

RESEARCH

Open Access

# An effective queuing architecture for elastic and inelastic traffic with different dropping precedence in MANET

Iyyapillai Ambika<sup>1\*</sup>, Velayudhan Pillai Sadasivam<sup>2</sup> and Perumal Eswaran<sup>3</sup>

## Abstract

Mobile *ad hoc* network (MANET) is a group of self-organized mobile nodes that are associated with comparatively low bandwidth wireless links. This paper proposes an effective queuing architecture, which supports both elastic and inelastic traffic. The packets of inelastic flows are always stored ahead of those of the elastic flows. If a link is critically loaded by the inelastic traffic, it results in large delays and elastic traffic may have some delay constraints that are nonnegligible. The virtual queue algorithm reduces the experienced delay by comprising virtual queues that are served at a fraction of the actual service rate and by using the virtual queue-length values in utility function. Then, optimization framework is used in which scheduling algorithm allocates resource fairly in the network to meet the fairness objectives of both elastic and inelastic flows. Finally, priority dropping active queue management algorithm is designed based on proportional-integral design (PID) mechanism which provides the differentiated services for the different layers or frames according to their priority.

## 1 Introduction

Mobile *ad hoc* network (MANET) is a group of self-organized mobile nodes that are associated with comparatively low bandwidth wireless links. Each node has its own area of control, which is called a cell, only within which others can receive its transmissions. In MANET, there is no fixed infrastructure [1-3]. Consequently, when nodes are free to ramble, the network topology may change rapidly and randomly over time, and nodes automatically make their own accommodating infrastructures [4]. There is a range of applications of MANET like video conferencing, rescue operations, military applications, disaster management, etc.

### 1.1 Queuing in mobile *ad hoc* network (MANET)

Queuing resolves the discipline for ordering entities in a queue. It describes the way in which resources are divided as packets and the order in which they are served. Queuing mechanisms control the transmission process by indicating which packets must be transmitted and which packets must be dropped. It affects the

packets latency qualified by finding the waiting time of the packet. Some examples of queuing are first in, first out (FIFO); last in, first out (LIFO); priority queuing (PQ); the shortest is served first; service in random order; round robin, random exponential marking (REM); etc. [5].

### 1.2 Advantages of queuing in mobile *ad hoc* network (MANET)

There are some advantages of queuing in MANET since it resolves the discipline for ordering entities in a queue and traffic distribution; packet scheduling affects the performance of multipath routing in mobile *ad hoc* networks. Therefore, queuing can reduce the resequencing delay. The packet scheduling intends to assign packets in a proper order to minimize the resequencing delay [6]. It can also evade the transmission delay and packet loss in the network. Nodes should delay the assembly of new route passing through them when their load level is high. By proper scheduling, as the channel is shared fairly, the throughput is high even at moderate mobility [7]. Scheduling can provide strict bandwidth allocation assurance since no transmission conflicts exist [8,9].

\* Correspondence: ambika0584@gmail.com

<sup>1</sup>Anna University, Chennai 600025, India

Full list of author information is available at the end of the article

### 1.3 Issues of queuing in mobile *ad hoc* network (MANET)

There are some issues of queuing in MANET. The queuing techniques play the most important factor in service differentiation. Implementation of conventional priority queuing strategy in MANET is considerably complex. Taking an example, firstly, a simple priority queue makes sure that high-priority packets are given unqualified preference over low-priority packets as proposed in the flexible quality of service (QoS) model for MANETs [10]. Secondly, they are regarded as a FIFO queue, which they improve with a mechanism called random early discard with in/out buffer management. In the same way, service differentiation in stateless wireless *ad hoc* networks [11] also theoretically employs a priority queue but confines the amount of real-time traffic for protecting the lower-priority traffic from starvation [12].

### 1.4 Elastic traffic

Elastic traffic has the ability of making adjustment of wide-ranging changes in delay and throughput across the internet and still meets the needs of the applications. It adjusts its throughput between end hosts in response to network condition. Network load or congestion may cause packet loss. Congestion occurs when the aggregated demand for a resource exceeds the available capacity of the resource [13]. To avoid this, transmission control protocol (TCP) implements its congestion avoidance algorithm and reduces the rate at which packets are sent over the network.

### 1.5 Inelastic traffic

Inelastic traffic does not easily adapt to changes in delay and throughput. Real-time multimedia (audio streaming, video, VoIP) is an example of inelastic traffic. Inelastic traffic needs special treatment, while elastic traffic could perceptibly also benefit from such treatment. In general, the quality of wireless links would be affected by many factors like collisions, fading or the noise of environment [14] [15].

