

A Distributed Scheduling Scheme Providing QoS in Multimedia Ad Hoc Wireless Networks

Hyunho Yang

School of Electronic & Information Engineering, Kunsan National University,
San 68, Miryong-dong, Kunsan, Jeonbuk, 573-701, Republic of Korea.
hhyang@kunsan.ac.kr

Abstract. Providing Quality of Service(QoS) in distributed networks, such as multimedia ad hoc wireless network, also requires well defined scheduling schemes. However, due to the distributed nature of ad hoc networks, nodes may not be able to determine the next packet to transmit as in the centralized networks. Thus, it is a non-trivial issue to provide bounded delay guarantee, with fair share of resources. In this paper, we implement a scheduling scheme named delay guaranteed fair queueing (DGFQ) in a multimedia ad hoc wireless network with distributed manner. According to the performance evaluation results, both average and maximum delay could be controlled with varying *service differentiation coefficient*. In summary our new scheme can manage the delay performance of multimedia traffic in the distributed network environment.

Index terms-- Fair queueing, Ad hoc network, Quality of Service (QoS), Multimedia network.

1 Introduction

The multimedia ad hoc wireless network is quite an attractive issue since it offers a flexible solution to enable delivery of multimedia services to mobile end users without fixed backbone networks. As a distributed network technology, it also required to provide a set of applications, e.g., both error-sensitive and delay-sensitive applications, over the bandwidth-constrained wireless medium. In practice, to implement those applications over the distributed networks aforementioned, the issue of providing fair and delay bounded channel access among multiple contending hosts over a scarce and shared wireless channel is essential.

Fair queueing has been a popular scheme to provide fair share of resources among nodes according to their application requirements in both wireline and packet cellular networking environments [1]-[6]. However, the problem of designing fully distributed, scalable, and efficient fair scheduling algorithms in the shared-channel ad hoc wireless network remains largely unaddressed. In essence, the unique characteristics of ad hoc wireless networks such as location-specific contention create spatial coupling effects among flows in the network graph, and the fundamental notion of fairness may require non-local computation among contending flows. Adding these features together, fair queueing in shared-channel multihop wireless environments is no longer a local property at each output link

and has to exhibit global behaviors; this has to be achieved through distributed and localized decisions at each node.

In some related works the fair packet scheduling issues have addressed, in particular, on the aforementioned problems in ad hoc wireless networks,[7]-[9]. The focus of [7], [8] has been the problem formulation and an appropriate ideal centralized model for fair queueing in shared-channel multihop wireless networks. They also proposed a distributed fair scheduling implementation scheme, which merely approximate the centralized model. In [9], they devised distributed and localized solutions such that local schedulers self-coordinate their local interactions to achieve the desired global behavior. They also propose a suite of fully distributed and localized fair scheduling models that use local flow information and perform local computations only. Though the contributions stated above, [9] mainly addressed on the fairness of the overall throughput performance for the various usage scenarios without consideration of the QoS factors such as delay performance especially for the multimedia ad hoc wireless networks.

In [10], they propose a new fair queueing scheme i.e., delay guaranteed fair queueing (DGFQ), guaranteeing bounded delay for multimedia services. DGFQ scheme is basically a generalized process sharing(GPS) based fair queueing scheme with some modifications to guarantee bounded delay. In detail, the *service differentiation coefficient* was introduced to apply additional weight factor for the delay guaranteed (DG) class over non-delay guaranteed (NG) class. With this policy, DGFQ provides better delay performance for DG class at the same fairness guarantee without serious increase of computational complexity. However [10] has focused on the centralized network, rather than distributed one e.g., ad hoc wireless networks.

In this paper we implement the delay guaranteed fair queueing (DGFQ) in the multimedia ad hoc wireless network using the distributed fair queueing protocol to verify the controllability and adaptability of DGFQ on the bounded delay requirement in multimedia ad hoc wireless networks. Through the results of performance evaluation, we can conclude that DGFQ also performs well to control bounded delay in multimedia ad hoc wireless networks

The rest of the paper is organized as follows. Section 2 summarizes the delay guaranteed fair queueing (DGFQ). Section 3 describes the network model for ad hoc fair scheduling. In Section 4 we describe on distributed implementation of delay guaranteed fair queueing (DGFQ) in the multimedia ad hoc wireless network. Section 5 presents a simulation-based performance evaluation of the implementation, and, finally in Section 6 we conclude our work.

