A Scheduling Scheme for QoS Supports in the Heterogeneous Wireless Network Environment

D-H. Hoang*, D. Reschke

TU Ilmenau, Germany

Abstract. Providing Quality of Service (QoS) is a challenging issue especially in the heterogeneous wireless network environment. Traffic scheduling is considered as the key mechanism to support QoS with respect to protection of flows and fair allocation of the limited link bandwidth. However, scheduling schemes developed for wired networks can not be directly applied to wireless networks due to bursty and location-dependent channel errors. In this paper, we propose a scheduling scheme called Packetized Wireless Scheduler (PWS) for providing both fairness and protection. The key idea of PWS is to keep track of the channel state and to compensate the flows which have received less service due to poor quality of their wireless channel. Since our scheme uses a portion of the link bandwidth for giving an extra compensation service, our scheme can achieve fairness guarantees and protection to individual flows. Furthermore, this bandwidth portion facilities the handoff while making efficient use of the link bandwidth.

1 Introduction

Next generation wireless networks will have to support a wide range of applications requiring different Quality of Service (QoS) requirements in term of delay, jitter and loss rate. Typically, the expected traffic will be a mix of real-time sources such as multimedia applications and data transfers. Recently, there has been widespread interest in the delivery of multimedia services over the heterogeneous environment of wired and wireless networks.

The problem of providing QoS to contending applications involves suitably allocating networks resources such as buffer and link bandwidth depending on each application. To solve this problem, a suitable scheduling scheme at the network node is necessary. Scheduling is seen as the key mechanism for QoS support at the packet level. This mechanism is responsible for the transmission order of packets from different competing flows on a output link of a network node.

Several scheduling schemes have been studied in the past in the context of wired networks (see [1] for references). Fair scheduling schemes have been argued to be a popular paradigm for achieving bounded delay, throughput and fairness guarantees. However, these schemes assume error-free conditions. They do not address the cause of performance impairments due to the link errors. That is why they can not be applied for the heterogeneous network environment. The transmission over wireless

* With a Fellowship of the Alexander-von-Humboldt Foundation

links may be adversely affected by location-dependent and bursty channel errors. In the presence of channel errors it is not possible to reach the mobile host temporally as well as some mobile hosts may not be able to transmit data to the base station. In this situations, fair scheduling is neither fair nor able to provide service guarantees.

Recently, many efforts have been made in order to adapt fair scheduling schemes to the wireless network environment. Several new schemes have been proposed such as Wireless Fair Service (WFS)[2], Channel-Independent Fair Queuing (CIF-Q)[3], Server-Based Fairness Approach (SBFA)[4], Channel State Dependent Packet Scheduling (CSDPS)[5]. The common strategy of these schemes is to use compensation to remedy the fairness guarantees. Various degrees of service guarantees have been shown in these works. However, most of these schemes assume a prior knowledge of the channel condition according to a bad or good state. Moreover, there is remaining a trade-off between efficient use of the link bandwidth and service guarantees. On the other hand, the compensation of some flows may disturb the service guarantees of other flows. In case of bursty channel errors, existing schemes provide neither protection nor bounded delay guarantees to competing flows.

In this paper, we propose a scheduling scheme called Packetized Wireless Scheduler (PWS) for providing both fairness and protection in the heterogeneous network environment. The key idea of PWS is to keep track of the channel state and to compensate the flows which have received less service due to poor quality of their wireless channel. When a flow experiences channel errors, its lost service will be credited. This flow will be given an extra compensative service after its channel has been recovered from the error state. A portion of the link bandwidth is allocated for this compensation purpose. This portion is not waste since it will be fairly distributed to all existing flows in case of error-free channel conditions. Furthermore, this portion will be used for handoff purpose.

The rest of this paper is organized as follows. In section 2, we describe the model of fair scheduling and its issues. In section 3, we introduce the fluid model that is used as a formal model for PWS. Section 4 presents the packetized model for PWS and some analytical results. Simulation results for PWS are shown in section 5. Section 6 concludes the paper.