### 1.6 Previous work

A token-based resource allocation technique is proposed [16] for multiservice flows in MANET. In that technique, it is assumed that the nodes cycle has three states such as noncritical section (NCS), entry section (ES) and critical section (CS). During deployment, the node is in NCS state, and after receiving the unique tokens, it enters into CS state. The scheduler sends the resource request message in different queues using fuzzy-based flow prioritization technique. If the available resource exceeds the required resource, the scheduler allocates the inelastic service similar to the available resource until the inelastic queue gets empty. Then the token is

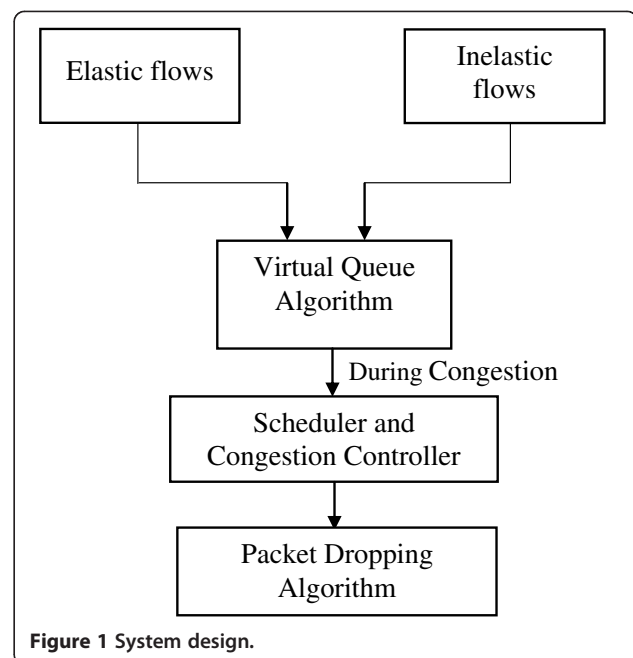


Figure 1 System design.

passed to the queue that contains elastic service flows. Based on simulation results, the proposed approach allocates the resources efficiently.

As an extension, this work proposes an effective queuing architecture that can handle both elastic and inelastic traffic flows and assign different dropping precedence for different priority of traffic. The solutions of this work prove that the proposed architecture offers better fairness and delivery ratio with reduced delay and drop.

## 2 Related work

Guo and Kuo [6] discussed about the problem connected to traffic assignment and packet scheduling in MANET.

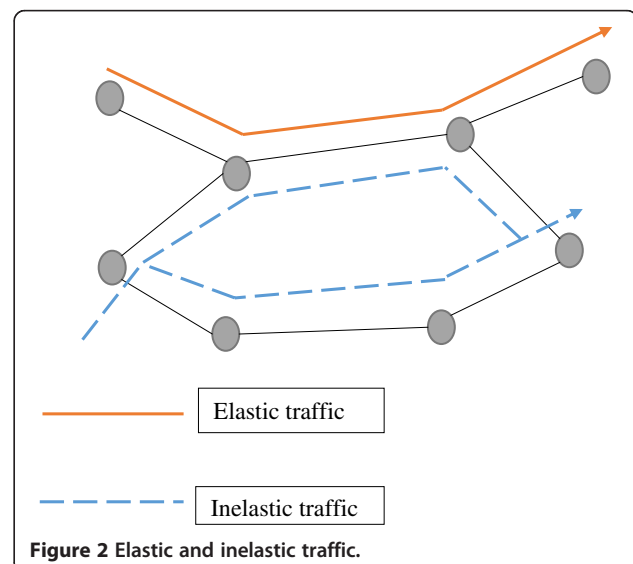


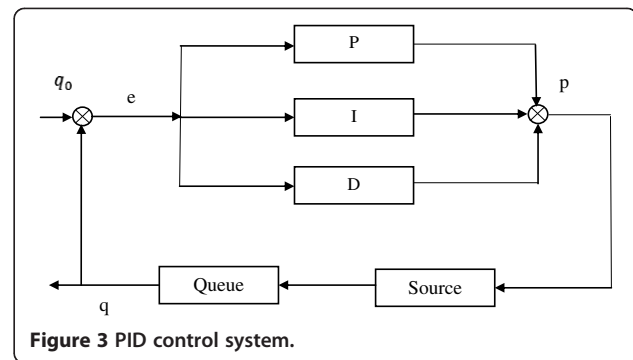
Figure 2 Elastic and inelastic traffic.

This work proposes a packet scheduling framework to study the effect of scheduling strategy on the resequencing delay. Two packet-scheduling schemes uniform round scheduling (URS) and non-uniform round scheduling (NURS) based on the optimal traffic distribution were studied in this work, and it was analysed that URS scheme outperforms the NURS one. Furthermore, by increasing the round length, the URS scheme supplementary decreases the resequencing delay. The authors modelled every path as a multiple-node M/M/1 tandem network. They assume that the end-to-end path delay follows the normal distribution. The performance metric like end-to-end path delay and resequencing delay is discussed in this paper. When average arrival rate  $\lambda$  is increased, the time in every queue is increased by which resequencing delay is also increased

Patil et al. [7] proposed a cross-layer mechanism for scheduling. Cross-layer mechanism is able to overcome many challenges for QoS due to excessive channel sharing. By adopting cross-layer approach to determine the order of the nodes, the packets will be scheduled to give a very high throughput. In this mechanism, when packet loss and retransmission is essential, still the nodes get sufficient time to serve those nodes. Because of time loss, all the other nodes are finished. These techniques significantly reduce latency and losses. The mechanism needs to improve this technique by adopting a suitable bandwidth estimation mechanism as one of the parameters for scheduling.