2 Delay Guaranteed Fair Queueing (DGFQ)

In delay guaranteed fair queueing (DGFQ)[10], two tags i.e., a start tag and a finish tag, are associated with each packet. Packets are scheduled in the increasing order of the start tags of the packets. Furthermore, $v(t)$ is defined as a virtual time function which calculates the start tag of the packet in service at time t .

Finally, in DGFQ scheme, there is a certain interval of time in which all flows are scheduled at least once, we call it *scheduling interval*.

All flows are classified into a number of classes according to their delay bound requirements. The simplest and basic classification is to make two classes, one for delay guaranteed (DG) flows and the rest for non delay guaranteed (NG) flows. In our scheme, we introduce the *service differentiation coefficient*, α ($0 < \alpha \leq 1$), to handle each flow class differently. By varying α , we can customize delay bound for individual flows i.e., adjust the relative service order of each flow in a scheduling interval.

The complete algorithm is defined as follows.

1. On the arrival of p_f^j , the j^{th} packet of flow f , is stamped with start tag $S(p_f^j)$, computed as

$$S(p_f^j) = \max\{v[A(p_f^j)], F(p_f^{j-1})\} \quad j \geq 1 \quad (1)$$

where $A(p_f^j)$ is the arrival time of packet p_f^j , $v[\cdot]$ is the virtual time function for the given arrival time and $F(p_f^j)$ is the finish tag of packet p_f^j .

The finish tag of packet p_f^j is defined as

$$F(p_f^j) = S(p_f^j) + \alpha_f \frac{l_f^j}{\phi_f} \quad (2)$$

where $F(p_f^0) = 0$, ϕ_f is the weight of flow f , l_f^j is the length of packet p_f^j , and α_f ($0 < \alpha_f \leq 1$) is the *service differentiation coefficient* for flow f . $\alpha_f=1$ for NG class or appropriate value for DG class.

2. Initially the system virtual time is 0. During a busy period, the system virtual time at time t , $v(t)$, is defined to be equal to the start tag of the packet in service at time t . At the end of a busy period, $v(t)$ is set to the maximum of finish tag assigned to any packets that have been serviced by then.
3. Packets are serviced in the increasing order of the start tags; ties are broken arbitrarily.

3 System Model

3.1 Network Model

In this paper, we consider a packet-switched multihop wireless network in which the wireless medium is shared among multiple contending users, i.e., a single physical channel with capacity C is available for wireless transmissions. Transmissions are locally broadcast and only receivers within the transmission range of a sender can receive its packets. Each link layer packet flow is a stream of packets being transmitted from the source to the destination, where the source and destination are neighboring nodes that are within transmission range of each

other. Two flows are contending with each other if either the sender or the receiver of one flow is within the transmission range of the sender or the receiver of the other flows [11]. We make three assumptions [11]-[13]: (a) a collision occurs when a receiver is in the reception range of two simultaneously transmitting nodes, thus unable to cleanly receive signal from either of them, (b) a node cannot transmit and receive packets simultaneously, and (c) neighborhood is a commutative property; hence, flow contention is also commutative.

In addition, we do not consider non-collision-related channel errors. For simplicity of presentation, we only consider fixed packet size in this paper, which is a realistic assumption in typical wireless networks.

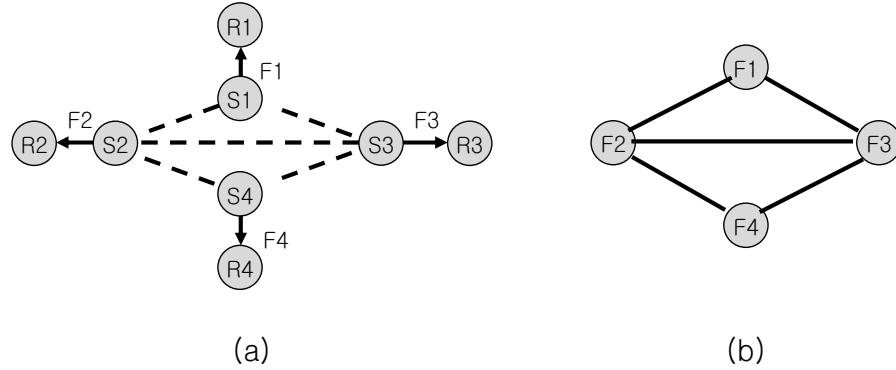


Fig. 1. Node graph and flow graph in location dependent contention.
 (a) Original node topology graph (b) Flow graph

