New Queuing Algorithm for Ad Hoc Networks (Custom Queuing Algorithm)

[1]Ashraf Diaa El-Den; [2]Hosny Ahmed Motaafy, [3]Ibrahim Fathy Torad; [1]Atef Lofty Salama and [2]Adel Fouad El-Kabbany

[1]Military Technical College, [2]Higher technology Institute, [3]Faculty of Engineering, Al Azhar University
kabany_2008@yahoo.com

Abstract: Multi-hop wireless networks are challenging, due to the unique characteristics of these networks. In this paper we consider the presence of both guaranteed and best-effort flows in the network. The aim is to ensure minimum required bandwidth to the best-effort flows and provide an equal share of residual bandwidth to all the flows. Such a network is expected to support advanced applications such as communications in emergency disaster management, video conferencing in a workshop or seminar, communications in a battlefield. This class of mission-critical applications demands a certain level of quality of services (QoS) for proper operations. In Fair scheduling each flow f is allowed to share a certain percentage of link capacity based on its flow weight indicated as W(t1, t2) denote the aggregate resource received by flow f and g respectively in time interval [t1, t2]. The allocation is ideally fair if it satisfies (1)

$$\left|\frac{W_f(t_1,t_2)}{w_f} - \frac{W_g(t_1,t_2)}{w_g}\right| = 0 \qquad (1)$$

For all flows f and g. Adapting fair queuing to an ad hoc network is challenging because of the unique issues in such a network **[7].**

[Ashraf Diaa El-Den; Hosny Ahmed Motaafy, Ibrahim Fathy Torad; Atef Lofty Salama and Adel Fouad El-Kabbany **New Queuing Algorithm for Ad Hoc Networks (Custom Queuing Algorithm).** New York Science Journal 2012;5(10):1-7]. (ISSN: 1554-0200). http://www.sciencepub.net/newyork. 1

Keywords: Queuing; Algorithm; Ad Hoc Networks

1 Design Issues in QoS Supported Fair Scheduling

This section identifies issues unique to fair scheduling in ad hoc wireless networks.

1.1 Defining Fairness for spatially contending Flows.

1.2 Conflict between Fairness and Maximal Channel Reuse.

1.3 Distributed Nature of Ad Hoc Fair Scheduling.

1.4 Providing QoS and State Maintenance:

Providing QoS to certain flow requires availability and reservation of resources for that flow. Since, the single wireless medium is shared by the contending nodes; a node can ensure the availability of certain resources if it has the flow information of all the contending nodes and if the information is updated regularly. Also if there is break of route the resource of the flow should be released as early as possible so that the resource can be allocated to a new flow.

Most research work in the area of wireless ad-hoc networks attempts to balance the trade-off between fairness and channel utilization. In this paper, we first propose a topology-independent methodology to predict maximum achievable channel utilization under fairness constraint by two performance bounds. Based on the notion of bottlenecks introduced in prediction, we design a centralized and improved fair scheduling algorithm for wireless ad hoc networks. We capture traffic load characteristics by using a proposed parameter that represents the "contending power" of nodes in the weighted flow contention graph. Finally, we demonstrate the effectiveness of our pro-posed algorithm through both provable analysis and simulations, and discuss natural derivations of a fully distributed algorithm using our bottleneck-based analytic model. In recent research, various resource management algorithms and protocols for mobile networking environments are proposed to devise effective management schemes to support Quality-of-Service (QoS) in capacity constrained and highly dynamic wireless networks. Typical proposals include QoS-oriented MAC layer design, packet scheduling, and ad-mission control schemes, where fair distribution of band-width and maximization of resource utilization have been identified as two important design goals. However, as identified, achieving both fairness and maximization of channel utilization is particularly challenging in wireless ad-hoc networks. They have proposed various distributed schemes that seek to maximize the aggregate throughput with a basic fairness guarantee. Although these are effective solutions, it is not clear exactly what levels of fairness or throughput they are able to achieve before simulating the algorithms. In this paper, our major contributions are the following:

First, we propose a novel prediction methodology, based on lower and upper bound analysis, to reveal the maximum achievable throughput under the strict notion of fairness for any network topology. Such predictions provide essential guidelines during the design of new fairness-aware protocols.

Second, along with our prediction methodology, we present a key observation with respect to bottleneck considerations in multi-hop wireless networks. Such bottlenecks should receive full attention during analysis and scheduling.