Cui and Wei [17] converse the problem related to efficiency and fairness of *ad hoc* networks. This work proposes a novel and efficient contention-based backoff mechanism for wireless *ad hoc* networks, which is adaptive efficiency-fairness tradeoff (AEFT) backoff algorithm. The authors increase the contention window at the time when channel is busy, then use an adaptive window to reduce the backoff time when the channel is idle by fair scheduling. The fair scheduling principally adopts maximum successive transmission and collision limit to terminate the fairness. This algorithm provides a larger fairness index and a tradeoff between efficiency and fairness. The proposed algorithm can improve total throughput. Performance metrics like backoff time, threshold and efficiency have been discussed in this paper. The proposed algorithm needs to address the continuous maximum successive transmission and the deferring or collision limit problem.

Shi et al. [18] discussed about the problem of head of line (HOL) blocking in the smart antenna system in wireless *ad hoc* networks. The authors propose a novel directional network allocation vector-based packets scheduling (DBPS) algorithm in this paper. The proposed DBPS algorithm uses the DNAV information and chooses the fittest packet in the smart antenna



system. It makes greatest of the communication status of the neighbour nodes and is further adaptive to the network topology. Hence, nodes can efficiently extend the spatial reuse and address the HOL blocking problem. The proposed algorithm improves the throughput greatly and decreases the interference. It needs to study the performance of the DBPS algorithm in the more complex network topology and to extend to some multihop scenarios.

Marbach [19] proposed distributed scheduling and active queue management mechanism for wireless *ad hoc* networks. This approach is based on a random access scheduler where the transmission-attempt probabilities depend on the local backlog. This mechanism is simple and able to be implemented in a distributed fashion. The proposed scheduling mechanism needs only a re- definition of the transmission probabilities at personage nodes. This can be done by redefining the contention window (CW) size of the current 802.11 protocol. The proposed algorithm shows high throughput and fair bandwidth allocation. This algorithm suffers from the exposed terminal problem. Approaches to avoid this problem by improving the channel feedback need to be investigated.

Jaramillo and Srikant [20] converse the predicament of congestion control and scheduling in *ad hoc* wireless networks that must support a combination of best-effort and real-time traffic. This work proposes an

**Table 1** Simulation settings

Number of nodes	100
Area size	1,500 × 300 m <sup>2</sup>
Mac	802.11
Radio range	250 m
Simulation time	10,20,30,40 and 50 s
Traffic source	VBR, CBR,TCP
Packet size	512 bytes
Mobility model	Random way point
Rate	100 kb, 200 kb,..., 500Kb

**Table 2 EQADDP**

Rate (kb/s)	Bandwidth utilization (Mb/s)	Fairness	Delivery ratio	Delay(s)	Drop (packets)
100	2.096256	0.419251	0.910192	2.235838	3,694
200	0.978416	0.195683	0.71411	6.679917	10,282
300	0.949984	0.18999	0.577505	8.603968	17,913
400	1.126	0.22519	0.43863	12.87078	27,380
500	1.083696	0.21673	0.374322	14.61645	35,874

optimization framework for the problem of congestion control and scheduling of elastic and inelastic traffic in *ad hoc* wireless networks. Authors presented a decomposition of the problem into an online algorithm that can make best possible decisions at the same time as keeping the network stable and satisfying the inelastic flow's QoS constraints. The scheduling problem for elastic and inelastic flows in a common framework by using deficit counters. Performance metrics like throughput are discussed in this paper. The channel state is considered constant during the entire frame in this work; study of this framework in the unknown channel state case is still needed. Traffic model for inelastic packets assumes that packets arrive at the beginning of the frame and all have the same delay, but it is not possible that all frames have the same delay, so it should be discussed in this framework with regards the difference in frame delay.

### 3 Proposed work

This paper proposes a queuing architecture, which supports both elastic and inelastic traffic. In this architecture, a single priority queue is maintained at the transmitting node. The priority queue holds all the packets whose routes traverse. It uses virtual queue algorithm to reduce the experienced delay by comprising virtual queues that are served at a fraction of the actual service rate and using the virtual queue-length values in utility function. Then, the optimization framework is used where the scheduling algorithm allocates the resource fairly in the network for both elastic and inelastic flows. Finally, priority dropping active queue management algorithm is applied based on proportional-integral design (PID) mechanism. This algorithm provides

the differentiated service for the different layers or frames according to their priority. When network congestion arises, the low-priority packet is dropped initially. After that, the second low-priority packet is dropped and so on.

#### 3.1 System design

System design of the proposed work consists of many steps like virtual queue algorithm, scheduler and congestion controller and active queue management algorithm. These steps will occur one after the other as shown in the Figure 1.