3.2 Flow Contention Graph

To visualize the contending flows in the network, we introduce the flow contention graph which precisely characterizes the spatial-domain, as well as the time-domain contention relationship among transmitting flows. In a flow graph, each vertex represents a backlogged flow, and an edge between two vertex denotes that those two flows are contending with each other. If two vertices are not connected, spatial reuse is possible because those two flows can transmit simultaneously.

As an example, Figure 1 shows the simple network consists of 8 nodes and 4 flows. The dashed lines in the node graph represents tow nodes are in the communication range. Each node in an ad hoc wireless network maintains information for flows within one-hop neighborhood in the flow contention graph. In Figure 1, one-hop neighborhood of flow F_1 includes F_2, F_3 . Therefore, for given flow f , it

is required to maintain flow information for flows that are within the transmission range of either f 's sender or its receiver. However, for any given node, our goal is to maintain flow information (e.g., service tags) for flows only within its one-hop neighborhood in the node graph, even though one hop neighborhood in a flow graph will translate to the two-hop neighborhood in the real node graph in practice. This means that no node needs to be aware of flow information at nodes that are more than one hop away in the node graph.

4 Distributed Implementation of DGFQ in Ad Hoc Wireless Networks

4.1 Basic Scheduling Operations

The detailed operations for distributed implementation of delay guaranteed fair queueing (DGFQ) in multimedia ad hoc wireless network consist of the following four parts:

- *Local state maintenance*: Each node n maintains a local table E_n , which records each flow's current service tag for all flows in its one-hop neighborhood of the flow graph. Each table entry has the form of $[f, T_f]$, where T_f is the current service tag of flow f , e.g., the most recent start tag of flow f .
- *Tagging operations*: Two tags, i.e., a start tag and a finish tag, are assigned for each arriving packet, using DGFQ algorithm described in Section ??, for each flow f in the local table.
- *Scheduling loop*: After the tagging operation, at the sender node n of a flow f , the following procedure is performed, whenever the node n hears that the channel is clear,
 - (a) if the flow f has the smallest service tag in the table E_n , of node n , transmit the head-of-line packet of flow f immediately;
 - (b) otherwise, set the backoff timer B_f of flow f as

$$B_f = \sum_{g \in S} I(T_g(t) < T_f(t)),$$

where g is a flow entry of table E_n and $I(x)$ denotes the indicator function, i.e., $I(x) = 1$, if $x > 0$; $I(x) = 0$, otherwise. Consequently, the value of B_f is equal to the number of flows in a table E_n which has smaller service tag than flow f .

- (c) if flow f 's backoff timer expires, i.e., waits for B_f timeslots, and the channel is idle, transmit the head-of-line packet of flow f .
- *Table updates*: whenever node n hears a new service tag T'_g for any flow g on its table E_n , it updates the table entry for flow g to $[g, T'_g]$. Whenever node n transmits a head-of-line packet for flow f , it updates flow f 's service tag in the table entry.

Table 1. Table updates between transmission of flows 1 and 4.
(assume packet transmission time = 10)

	Table for F1	Table for F2	Table for F3	Table for F4
Before F1 and F4 Transmit	F1: T1=1	F1: T1=1	F1: T1=1	F2: T2=2
	F2: T2=2	F2: T2=2	F2: T2=2	F3: T3=3
	F3: T3=3	F3: T3=3	F3: T3=3	F4: T4=4
	F4: T4=4	F4: T4=4	F4: T4=4	
	Backoff=0	Backoff=1	Backoff=2	Backoff=2
After F1 and F4 Transmit	F1: T1=11	F1: T1=11	F1: T1=11	F2: T2=2
	F2: T2=2	F2: T2=2	F2: T2=2	F3: T3=3
	F3: T3=3	F3: T3=3	F3: T3=3	F4: T4=14
	F4: T4=14	F4: T4=14	F4: T4=14	
	Backoff=2	Backoff=0	Backoff=1	Backoff=2