Finally, from such observations, we propose a new QoS parameter, based only on local flow weights and topology information, to integrate the degree of contention among flows into our fairness mode. With the parameter we design a centralized packet scheduling algorithm that achieves optimal channel utilization and fairness for each **flow**. The fact that only local state information is used promotes a fully distributed version of the scheduling algorithm.

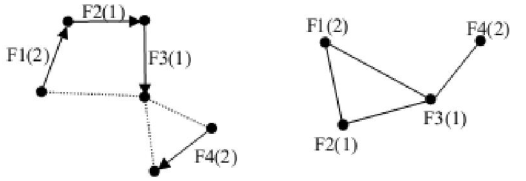
2 System Model and Throughput Prediction

In shared-medium multi-hop wireless networks, fair scheduling amounts to unbiased scheduling of spatially contending flows. Based on widely accepted definitions of fairness **[1]**, various scheduling disciplines from wire line networks have been adapted in the multi-hop wireless domain. On the other hand, non-contending flows that are spatially far apart could potentially be scheduled together, leading to effective channel utilization. A common strategy to arbitrate the conflicts between the two inherently incompatible design goals has been to maximize channel utilization under a certain fairness constraint. Taking this strategy into account, we propose our throughput prediction methodology based on Weighted Flow Contention Graphs.

2.1 Weighted Flow Contention Graph

A flow contention graph (or flow graph) represents spatial contention relationships among contending flows. Vertices are mapped to backlogged flows represented by edges in the network node graph. An edge in the flow graph connects two vertices whenever the represented flows are within a two-hop distance. Thus the resulting undirected flow graph precisely illustrates the location dependency of spatial contention. Fig.1 shows a conversion from a node graph to the corresponding flow contention graph. When flows have unequal rights to channel resources, flow weights are often associated to represent their relative share.

In our analysis we consider positive integer weights 2 w = {w1,. …, wn} flow to be associated with the n vertices of the flow graph G, resulting in a weighted contention graph (G, w) for the topology. It should be noted that multi-hop flows are being modeled as multi-single-hop flows in our formulation. This can be understood as a per-hop behavior of packet scheduling.



(a) weighted node graph (b) weighted flow contention graph

Fig. 1. A simple topology and its weighted flow contention graph

2.2 Channel Reuse Index

We define the average throughput or transmission rate, u, for a system of flows in the multi-hop wireless network as transmission time no. of packets transmitted in the network

$$u = \frac{no.\ of\ packets\ transmitted\ in\ the\ network}{transmission\ time}. \qquad (1)$$

For simplicity, transmission is assumed to occur in discrete time slots. For fixed packet length the average throughput can be seen to be proportional to the number of packets transmitted per time slot. And if such denotes the average throughput attained by a scheduling discipline shed, and Uno reuse denotes the average throughput without channel reuse (e.g., from a strict fair queuing discipline), we define a channel reuse index **(CRI)** for the discipline to be such

$$CRI = u_{sch} / u_{no\_reuse}. \qquad (2)$$

CRI can be seen to precisely measure the performance boost of a scheduling discipline with channel reuse considerations, in terms of its channel utilization.

2.3 Throughput Prediction

We study two common graph-theoretic techniques to predict maximal throughput in a generic multi-hop wireless network. Both techniques take fairness constraint into account when striving for maximal throughput. In the forthcoming discussions, mutually contending flows in G share a single channel of capacity C.

2.3.1 Weighted Graph Coloring

Graph or vertex coloring in graph theory finds widespread applications in many day-to-day scheduling problems, such as timetabling, register allocation and frequency assignment. Most of them have to do with avoidance of scheduling conflicts. In our context, we need to schedule weighted flows in a multi-hop network by segregating them into multiple non-contending sets, thereby exploiting channel reuse. The ultimate goal is to come up with a partitioning strategy that results in a minimal number of non-contending sets.

A linear programming approach is, however more commonly used to formulate generalized

Version of the problem: Suppose L denotes of all stable sets of a weighted graph **(G, W).**
Find positive **Y**s for each S L to solve

Min $\sum_{S \quad L} \mathbf{Ys}$ subject to

$\sum_{S \quad L} \quad i \quad s \; \mathbf{Ys} \geq \mathbf{w_i} \quad i \geq \mathbf{V}$ (3)

Assuming a packet can be transmitted in unit time slot, by minimum coloring techniques we could deliver all packets within one scheduling cycle in only $w(G)$ time slots. The throughput gain based on this approach, or in other words, its channel reuse index is therefore given by $\boldsymbol{CRI}_{col} \; \sum_{I} \boldsymbol{w}_{i/X\,w(G)}$ **(4)**

From the optimality of minimum coloring, we contend that ***CRI col*** *sets out a feasible lower* bound on maximum channel reuse or throughput prediction. This can be conveniently formulated as a minimum weighted graph coloring problem.

