A Survey of Wireless Fair Queuing Algorithms with Location-Dependent Channel Errors

Anca VARGATU, Ionut BOSOANCA
Department of Computer and Communications Engineering,
Faculty of Automation, Computers and Electronics, University of Craiova
ankutza_vargatu@yahoo.com, ionutbosoanca@yahoo.com

The rapid development of wireless networks has brought more and more attention to topics related to fair allocation of resources, creation of suitable algorithms, taking into account the special characteristics of wireless environment and insurance fair access to the transmission channel, with delay bound and throughput guaranteed. Fair allocation of resources in wireless networks requires significant challenges, because of errors that occur only in these networks, such as location-dependent and bursty channel errors. In wireless networks, frequently happens, because interference of radio waves, that a user experiencing bad radio conditions during a period of time, not to receive resources in that period. This paper analyzes some resource allocation algorithms for wireless networks with location dependent errors, specifying the base idea for each algorithm and the way how it works. The analyzed fair queuing algorithms differ by the way they treat the following aspects: how to select the flows which should receive additional services, how to allocate these resources, which is the proportion received by error free flows and how the flows affected by errors are compensated.
Keywords: *Fair Scheduling, Wireless Networks, Location Dependent Channel Errors, Scheduling Algorithms*

1 Introduction

Wireless networks have been increasingly developed, and issues related to ensuring fair queuing have become important. Based on algorithms for wired networks, other algorithms have been developed for wireless networks, which deal with properties characteristic of these networks, such as location-dependent or bursty errors, which are not dealt in the algorithms for wired networks as there do not appear. Bursty errors can interrupt continuous services of a flow, while location-dependent errors can make flows affected by errors to receive more services than they would have normally received, this violating the fairness and bounded delay properties. Thus, because of possible errors, fair allocation of resources in wireless networks involves significant challenges. In wired networks fluid fair queuing became a known concept for providing quality of service and fair channel access with bounded delays. As it is not valid for wireless networks, the subsequently proposed algorithms are designed to reallocate resources if there are location-dependent errors for one or more users in a transparent way for all users. In wireless networks, frequently happens, because interference of radio waves, that a user experiencing bad radio conditions during a time period, not to receive resources in that period. These resources are allocated to other users, which will receive more than would have received in an ideal network without errors. Therefore, it may be necessary to provide additional channel capacity for the users that have encountered transmission errors, once they have good conditions for packets transmission. The created algorithms for wireless fair queuing treat different the way how it is made the compensation of those flows which encountered channel errors and have lost their allocated services. Also, another feature of wireless channels is the capability of communication in multiple rates. Among the important features that should be presented by a scheduling algorithm for wireless networks are: efficient use of channel- the algorithm should not allocate a slot for transmitting to a user which is affected by channel errors, fairness-the algorithm must allocate resources properly to users, bounded delay -

for each user must be guaranteed a limited delay in sending data, support for QoS, application. Increasing demands to achieve real-time applications and to ensure quality of services led to the development of a growing number of algorithms which provide fair allocation of the resources. The difference between the various proposed algorithms consists in how to make compensation to those flows which have encountered channel errors and have not received the appropriate quantity of services. So if there is a backlogged flow (a flow which has data to send) which can not send due to channel errors, its appropriate resources are shared to the backlogged flows, with an error-free channel. Those flows gain additional services, while the one affected by errors lose them and the delays of its packets increase. When leaving the error state, it has to recover lost service taking back resources from flows which profited, but a certain proportion, so its return from error state should not affect the flows which have good conditions to transmit.

2 Wireline fair queuing algorithms

Initially, the problem of fair queuing was encountered in routers, which used a single queue for all outputs, and the server sends the first packet from queue; if a flow sends many packets in queue, increases the waiting time of the packages belonging to the other flows. Thus, it was proposed the existence of a queue for each flow. The server verifies all queues and each queue which is not empty sends the first packet. A round robin method does not ensure fairness because packets have not equal lengths and a flow that sends large packets occupies more time the transmission channel and increases delay for flows with small packets. Wang et all remember in [1] that it was used "bit round fair queuing" based on processor sharing to keep track of the amount of services received by each flow, by sending one bit in a round from each queue. This method was not efficient because it does not take into account the insurance of different qualities of service.

GPS scheme [1] was developed after an ideal model, but it can not be implemented because it considers that the server can send packets simultaneously from all non-empty queues and the traffic is infinitely divisible. GPS scheme associates each flow a weight r and the channel capacity is shared among flows according to their weights. Fair queuing algorithms created for wired networks provide the fair channel allocation and bounded delay properties. They were developed on GPS model, which could not be applied in reality. The most important algorithms that approximate GPS model are WFQ (Weight Fair Queuing), SFQ (Start-time Fair Queuing), WF^2Q (Worst-case Fair Weighted Fair Queuing).

2.1 Weighted Fair Queuing (WFQ)

The first algorithm developed for wired networks, based on GPS model was WFQ [1][2]. It associates each packet of a flow two tags, a start one and a finish one, calculated by formulas:

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

$$F(p_f^j) = S(p_f^j) + {l_f^j}/{r_f}, j \geq 1 \quad (2)$$

In these formulas (1),(2) $v(A(p^j_f))$ represents the virtual time, calculated for the actual arrival time (A-arrival time) of the packet j from user f; l^j_f is the length of that packet, and r_f is the weight associated to user f. WFQ sends packets in increasing order of finish tags. Initially, the finish tag is considered to be 0 ($F(p^0_f)=0$). Virtual time v(t) is defined in [3] as "the normalized fair amount of service time that each session should have received by time t ". Thus $S(p^j_f)$ and $F(p^j_f)$ represent the virtual time when it is sent the first, respectively the last bit from packet j of flow f in GPS model. Virtual time advances as:

$$dV(t)/dt = C(t)/\sum i \in B(t) r_i \quad (3)$$

Where C is the channel capacity and B(t) is the set of backlogged flows at time t. The advantage that it shows the WFQ versus the GPS is that the maximum delay reached by a packet of a flow is equal to a packet length (the packet with maximum length among all flows).