2 Fair Scheduling and Issues

As the background, we now describe shortly the model of fair scheduling in wired networks.

Consider an output link that is being shared by N flows. Let R_i denote the *reserved rate* (the weight) of a flow *i*. At each time instant *t*, a backlogged flow *i* will be allocated a fair shared rate of $R_i * C / \sum_{j \in S_B(t)} R_j$, where *C* is the link bandwidth and $S_B(t)$ is the set of non-empty (*backlogged*) queues at time *t*. That is, the fluid queuing scheme serves backlogged flows in the proportion of their reserved rates. The fluid model guarantees that for any time interval $(t_1,t_2]$, during which two flows *i* and *j* are continuously backlogged, the following property is satisfied:

$$\left| W_i(t_1,t_2)/R_i - W_j(t_1,t_2)/R_j \right| = 0 \text{ , } \forall i, j \in S_B(t), t \in (t_1,t_2]$$

where $W_i(t_1,t_2)$ is the service (bits) received by the flow i in the time interval $(t_1,t_2]$. That is, each backlogged flow will have the same quotient of its received service (bits) and its weight regardless of the load of other flows. This quotient is called *normalized service* offered by the scheduler to *flow i*.

The assumption of the fluid flow queuing is that the packets are infinitely divisible and all flows can be served simultaneously. Since this assumption is not realistic in packet-switching networks, packetized fair queuing schemes must approximate the fluid flow model. The goal of a packetized scheme such as Weighted Fair Queuing (WFQ) [6,7] is to keep the normalized service of all backlogged flows equal to each other, i.e. to keep the difference $\left|W_i(t_1,t_2)/R_i - W_j(t_1,t_2)/R_j\right|$ of any two backlogged flows i and j during their backlogged interval (t_1,t_2) as small as possible. In order to achieve this goal, packetized schemes such as WFQ maintain a start tag and a finish tag for each packet, and a *virtual time function V(t)*. The start tag T_{Si}^{k} and finish tag T_{Fi}^{k} are assigned to an arriving packet k of flow i as follows:

$$T_{S_i}^{k} = \mathbf{max}\left(V(ta_i^k), T_{Fi}^{k-1}\right)$$

$$T_{Fi}^{k} = T_{Si}^{k} + \frac{L_i^k}{R_i}$$

where L_i^k is the length of the packet k of flow i. The virtual time function is given by:

$$\frac{dV(t)}{dt} = \frac{C}{\sum_{j \in S_B(t)} R_j}$$

The scheduler then serves the packet with the minimum finish tag in the system.

It has been proved in [7,1] that fair scheduling schemes in wired networks provide: (1) fairness to all flows sharing a link; (2) protection and service guarantees to each flow. However, they assume the error-free condition of flows, i.e. all transmitted packets are received correctly, which is not true in a wireless environment. A straight-forward application of these schemes to wireless networks can not provide the same notation of performance guarantees. We illustrate this issue by the following example.

Example: Suppose that there are two backlogged flows f_1 and f_2. Flow f_1 has an error-free channel while flow f_2 has bad channel condition during a time interval $[t_1,t_2]$. Due to the channel errors flow f_2 cannot be served in this time interval, but its timestamp is continuously maintained. When the flow f_2 exists from the error upon t_2, its timestamp will be the smallest. The scheduler will serve flow f_2 exclusively until its timestamp catches up with that of flow f_1. In the meantime, flow f_1 will not be served. Thus, there is unfairness between the flows and a fair scheduling scheme fails to provide fairness guarantee. Moreover, the service guarantees of flow f_1 are affected. On the other hand, if flow f_2 exists from the error and its timestamp is updated immediately to the virtual time function $V(t)$, this flow will never re-gain the service it has lost in the past due to channel errors. The result is again unfairness between flows. The example shows that a suitable scheduling mechanism is needed for QoS control in the wireless environment.