2.3.2 Maximum Weighted Clique (Bottleneck Analysis)

In wireless ad-hoc network, localities of intense spatial contention, or bottlenecks, should be identified and honored when predicting maximum throughput under the fairness constraint. In this context we are particularly interested in severe bottlenecks, identified as maximal weighted cliques in a given weighted flow graph. Consider the simple flow graph of Fig. 1. A feasible fair scheduling scheme would allocate 1/2, 1/4 and 1/4 of the channel capacity to the bottleneck flows F1-F3, while F4 would not be given a larger share than F13 (Fig. 2).

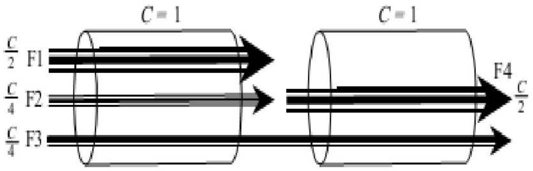


Fig. 2. Flow contention situation for flow graph of Fig. 1: F1, F2 and F3 fully consume one channel; F3 and F4 partially consume a different channel. Resource allocation, however, has complied with the fairness constraint.

A proper prediction of maximum throughput of 3/2 of single channel capacity. This is in contrast with max-min fairness, where in this scenario flow 4 will be allocated a share of 3/4. The network is thus 3/2 of the single channel capacity. However, a careless prediction without bottleneck consideration would have claimed 1/3 and 2/3 of the capacity for F3 and F4, leading to an unrealistic conclusion that F1-3 combined consumes 4/3 of the single channel capacity.

Formally, suppose uo is the throughput per unit weight in the weighted flow graph G = (G**,** W)

Observing the capacity constraint for each identified weighted clique, we write down a system Of inequalities: $\mathbf{i \quad J \quad u \quad C \quad J \quad G}$ **(5)**
where *G* is a set of identified cliques of *G*. To ensure feasibility in **(5),** the throughput per unit weight must satisfy: **uo C** / max**(iJ Wi)JG (6)**

The denominator of **(6)** is naturally the *weighted clique number* w(G) of the weighted flow graph, frequently used as a lower bound for ***w*(*G*).** Hence, an upper bound on the channel
reuse index easily follows:

CRI clq = uo Wi / **c** = **Wi** /ωw**(G) (7)**

Since capacity constraints can never be violated, we contend that **CRI clq** sets out a theoretical upper bound on maximum channel reuse or throughput prediction. With the two bounds for prediction in place, we discuss their significance. **First**, in cases where $\chi_w(G) = \omega_w(G)$

We have the tightest bounds; otherwise other theories are also known to obtain a tighter lower bound for **CRI col**. **Secon**d, we expect scheduling disciplines claimed to deliver optimal throughput under fairness constraint to observe the two bounds. This expectation, however, may just be too optimistic in our opinion.

Third, we also note that the theoretical bound of ***CRIclq*** is maximal on condition that the fairness constraint is honored. There is absolutely no reason why an exceeding CRI cannot be realized through flow starvation.

3 Centralized Fair Scheduling With Bottleneck

3.1 Considerations:

In this section, we describe a centralized scheduling algorithm that takes our bottleneck notion into consideration for multi-hop networks. Recall that bottlenecks are localities where special attention is required. We would simply prefer to pay such attention when designing a scheduling discipline. In particular, such scheduling discipline needs to give priority services to flows belonging to a bottleneck locality. In order to differentiate between the severities of bottlenecks to which flows belong so as to assign the appropriate priorities, we devise a metric known as the contending power of flow for the purpose

3.2 The Flow Contending Power

We define the flow contending power Pi for a flow i as

Pi max j w k)i j , jG (8)

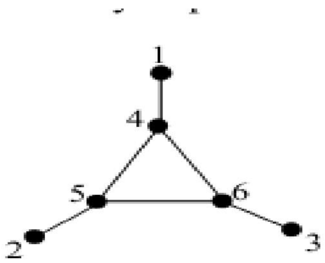


Fig. 3. Flow graph

TABLE I
WEIGHTS AND CONTENTION POWER
FOR FIG. 3.