2.2 Start-time Fair Queuing (SFQ)

Another algorithm developed for fair queuing in wired networks is SFQ [4]. Like WFQ, it associates two tags for each packet arrived in the queue, an arrival tag (start tag - S) and a leave tag (finish tag - F), calculate by the same formulas above (1),(2). The packet with the smallest start tag will be sent first in the network. Initially virtual time of server is 0. How long there are packets in a queue, virtual time of server, for real time t, v (t) is equal to the start tag of the packet in service at the moment t, so virtual time changes only when it ends to send a packet. When there are no packets in queues, v (t) is equal to the maximum finish tag of the associated packets that were sent up to the time t. The packets are sent in increasing order of start tags. The algorithm demonstrates fairness among users for that each packet from a user is scheduled behind the packets from the same user. If a user transmits more than it should then its packets won't pass in front of the packets from other users.

Goyal et al show in [4] that the WFQ simulation is computationally expensive and requires constant server capacity; if the server capacity varies WFQ algorithm do not fulfill the property of fairness, this happens for variable rate servers too. Unlike WFQ, SFQ algorithm ensures fairness in the allocation of resources even when server rate and capacity change; also, SFQ is computationally efficient.

2.3 Worst-Case Fair Weighted Fair Queuing (WF^2Q)

To solve the problem that services offered by WFQ can be long before the services approximated by GPS, WFQ algorithm was extended to the Worst-case Fair Weighted Fair Queuing (WF^2Q) [5], which is neither forward nor back more than one package length. For avoiding oscillations between high and low services for a flow, WF^2Q is also more suitable for the resumption of the congestion and control algorithms. In a system that uses WF^2Q, the server selects the following packages which must be processed among those packages the GPS system would have sent. Among these packages it is chosen that one who has minimum reference time in GPS. WF^2Q is a better packet approximation algorithm of GPS than WFQ. It provides almost identical service with GPS, the maximum difference is no more than one packet size. The problem with WF^2Q is the time complexity for computing the virtual time.

3 Wireless fair queuing algorithms

The problem of resource allocation in wireless network is more complicated than in wired network because wireless channel is influenced by additional factors, which a wireless scheduling mechanism needs to take into account. Some of these factors are: location dependent errors, higher probability of transmission errors/error bursts, interferences with other radio sources. In a wireless network, it may happen that some users have good channel conditions while others haven't and after a while, a different set of users have available channels. Therefore, it may be necessary to provide additional channel capacity for the users that have encountered transmission errors, once they have good channel state.

The above presented algorithms can not be applied in this case because if a flow keeps unchanged its packets tags when it can't send data their values remain very small. After the flow leaves the error state the server selects only it until the values of tags associated to packets are equal or larger that other packets tags values belonging to other flows. This increases the delay for all flows that have good channel condition. If it would update the tags values of a flow while its channel state is bad, it couldn't ever recover lost services from other flows that have benefited. Figure 1 below illustrates the case when there are four users connected to the network, one of them encounters channel errors and no longer receives resources.

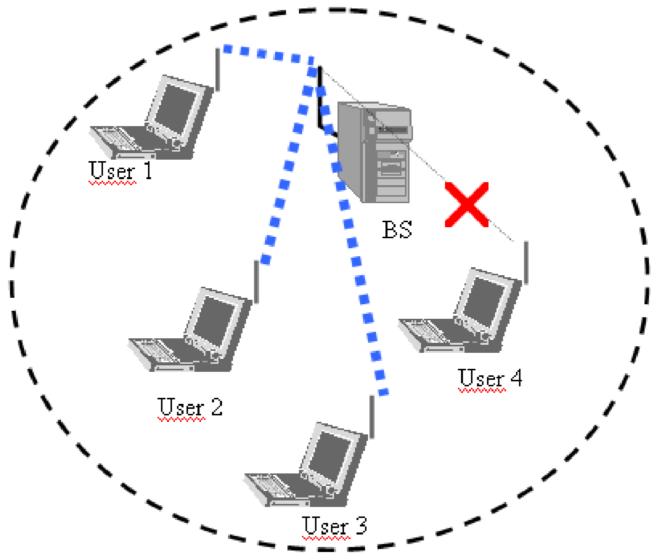

Fig. 1. Example of wireless network with location dependent errors: the fourth user loses the contact with the base station because of an error

The main categories of scheduling algorithms in wireless networks are: algorithms that are based on an error free service model, algorithms that use a compensation model, that consider traffic types and that use adjustment weights. More information can be found in [1]. There are used two types of systems, an ideal or reference one and a real one; in ideal system no errors occur and runs a fair scheduling algorithm for wired networks and in the real system errors affect the good transmission of information and runs a fair scheduling algorithm for wireless networks. In this way it can be observed the difference between the resources that a user should receive if it never encounters channel-errors and the resources that it actually receives when the channel condition is sometimes affected by location-dependent channel errors. Based on this fact users are classified into three categories, depending on the resources allocated to them. If we assume that it exists the ideal system in which there is no error, and compare the resources received by a user with the resources received by the same user in the real system, affected by errors, result that a user can be:

- *In advance (leading):* if it receives more resources than it would have received in the ideal system;
- *Synchronous (satisfied):* if the resources received are equal in both systems;
- *Behind (lagging):* if it receives fewer resources than it would have received in the ideal system;

To identify the users requirements in a radio network, it is defined a wireless fair service model for fair allocation of radio channels, with the following properties [6]:

1. Short-term fairness for synchronous backlogged flows that perceive an error-free channel;
2. Short-term limitations of bandwidth capacity for flows with channels unaffected by errors;
3. Limitations of packet delay conditioned by channel state;
4. Long-term fairness for backlogged flows with limited channel errors;
5. Long- term limitations of bandwidth capacity for all flows with limited channel errors;
6. Support for both delays and errors “sensitive” from data flows;