As the result, a wireless scheduling scheme needs to deal with the following issues:

- The failure of traditional schemes in the presence of channel errors.
- The issue of detecting the channel state.

- The compensation for flows that perceive channel errors.
- The trade-off between service guarantees and efficient use of the limited wireless bandwidth.

In the following sections, we propose a simple approach called Packetized Wireless Scheduler (PWS) which addresses the issues presented above. The development of PWS consists of two steps: definition of a wireless scheduler based on the ideal fluid model, emulation of the fluid model to achieve the packetized scheme.

3 The Fluid Wireless Scheduler

We consider a wireless scheduler, which resides at the base station and is responsible for either uplink or downlink. In a packet cellular network, a base station can serve one or more cells. Thus, the scheduler at the base station can serve flows from different mobile hosts. We introduce the following definitions for the description of the wireless scheduler.

Definition 1: **Backlogged flow**. *A flow is said to be backlogged if there are packets of this flow in queue waiting for transmission.*

Definition 2: **Feasible flow**. *A flow is feasible if it is backlogged and its channel condition is error-free.*

Definition 3: **Lagging flow**. *A flow is said to be lagging if it has received less service than its error-free service due to the lost service in the time of channel errors. This flow needs to be compensated.*

The fluid model is a work-conserving scheduler that is never idle when there are packets in queue waiting for transmission. Thus, during a period with channel errors, error-free flows will receive more service than their fair share, while flows with channel errors will receive no service. To address the issue of service guarantees, we need to detect the channel errors, credit the lost service, and to give a compensative service for flows with channel errors after the flow's channel has been recovered from the error state. The scheduling scheme should be designed so that the flow with channel errors will catch up with other error-free flows, while a fairness between flows is obtained and service of all flows will be guaranteed. Furthermore, the issues of fairness, delay guarantees and graceful degradation of service should be re-defined for the wireless environments.

Most of the existing works such as [2,3,4,5] suggested to compensate lagging flows by forcing the other flows to give up some bandwidth. However, if the scheduler offers an extra compensation service to lagging flows, the service guarantees of other flows may be worse. Furthermore, a static reservation of a bandwidth fraction for compensation purpose is not efficient, since the unused link bandwidth is waste.

We propose therefore to allocate a portion of the total bandwidth αC for the compensation purpose, where α is an administrative parameter and $0 \le \alpha < 1$ (see Figure 1). Let S_L ($S_L \subset S_B$) the set of lagging flows that need to be compensated, S_F the set of feasible flows and S_C the set of flows being compensated. Then the bandwidth portion αC is allocated to the lagging flows, while the rest of bandwidth, i.e. $(1-\alpha)C$ is allocated to all backlogged flows. If $S_L=\varnothing$, i.e. no lagging flow, all backlogged flows will share the total bandwidth C (that is, $\alpha=0$). In this manner, the service

guarantees of all feasible flows are maintained, even when the whole portion αC is being used for compensation. This bandwidth portion is not waste since it will be fairly distributed to all backlogged flows in case of error-free conditions. Furthermore, this bandwidth portion will be useful for handoff purpose. When a mobile host moves to a cell belonging to a foreign base station, its flow will be assigned immediately a fraction of this bandwidth portion. The choice of α depends on the service requirements of flows sharing the wireless link, the transmission characteristics of the actual wireless environment and the frequency of the handoffs. There is a trade-off between compensative capability and the number of flows sharing the wireless link, since large α means that fewer flows can be concurrently served on the wireless link.