##### 3.1.1 Virtual queue algorithm

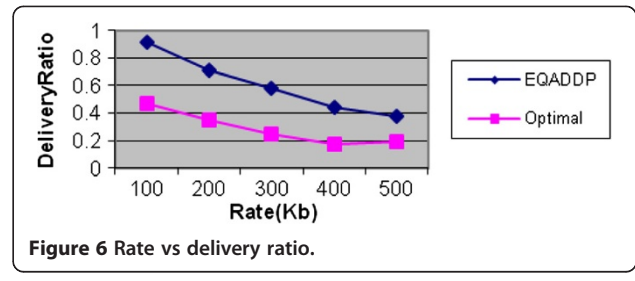
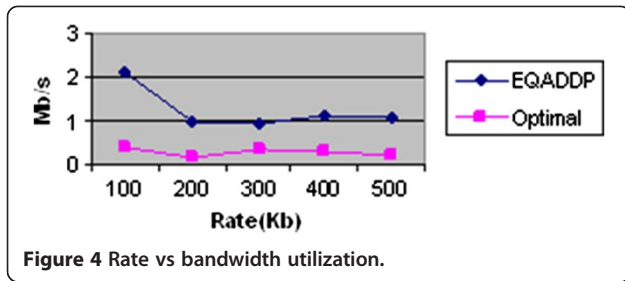
The packets from the inelastic flows have strict priority over their elastic counterparts because the inelastic applications are delay sensitive. Hence, the inelastic flows are not able to see the elastic flows in the queues in which they traverse. However, in some situations, the link might be critically loaded by the inelastic traffic itself resulting in huge delays. The elastic traffic also has some slight delay constraints. By applying virtual queues, which serves at the fraction of the actual service rate, and using the virtual queue-length values in utility function, the experienced delay can be reduced.

##### 3.1.2 Joint congestion control and load balancing algorithm

Joint congestion control and load balancing algorithm [21] is used to maximize the utilization of elastic traffic while guaranteeing the support of inelastic traffic. Consider the fluid model where dynamic behaviour and randomness is ignored. The elastic and inelastic traffics are illustrated in Figure 2. The load balancing algorithm transfer the inelastic flows to less heavily loaded routes in order to provide maximum network utilization for elastic flows.

**Table 3 Optimal**

Rate (kb/s)	Bandwidth utilization (Mb/s)	Fairness	Delivery ratio	Delay (s)	Drop (packets)
100	0.387664	0.077532	0.47328	13.3032	6,906
200	.1704	0.034079	0.347072	13.29301	15,033
300	0.34064	0.0681279	0.246597	18.57866	25,215
400	0.335408	0.067081	0.174292	18.26814	31,146
500	0.237968	0.047593	0.187054	18.01323	42,592



Here, a source must have the knowledge of all the queue information along its route. The source sends the queue information hop by hop to achieve stability even though this information is delayed. Initially, virtual queue is evolved for both elastic and nonelastic flows. After this, congestion controller for elastic flow and load balancing for inelastic flows are performed using equations developed by Li et al. [21].

**Algorithm:**

Step 1: Virtual queue evolution for a link  $l$  is given by

$$\theta_l(t) = (z_l(t) + y_l(t) - \alpha_1 c_l)_{\theta_l(t)} \quad (1)$$

where  $(t)$  is continuous time index, the aggregated elastic and inelastic rates are denoted by  $y_l$  and  $z_l$ .  $\alpha_1$  and  $\alpha_2$  are two types of virtual queues, which control the total load and the inelastic flow load, respectively.  $c_l$  is the capacity of link  $l \in L$ .

Virtual queue evolution for a link  $l$  for inelastic flow is given by

$$\gamma_l(t) = (z_l(t) - \alpha_2 c_l)_{\gamma_l(t)} \quad (2)$$

Step 2: Congestion controller for elastic flow

$$x_e(t) = U_e^{t-1}(S_{R_e}(t)) \quad (3)$$

where  $S_{R_e}$  is the aggregated virtual queue length of the elastic flow and  $U$  is the utility function.

Step 3: Load balancing implemented for inelastic flow

The number of packets at flow  $i$  at route  $r$  is given by

$$x_i^{(r)}(t) = \left( \mu_i'(t) - \mu_{R_i^{(r)}}(t) \right)_{x_i^{(r)}(t)} \quad (4)$$

where  $\mu_i'(t)$  satisfies  $\sum_{r=1}^{|R_i|} \left( \mu_i'(t) - \mu_{R_i^{(r)}}(t) \right)_{x_i^{(r)}(t)} = 0$  and  $\sum_{r=1}^{|R_i|} x_i^{(r)}(0) = a_i$ ,  $a_i$  denotes the arrival rate of inelastic flows.