Property 1 ensures that resource allocation is fairly realized among flows that have packets in queue, are in accordance with their associated error-free service and can send packets. Property 2 specifies that if a flow has received additional services in an earlier moment of time, the return of the service in the following moments of time must be gradual so that the flow that has received more services than it should not be missed of access to channels at any time in the future. The requirement for bounded delay specified in property 3 is conditioned by the fact that the channel errors are limited for any flow during a period of time so that each flow i encounters at least e_i errors in any time interval T_i where e_i and T_i are parameters of the flow. Also property 3 specifies that as long as a flow has limited channel errors, no packets of that flow should wait indefinitely to be sent. Property 4 states that, if it is ensured long-term fairness, this is not violated as long as each backlogged flow has a sufficient number of error-free slots during which the flow can transmit its packets. Property 6 is useful for handling with delays and errors "sensitive" in the flows with channels prone to errors. A wireless channel model commonly used in the study of scheduling algorithms is the Markov model [7], which proposes two channel states: a state without errors (good) and a state with errors (bad). When the link is in good condition throughout the transmission of a packet, this is sent / received successfully. Information such as number of users, their weights, channel state, number of packets from the queues of users must be known by the scheduler in order to take some decisions. On the downlink, this information is easy to find, if the scheduler is at the base station (BS) level, while the uplink should be provided some means to collect information about packages in the queue of users (mobile stations, MS) or to inform them when it's time to send. There are several essential elements [2] that a scheduler has to include for efficiently resource allocation in a wireless environment, namely: error-free service model, compensation model, lead / lag model, separate slots and packets queues, means for monitoring and predicting the channel state.

Error-free service model: To highlight the occurrence of errors, there are considered two systems, one real and one ideal, which usually runs an algorithm for wired networks, the difference is given by the fact that in the real system, some users can't send packets when it is their turn because of errors and the algorithm from the ideal system is not implemented for the case when errors occur. In the ideal system (or reference), each time when a user is selected to send data, it sends the first packet in the queue. In contrast, when a user is selected in the real system, it is possible to be sent the package from the head of another queue, corresponding to another user. This happens when the selected user is affected by errors or is in advance and must give up some of its services.

Compensation model: represents the essential component of a resource allocation algorithm in wireless networks, which determines how to give up the leading flows to the surplus of services received and how to recover the lagging flows the lost services. A lagging flow regains its services when channel condition becomes good. In some cases, leading flows do not give up to all services held at the time when a lagging flow returns from an error state, but only a part of them. There are three possible situations in which leading flow gives up the additional services:

- *Leading flow:* gives up to all services until it becomes in sinc. In this case, there is the danger for that flow, if it has a high lead value due to errors that have affected other flows for a longer time, to accumulate large delays when lagging flows regain access to the channel;
- *Leading flow:* gives up only a fraction, constant or proportional to the lead value, of the services allocated to its. An important property which is respected in this case is partial degradation, which says that is not lost suddenly and for a long time the entirely access to resources, but only a fraction of them;
- *Leading flow:* doesn't give up to additional services gained. Disadvantage is that lagging flows can't recover the services lost while the access to channel has been affected by errors, unless there is a part of the channel width used to compensate those flows;

Lead / lag model: Each user has an associated lead and lag parameters to indicate the state in which the user is at any given time:

in advance (leading), behind (lagging) or synchronous (in sync).

Separate queues for slots and packages: In scheduling algorithms for wired networks, when a package arrives in the queue of a user, a tag is assigned, that specifies the arrival time. Due to absence of channel errors in these networks, the packet is sent and received successfully. Instead, wireless channel can cause some packets not to be transmitted successfully and be required to retransmit them, this leading to the association of new tags, and placing those packages in the end of waiting packets queue, this no longer keeps the initial order.

The scheduler should treat the case when the next packet is sent and the decision which package to be sent must belong to the flow. To make the difference between slots, which are units of channel allocation and packets, representing units for transmission of the information, it must exist separate queues. When a packet arrives in queue of a user is also generated a corresponding slot in the slots queue, that is assigned a tag based on the scheduling algorithm used. Each time when the scheduler determines which slot gains access to the channel, it is sent the first packet from packets queue associated with the corresponding slots queue. In every moment of time the number of slots from the slots queue is equal with the number of packets from the packets queue for a flow. Separating the slots and the packets in different queues, the scheduler can ensure fairness for both flows sensitive to errors, as well as those sensitive to delays.

Means for monitoring and predicting the channel state: the fair allocation of resources in wireless networks is possible if it is known the channel state for flows that have packets to send (they are backlogged). Location-dependent channel errors make that channel condition to be monitored continuously by each backlogged flow and according to a prediction of future state of the channel is sent or not the data to the scheduler. Usually, if slot i is good, it is considered that the next slot i+1 will be unaffected by errors and vice versa. The most important algorithms in this category are presented in the next section.

3.1 Idealized Wireless Fair Queuing (IWFQ)

Idealized Wireless Fair Queuing (IWFQ) [1] has as reference model WFQ algorithm and associates two tags for each packet, in the same way as WFQ does. The packet with the lowest finish tag is sent. When all flows have error-free channel state the two algorithms work identically. Virtual time is equal to virtual time of error free service. If a flow is selected to send data, and can't do this because encounters a location dependent channel error, IWFQ algorithm select other packet with the smallest finish tag. This process is repeated until it is find a packet that can be sent. After it was sent, packets tags are recalculated, then there are established service tags as being the finish tag of the first packet in queue for each flow. If a flow has no data in the queue, its service tag is set to ∞.