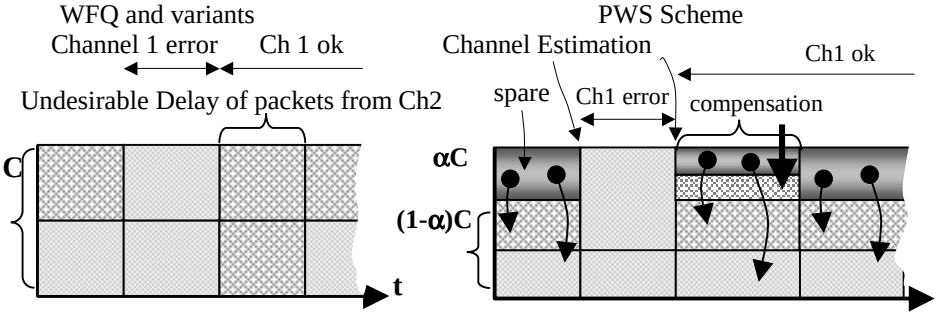


Figure 1 Comparison of WFQ and PWS by channel error.

In order to maintain the fairness between all backlogged flows, a virtual time function is maintained as described above. However, an extra service is added for the lagging flow so that this flow will get the service it has lost in the past due to channel errors. The wireless scheduler provides this extra service by giving a weight increase r_i as follows:

$$R_i(t) = \begin{cases} R_i + r_i & \textit{during the compensation} \\ R_i & \textit{otherwise} \end{cases}$$

The compensation weight increase r_i can be chosen according to the need of flow being compensated. The compensation is faster by large r_i, and slower by small r_i. Since the wireless scheduler only uses the bandwidth portion αC for compensation, the service of other error-free flows will not be influenced. It is obviously that a short-term unfairness exists when a flow is being compensated. In order to bound this short-term unfairness, we choose r_i according to:

$$r_i = \frac{\alpha C}{\sum_{j \in N} R_j} * R_i \quad , \text{ and } \quad \alpha C \le \sum_{j \in N} R_j$$

The principle of compensation is further illustrated with respect to the wireless scheduler in Figure 2. Suppose that there are two flows with weights $R_1=R_2$ that are continuously served in a system busy period. In the time interval (t_1,t_2), two flows are served proportional to their weights according to the virtual time function $V(t)$. During the interval $[t_2,t_3)$, flow *2* experiences a channel error, thus it receives no service. Flow *1* will be exclusively served during this interval. A credit counter $CC_2(t)$ is

activated for flow *2* to count the service it has lost during this interval. The evolution of the credit counter of flow *i* is given by:

$$\frac{\partial CC_i(t)}{\partial t} = R_i * \frac{\partial V(t)}{\partial t}, \quad i \in S_B(t) \text{ and } i \notin S_F(t)$$

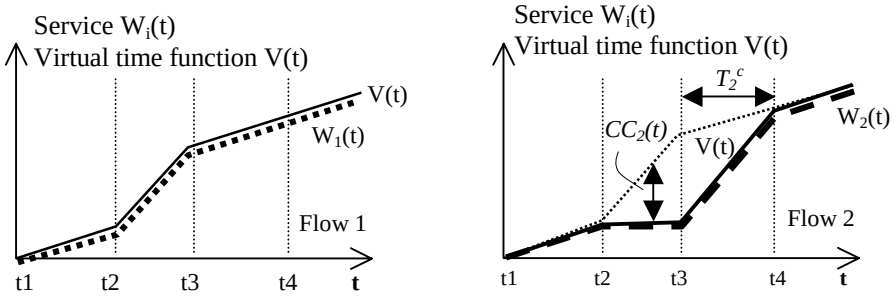


Figure 2 Comparison of WFQ and PWS by channel error.