We provide an illustrative example to show how the algorithm works. In the example, as shown in Figure 1, four flows are scheduled from the sender node to its respective receiver node and the dotted line denotes the two nodes are within the communication range. It is assumed that the initial virtual time $V = 0$, and the initial service tags for the four flows are $T_1 = 1$, $T_2 = 2$, $T_3 = 3$, $T_4 = 4$. The table maintained at each sender of the four flows and the backoff calculation and table updates before and after transmission of flows 1 and 4 are shown in Table 1. Flows F_1 and F_4 could transmit simultaneously because they are not neighboring flows (see Figure 1). After the transmission of F_1 and F_4 , the service tags of two flows are increased by 10, the packet transmission time, and subsequently, the backoff value of each flow table updated to the number of other flows which have smaller service tag (T) value.

4.2 Protocol Description

In the distributed implementation protocol, each data transmission follows a basic sequence of RTS-CTS-DS-DATA-ACK handshake, and this message exchange is preceded by a backoff for certain number of timeslots. When a node has a packet to transmit, it waits for an appropriate number of timeslots before it initiates the RTS-CTS handshake. In particular, the node checks its local table and sets a backoff timer for flow f to be the number of flows with tags smaller than the tag of flow f . This way, the local minimum-tag flow backs off for zero minislot and contends for the channel immediately. If the backoff timer of f expires without overhearing any ongoing transmission, it starts RTS carrying B_f , the backoff time of flow f according to the table, to initiate the handshake. If the node overhears some ongoing transmission, it cancels its backoff timer and defers until the ongoing transmission completes; In the meantime, it updates its local table for the tag of the on-going neighboring transmitting flow. When other nodes hear a RTS, they defer for one CTS transmission time to permit the sender to receive a CTS reply. When a receiver receives a RTS, it checks

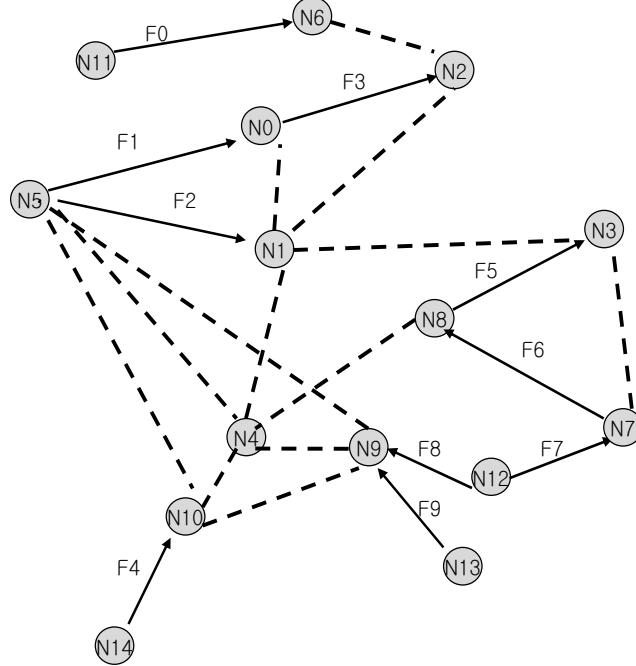


Fig. 2. Node graph of simulated multimedia ad hoc wireless network.

its local table. If B_f is greater than or equal to the backoff value for flow f in the receiver's local table, it responds with CTS. Otherwise, the receiver simply drops RTS. This procedure is required for maintaining the table information at both sender and receiver nodes. Once a sender receives the CTS, it transmits DS. When hosts hear either a CTS or a DS message, they will defer until the DATA-ACK transmission completes.

In order to propagate a flow's service tag to all its one-hop neighbors in the node graph and reduce the chance of information loss due to collisions during this service tag information propagation, the tag T_f for flow f is attached in all four packets RTS, CTS, DS and ACK, i.e., the *old tag* in RTS and CTS packets, and *updated tag* in DS and ACK packet.