After the service tags were calculated for all flows, it's chosen that flow whose service tag has the lowest value and meets no channel errors and the first packet from its queue will be sent in the network. The process is resumed by recalculation of packets tags for all flows, as follows: consider a lagging flow is not allowed to recover more than b_i bits, computed as:

$$b_i = Bx(r_i / \sum_{k \in F} r_k) \qquad (4)$$

Where r_i is the flow weight, F is the set of flows, and B is a constant that limits the total lag (amount of lost services) for all flows. The length of packets with the finish tag value lower then virtual time is added for each lagging flow. The result should be bounded by b_i bits. There are kept only the first lowest tagged packets in queue, the rest are removed. This is the bound on lag set in IWFQ, together with bound of lead [1], to prevent those flows which accumulate a lot of packets with small finish tags values, unsent because of errors, affecting the other flows. Bound on lead refers to leading flows. If the start tag of the first packet (HOL- head of line) in queue of a leading flow is higher than virtual time with a l_i/r_i value, then the start tag and finish tag of the first packet in queue will be updated as follows:

$$S_{i,hol} = v(t) + l_i / r_i \qquad (5)$$

$$f_{i,hol} = s_{i,hol} + L_{i,hol} / r_i \qquad (6)$$

Where $L_{i,hol}$ is the length of the first packet of flow i and l_i represents the maximum value for the amount of additional services which can be gained by o flow. More details about this algorithm can be found in [8].

3.2 Channel-condition Independent Fair Queuing (CIF-Q)

Scheduling algorithm Channel-Condition Independent Packet Fair Queuing (CIF-Q) is an adaptation of the Start-time Fair Queuing (SFQ) algorithm for the radio channels. It was developed to provide the following properties [3]:

1) Delay bound and throughput guarantees: delay bound and throughput are guaranteed for users that don't encounter errors and they are not affected by the fact that other users encounter errors in transmission.
2) Long term fairness: if a user leaves the error state after a period, it should recover the service that it had lost because of errors. But this recovery should not affect the services guaranteed to users which don't experience errors, according to the first property.
3) Short term fairness: the difference between the normalized service received by any two users unaffected by errors, which have continuously packets in queues and are in the same state (leading, lagging, satisfied) should be limited.
4) Graceful degradation: During any period in which no error occurs, a leading user must have guaranteed at least a minimum fraction from the resources it would have received in the ideal system, without errors.

Based on these four properties, Ng et al present in [3] the way CIFQ algorithm works. To highlight the occurrence of errors, they consider two systems, one real and one ideal, in which runs the CIFQ, respectively SFQ algorithm, the difference being that in the real system, some users may not send packets when it's their turn because of errors and SFQ algorithm is not implemented for the case when errors occur. In the ideal system (or reference system), whenever a user is selected to send data, it will send the first packet in the queue. However, when a user is selected in the real system, it is possible to be sent another packet from the head of the queue corresponding to another user. This happens when the selected user is affected by errors or it is leading and must give up a part of its services.

In order to provide all the properties mentioned above CIFQ assigns a lag parameter to each flow, with a positive value for lagging flows and negative for leading flows. The CIFQ is work-conserving because all the time it makes true the formula:

$$\sum i \in A \;\; lag_i = 0 \qquad (7)$$

Where A represents the set of active flows and a flow is active when it is backlogged and has an error-free channel state. If a lagging or in sync flow i is selected to transmit a packet and it has bad channel, another backlogged flow j with good channel transmits, thus the lag parameter of flow i is increased and the lag parameter of flow j is decreased. To monitoring the degradation of service for the leading flows, CIFQ introduces an α parameter, means the fraction of service retained by a leading flow i , which gives up only a fraction equal to 1-α from its services.

3.3 WFQ with link level Retransmission (WFQ-R)

Kim &Yoon [17] proposed an algorithm to improve the WFQ-R with window-based retransmission because they consider that the existing algorithms need to predict perfectly the channel state and they don't consider a MAC algorithm. In the MAC layer the scheme of link level retransmission is frequently used for treat the channel errors.

WFQ-R doesn't need a channel prediction because it works well, providing fairness and throughput with the link level retransmission scheme. This algorithm is based on the fact that the share used for retransmission of data is considered debt of the retransmitted flow. Hence, when data must be retransmitted, the wireless resource used for resend this data is assign to the flow that retransmit. There are considered two kind of compensation types: Flow-In-Charge(FIC) and Server-In-Charge(SIC). The first one considers that the entire overhead used for retransmission is charge for the retransmitted flow, so if a flow that has encountered channel errors consumes more resources that has allocated for retransmit data, it won't use the next resources allocated to its and will make a concession to other flows. This type of compensation assures fairness among users, but it is too severe with the flows that experience frequently channel errors. The second type of compensation considered by this algorithm is SIC and supposes that all flows that have data to transmit are responsible for channel errors, depending on their weights, so the overhead is distributed over all flows that retransmit data. This is better then the FIC because the charging overhead is reduced and is correctly divided among flows that encountered channel errors according to their weights. The algorithm WFQ-R (*WFQ with link level Retransmission*) distributes the scarce wireless resources among all flows according to their weights, but considering also the resource consumption of the retransmissions. The compensation could be charged to the retransmitted flow only (an error-prone flow should take responsibility for its own channel condition) or the compensation is distributed over all flows proportionally to their weights. It combines the CIF-Q algorithm with a compensation system where the share used for retransmissions is regarded as a debt of the retransmitted flow to the others. Based on this algorithm proposed by Kim &Yoon [17], Elshafei & Baroudi proposed new wireless fair queuing algorithm to improve the WFQ-R with window-based retransmission. This scheme calculates the amount of extra resources a flow used during retransmission and later provides compensations to the other flows which used less resource. The amount to be charged on a flow is calculated based on the weight of the flow and extra resources used. Then it distributes this amount to backlogged flows in proportion to their weights. The authors showed that this algorithm can improve throughput and decrease queuing delay while maintaining fairness in the presence of errors [15], [16].