**3.1.3 Scheduler and congestion controller**

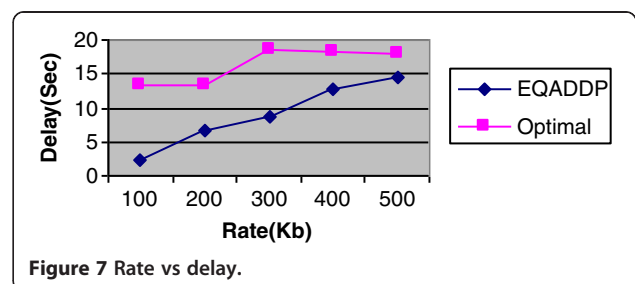
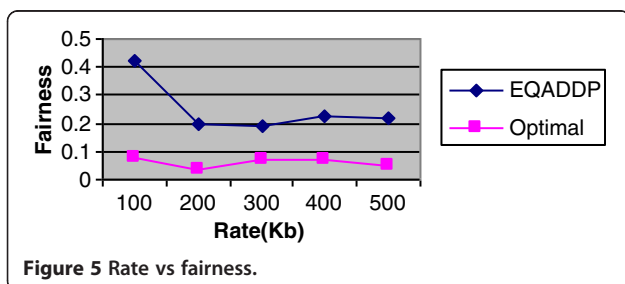
Let  $S_{il}$  and  $S_{el}$  be the number of inelastic and elastic packets, respectively, that can be scheduled for transmission at link  $l$  for the time slot  $t \{1, 2, \dots, T\}$

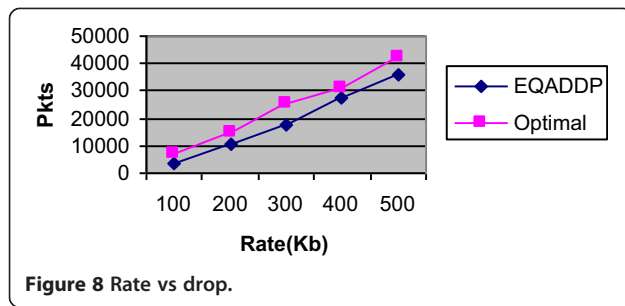
Let  $S(a_p, c)$  be the feasible schedule where  $c$  is the channel state.

In the congestion control algorithm [20], the queue length of elastic flows and inelastic flows at link  $l$  is given by  $q_l(k)$  and  $d_l(k)$ , respectively. Here,  $k$  is the current frame composed of time slot  $t$ . The congestion control algorithm is given by

$$\tilde{x}_{el}^*(k) \in \arg \max_{0 \leq x_{el} \leq X_{max}} \frac{1}{\epsilon} U_l(x_{el}) - q_l(k) x_{el} \quad (5)$$

The elastic arrival rate, which is a nonnegative real number, is converted into a nonnegative integer. This will indicate the number of elastic packets allowed to enter the network in a given frame  $k$ . Let us assume that the elastic arrival at link  $l$  is  $a_{el}(k)$  is a random variable and  $\Pr$  is a Probability. This satisfies  $\Pr(a_{el}(k) = 0) > 0$  and  $\Pr(a_{el}(k) = 1) > 0$  for all  $l \in L$  and all  $k$ . These assumptions guarantee the Markov chain that is defined below is irreducible and a periodic.





Consider the number of inelastic arrivals be  $a_i(k)$  and the channel state be  $c(k)$ . The scheduling algorithm is as given by

$$\begin{aligned} \bar{s}^* a_i(k), c(k), d(k), q(k) \in \arg \max_{s \in S(a_i(k), c(k))} \sum_{l \in L} \left\{ \left[ \frac{1}{\epsilon} w_l + d_l(k) \right] \sum_{t=1}^T S_{il,t} \right. \\ \left. + q_l(k) \sum_{t=1}^T S_{el,t} \right\} \end{aligned} \quad (6)$$

Here, the number of inelastic arrivals at  $l(a'_{il}(k))$  is a binomial random variable with parameters  $a_{il}(k)$  and  $1 - p_l$ . The quantity  $a'_{il}(k)$  can be generated by the network as follows: on each inelastic packet arrival, toss a coin with probability of heads equal to  $1 - p_l$  and if the outcome is heads, add a 1 to the deficit counter. The optimal scheduler is a function of  $a_i(k)$ ,  $c(k)$ ,  $d(k)$  and  $q(k)$ .  $d_l(k)$  is interrupted as a virtual queue that counts the deficit in service for link to achieve a loss probability due to deadline expiry less than or equal to  $p_l$ .

### 3.1.4 PID control

PID [22] is a power controller. It is composed of proportion, integral and derivative controller. PID will compute a control action based on the input state and feedback gain multipliers that control stability, error and response. The proportional-integral design will avoid the steady-state error, but it will decrease the responsiveness by almost  $1^\circ$  of magnitude. A derivative part helps to reduce the overshoot and the settling time. The network feedback control based on PID is as shown in Figure 3 [23].

Here  $q_0$  is the expected queue length,  $q$  is instantaneous queue length,  $e = q - q_0$  is the error signal.  $p$  is the packet

loss rate at some time which is the output of the PID controller. The input given to the PID controller is  $e$ .