Flow	w_i	P_i	w_i	P_i
1	1	2	1	2
2	1	2	2	4
3	1	2	3	6
4	1	3	1	6
5	1	3	2	6
6	1	3	3	6

Intuitively, it measures indirectly the level of contention a flow perceives in its neighborhood. By comparing the contending powers of respective flows, we can identify locations of bottlenecks and assign priorities accordingly. Under such notion of contending power, we do not intend to single out at all times the flow that experiences the most contention, but only the one from a particular subset currently under scheduling consideration. We claim that even in a fully distributed environment, a node can "learn" its Pi by exchanging topology and weight information with only the nodes in its neighborhood.

3.3 Considerations in a Centralized Scheduling Algorithm

While bottleneck consideration is a valuable methodology for prediction, it does not constitute sufficiency in scheduling decisions per se. In our case, it is only being used as a supplementary tool within a fair scheduling discipline that provides basic fairness. We adopt **3.3.1 Start-time Fair Queuing** (**SFQ**) to assign two tags to each arriving packet: a start tag and a finish tag. Specifically, a packet with sequence number n of flow i arriving at time **A(ti,n)** is assigned a start tag **si,n** and a finish tag **fi,n**, defined as follows:

$$s_{i,n} = \max\{V(A(t_{i,n})),\ f_{i,n-1}\}; \quad f_{i,n} = s_{i,n} + L_p / w_i\ , \quad (9)$$

Where ***Lp*** denotes the packet size in bits. Basically, the scheduling window should always be filled up to full capacity to expedite delivery of packets.

3.3.2 In summary, the following scheduling rules are enforced within the scheduling window:

Rule 1: Bottleneck Consideration. The packet from a flow f carrying the highest contending power Pf is always given the priority for transmission.

Rule 2: Start Tag Usage. For the packets from flows carrying the same maximum contending power, the one with the smallest start tag from flow f is given the priority to transmit.

Rule 3: Maximal Independent Flow Set. To optimize network utilization, the maximal set of packets that are not contending with flow f is selected to transmit simultaneously.

Rule 4: Secondary Usage of Contending Power. If there are several such flows sets in Rule 3,
we compute the total contending power for each set and select the highest one to transmit with flow f. Among multiple sets with the highest contending power, the one with the largest cardinality will be selected to transmit with flow f. Further ties are broken arbitrarily.

4 The Centralized Scheduling Algorithm

The algorithm comprises of **five steps**:

Step 1: Compute the contending power Pi, pre-compute start tag and finish tag for each flow in the flow graph.

Step 2: Pre-fill the scheduling window with packets from the scheduling queue in ascending order of start tags.

Step 3: Within the scheduling window, apply Rule 1 to grant transmission priority to the flow f with the largest **Pf**. Apply **Rule 2** when necessary.

Step 4: Apply Rule 3 to select the appropriate non contending flow set containing Pf. Apply Rule 4 accordingly to seek additional resolution. Transmit the resulting flow set Simultaneously with **Pf**.

Step 5: Refill packets into the scheduling window from the scheduling queue.

5 Algorithmic Properties:

5.1 Fairness Guarantee

In our design, we use **SFQ** to achieve the basic fairness. In attaining channel reuse, we swap service order of the queuing packets. In order to guarantee long-term inter-flow fairness,

We adopt the scheduling window mechanism to constrain potential unfairness due to channel reuse by only rescheduling queuing packets within the scheduling boundary. Even within the scheduling window, the notion of fairness is not totally abandoned. We note that the minimum start tag mechanism is still being adopted to resolve selection conflict between two candidate flows with maximum contending power. In addition, we claim that stricter short-term fairness can easily be achieved by simply adding a counter to each of the packets in the scheduling window, keeping track of its sojourn time.

We then give a packet the highest priority when its counter exceeds a time bound ζ so that short-term unfairness is effectively bounded by **f(ζ)**.

5.2 Maximal Throughput

Our algorithm has an edge over others that consider channel reuse in that we pay more attention to the highly congested areas in the topology. We always select the bottleneck flow within the scheduling window to realize channel reuse. We compare contending powers of the independent flow sets containing the bottleneck flow to identify the maximal non contending one. This strategy allows efficient channel utilization while staying in line with our design tenet those localities of high contention should always be honored.

