Opportunistic Weighted Fair Queueing

Kinda Khawam, Daniel Kofman
GET/ENST Telecom Paris
Email: {khawam, kofman}@enst.fr

***Abstract*— The ongoing evolution towards the generalization of wireless access to multi-service networks stresses on the need for optimizing the control of radio resources and in particular, for designing efficient scheduling approaches. The scheduling mechanisms proposed for wireline links do not carry over to wireless interfaces due to the variability of radio links capacity. Efficient schedulers for the radio channel have to take into account information relative to the channel state when allocating resources to the various connected equipments. For the case of delay tolerant traffic, the scheduler may take advantage of terminals diversity. We propose in this paper a modified version of the Weighted Fair Queueing (WFQ) algorithm, termed OWFQ (Opportunistic WFQ), that notably increases the average system performance through opportunistic scheduling while fulfilling users' QoS needs in terms of minimum realized throughput.**

I. Introduction

The number of devices accessing the network through wireless interfaces is growing fast and will overtake, in the coming years, the number of wireline connected devices. This evolution exacerbates the difficulties raised by the scarcity of radio resources and increases the need for optimal utilization of the latter, especially in the case of nowadays multi-service networks. Scheduling is a key tool for optimizing resource allocation under the QoS constraints imposed by multi-service networks. The algorithms that have been proposed for adapting fair queueing to the wireless domain (e.g. [5]) make the assumption that the channel capacity is constant and try to make short bursts of channel errors transparent to flows by a dynamic reassignment of channel allocation over small time scales. In this paper we follow a different approach: rather than falsely assuming that the shared capacity is constant, we propose to enhance the radio link utilization by allowing the scheduler to make its decisions based on the knowledge of the various terminals' channel state. More precisely, we propose a modified version of the Weighted Fair Queueing (WFQ) algorithm [1], termed OWFQ (Opportunistic WFQ), that notably increases the average system performance through opportunistic scheduling while fulfilling users' QoS needs in terms of minimum realized throughput. Our mechanism can enhance the performance of existing and future systems, as those based on the 802.16 ([6], [7]) technology.

The rest of the paper is organized as follows. In Section II-C, we present the proposed OWFQ mechanism and in Section III, we prove analytically that OWFQ guarantees fairness among competing flows. Section IV is devoted to performance comparison between WFQ and OWFQ in a wireless environment impacted by fast fading. We conclude in Section V.

II. Opportunistic Weighted Fair Queueing

Most scheduling algorithms designed for radio access divide the flows in a simplistic and binary way into two categories: good channel-state flows that can be scheduled and bad channel-state flows that cannot be scheduled and consequently relinquish their bandwidth to the former category. The scheduler keeps track of the excess bandwidth obtained by good-channel state flows in order to restore it to bad channel-state flows. Existing algorithms differ mainly by the compensation process they propose. The main limitation of these works is that channel condition is modelled as either "good" or "bad" which is too simple to characterize realistic wireless channels. Contrary to existing schemes, our approach profits from the channel fluctuations by way of opportunistic scheduling, seeking to augment overall Throughput under users' QoS requests. In this section, we present the radio resource model and then introduce our scheduling mechanism.

A. The radio model

We consider a wireless system where the scheduling is performed by a Base Station (BS) serving a multitude of flows in a downlink channel of mean capacity C.

Let x_i be the i.i.d. random variables (of unit mean) representing the effect of fast fading experienced by flow i, then the channel capacity of this flow at time t is:

$$C_i(t) = C \cdot x_i(t) \tag{1}$$

As we consider Rayleigh fading, the random variables x_i are exponentially distributed. However, we choose to bound the minimum value taken by x_i to $x_{min} \neq 0$ to obtain a stable system. Indeed, since $\mathbb{E}[\frac{1}{x_i}] = \infty$, the mean service time is also infinite and the system is not stable. Thus, we replace (1) by the following:

$$C_i(t) = max\big(C \cdot x_i(t), C \cdot x_{min}\big) \tag{2}$$

B. Reminder on Weighted Fair Queuing

WFQ is a packet scheduling technique allowing guaranteed bandwidth services. It is an approximation of the Generalized Processor Sharing (GPS) scheduling which allows different flows to have different service shares. In a link of capacity C, WFQ guarantees to each flow i of weight r_i, a minimum rate $\Re_i$ given by:

$$\Re_i = C \cdot \frac{r_i}{\sum_j r_j}$$

The WFQ scheduler assigns a start tag and a finish tag to each arriving packet and serves packets in the increasing order of their finish tags. We denote by p_i^k the k^{th} packet of flow i, by $A(p_i^k)$ its arrival time and by $S(p_i^k)$ and $F(p_i^k)$, respectively, the start and finish tags assigned to p_i^k. The WFQ behaviour is defined by the following equations:

$$S(p_i^k) = max\big(V(A(p_i^k)), F(p_i^{k-1})\big) \tag{3}$$

$$F(p_i^k) = S(p_i^k) + \frac{L_i^k}{r_i} \tag{4}$$

where L_i^k is the size of packet p_i^k and $V(t)$ is the virtual time at time t defined by:

$$\frac{dV(t)}{dt} = \frac{C}{\sum_{i \in B(t)} r_i} \tag{5}$$

with $B(t)$ being the set of active flows at time t.

C. Opportunistic Weighted Fair Queuing

In order to account for the variability of the channel capacity, we propose a new scheduler, named Opportunistic WFQ (OWFQ), defined as follows (as in WFQ, packets are scheduled in increasing order of their finish tags):

$$S(p_i^k) = max\big(V(A(p_i^k)), S(p_i^{k-1}) + \frac{L_i^{k-1}}{r_i}\big) \tag{6}$$

Whenever the BS finishes serving a packet at time TS, in order to schedule the next packet, it computes the finish tag for HOL (Head Of Line) packets of active flows according to the following equation:

$$F(p_i^k) = S(p_i^k) + \frac{L_i^k}{r_i \cdot x_i(TS)} \tag{7}$$

The rationale behind OWFQ is as follows: the better the channel quality experienced by a flow i, the greater the value taken by x_i and the lower the value taken by its finish tag. As a result, its chance to be scheduled will increase. Thus, potential candidates for accessing the channel may be compelled to pass their turn in favour of flows with better channel state which will obviously increase global Throughput. Besides, flows that "missed" their turn will not wait "too long" even if they experience persisting bad channel state because we do not change the definition of the start tag and therefore the state of the channel only impacts one of the terms defining the finish tag. Indeed, the start tag is still computed as in the original model and reflects how frequently the present flow is served in respect to its weight. For that reason, the penalized flow will eventually have the smallest finish time among contending flows despite its large virtual service time.

As we consider fast fading, the approach reaches its optimal behaviour when the packet duration time $\frac{L_i}{C_i}$ is in the same or lower order of magnitude of fast fading variations. This is the case when large capacities are provided to flows. Otherwise, fragmentation of packets will enhance performances.

III. FAIRNESS GUARANTEE

Different metrics of fairness can be defined. In this section, we use the weighted fairness concept as defined in [4] and prove that, when using OWFQ, $\left|\frac{W_i(t_1,t_2)}{r_i} - \frac{W_j(t_1,t_2)}{r_j}\right|$ is bounded for any interval $[t_1, t_2]$ in which both flows i and j are backlogged, where $W_i(t_1, t_2)$ is the aggregate service (in bits) received by flow i in the interval $[t_1, t_2]$.

The proof is obtained by establishing an upper and lower bound on $W_i(t_1, t_2)$ in Lemmas 1 and 2 respectively (detailed proofs are found in VI).

Lemma 1: In an OWFQ server, if flow i is backlogged in the interval $[t_1, t_2]$, then:

$$W_i(t_1, t_2) \geq r_i \cdot (v_2 - v_1) - L_{max} - \frac{L_{max}}{x_{min}}$$

Where $v_1 = V(t_1)$ and $v_2 = V(t_2)$.

We conclude that OWFQ guarantees a minimum throughput for each flow.

Lemma 2: In an OWFQ server, during any interval $[t_1, t_2]$, we have that:

$$W_i(t_1, t_2) \leq r_i \cdot (v_2 - v_1) + L_{max}$$

Theorem 1: For any interval $[t_1, t_2]$ in which two flows i and j are backlogged during the entire interval, the difference in the service received is bounded by:

$$\begin{aligned}\left|\frac{W_i(t_1,t_2)}{r_i} - \frac{W_j(t_1,t_2)}{r_j}\right| \leq & L_{max}\left(\frac{1}{r_i} + \frac{1}{r_j}\right) + \\ & \frac{L_{max}}{x_{min}} \cdot max\left(\frac{1}{r_i}, \frac{1}{r_j}\right)\end{aligned}$$

The result is easily obtained from Lemmas 1 and 2. We conclude that OWFQ provides fairness guarantees.