PID control system estimates the packet loss rate  $p$  of every arriving packet based on the variance of queue length of the router. Source detects the packet loss rate after a link delay time. Source then judges the congestion state according to  $p$  and adjusts its sending rate to control the length of the router. The dropping probability  $p$  is given by:

$$p = \begin{cases} 0 & p < 0 \\ p & 0 \leq p \leq 1 \\ 1 & p > 1 \end{cases} \quad (7)$$

From Equation 7, it is clear that  $p$  is always in between 0 and 1.

The implementation process of priority dropping can be explained as follows: First, the packet priority number is defined when the data is packetized in the application layer. The priority number is then written to the priority field of the packet. The priority number for other background flows is set to 0. Here, the router maintains a packet queue. The queue is updated when packets enter queue or depart queue. For the newly arriving packet, dropping probability is calculated according to (7). If the current packet is determined to drop, then the packet whose priority number is less than the current packet in the queue is found. If there is any lower-priority packet in the queue, then that packet is dropped and the current packet enters the queue. Else, the current packet will be dropped.

### 3.2 Advantages

The main advantage of the proposed approach is that it is an effective queuing architecture, which will handle both elastic and inelastic traffic flows and assign different dropping precedence for different priority of traffic.

### 4 Simulation results

NS-2 [24] is used for simulation for proposed effective queuing architecture with different dropping precedence (EQADDP) technique. By simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. In simulation, 100 mobile nodes move in a  $1,500 \times 300 \text{ m}^2$

**Table 4 EQADDP**

Time (s)	Rate (Kb/s)	Bandwidth utilization (Mb/s)	Fairness	Delivery ratio	Delay(s)	Drop (packets)
10	100	1.538864	0.307772	0.892803	0.629921	699
20	100	1.874672	0.374934	0.915715	0.887353	1,286
30	100	1.790032	0.358006	0.904152	1.308502	2,176
40	100	1.819456	0.363891	0.903085	1.826787	3,028
50	100	2.096256	0.419251	0.910192	2.235838	3,694

**Table 5 Optimal**

Time (s)	Rate (Kb/s)	Bandwidth utilization (Mb/s)	Fairness	Delivery ratio	Delay(s)	Drop (packets)
10	100	0.37584	0.0751	0.36596	3.445484	1,063
20	100	0.428608	0.0857	0.459351	6.360243	2,469
30	100	0.402048	0.0804	0.474589	8.430253	3,750
40	100	0.436816	0.0873	0.478822	10.82739	5,363
50	100	0.387664	0.07753	0.47328	13.3032	6,906

rectangular region for different time simulation times. Initial locations and movements of the nodes are obtained using the random waypoint (RWP) model of NS-2. Assume that each node moves independently with the same average speed. In this mobility model, a node randomly selects a destination from the physical terrain. In simulation, the time varies from 10 to 50 s. The simulated traffics are variable bit rate traffic (VBR) and constant bit rate (CBR) for inelastic traffic and TCP for elastic traffic.

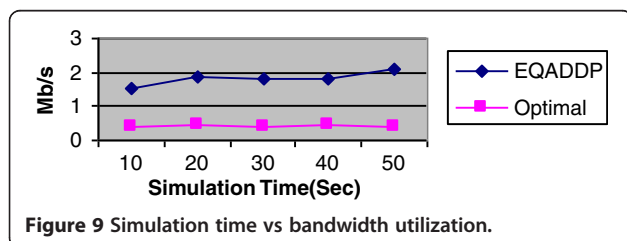
Simulation settings and parameters are summarized in Table 1.

**4.1 Performance metrics**

Comparative study made to prove the performance of this proposed (EQADDP) with the optimal scheduling algorithm. The following metrics were used for performance evaluation:

- Average end-to-end delay: The end-to-end delay is averaged over all surviving data packets from the sources to the destinations.
- Average packet delivery ratio: It is the ratio of the number of packets received successfully and the total number of packets transmitted.
- Drop: It is the total number of packets dropped during the transmission.
- Bandwidth: It is the measure of received bandwidth for all traffic flows.
- Fairness: For each flow, measure the fairness index as the ratio of throughput of each flow and total no. of flows.

The performance results are presented graphically in the next section.



**4.2 Results**

**4.2.1 Based on rate**

In the first experiment, vary the rate as 100, 200, 300, 400 and 500 kb and keep the simulation time at a constant value (50 s) and measure the selected metrics. The results obtained for the proposed algorithm and the algorithm taken for comparison are shown in Tables 2 and 3, respectively.

Figure 4 shows the received bandwidth of EQADDP and optimal techniques for different rate scenarios. It is concluded that the bandwidth of proposed EQADDP approach is 337% higher than optimal approach.

Figure 5 shows the fairness of EQADDP and optimal techniques for different rate scenarios. It is concluded that the fairness of proposed EQADDP approach is 337% higher than optimal approach.

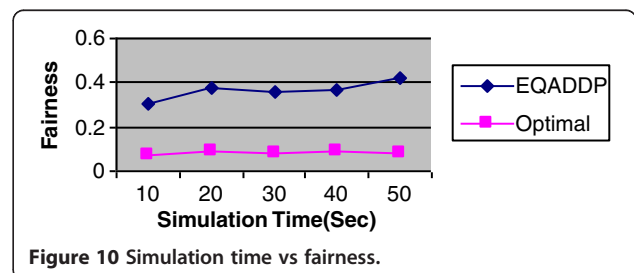
Figure 6 shows the delivery ratio of EQADDP and optimal techniques for different rate scenarios. It is concluded that the delivery ratio of proposed EQADDP approach is 106% higher than optimal approach.