5.3 Conflict between Fairness and Maximal Throughput

Based upon the aforementioned properties, we argue that our algorithm can find a balance **[2]** spot between the two seemingly incompatible design goals: fairness and maximal throughput. We examine the queuing dynamics at nodes in an ad hoc mobile network and evaluate net. performance under different packet scheduling algorithms using Dynamic Source routing and Greedy Perimeter Stateless Routing (GPSR) as the underlying routing protocols. Typically, packet schedulers in ad hoc networks give priority to control packets over data packets and serve data packets in FIFO order.

5.4 Scheduling Algorithms Studied

Scheduling algorithms determine which packet is served next among the packets in the queue**.** The scheduler is positioned between the routing agent and above the MAC layer **[4]**. All nodes use the same scheduling algorithm. We consider the conventional scheduling (Priority scheduling) typically us in mobile ad hoc networks and also proposed other applicable scheduling policies to study. All scheduling algorithms studied are non-preemptive

As traffic load in the network increases, the performance degradation gets worse.

5.5 Performance Evaluation

In this section, we use simulations to evaluate the performance of EMLM-FQ **[3]**. with different overlaying applications. The radio model is based on the existing commercial hardware with a wireless transmission range of 250 meters and channel capacity of 2Mbps.

FIFO and IEEE 802.11 MAC. how to route packets hop by hop as efficiently as possible and medium access control (MAC), how to share the medium efficiently.

6 Distributedpriorityscheduling

6.1 Preliminaries

In this section, we devise a scheme for approximating a dynamic priority scheduler within a broadcast region (a region in which all nodes are within radio range of all other nodes controlled by a CSMA/CA scheme **[5].**

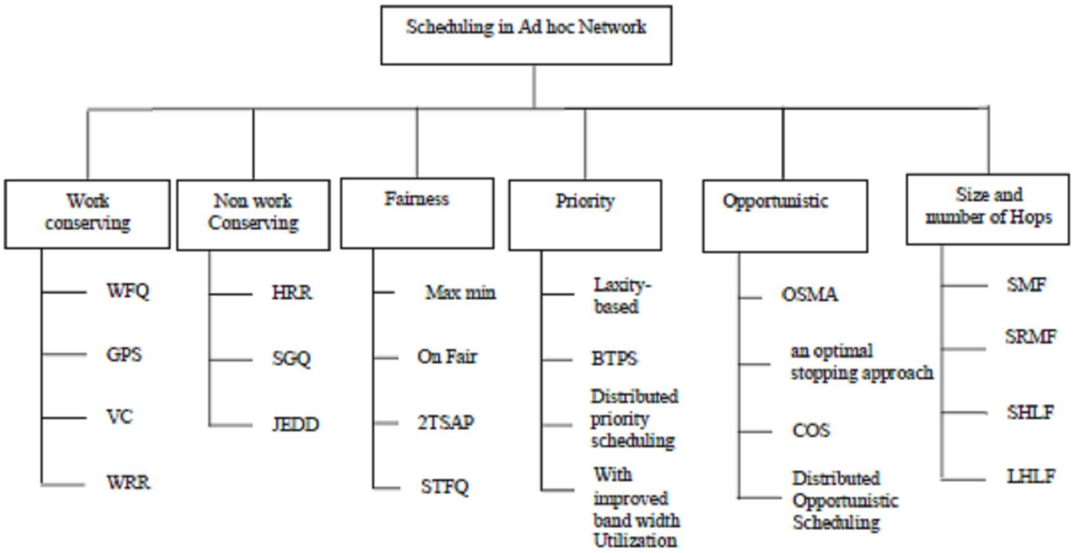


Figure 1 Classification of scheduling algorithms in Ad hoc Network

6.2 Classification of scheduling algorithms in Ad hoc Networks [10]

Ad hoc networks have several features including possible frequent transmissions of control packets due to mobility, the multi-hop forwarding of packets, and the multiple roles of nodes as routers, sources, and sinks of data that may produce unique queuing dynamics. We believe that the choice of scheduling algorithm to determine which queued packet to process next may have a significant effect on overall end-to end performance when traffic load is high. This belief motivated us to evaluate several applicable scheduling algorithms.