IV. NUMERICAL EXPERIMENTS

The values chosen in the following numerical analysis are inspired by the properties of 802.16 systems. We consider a wireless channel of mean capacity equal to $C = 10Mbps$ subject to Rayleigh fading. Packet size follows the Bounded Pareto (BP) distribution which is commonly used in analysis because it can exhibit the high variance property as observed in the internet traffic [8]. We denote the BP distribution by $BP(p, q, \alpha)$ where p and q are respectively the minimum packet size (50 bytes) and maximum packet size (1500 bytes) and α is the exponent of the power law. The probability density function of the BP distribution is:

$$f_{BP}(x) = \frac{\alpha \cdot p^\alpha}{1 - \left(\frac{p}{q}\right)^\alpha} \cdot x^{-\alpha-1}, \quad p \leq x \leq q, 0 \leq \alpha \leq 2$$

We take $\alpha = 1.16$ and therefore the mean packet size equals 155 bytes.

We consider three independent permanent Poisson processes of intensity $\lambda/3$ with the following weights $r_0 = 0.5$, $r_1 = 0.25$ and $r_2 = 0.1$. The global arrival process is therefore a Poisson process of intensity λ. In order to vary the load in the cell, we vary λ. We take $x_{min} = 0.1$ and we normalize the mean capacity to $C = 1.0$. The buffer capacity of each flow is limited to 10^6 packets. New packets generated when the buffer is full are lost.

We consider two independent models, in the first, flows are served according to the WFQ algorithm and in the second, flows are served according to our Opportunistic WFQ. The total load ρ is defined as the ratio of the total arrival rate, λ, over the average capacity C. We first analyze and compare, for both models, the impact of total load on the throughput, the packet loss ratio and the percentage of packets served at the minimum rate for each individual flow. Then we analyse the impact of total load on delay.

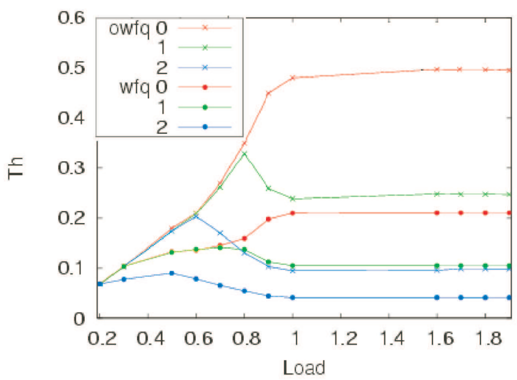

Fig. 1. Mean Throughput per flow as a function of ρ

As we can see in Figure 1, for $\rho \leq 0.2$, the OWFQ does not realize any gain in terms of throughput compared with WFQ which is natural since the probability of having more than one bottleneck flow is negligible and therefore no advantage is taken from taking into account the radio channel state. Any scheduling policy based on opportunism will bring no improvement in comparison with a non-opportunistic scheme if there are not at least two active users present in the system when the scheduling decision is made. For $\rho \leq 0.3$, OWFQ does not realize any gain in terms of throughput for flows 0 and 1 since only flow 2 has bottlenecked packets with non negligible probability; according to simulation results, the mean number of packets enqueued for flows 0 and 1 is strictly inferior to 2 for this range of the total load. Nevertheless, while the number of packets served at the minimum rate in WFQ remains constant when ρ varies in the cited range, (approximately equal to 9.6% of the total number of served packets), the latter decreases in OWFQ as load increases passing from 8% for $\rho = 0.2$ to 6% for $\rho = 0.3$. As for flow 2, the mean number of packets enqueued is around 2 which means that the latter is often active and served simultaneously either with flow 0 or flow 1. As a result, flow 2 realizes a gain in mean throughput of approximately 20%.

For $0.4 \leq \rho \leq 1.0$, the gain obtained from multi-user diversity is tangible: for flow 0, the gain in mean throughput as compared to WFQ varies from 6% for $\rho = 0,4$ to 128% for $\rho = 0.4$. For flow 1, the gain varies from 6% to 127% and for flow 2, the gain varies from 25% to 128%. Moreover, at $\rho = 1.0$, flow 0 and flow 1 lose respectively 30% and 70% of their packets in WFQ due to buffer overflow while no packets losses have been observed during the simulation time in the OWFQ case. Besides, 9.6% of the packets are served at C_{min} in WFQ against respectively 4% and 2% in OWFQ. As for flow 2, the number of lost packets, at $\rho = 1.0$, is 85% in WFQ against 55% in OWFQ while the number of packets served at C_{min} is limited to 0.4% in OWFQ (their number remains the same in WFQ for all flows and equals 9.6% whatever the load is). Besides, the differentiation in the service received by flows 0 and 1 appears in OWFQ at $\rho > 0.8$ while it appears earlier in WFQ at $\rho = 0.7$ which highlights the efficient allocation of resources in OWFQ.