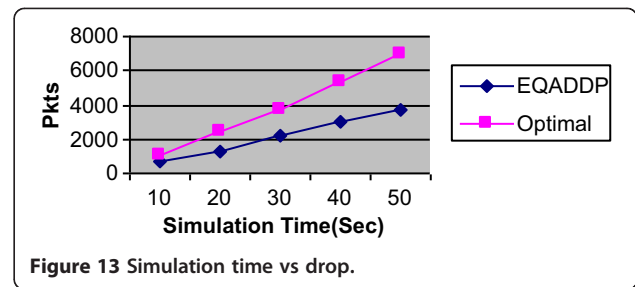
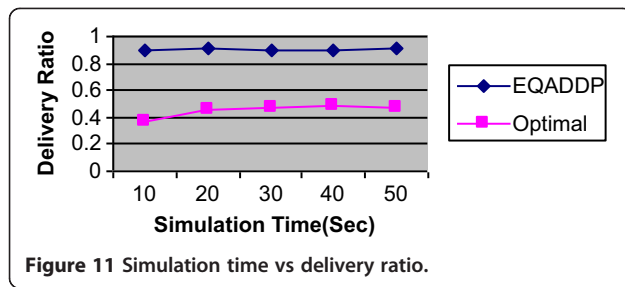
Figure 7 shows the delay of EQADDP and optimal techniques for different rate scenarios. It is concluded that the delay of proposed EQADDP approach is 47% less than optimal approach.

Figure 8 shows the drop of EQADDP and optimal techniques for different rate scenarios. It is concluded that the drop of proposed EQADDP approach is 27% less than optimal approach.

**4.2.2 Based on time**

In second experiment, vary the time as 10, 20, 30, 40 and 50 s and keep the rate at a constant value (100 kb/s) and measure the selected metrics. The results obtained





for the proposed algorithm and the algorithm taken for comparison are shown in Tables 4 and 5, respectively.

Figure 9 shows the received bandwidth of EQADDP and optimal techniques for different time scenarios. It is concluded that the bandwidth of proposed EQADDP approach is 349% higher than optimal approach.

Figure 10 shows the fairness of EQADDP and optimal techniques for different time scenarios. It is concluded that the fairness of proposed EQADDP approach is 278% higher than optimal approach.

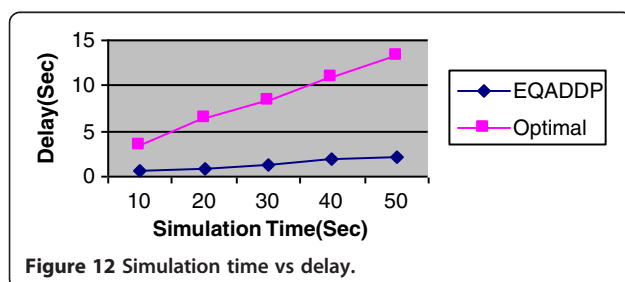
Figure 11 shows the delivery ratio of EQADDP and optimal techniques for different time scenarios. It is concluded that the delivery ratio of proposed EQADDP approach is 103% higher than optimal approach.

Figure 12 shows the delay of EQADDP and optimal techniques for different time scenarios. It is concluded that the delay of proposed EQADDP approach is 84% less than optimal approach.

Figure 13 shows the drop of EQADDP and optimal techniques for different time scenarios. It is concluded that the drop of proposed EQADDP approach is 43% less than optimal approach.

## 5 Conclusions

This paper proposed a queuing architecture for elastic and inelastic traffic. If a link is critically loaded by the inelastic traffic, then large delays may occur. Elastic traffic also has delay constraints. Virtual queue algorithm is used to reduce the delay using virtual queues and virtual queue-length values. The optimization framework is used where scheduling algorithm allocates the resource fairly in the network. Based on priority, the packets are classified as



low-, medium- and high-priority data packet for drop preference. Based on PID mechanism, priority dropping active queue management algorithm (PID\_PD) provides the differentiated service for the different layers or frames according to their priority. Simulation results proved that the proposed architecture offers better fairness and delivery ratio with reduced delay and drop.

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>Anna University, Chennai 600025, India. <sup>2</sup>PSN College of Engineering and Technology, Melathediyoor, Tirunelveli 627152, India. <sup>3</sup>Department of computer Science and Engineering, Allagappa University-Karaikudi, Karaikudi 630003, India.