6.3 The most common queuing disciplines:

As part of the resource allocation mechanisms, each router must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. Various queuing disciplines can be used to control which packets get transmitted (bandwidth allocation) and which packets get dropped (buffer space). The queuing discipline also affects the latency experienced by a packet, by determining how long a packet waits to be transmitted. Examples of the common queuing disciplines are first-in first-out (FIFO) queuing, priority queuing (PQ), and weighted-fair queuing (WFQ).

We will study how the choice of the queuing discipline in the routers can affect the performance of the applications and the utilization of the network resources.

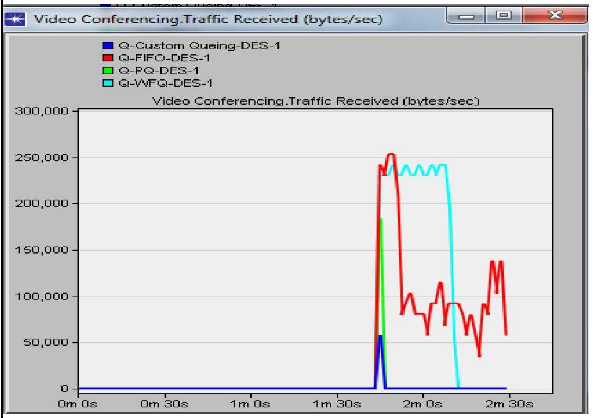
7 View the results:

Figures A, B, C, D and E show the results.

8) Simulation and Analysis

8.1 Traffic Received:

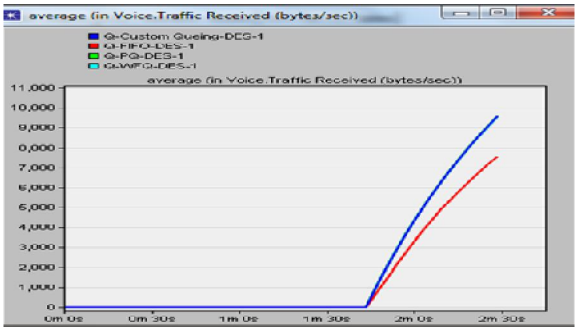
8.1.1 Fig. (A) Video Conferencing: Traffic Received (bytes/second)



-Is the average bytes per second forwarded to all video conferencing applications by the transport layers in the network.

-Fig. (A), shows traffic Received statistics for Video conferencing, where it can be observed that in cases of CQ, FIFO, PQ and WFQ video receiving rate graph WFQ is always higher than the performance graph of FIFO and PQ is lower and CQ is the lower.

8.1.2 Fig.(E) Traffic Received (bytes/second)



Figures (E) shows traffic Received statistics for VoIP, where it can be observed- that as the traffic increased the performance graph line increased in both group of CQ, FIFO, The performance graph line of FIFO group is always lowered compared to CQ

8.2 Traffic dropping: Four simulations have been executed using OPNET 17.1 software for every queuing scheme in terms of packet dropping, traffic receiving, packet delay variation and packet end-to-end delay and it is tested for Video Conferencing and Voice Traffics **[8, 9]**.

8.2.1 Fig.(B) IP Traffic Dropped (packets/second)

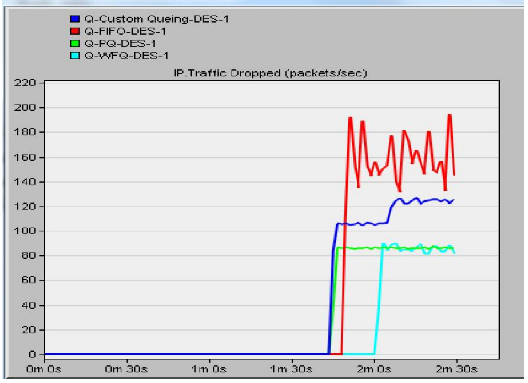


Fig (B): shows traffic dropping statistics.

-Is the number of IP Datagrams dropped by all nodes by in the network across all IP interface.