At time t_3, the channel condition of flow *2* has been recovered, and two flows are then served. Since flow *2* has lost some service in the interval (t_2,t_3) due to channel error, it needs to be compensated. The service of flow *2* should increase with a rate that is proportional to the additional weight R_2+r_2 during $[t_3,t_4)$. In this interval $[t_3,t_4)$, the credit counter of the corresponding flow *i* decreases with a rate as follows:

$$\frac{\partial CC_i(t)}{\partial t} = -r_i * \frac{\partial V(t)}{\partial t}, \quad i \in S_C(t) \text{ and } i \in S_F(t)$$

At time t_4, the credit counter reaches zero. The flow *2* has been fully compensated. Upon t_4, the weight of flow *2* returns to the normal value R_2, and both flows will be served proportional to their normal weights R_1 and R_2 respectively. The time interval $T_2^c = t_4 - t_3$ is called the compensation period for flow *2*.

Any flow *i* that is feasible in the interval $(t_1,t_2]$ will be served proportional to the virtual time function *V(t)*. Its amount of service $W_i(t_1,t_2)$ received in this interval satisfies:

$$W_i(t_1,t_2) = \begin{cases} (R_i + r_i) * (V(t_2) - V(t_1)) & \text{if } CC_i(t) > 0 \\ R_i * (V(t_2) - V(t_1)) & \text{otherwise} \end{cases}$$

The fluid scheduler has following properties.

Property 1: *The total amount of weight increase given to all compensated flows satisfies:* $\sum_{i \in S_C(t)} r_i \leq \alpha C$

Property 2: *Suppose that flow i has channel error in the time interval (t_b,t_r), and its channel is recovered at time t_r. Let us denote with t_c the time at which this flow has been completely compensated, then t_c satisfies:*

$$V(t_c) = V(t_r) + \frac{R_i}{r_i}(V(t_r) - V(t_b))$$

moreover, $t_c \leq tr + \frac{R_i}{r_i}(V(t_r) - V(t_b))$

if flow i is continuously feasible in the time interval $[t_r,t_c]$.

Property 3: *The fairness of the wireless scheduler is defined as (Ω is also called the fairness index):* $\Omega = \left| W_i(t_1,t_2)/R_i - W_j(t_1,t_2)/R_j \right|$ *for any two flows in the time interval* $(t_1,t_2]$. *The scheduler is ideal fair if Ω is zero, i.e. the difference in normalized service of any two backlogged flows in the interval* $(t_1,t_2]$ *is zero. The scheduler is considered close to fair if Ω is close to zero. If flow i and flow j are continuously feasible in the time interval* $(t_1,t_2]$, *then:*

$$\Omega = 0 \qquad \text{if } i, j \in S_C(t) \text{ or } i, j \notin S_C(t)$$

$$\Omega = \frac{\alpha C}{\sum_{k \in N} R_k} * \left(V(t_2) - V(t_1) \right) \qquad \text{if } i \in S_C(t), j \notin S_C(t) \text{ or } i \notin S_C(t), j \in S_C(t)$$

where $S_C(t)$ *is the set of flows being compensated, and* $t \in (t_1,t_2]$.

4 The Packetized Wireless Scheduler

The packetized model approximates the fluid model by using the virtual time functions to define the sending order of packets. The natural way to do this is to timestamp packets by the virtual finishing time based on the virtual time function *V(t)* and the packetized scheduler transmits packets according to the increasing order of their time-stamps. Intuitively, the scheduler needs only to calculate the timestamp each time when a packet of this flow becomes the head-of-line (HOL) packet of the flow's queue. Let T_{Si}, T_{Fi} be the Start Tag and the Finish Tag of flow *i*, respectively. We can have:

$$T_{Si} = \max \left(V(ta_i^k), T_{Fi} \right)$$

$$T_{Fi} = T_{Si} + L_i^k / R_i$$

where $V(ta_i^k)$ is the virtual time function at the time, when packet *k* of flow *i* becomes the HOL-packet.

In order to credit the lost service, PWS should keep track of the channel state. We use our method developed in [8] to monitor and predict the channel state for PWS.

The packetized wireless scheduler has following properties.