For $1.0 < \rho \leq 1.5$, the gain in mean throughput for flow 0 is around 137%, while at $\rho = 1.5$ half of packets are lost in WFQ and only 10% in OWFQ. The number of packets served at C_{min} for flow 0 is around 5% in OWFQ. For flow 1, the gain is around 136% and while the number of lost packets is only lowered from 90% to 80% at $\rho = 1.5$, the number of packets served at C_{min} is around 1% in OWFQ. As for flow 2, the gain in throughput is around 129% and although the blocking probabilities are alike in both scenarios, the number of packets served at C_{min} is only 0.1%, at $\rho = 1.5$, in OWFQ. We conclude that, in addition to notably increasing performances in terms of realized throughput, our approach protects flows at much higher rates than in plain WFQ.

The last remark allows predicting that Opportunistic WFQ will lead to a reduction in average sojourn times. We analyse now the impact of the total load on the relative deviation dT of the Mean Sojourn Time in the WFQ case, termed T_{WFQ}, from the Mean Sojourn Time in the OWFQ case, termed T_{OWFQ}, which we define as $dT = |T_{WFQ} - T_{OWFQ}| / T_{WFQ} \times 100$. We can see from Figure 2 that the Mean Sojourn Time in OWFQ is dramatically reduced as compared to WFQ. Therefore, the gap in the realized mean throughput between OWFQ and WFQ will widen even more in a realistic model based on the TCP protocol where the delay experienced by packets has severe negative repercussions on the overall performances.

We analyse now the Total Throughputs. We denote by C_{WFQ} and C_{OWFQ} the Total throughput for the WFQ and the OWFQ cases respectively. From Figure 3, we can see

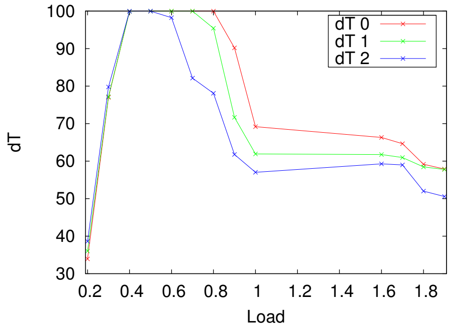

Fig. 2. dT as a function of ρ

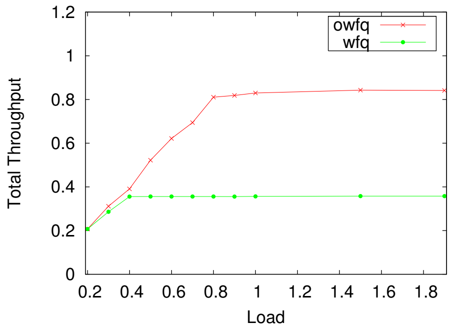

Fig. 3. Total Throughput as a function of ρ

the remarkable gain realized in terms of Total throughput in OWFQ in comparison with WFQ. We notice also that C_{WFQ} converges to 0.36 which is very inferior to the mean capacity, given by the following:

$$\mathbb{E}[C(t)] = C \cdot \int_0^\infty max(x, 0.1) \cdot e^{-x} dx \approx C$$

which in our experiments is approximately equal to 1.0 (we chose $C = 1.0$). To interpret this result, we observe that our model is equivalent to a model of constant capacity C fed by the same arrival process as in the original model but with packets of length $\frac{L_i}{x_i}$. We thus have the following stability condition:

$$\lambda \cdot \mathbb{E}[\frac{L_i}{x_i}] \leq C \tag{8}$$

From (8), we have that:

$$\lambda \cdot \mathbb{E}[L_i] \cdot \mathbb{E}[\frac{1}{x_i}] \leq C \Rightarrow \lambda \cdot \mathbb{E}[L_i] \leq \frac{C}{\mathbb{E}[\frac{1}{x_i}]}$$

Hence, the actual mean capacity of the system is:

$$\frac{C}{\mathbb{E}[\frac{1}{x_i}]} = \frac{C}{\int_0^\infty \frac{1}{max(x,0.1)} \cdot e^{-x} dx} \approx 0.36 \cdot C \tag{9}$$

which explains the result obtained in WFQ.