3.4 Server Based Fairness Approach (SBFA)

The basic idea of this algorithm [9] is to allocate an additional bandwidth for those flows that have been affected by location-dependent errors in order to be compensated without affecting the other flows. The reserved service is called long-term fairness server (LTFS). It is a virtual data flow, which shares bandwidth with the other flows. This algorithm makes the difference between slots queue (SQ) and packets queue (PQ) for a flow, except LTFS. When a packet arrives from a flow i, it is placed in the packet queue corresponding to that flow (PQ_i). In the same time a slot is created in the slots queue corresponding to flow i. For SBFA, any resource allocation algorithm for wired networks can be used as a reference model, but it is applied to the slots queue. Thus, if a slot is selected from SQ_i queue, the first packet from the PQ_i queue is sent, if the channel status is good for the user i. After the packet was sent both selected slot and sent packet are deleted from the queues. If they are considered, for example, two flows in the system, f1 and f2, using a round-robin mechanism to send packets, when the scheduler selects flow f1 to send, but the transmission channel for this flow has bad condition, it is moved a slot from the slots queue of f1, SQ1, in LTFS. The scheduler selects to send flow f2 instead of f1, which having good condition sends the first package from queue PQ2. This packet and the first slot from SQ2 are deleted. Next, it is also selected flow f2 because this time was its turn to send and sends another packet in the network. The next flow selected by the scheduler won't be f1, but the third flow in the system, LTFS, which is virtually. It is found that the first slot of LTFS belonging to f1 and is trying to send a package from PQ1. If the flow f1 is still affected by errors, is seeking the next flow that has good channel conditions.

SBFA algorithm provides a mechanism to compensate lagging flows, considering the order they should be compensated in the future. Depending on the requirements of flows that share the wireless channel, there may be one or more LTFS. Preferable is that flows with similar requirements to be distributed at the same LTFS.

3.5 Effort-Limit Fair scheduling algorithm (ELF)

ELF algorithm [10] belongs to the category of algorithms that modify the weights of the users. Most scheduling algorithms in wireless networks consider that the weights of the users are static and they don't change them during the entire resources allocation process, but this algorithm adjusts the weight of each user based on the errors rate that the user has met. There is a maximum weight which can be reached, defined by the power factor of each user. The basis of this algorithm is WFQ, which is extended by a dynamic weights adjustment mechanism.

Dynamic capacity of wireless link creates a series of problems for both users and network administrators. When the bandwidth drops significantly it may not be possible to ensure the bandwidth reserved to the users, and the problem is to which flow should be allocated less bandwidth. Also, if considers a wireless network with location-dependent errors, the decision that must be taken is related to how much extra transmission time should be allocated to users with a high rate of errors in disfavor of the others. Based on these two issues arises the question how the scheduler responds when it loses from the bandwidth.

ELF algorithm was designed to fulfill the following: in an error-free environment, the result obtained by a wireless scheduler should be identical to that obtained by a scheduler in wired networks, the services lost by a flow should be configured through administrative controls in the network and should not be proportional with the bandwidth or rate of errors for that flow, the channel capacity of a flow, which is lost due to location-dependent errors should be administrative limited, flows that have the same rates of errors should have the same amount of lost services, if a flow does not use its allocated bandwidth, it should be equitably distributed to other flows in the network. Figure 2 presents the basic concept behind this algorithm.

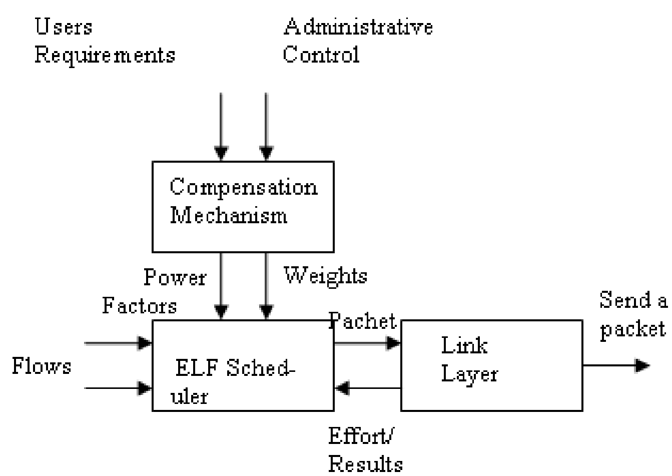

Fig. 2. System model for ELF algorithm

Considering that there are several flows that share the same channel, each flow i having a weight r_i, a power factor p_i and an error rate e_i, then the ELF algorithm defines a modified weight a_i associated to the flow i, calculated as follows:

$$a_i = \min(r_i/_{1-e_i}, p_i \times r_i) \quad (8)$$

By dynamic adjustment of the weight, the behavior of the scheduler can be controlled in the presence of errors. The algorithm is called effort-limited fair because it changes the users' weights to get fair results with a limited effort for each flow.

3.6 Wireless Fair Service (WFS)

The basic idea of "fluid fair queuing" algorithms is not to compensate flows that didn't have data to send when it was their turn. Thus, if a flow had no packets in the queue, and it was its turn to send, it can't claim additional services when it receives packets. The WFS algorithm [11] makes the difference between a service lost due to the lack of packets for being sent and a service lost due to channel errors. If a flow is backlogged but it can't send packets either because it faces a bad channel state, either because of network congestion avoidance, then it should be compensated, but the flows that are neither leading nor lagging should not be affected. A backlogged flow is compensated if it can't send data when the slot corresponding to its queue is selected only if another flow has sent a packet during this slot.

The compensation model is the following: if a flow doesn't have packets in queue or is affected by channel errors but no other flow can send data while its appropriate slot is selected, then it is not compensated. If a flow encounters channel errors but another flow can send instead then the first will be compensated later when it will exit from the error state, and that which has sent in its place will give up a slot. Thus, the leading flows give up to additional services gained by removing the transmission slots, and in case of those lagging they receive the lost services, transmitting data during the slots discarded by leading flows. This scheduling algorithm ensures transparency for short location-dependent errors, changing slots between different flows. A last case refers to in sync flows, which remain unaffected by the compensation model proposed by WFS.