Property 4: *Let* T_{AFi}^k *be the actual service finish time of packet k of flow i in the packetized system,* td_i^k *be the virtual finish time of that packet in the fluid system,* $L_{i,max}$ *be the maximum packet size of flow i. If flow i is feasible, the following unequation holds:*

$$T_{AF}{}_i^k \le td_i^k + \frac{L_{i,\max}}{C} \qquad \forall i,k \text{ if connection } i \text{ feasible}$$

Property 5: *When flow i is feasible, the extra compensation service given to packet k of this flow is given by:*

$$Wa_i^k = \begin{cases} \dfrac{L_i^k * r_i}{R_i + r_i} & \text{if } CC_i(ts_i^k) \ge \dfrac{L_i^k * r_i}{R_i + r_i} \\ CC_i(ts_i^k) & \text{otherwise} \end{cases}$$

where Wa_i^k is the extra compensation service given to packet *k* of flow *i*, $CC_i(ts_i^k)$ is the credit counter at the virtual start time ts_i^k of packet *k* (see Figure 3).

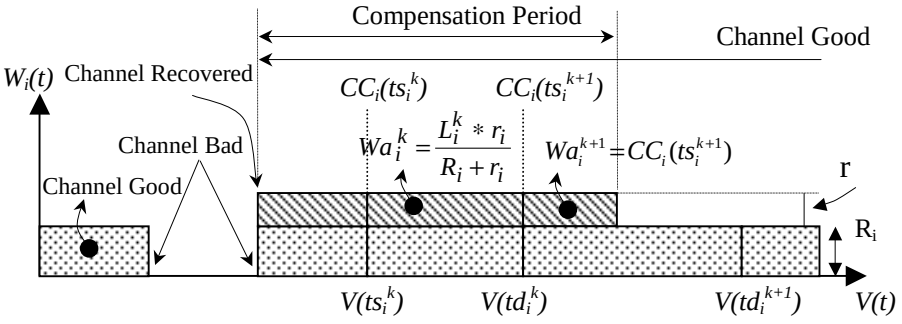


Figure 3 Compensation service.

Property 6: *Let ta_i^k, ts_i^k be the arrival time and the service start time of packet k of flow i in the fluid system, respectively. When flow i is feasible, the start tag T_{Si}^k and finish tag T_{Fi}^k of packet k in the packetized system are given by:*

$$T_{Si}^k = \max\left(V(ta_i^k),\ T_{Fi}^{k-1}\right)$$

$$T_{Fi}^k = \begin{cases} T_{Si}^k + L_i^k/(R_i + r_i) & \text{if } CC_i(ts_i^k) \geq L_i^k * r_i/(R_i + r_i) \\ T_{Si}^k + L_i^k - CC_i(ts_i^k)/R_i & \text{otherwise} \end{cases}$$

The property 6 shows that, if no compensation is required for flow *i*, i.e. $CC_i(ts_i^k) = 0$, the service tags can be calculated just as in the case of error-free service. During compensation period, packet *k* of flow *i* will have precedence by scheduling, since its service tags are smaller in comparison to those in error-free service.

Property 7: *Suppose that flows i and j are continuously feasible in the time interval $(t_1,t_2]$. If $i,j \in S_C(t)$ or $i,j \notin S_C(t)$ then:*

$$\hat{\Omega} \leq L_{\max} * \left(1/R_i + 1/R_j\right)$$

If $i \in S_C(t)$, $j \notin S_C(t)$ or $i \notin S_C(t)$, $j \in S_C(t)$ then:

$$\hat{\Omega} \leq L_{\max}\left(1/R_i + 1/R_j\right) + \alpha C * \left(V(t_2) - V(t_1)\right)/\sum_{k \in N} R_k$$

where $\hat{\Omega} = \left| W_i(t_1,t_2)/R_i - W_j(t_1,t_2)/R_j \right|$ is the fairness index of the packetized scheduler, $S_C(t)$ is the set of flows being compensated, L_{max} is the maximum packet size of flows i and j.