In OWFQ, the behaviour of the system is much more complex so we cannot provide a similar analytical evaluation of the capacity. The gain shown in Figure 3 is of course due to the usage of an opportunistic approach.

OWFQ	Th_i	Th_j	C_{OWFQ}
6	0.3088	0.1544	1.39
10	0.2367	0.1184	1.78
14	0.1798	0.0896	1.89

TABLE I

WFQ	Th_i	Th_j	C_{WFQ}
6	0.0798	0.0398	0.359
10	0.0475	0.0238	0.357
14	0.0342	0.0171	0.359

TABLE II

To show that the gain obtained increases with the number of users (multi-user diversity), we run three sets of simulations with respectively $n = 6$, 10 and 14 flows at overload. For every value of n, flows are divided into two categories with equal number of flows ($\frac{n}{2}$). All flows of a given category have the same weight and the weight of flows of the first category is twice the weight of flows of the second category. We compute the Total Throughput C_{WFQ} and C_{OWFQ} for both schedulers, given by $(Th_1 + Th_2) \cdot \frac{n}{2}$, where Th_1 and Th_2 denote, respectively, the average rate of flows of categories 1 and 2.

First, we notice that C_{OWFQ} increases significantly with the number of flows. We also observe that C_{WFQ} is approximately equal to 0.36 for all values of n, which was expected from formula (9). To estimate the gain obtained in terms of realized Total Throughput, we compute the relative deviation of C_{OWFQ} from C_{WFQ} by applying the following formula $G_n = |C_{WFQ} - C_{OWFQ}| / C_{WFQ}$. We get the subsequent results $G_6 \approx 2.87$, $G_{10} \approx 3.99$ and $G_{14} \approx 4.26$. Hence, we can see that the achieved gain is significant and that it increases with the number of served flows, which means that our scheduler takes advantage of flows diversity. Moreover, we see that for both schedulers, we have $Th_i \approx 2 \cdot Th_j$ and thus, the differentiation in the realized throughputs is achieved.

V. Conclusion

Fair queueing has long been a popular paradigm for guaranteeing minimum throughput for users or flows sharing a wireline link. In this paper, we propose a new scheduler that couples opportunistic scheduling with weighted fair queueing to enhance overall performances in the case of wireless links by taking advantage from the radio channel variations. We proved analytically that the Opportunistic WFQ guarantees fairness among users and we showed through simulations that our solution provides significantly better performances than the WFQ approach. In future work, we will evaluate the interactions of the proposed scheduler with the dynamics of TCP and we will integrate to its mechanism previous proposed ideas allowing to deal with the case where certain flows are blocked due to very bad channel quality.

VI. Appendix

A. Proof of Lemma 1:

If $r_i \cdot (v_2 - v_1) - L_{max} - \frac{L_{max}}{x_{min}} \leq 0$, Lemma 1 holds trivially since $W_i(t_1, t_2) \geq 0$. Hence, we consider the case where:

$$v_2 > v_1 + \frac{L_{max}}{r_i} + \frac{L_{max}}{r_i \cdot x_{min}} \quad (10)$$

Let packet p_i^k be the first packet of flow i to receive service in (v_1, v_2). To observe that such a packet exists, we consider the following two cases:

- Packet p_i^n such that $S(p_i^n) < v_1$ and $S(p_i^n) + \frac{L_i^n}{r_i} > v_1$ exists: since flow i is backlogged in $[t_1, t_2]$, we conclude that $V(A(p_i^{n+1})) \leq v_1$. From (6), we get $S(p_i^{n+1}) = S(p_i^n) + \frac{L_i^n}{r_i}$. Using the fact that $S(p_i^n) < v_1$, we get that $S(p_i^{n+1}) < v_1 + \frac{L_{max}}{r_i}$. Also, using (10), we deduce:
$$S(p_i^{n+1}) < v_2 \quad (11)$$
Since $S(p_i^{n+1}) = S(p_i^n) + \frac{L_i^n}{r_i} > v_1$, using (11), we conclude that $S(p_i^{n+1}) \in (v_1, v_2)$.
- Packet p_i^n such that $S(p_i^n) = v_1$ exists: p_i^n may finish service at time $t < t_1$ or $t \geq t_1$. In either case, since flow i is backlogged in $[t_1, t_2]$, we conclude that $V(A(p_i^n)) \leq v_1$. Hence $S(p_i^{n+1}) = S(p_i^n) + \frac{L_i^n}{r_i}$. Using the fact that $S(p_i^{n+1}) < S(p_i^n) + \frac{L_{max}}{r_i}$ and $S(p_i^n) = v_1$, we get from (10) that $S(p_i^{n+1}) < v_1 + \frac{L_{max}}{r_i} < v_2$. Since $S(p_i^{n+1}) = v_1 + \frac{L_i^n}{r_i} > v_1$, we conclude that $S(p_i^{n+1}) \in (v_1, v_2)$.