As error-free service model, this algorithm uses fluid fair queuing model, but modified in that it establishes two types of weights for each flow, namely: a weight of transfer rates and a weight of packets delay. Considering a channel shared by a set of flows F, and each flow in this set has associated a rate weight r_i and a delay weight Φ_i. In addition it has associated two tags, a start one (S) and a finish one (F) for each packet from queue. For the packet k of the flow i, the two tags are calculated as follows:

$$S(p_i^k) = \max\{v(A(p_i^k)), S(p_i^{k-1}) + L_i^{k-1}/r_i\} \quad (9)$$

$$F(p_i^k) = S(p_i^k) + L_i^k/\Phi_i \quad (10)$$

$$dv/dt = C(t)/\sum_{i \in B} r_i \quad (11)$$

B (t) denotes the set of backlogged flows at time t, C (t) is the channel capacity at time t. At each time moment t, is transmitted the package with the lowest finish tag from those packets whose start tag is not greater than the virtual time v (t) to which is added the value of a parameter noted ς and named "lookahead" parameter of the system. This provides a measure of the number of packets over which the scheduler can jump without affecting long-term transfer rate. The parameter ς allows to the scheduler to advance the virtual time with ς rounds, having an important role in determining the flows that can send packets. Its value can vary between 0 and infinity allowing flexibility in the scheduling area control.

Virtual time advances similarly as in fluid fair queuing, each packet having associated a finish tag, which is the sum of the start tag of the package and the maximum number of rounds, in virtual time, which that packet can wait before it is sent.

In WFS packets are considered fixed-length. For each flow is assigned a parameter "lead counter" E(i) and a "lag counter" G(i). At least one of these parameters is set to 0, and both are limited by a maximum value $E_{max}(i)$ and $G_{max}(i)$. The set B (t) consists of those flows which have Q(i)> 0 or E(i)> 0, where Q(i) is the number of packets waiting in the

queue of flow i. Thus, the slots are generated for backlogged or leading flows. According to this algorithm, at each moment of time, a slot s is selected for transmission, which has the smallest finish tag among those slots that have the start tag less than the current virtual time plus the value of the look ahead parameter. The flow i to which corresponds the selected slot s can be in one of the following states: leading and wants to send a packet in slot s, leading and wants to gives up the slot s in favor of a lagging flow, synchronous or lagging.
Unlike the fluid fair queuing, WFS supports flows with high bandwidth requirements and large delay, and flows with low bandwidth demands and small delay. Questions that arise are: how a leading flow knows when to give up a slot for a lagging flow, which of the lagging flows must send a packet during the slot dropped by the leading flow, which is the channel allocation algorithm that must be applied. For all the cases before mentioned, there is a channel allocation algorithm.
Thus, when there is a leading flow i and is selected one of its appropriate slots, during which the flow wants to send a packet from the queue, the possible cases are presented bellow:
Case 1) If the flow is not affected by channel errors, the first packet in the queue is sent in the network. Otherwise, if there is a backlogged and lagging flow j, that slot is allocated to j, and E (i) and G (j) are decremented with 1. Otherwise, if there is a backlogged and leading flow j which has good channel condition and the value of the parameter "lead" does not exceed the maximum allowed value, that slot is assigned to its, and E (i) and E (j) is decremented, respectively incremented with1.
Otherwise, if there is no leading or lagging flow that can send packets and there is an error-free synchronous flow j, it receives the slot s and decrements with 1 the value of E(i) and increments with 1 the value of E (j). In the last case, if there is an error-free backlogged flow, it sends, and if there is no backlogged flow with a channel state unaffected by errors, the slot is deleted, without being assigned.
Case 2) If the flow i is leading and gives up that slot to compensate a lagging flow, there are the next situations:
If there is a backlogged error-free lagging flow j, it receives the slot s and decrements E(i) and E(j) with 1.
Otherwise, the slot is reassigned to flow i, if it has good channel conditions and this sends its first package from the queue.
Otherwise, if there is another leading flow j, backlogged and with an error-free channel, and the relationship $0 < E(j) < E_{max}(j)$ is true, flow j sends a packet, E (i) is decreased and E(j) is increased with 1.
Otherwise, it is searched a synchronous flow j, with an error-free channel state and the slot s is allocated to its, E (i) decreases its value with 1 and E (j) increases with 1. Otherwise, it proceeds as in the last situation from case 1.
Case 3) Flow i is synchronous or lagging: If i has an error free channel, sends the first packet in queue. Otherwise, if there is a backlogged and lagging flow j, no channel errors, that slot is allocated to its, and G (i) is incremented with 1 and G (j) is decremented with 1.
Otherwise, if there is a leading flow j, backlogged and error-free channel state and the relationship $0<E(j)<E_{max}(j)$ is true, flow j sends a packet and G(i) and E(j) are increased with 1.
Otherwise, if there is no leading or lagging flow that can send but there is an error-free synchronous flow j, this receives the slot s and increments with 1 the values of G (i) and E (j).
Otherwise, if there is an error-free backlogged flow, it sends a packet, and if there is no backlogged flow with a channel state without errors, the slot is deleted, without being assigned to anyone.

3.7 Traffic – Dependent wireless Fair Queuing (TD-FQ)

Most resource allocation algorithms in wireless networks don't take into account the difference among the nature of network traffic, leading to large delays for real-time flows. TD-FQ algorithm [12] proposes a scheduling method which takes into account the type of traffic for each flow. Its authors have based on the fact that most resource allocation schemes don't consider the difference among the nature of traffic, which can lead to long delays for real time traffic. This algorithm aims to reduce the delays of real-time packets caused by long waiting times in queues and to ensure fairness and bounded delays for all users in the network. Usually, real-time applications are delay-sensitive and if they are not treated more special than the others, the problem of connection between delays and weights could affect them and also the performance of the entire system. The traffic that occurs in the base station is separated into real-time applications and non-real-time applications.