5 Performance Evaluation

5.1 Simulation Environment

We use simulations to evaluate the performance of our distributed implementation of DGFQ in multimedia ad hoc wireless networks. The following is the simulation environment used in this simulation.

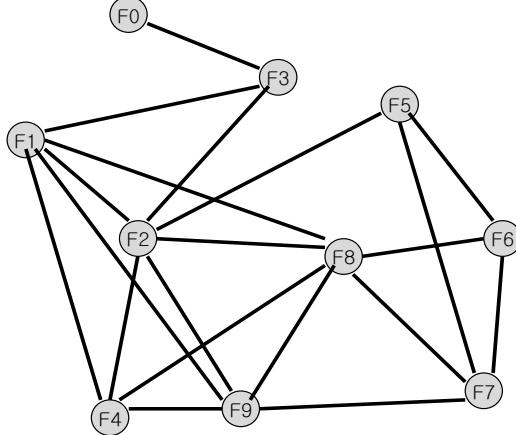


Fig. 3. Flow graph of simulated multimedia ad hoc wireless network.

The radio model is based on existing commercial wireless network with a radio transmission range of 250 meters and channel capacity of 2Mbit/sec which is typical capacity of current wireless mobile networks. Moreover, for the distributed implementation of DGFQ scheme, error free channel model is assumed to concentrate our evaluation work on the key features of proposed scheme, i.e., the controllability and adaptability of DGFQ scheme in distributed network environment such as multimedia ad hoc wireless networks to provide delay guaranteed service.

As the traffic source model, we choose the modified MPEG video source, described in [14]. Moreover, we assumed that all the sources have identical characteristics. In this video flow model, there are three types of frame, i.e., I, B and P frames. Each frame size is determined by a Lognormal distribution with a specified mean and standard deviation. A video source generates 24 frames per second.

Further, we consider a wireless ad hoc network which includes 14 nodes transmitting 10 flows. Figures 2 and 3 show the node graph and flow graph of simulated network respectively. To testify the controllability of DGFQ for guaranteed delay provision in distributed network environment, flow F_4 is controlled with varying the value of α , *the service differentiation coefficient*, for the range of $0 < \alpha \leq 1$, while other flows are assumed to have α value of 1. Finally, the simulation results for flow F_4 are compared with that of other contending flows and overall average.

Each simulation is run for 1000 seconds, and we selected average delay, maximum delay and throughput as the performance measures as in [10]. Detailed definitions and discussions for these measures are described in the following section.

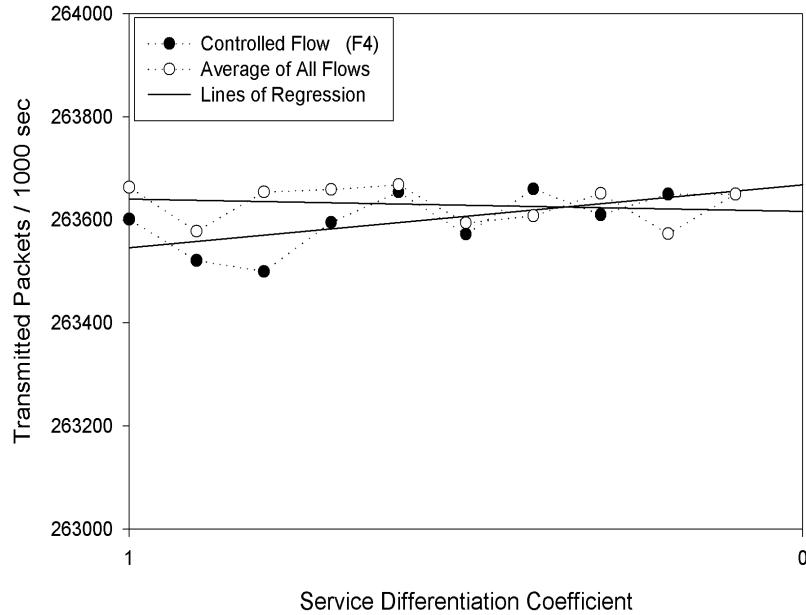


Fig. 4. Total transmitted packets with varying the value of *service differentiation coefficient* (α).