Received: 5 December 2013 Accepted: 24 June 2014

Published: 26 September 2014

### References

1. A. Moon, H. Cho, Energy-efficient replication extended database state machine in mobile ad-hoc network, in *IADIS International Conference on Applied Computing*, 2004, pp. 224–228
2. Z. Xing, L. Gruenwald, "Issues in Designing Concurrency Control Techniques for Mobile Ad-hoc Network Databases", *Technical Report, School of Computer Science* (University of Oklahoma, 2007), pp. 1–37
3. P. Mukilan, D.A. Wahi, EENMDRA: efficient energy and node mobility based data replication algorithm for MANET. *IJCSI Int. J. Comput. Sci. Issues* **9**, 357–364 (2012)
4. Y.-S. Yen, H.-C. Chao, R.-S. Chang, A. Vasilakos, Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Math. Comput. Model.* **53**, 2238–2250 (2011)
5. S.T. Hasson, E. Fadil, Queuing approach to model the MANETs performance. *Br. J. Sci.* **6**, 18–24 (2012)
6. Y.-F. Guo, G.-S. Kuo, A packet scheduling framework for multipath routing in mobile ad hoc networks, in *Vehicular Technology Conference (IEEE, 2007)*, pp. 233–237
7. R. Patil, A. Damodaram, R. Das, Cross layer fair scheduling for MANET with 802.11 CDMA channels, in *First Asian Himalayas International Conference, IEEE* (Kathmandu, 2009), pp. 1–5
8. T. Salonidis, L. Tassiulas, Distributed on-line schedule adaptation for balanced slot allocation in wireless ad hoc networks, in *Quality of Service, Twelfth IEEE International Workshop (IEEE, 2004)*, pp. 20–29
9. Y. He, R. Yuan, J. Sun, W. Gong, Semi-random backoff: towards resource reservation for channel access in wireless LANs, in *17th IEEE International Conference on Network Protocols (IEEE, Princeton, 2009)*, pp. 21–30
10. H. Xiao, W.K.G. Seah, A. Lo, K.C. Chua, A flexible quality of service model for mobile ad-hoc networks, in *IEEE Vehicular Technology Conference (Tokyo, 2000)*, pp. 445–449
11. G.-S. Ahn, A.T. Campbell, A. Veras, L.-H. Sun, SWAN: service differentiation in stateless wireless ad hoc networks, in *Twenty-First Annual Joint Conferences of the IEEE Computer and Communications Societies, IEEE, 2002*, pp. 457–466
12. J. Parvez, M.A. Peer, A comparative analysis of performance and QoS issues in MANETs. *World Acad. Sci. Technol.* **48**, 937–948 (2010)



13. N. Xiong, A.V. Vasilakos, L.T. Yang, C.-X. Wang, R. Kannan, C.-C. Chang, Y. Pan, A novel self-tuning feedback controller for active queue management supporting TCP flows. *Inform. Sci.* **180**, 2249–2263 (2010)
14. P. Li, S. Guo, S. Yu, A.V. Vasilakos, CodePipe: an opportunistic feeding and routing protocol for reliable multicast with pipelined network coding. *IEEE Infocom* 100–108 (2012)
15. S. Wang, A. Vasilakos, H. Jiang, X. Ma, W. Liu, K. Peng, B. Liu, Y. Dong, Energy efficient broadcasting using network coding aware protocol in wireless ad hoc network, in *Communications (ICC), 2011 IEEE International Conference on* (Kyoto, 2011), pp. 1–5
16. I. Ambika, P. Eswaran, Improved token based resource allocation technique for multi-service flows in MANET. *Int. Rev. Comput. Softw.* **8**, 1486–1496 (2013)
17. H.-X. Cui, G. Wei, A novel backoff algorithm based on the tradeoff of efficiency and fairness for ad hoc networks. *WRI Int. Conf.* **2**, 81–86 (2009)
18. C. Shi, X. Dai, L. Luo, M. Cui, A novel directional-NAV-based packets scheduling algorithm for ad hoc networks, in *International Conference on Wireless Communications & Signal Processing* (Nanjing, 2009), pp. 1–4
19. P. Marbach, Distributed scheduling and active queue management in wireless networks, 26th IEEE international conference on computer communications. *IEEE* 2321–2325 (2007)
20. J.J. Jaramillo, R. Srikant, Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic. *IEEE/ACM Trans. Networking* **19**, 1124–1136 (2011)
21. R. Li, L. Ying, A. Eryilmaz, N.B. Shroff, A unified approach to optimizing performance in networks serving heterogeneous flows. *IEEE/ACM Trans. Networking* 223–236 (2011)
22. N. Xiong, X. Jia, L.T. Yang, A.V. Vasilakos, A distributed efficient flow control scheme for multirate multicast networks. *IEEE Trans. Parallel Distr. Syst.* **21**, 1254–1266 (2010)
23. Y. Xiaogang, L. Jiqiang, L. Ning, K. T-h, Priority dropping for scalable video. *Int. J. Multimed. Ubiquitous Eng.* 119–129 (2007)
24. Network Simulator, *Network Simulator*, <http://www.isi.edu/nsnam/ns/>

doi:10.1186/1687-1499-2014-155

**Cite this article as:** Ambika et al.: An effective queuing architecture for elastic and inelastic traffic with different dropping precedence in MANET. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:155.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---