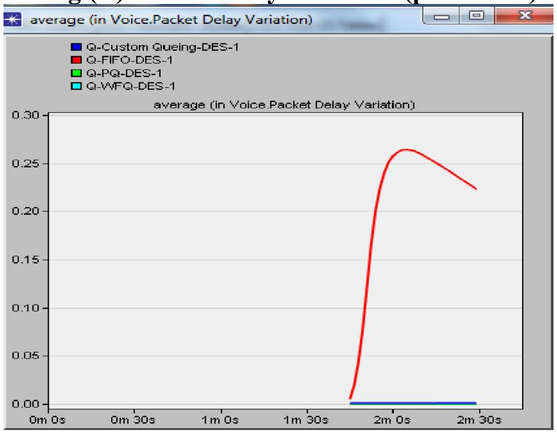
-In case of FIFO,CQ, PQ, WFQ (Fig B) the packet drop starts at near 120 sec. Packet drop for FIFO in this case is higher, CQ is semi lower, PQ and WFQ is lower.

8.2.2 The reasons for dropping an IP datagram can be any one of the following:

- insufficient space in the central processor's queue.
- insufficient space in a slot processor's buffer (only when slot based is enabled).
- Maximum number of hops exceeded by an IP datagram.
- On non-routing nodes, for destination that are more than one hop away, a local router Interface wasn't found to be used as next node.

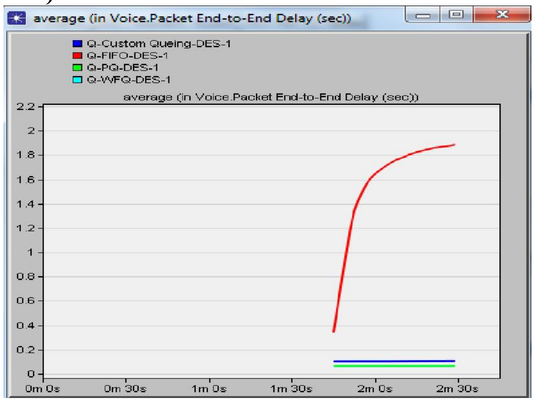
-On routing nodes, the route table looks up failed to yield a route to the destination.

8.3 Fig.(C) Packet Delay Variation (packet/sec)



-Figures (C) shows Packet delay variation time for VoIP, for all the cases such as time increase or traffic increase CQ, PQ and WFQ groups packet delay time line always showsthe same characteristics that is packet delay time is nearly zero and for FIFO group.it is always higher.

4. 8 Fig. (D) Packet End-To-End Delay (For VOIP)



-

(Fig.: D) shows Packet end to end delay time for VoIP. For all cases, such as time increase or traffic increase, PQ and WFQ groups packet end to end delay line always shows the same characteristics that is packet end to end time delay is nearly zero and FIFO group is always higher.

9 Conclusion

From Figures A, B, C, D and E:

It's found that the new algorithm "Custom Queuing Algorithm" is the best than FIFO, in packet end-to-end delay, packet delay variation and is better in IP PQ and WFQ traffic dropped.

Corresponding author
Adel Fouad El-Kabbany
Higher technology Institut
kabany_2008@yahoo.com

10 References:

(1) Xinran Wu, Clement, Yuen Yan Gao Hong Wu and Baochun Li.: "Fair Scheduling with Bottleneck Consideration in Wireless Ad-Hoc Networks", Department of Computer Science Department, University of Toronto.
(2) Luo H., S. Lu and V. Bharghavan (2000): "A new model for packet scheduling in multi-hop wireless networks", ACM MOBICOM'00, August.
(3) Luo H., P.Medvedev, J. Cheng, S. Lu. (2001): "A self-coordinating approach to distribute Fair queuing in ad hoc wireless networks", IEEE INFOCOM' 01, April.
(4) Byung Gon and Chun Mary Baker, Evaluation of Packet Scheduling Algorithms x in Mobile Ad Hoc Networks. Computer Science Department, Stanford University.
(5) Kanodia V, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly(2001):" Distributed Multi- Hop Scheduling and Medium Access with Delay and Throughput Constraints". In Proceedings of ACM MOBICOM, Rome, Italy, July.
(6) Byung Gon Chun. (2002): "Evaluation of Scheduling Algorithms in Mobile Ad Hoc Networks", Stanford University, May.
(7) Muhammad Mahbub Alam, Md. Abdul Hamid, Choong Seon Hong: "QoS-Aware Fair Scheduling in Multi-hop Wireless Ad Hoc Networks", Department of Computer Engineering, Kyung Hee University (8) Queuing Disciplines, Lap - 9. (9) Opnet 17.1 (licensed - Authorized).
(10) Annadurai C. (2011): "Review of Packet Scheduling Algorithms in Mobile Ad hoc Net."International Journal of Computer Applications, Volume 15– No.(1), Feb.

7/30/2012