MANETs with their associated challenges and opportunities. Urgency aware Packet scheduling with priority based buffer management algorithm is implemented for AODV protocol and its effect on QoS of MANET is observed. UaPS-AODV is simulated with necessary parameters and its performance is compared with previously proposed protocols. The experimental results clearly revealed that UaPS-AODV significantly outperforms the other two existing algorithms.

It guarantees minimum latency for the most urgent packets while demonstrating tolerable fairness towards lowest-priority data. UaPS-AODV delivers packets with lesser delay than AODV and AODV-D by 17.02 and 8.12%, respectively. It outperforms AODV and AODV-D by 10.01 and 4.39%, respectively, in terms of average throughput. It also achieves greater packet delivery ratio than AODV and AODV-D by 10.59 and 2.73%, respectively. It is worthwhile to point out that these features make the protocol appropriate for reliable hard real time applications in MANET.

As a future research direction, the scheme can be executed in a real-world network. Further, a method of enhancing the accuracy of estimated packet release time can be studied. It is expected that the accurately estimating packet release time leads to good scheduling decisions.

REFERENCES

Bhatti, S.N. and J. Crowcroft, 2000. QoS-sensitive flows: Issues in IP packet handling. IEEE. Internet Comput., 4: 48-57.

Bose, P., P. Morin, I. Stojmenovic and J. Urrutia, 2001. Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks, 7: 609-616.

Boshoff, J.N. and A.S. Helberg, 2008. Improving QoS for real-time multimedia traffic in ad-hoc networks with delay aware multi-path routing. Proceedings of the Symposium on Wireless Telecommunications, April 24-26, 2008, IEEE, Pomona, California, ISBN: 978-1-4244-1869-5, pp: 1-8.

Cai, J., Z. Zhang and X. Song, 2010. An analysis of UDPtraffic classification. Proceedings of the 12th Intenational IEEE Conference on Communication Tehnology (ICCT), November 11-14, 2010, IEEE, Nanjing, Cina, ISBN: 978-1-4244-6868-3, pp: 116-119.

Chand, P. and M.K. Soni, 2012. Performane comparison of AODV and DSR ON-Demand Routin protocols for Mobile ad-hoc networks. Intl. J. Comput. ppl., Vol. 49,

Giagkos, A. and M.S. Wilson, 2010. BeeIP: Be-Inspired Protocol for Routing in Mobile Ad-Hoc Neworks. In: From Animals to Animats 11. Doncieux, S., B. Grard, A. Guillot, J. Hallam and J.A. Meyer (Eds.). pringer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-62-15192-7, pp: 263-272.

Krco, S. and M. Dupcinov, 2003. Improved neigbor detection algorithm for AODV routing protocol. EEE Communi. Lett., 7:

Lim, K.H. and A. atta, 2012. An In-depth analysis of the effects of imep on tora protocol. Proceedings of the IEEE. Conference on Wireless Communications and Networking (WCNC), April 1-4, 2012, IEEE, Shanghai, China, ISBN: 978-1-4673-0436-8, pp: 3051-3056.

Lin, C.R. and J.S. Liu, 1999. QoS routing in ad hoc wireless networks. IEEE. J. Sel. Areas Commun., 17: 1426-1438.

Macharla, P., R. Kumar, A.K. Sarje and M. Misra, 2008. A QoS routing protocol for delay-sensitive applications in mobile ad hoc networks. Proceedings of the 3rd International Conference and Workshops on Communication Systems Software and Middleware, January 1-3, 2008, IEEE, Bangalore, India, ISBN: 978-1-4244-1796-4, pp: 720-727.

Othmen, S., A. Belghith, F. Zarai, M.S. Obaidat and L. Kamoun, 2014. Power and delay-aware multi-path routing protocol for ad hoc networks. Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS), July 7-9, 2014, IEEE, Jeju, ISBN: 978-1-4799-4384-5, pp: 1-6.

Ramamoorthy, H.V. and H. Karthikeyani, 2014. Hybrid routing scheme of multi agent ant based system in MANET combination of proactive and reactive. Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES), February 27-28, 2014, IEEE, Chennai, India, ISBN: 978-1-4799-3835-3, pp: 1-7.

Sedrati, M., A. Bilami and M. Benmohamed, 2011. M-AODV: AODV variant to improve quality of service in MANETs. Intl. J. Comput. Sci., 8: 429-436.

Sengupta, A. and T.K. Pal, 2009. Fuzzy Preference Ordering of Interval Numbers in Decision Problems. Vol. 238, Springer, Berlin, Germany, ISBN: 978-3-540-89914-3,.

Sharma, D.K., C. Kumar, S. Jain and N. Tyagi, 2012. An enhancement of AODV routing protocol for wireless Ad hoc networks. Proceedings of the 1st International Conference on Recent Advances in Information Technology, March 15-17, 2012, Dhanbad, India, pp: 290-294.

Sun, L., H. Le and V.K. Prasanna, 2011. Optimizing decomposition-based packet classification implementation on FPGAs. Proceedings of the International Conference on Reconfigurable Computing and FPGAs, November 30-December 2, 2011, IEEE, Cancun, Mexico, ISBN: 978-1-4577-1734-5, pp: 170-175.

Syarif, A. and R.F. Sari, 2011. Performance analysis of AODV-UI routing protocol with energy consumption improvement under mobility models in hybrid ad hoc network. Intl. J. Comput. Sci. Eng., 3: 2904-2918.

Tabatabaei, S. and K. Tabatabaei, 2012. Routing and quality of service support for mobile Ad hoc networks. Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET), April 16-18, 2010, IEEE, Chengdu, China, ISBN: 978-1-4244-6347-3, pp: 548-548.

Yi, J., J.A.C. Fuertes and T. Clausen, 2013. Jitter considerations in on-demand route discovery for mobile ad hoc networks. Proceedings of the 16th International Conference on Network-Based Information Systems, September 4-6, 2013, IEEE, Gwangju, South Korea, ISBN: 978-1-4799-2509-4, pp: 170-177.