Since either of the two cases always holds, we conclude that packet p_i^k such that $S(p_i^k) \in (v_1, v_2)$ exists. Furthermore, we have the additional following result:

$$S(p_i^k) < v_1 + \frac{L_{max}}{r_i} \quad (12)$$

Let p_i^{k+m} be the last packet to receive service in the virtual time interval (v_1, v_2). Thus, $F(p_i^{k+m+1}) \geq v_2$. From (6) and (7), we know that at time TS:

$$F(p_i^{k+m+1}) = S(p_i^{k+m}) + \frac{L_i^{k+m}}{r_i} + \frac{L_i^{k+m+1}}{r_i \cdot x_i(TS)} \quad (13)$$

We deduce the following result:

$$S(p_i^{k+m}) \geq v_2 - \frac{L_i^{k+m}}{r_i} - \frac{L_{max}}{r_i \cdot x_{min}} \quad (14)$$

Using the tagging scheme in Section II-B, we can derive the following:

$$S(p_i^{k+m}) = S(p_i^k) + \sum_{j=0}^{m-1} \frac{L_i^{k+j}}{r_i} \quad (15)$$

Thus, from (14) and (15), we get that:

$$S(p_i^k) + \sum_{j=0}^{m} \frac{L_i^{k+j}}{r_i} \geq v_2 - \frac{L_{max}}{r_i \cdot x_{min}} \quad (16)$$

From (12) and (16), we get the following result:

$$\sum_{j=0}^{m} \frac{L_i^{k+j}}{r_i} \geq (v_2 - v_1) - \frac{L_{max}}{r_i} - \frac{L_{max}}{r_i \cdot x_{min}}$$

Since $W_i(t_1, t_2) \geq \sum_{j=0}^{m} L_i^{k+j}$, Lemma 1 follows.

B. Proof of Lemma 2:

The set of flow i packets during time interval $[t_1, t_2]$ have start tags at least v_1 and at most v_2. This set can be partitioned in two subsets:

- Set D consisting of packets that have start tags at least v_1 and strictly inferior to v_2. Formally, $D = \{k | v_1 \leq S(p_i^k) < v_2 \wedge F(p_i^k) \leq v_2\}$. For packets in D, using (6) and (7), we get $\sum_{k \in D} l_i^k \leq r_i \cdot (v_2 - v_1)$.
- Set E consisting of packets that have start tags equal to v_2 and finish tags strictly greater than v_2. It is obvious that at most one packet belongs to this set, because
$$S(p_i^k) = v_2 \Rightarrow S(p_i^{k+1}) = S(p_i^k) + \frac{L_i^{k+1}}{r_i} > v_2$$
Hence $\sum_{k \in E} l_i^k \leq l_{max}$.

We conclude that Lemma 2 holds.

References

[1] A. Parekh. and G. Gallager, A generalized processor sharing approach to flow control in integrated services networks: the multiple node case, IEEE/ACM Trans. on Networking, vol. 1-3, pp. 344-357, June 1993.
[2] P. Goyal, H. Vim, and H. Chen, Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks, IEEE/ACM Trans. on Networking, vol. 5, pp. 690-704,Oct 1997.
[3] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, Proceedings of IEEE, October 1995.
[4] S. Golestani, A self-clocked fair queueing scheme for broadband applications, Proceedings of the IEEE INFOCOM '94, pp. 636-646, 1994.
[5] V. Bharghavan, S. Lu, and T. Nandagopal. Fair queueing in wireless networks: Issues and approaches. IEEE Personal Communications Magazine, Feb 1999.
[6] A. Gosh, Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential, IEEE Communications Magazine.
[7] http://www.wimaxforum.org/
[8] Crovella M., Bestavros A., Self-Similarity in World Wide Web traffic: Evidence and possible causes, IEEE/ACM Transactions on Networking, pp 835-846, December 1997.