5 Simulation Results

We use the network configuration as showed in the Figure 4 for the experiments to show that the PWS scheme is feasible. In this configuration, the wireless link capacity is 1.0 Mbps that is shared by three UDP sources. The normalized capacity is 1, and the weights of sources are set to $R_1 = R_2 = R_3 = 0.2$. All flows start to send at t=0s. Flow 1 will experience channel error in the time interval (0.1s, 0.6s). We choose the weight increase of flow 1 equal to 0.04 for compensation. Three UDP sources send packets of length 1000 Bytes at the same rate of 400 Kbps. Under this condition, the scheduler

is continuing backlogged.

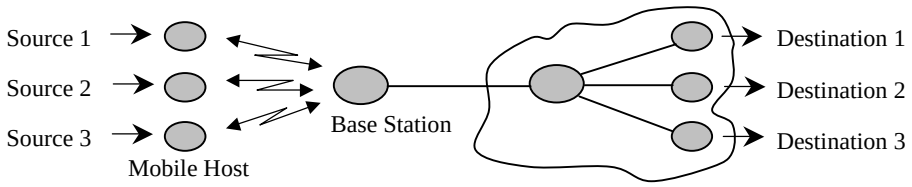


Figure 4 Model for simulation of PWS.

If Weighted Fair Queuing (WFQ) is used, flow 1 will loss its service in the time interval (0.1s,0.6s) due to its channel error and never receives this service again after its channel state has been recovered at time t=0.6s (Figure 5). In the PWS scheme, flow 1 will be compensated after its channel has been recovered. The compensation period takes as long as its credit counter is positive until t=2.6s (Figure 6). The length of the compensation period depends on the lost service and the weight increase.

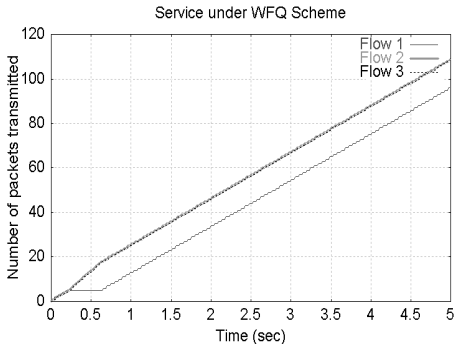


Figure 5 Service of Weighted Fair Queuing (WFQ)

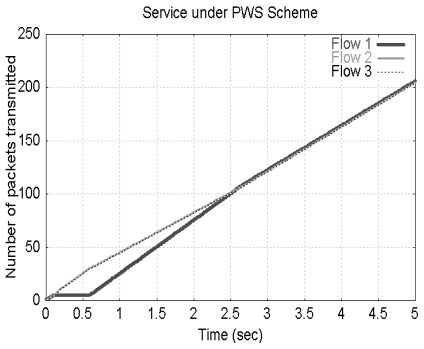


Figure 6 Service of PWS with error in (0.1,0.6), weight increase 0.04

As showed in the Figure 6, two flow 2 and 3 receive the same service according to their weights. From t=2.6s, three flows receive exactly the same service, since they have the same weights.

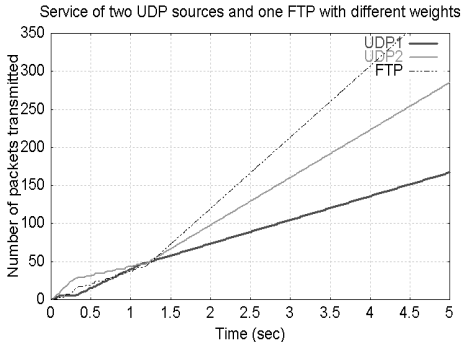


Figure 7 Service of PWS with two UDPs and one FTP. Flow 1 has error in (0.1,0.3).