TD-FQ algorithm is based on the CIF-Q algorithm, which has added an additional mechanism that offers higher priority to real-time flows to decrease their delays. Also, TD-FQ guarantees that treating especially real-time flows doesn't affect the others, providing bounded delays and fairness for these too. Figure 3 presents a simple system model for TD-FQ algorithm.

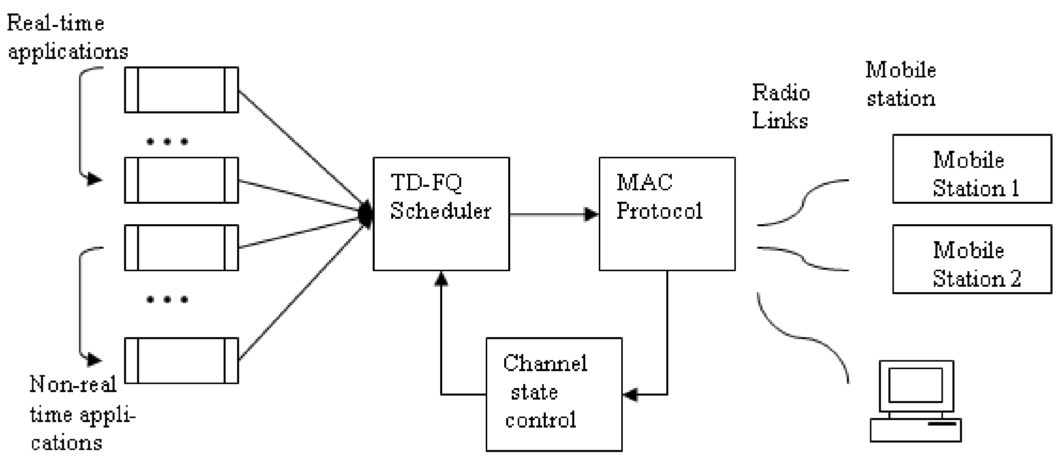

Fig. 3. System model for TD-FQ algorithm

One of the scheduler's roles is to verify the queues of the two types of flows and depending on the transmission channel state obtained from the "Channel state control" block, it sends the packet from the head of a queue into the network, using the MAC protocol, priority packets belonging to real-time flows. As CIF-Q algorithm and most scheduling algorithms, TD-FQ assigns a weight r_i to each flow i, representing the part of bandwidth that the system has to allocate to flow i in an ideal system. But in a real system the flow i might not receive all appropriate services. It also keeps the v_i and lag_i parameters.

3.8 Multi-rate wireless Fair Queuing (MR-FQ)

Wireless networks are characterized by bursty and location dependent errors. The most fair resources allocation methods in these networks take into account the occurrence of errors, most considering a simple channel model with two states, a good and a bad state. Multi-rates transmission, a technique increasingly used, leads to a question related to fairness among users of wireless networks, namely if the fair queuing algorithms should consider the average amount of time used by a user or the amount of received services.

Wang et al propose in [13] an algorithm called Multi-rate Wireless Fair Queuing which takes into account for resources allocation both average time that a user has access to resources and services received by a user, allowing it to transmit data in multiple transfer rates depending on radio conditions it faces and on the resources that it has lost because of errors. Considering multiple rates transmissions there are some problems related to fair allocation of resources. The first problem that arises is the difference between the amount of services that a user receives and the time allocated to that user by the server. The same amount of data is transmitted by a user with a lower transmission rate in more time. Thus it must be redefined the concept of virtual time. The second problem is that when a user affected by location dependent channel errors exits from this state, the system takes a while to compensate it, depending on the channel state, making difficult the creation of a compensation mechanism. The third problem relates to the performance of the system that can be degraded if there are many flows with low rates of transmission. Figure 3 presents a system model for MR-FQ algorithm.

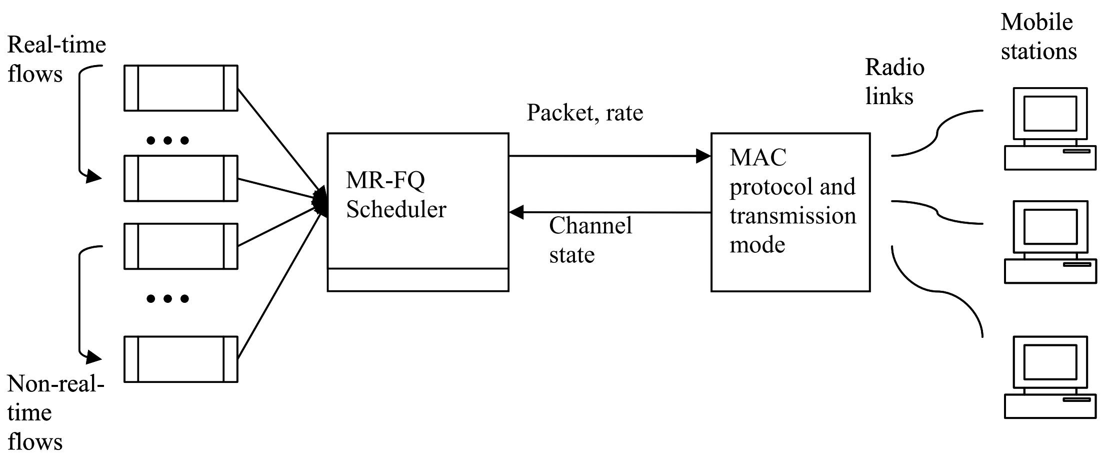

Fig. 4. System model for MR-FQ algorithm

In MR-FQ algorithm packets are sent using TDMA MAC protocol in a network which enables communication in multiple rates. The transmission rate is changed by the algorithm depending on the channel state and the lag parameter of each user. As a flow is behind in the real system from the ideal system, the transfer rate is lower, thus ensuring fairness among users and system performance, because the flows with low transmission rates don't lead to an increased transmission delay for the other flows .