5.2 Results and Discussions

Throughput We used *throughput* as a fairness measure, which is total transmitted packets during the whole simulation duration, say, 1000 seconds. Figure 4 shows the throughput of flows with scattered points and their regression. Basically, as reported in [10], there is only a minor differences in throughput between flows either controlled ($F4$) or not (all other flows). In Figure 4, the white circle points and their regression line represents the average throughput of all flows and the black circle points and their regression line shows the throughput of the controlled flow ($F4$). Specifically, as shown in the figure, the difference is several hundred packets over more than 263,000 packets. The number of transmitted packets is inverse proportional to α , it is because α controls $F4$ with the share of channel in some extend, and, subsequently, it affects to the throughput of contending flows. It should also be noticed that it is possible to control individual flow with varying α .

Average delay In our work *average delay* is defined as the average time interval between the arrival and departure of a packet for a certain time duration.

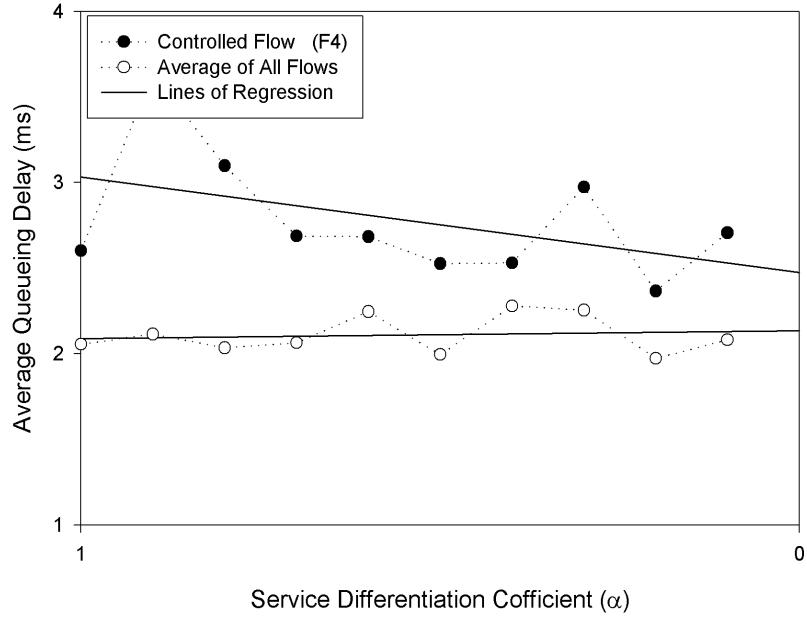


Fig. 5. Average delay with varying the value of *service differentiation coefficient* (α).

As shown in the Figure 5, the *service differentiation coefficient* α is the key parameter to manage delay performance. In the figure, the white circle points and their regression line represents the overall average delay, averaged for all flows, and the black circle points and their regression line shows the average delay of the controlled flow (F_4). With varying α we can control the average delay of flow F_4 . As shown in the figure, the control range of average delay could be 1ms. On the other hand, contrary to the throughput case discussed above, delay is proportional to α .

Maximum delay The *maximum delay* is another critical performance measure for real time multimedia flows. We define maximum delay as the maximum interval between the arrival and departure of a packet in the system in a certain duration of time, say, simulation duration. We can get the results simultaneously with average delay from the same simulation. As in the previous figures, in Figure 6, the white circle points and their regression line represents the overall maximum delay averaged for all flows, and the black circle points and their regression line shows the maximum delay of the controlled flow (F_4). From the figure, the control range of maximum delay is about 10ms. We can conclude that