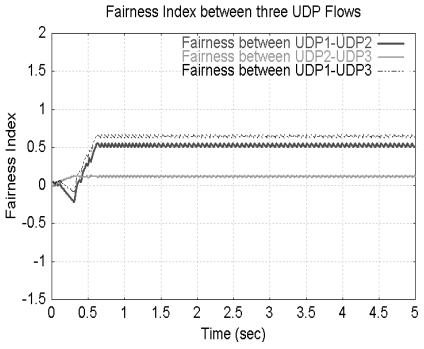


Figure 8 Fairness index of three UDPs with different weights.

Figure 7 shows the result of an experiment with two UDP sources and one FTP source. The weights of these sources are: 0.1 for flow 1, 0.2 for flow 2 and 0.3 for flow 3. Since we use the normalized link bandwidth of 1, the sum of weights should be less or equal to 1. The weight increase of flow 1 is 0.02. The wireless link has a capacity of 1.5 Mbps. Flow 1 has a rate of 400 Kbps, flow 2 has a rate of 800 Kbps. As shown in the figure, flow 1 has lost its service in the time interval (0.1s,0.3s) due to channel error. This service will be distributed to flows 2 and 3. At time t=0.3s, channel of flow 1 has been recovered and flow 1 will be compensated until t=1.15s. After that all flows will be served according to their weights. Figure 8 shows the fairness index of three UDP sources under PWS scheme with different weights (0.1, 0.2 and 0.3) sharing the wireless link of 1.0 Mbps. The rates of flows are 200 Kbps, 400 Kbps and 600 Kbps respectively. The channel of flow 1 has error in the time interval (0.1s, 0.3s). As shown in the figure, the fairness depends on the weight ratio of the flows. However, the fairness index stays within the allowed bound (i.e. between –1 and 1) as shown in Figure 8.

6 Conclusion

In this paper, we have identified the issues of fair scheduling in the heterogeneous environment of wired and wireless networks. We propose a simple approach which is based on the prediction of the channel state and a compensation model to remedy the performance impairments due to channel errors. Since our scheme uses a portion of the link bandwidth for compensation purpose and since the compensation is performed by giving an extra compensation service, our scheme can achieve fairness guarantees and protection to individual flows. Furthermore, the allocated portion of bandwidth facilities the handoff while making efficient use of the link bandwidth.

References

[1] D-H. Hoang, "*Adaptive Bandwidth Management for ATM-based Broadband Networks*", ISBN 3-631-35656-0, PETER LANG GmbH, Europäischer Verlag der Wissenschaften, Frankfurt am Main, 1999.

[2] S.Lu, T.Nandagopal, V.Bharghavan, *"A Wireless Fair Service Algorithm for Packet Cellular Networks"*, in 4th Annual ACM MOBICOM 1998, p.10-20, New York, NY, Oct.1998.

[3] T.S.Eugene Ng, I.Stoica, H.Zhang, *"Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors"*, in IEEE INFOCOM'98, vol.3, p.1103-1111, New York, NY., 1998.

[4] P.Ramanathan, P.Agrawal, "*Adapting Packet Fair Queueing Algorithms to Wireless Networks*", in 4th Annual ACM MOBICOM'98, p.1-9, New York, NY, Oct.1998.

[5] C.Fragouli, V.Sivaraman, M.B.Srivastava, *"Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queueing with Channel-State-Dependent Packet Scheduling"*, in proc of IEEE INFOCOM 98, p. 572-580, Apr.1998.

[6] A.Demers, S.Keshav, S.Shenker, "*Analysis and Simulation of a fair queueing algorithm*", In proc. of ACM SIGCOMM'89, p.1-12, 1989.

[7] A.K.Parekh, R.G.Gallager, *"A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case"*, IEEE/ACM Trans. On Networking, Vol.1, No.3, p.344-357, Jun. 1993.

[8] D.H.Hoang, D.Reschke, *"A Proactive Model for QoS Supports in Wireless Networks"*. In NetObjectdays 2001, Workshop on MIK, p.287-297, MIK'2001, Erfurt, 9-12 Oct.2001.