4 Conclusions

Based on mobility and connectivity needs, wireless communication has recorded an important development in recent years. Due to rapid growth of wireless data services, the problem of providing quality of services and fair access to various channels has become increasingly important. In wired networks there are many algorithms for resource scheduling, which can provide fairness and bounded delay. These algorithms are developed after the GPS model, which ensures fair queuing, but they can't be used in wireless networks because don't treat cases when errors occur, such as location-dependent and bursty channel errors, the most common in wireless environment. Thus many algorithms have been proposed for resources allocation in wireless networks (scheduling algorithms or queuing algorithms), with varying degrees of complexity.

Future wireless networks are designed to support voice and data communications in a wide band. Such a network will be the basis for a wireless information society where information and computer services are available anywhere, anytime and to anyone. The application of wireless communication systems has greatly increased, and because radio frequency is a scarce resource, its efficient use has become important. Thus, different schemes and algorithms must be created for equitable allocation of resources.

References

[1] Wang You-Chiun; Tseng Yu-Chee ,"Packet Fair Queuing Algorithms for Wireless Networks", *Design and Analysis of Wireless Networks, Nova Science Publishers*, chapter 7, pp. 113-128 2005.

[2] Bharghavan V.,Lu S., Nandagopal T., "Fair queuing in wireless networks: issues and approaches", *IEEE Personal Commun*, vol. 6, pp.44-53, 1999.

[3] Ng T.S.Eugene; Stoica Ion; Zhang Hui, „Packet Fair Queuing Algorithms for Wireless Networks with Location-Dependent Errors", *Proceedings of IEEE INFOCOM'98*, pp 1103-1111, San Francisco, CA, USA, 1998.

[4] Goyal Pawan; Vin Harrick M.; Cheng Haichen, "Start-time Fair Queuing: A scheduling Algorithm for Integrated Services Packet Switching Networks ", *Proceedings of ACM SIGCOMM*, pp 157-168, 1996.

[5] J.C.R. Bennett, H. Zhang, "WF2Q: Worst-case fair weighted fair queueing", *Procedings of the IEEE INFOCOM*'96, San Francisco, CA, March 1996.

[6] Nandagopal, Thyagarajan; Lu, Songwu; Bharghavan, Vaduvur; „A unified Architecture for the Design and Evaluation of Wireless Fair Queuing Algorithms", *Kluwer Academic Publishers. Manufactured* in The Netherlands, 2002.

[7] H.S. Wang and .Moayeri, "Finite-state Markov channel: a useful model for radio communication channels", *IEEE Trans. Vehicular Technology*, vol. 44, pp.163-171, 1995

[8] Lu S., Bharghavan V., Strikant R., "Fair scheduling in wireless packet networks", *IEEE/ACM Trans. Networking*, vol. 7, pp. 473-489, 1999.

[9] P. Ramanathan and P. Agrawal," Adapting packet fair queuing algorithms to wireless networks", *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking*, pp.1-9, 1998.

[10] D. A. Eckhardt and P. Steenkiste, "Effort-limited fair (ELF) scheduling for wireless networks," *in INFOCOM*, vol. 3, pp. 1097–1106, 2000.

[11] V.Bharghavan, S.Lu and T. Nandagopal, „A wireless fair service algorithm for packet cellular networks", *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking*, pp.10-20, 1998.

[12]Y. Wang, S. R. Ye, and Y. C. Tseng, "A fair scheduling algorithm with traffic classification in wireless networks", *Computer Comm*., vol. 28, no. 10, pp. 1225-1239, 2005.

[13]Y. Wang, Y. C. Tseng and W.T. Chen, „MR-FQ: A fair scheduling algorithm for wireless networks with variable transmission rates", *Proceedings of IEEE INFOCOM* 2005.

[14] Rong Yu, Yan Zhang, and Shengli Xie "QoS Differentiated and Fair Packet Scheduling in Broadband Wireless Access Networks" EURASIP Journal on Wireless Communications and Networking Volume 2009, Article ID 482764, 12 pages doi:10.1155/2009/482764.

[15] Abdul-Rahman Elshafei, Uthman Baroudi, "Wireless Fair Queuing algorithm for window-based Link Level Retransmission 2008 IEEE/ACS International Conference on Computer Systems and Applications Doha, Qatar, March 31-April 04, 2008, ISBN: 978-1-4244-1967-8

[16] Ahmed Riza, Ling Teck Chaw, Phang Keat Keong, Hoh Kian Pee, "Improving QOS In WLAN Using Dynamic Weighted Fair Scheduling, pp 85-101 Malaysian Journal of Computer Science, Vol. 23(2), 2010

[17] N. Kim and H. Yoon, "Wireless packet fair queuing algorithms with link level retransmission," Computer Communications, vol. 28, pp. 713-725, 2005.

Anca VARGATU has graduated the Faculty of Automation and Computers, Politehnica University of Timisoara, Computer Science specialization in 2009. In October 2009 she started to study in the Computer and Communication Engineering Master at University of Craiova, Romania. Her interests are Vertical Handoff algorithms and resources allocation in wireless network.

Ionut BOSOANCA has graduated the Faculty of Automation, Computers and Electronics, University of Craiova, Computer Science specialization in 2009. He received two external scholarships, one in Germany in 2009, where he achieved his license project at “Universität der Bundeswehr, München”, and another in 2008 at the “University T.E.I of Piraeus”, Greece. In October 2009 he started to study in the Computer and Communication Engineering Master at University of Craiova, Romania. His interests are Vertical Handoff algorithms and resources allocation in wireless network.