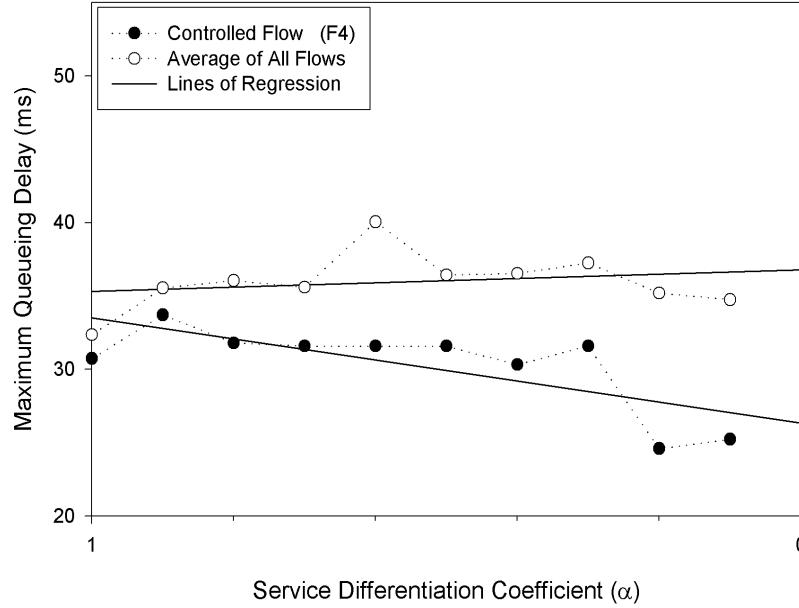


Fig. 6. Maximum delay with varying the value of *service differentiation coefficient* (α).

maximum delay could also be controlled with α , which means DGFQ controls the maximum delay also in distributed networks.

6 Conclusion

We implemented a delay guaranteed fair queueing(DGFQ) scheme, [10], distributively in the multimedia ad hoc wireless network environment. As far as throughput is concerned, there is only a minor differences between flows either controlled by *service differentiation coefficient* (α) or not. On the other hand, for delay performance, according to the simulation results, both average delay and maximum delay could be controlled by varying the value of α . In summary, it is clear that DGFQ can control the delay performance of multimedia traffic in the distributed network environment as well as centralized network.

We just consider about a limited network environment, i.e., stationary nodes with error-free wireless channel, which is too idealistic to apply our work in the practical systems. So, much more work should be done for the dynamic topology variation by mobile nodes in error-prone wireless channel case as a future work.

References

1. A. Parekh, "A generalized processor sharing approach to flow control in integrated services networks," *PhD Thesis*, MIT Laboratory for Information and Decision Systems, Technical Report LIDS-TR-2089, 1992.
2. J.C.R. Bennett and H. Zhang, "WF2Q: Worst-case fair weighted fair queueing," *IEEE INFOCOM'96*, 1996.
3. P. Goyal, H.M. Vin and H. Chen, "Start-time fair queueing: A scheduling algorithm for integrated service access," *ACM SIGCOMM'96*, August 1996.
4. M. Srivastava, C. Fragouli, and V. Sivaraman, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queueing with Channel-State-Dependent Packet Scheduling," *IEEE INFOCOM'98*, March 1998.
5. S. Lu, V. Bharghavan and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Trans. Networking*, August 1999.
6. S. Lu, T. Nandagopal, and V. Bharghavan, "Fair scheduling in wireless packet networks," *ACM MOBICOM'98*, October 1998.
7. N. H. Vaidya, P. Bahl and S. Gupta, "Distributed fair scheduling in a wireless LAN," *ACM MOBICOM'00*, August 2000.
8. H. Luo and S. Lu, "A topology-independent fair queueing model in ad hoc wireless networks," *IEEE ICNP'00*, November 2000.
9. H. Luo, P. Medvedev, J. Cheng and S. Lu, "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks" , *IEEE INFOCOM'91*, August 1991.
10. H. Yang and K. Kim, "Delay Guaranteed Fair Queueing (DGFQ) in Multimedia Packet Networks," *Proc. MMNS2003*, LNCS 2839, pp.170 - 182, Springer, September 2003.
11. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A medium access protocol for wireless LANs," *ACM SIGCOMM'94*, 1999.
12. J. Ju and V.O.K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Trans. Networking*, 6(3), June 1998.
13. Z. Tang and J.J. Garcia-Luna-Aceves, "A protocol for topology dependent transmission scheduling in wireless networks," *WCNC'99*, September 1999.
14. S. Lee, K. Kim, A. Ahmad, "Delay and data rate decoupled fair queueing for wireless multimedia networks," *Proc. GLOBECOM '02*, 2002, vol. 1, pp.946-950.