Providing Quality of Service over a Shared Wireless Link

Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, and Phil Whiting,
Lucent Technologies
Rajiv Vijayakumar, University of Michigan

Abstract

We propose an efficient way to support quality of service of multiple real-time data users sharing a wireless channel. We show how scheduling algorithms exploiting asynchronous variations of channel quality can be used to maximize the channel capacity (i.e., maximize the number of users that can be supported with the desired QoS).

Introduction

Providing quality of service (QoS), in particular meeting the data rate and packet delay constraints of real-time data users, is one of the requirements in emerging high-speed data networks. This requirement is particularly challenging in networks that include wireless links. Indeed, quality of a wireless channel is typically different for different users, and randomly changes in time on both slow and fast time scales. In addition, wireless link capacity is usually a scarce resource that needs to be used efficiently. Therefore, it is important to find *efficient* ways of supporting QoS for real-time data (e.g., live audio/video streams) over wireless channels (i.e., supporting as many users as possible with the desired QoS).

Efficient data *scheduling* is one of the ways to address the issue described above. In this article we consider the problem of scheduling transmissions of multiple data users sharing the same wireless channel so as to satisfy delay or throughput constraints of all, or as many as possible, users. This problem can be referred to as a *multi-user variable channel scheduling* problem. As mentioned above, the unique "wireless" feature of this problem is the fact that the capacity (service rate) of the channel varies with time randomly and *asynchronously* for different users. The variations in channel capacity are due to different interference levels observed by different users, and also to *fast fading* of the signal received by a user.

The multi-user variable channel scheduling problem arises, for example, in the third-generation (3G) code-division multiple access (CDMA) high data rate (HDR) system [1] (Fig. 1). (See also [2] for a background on CDMA wireless systems.) In HDR, multiple mobile users in a cell share the same CDMA wireless channel. On the downlink (the link from the cell base station to users), time is divided into fixed-size (1.67 ms) time slots. This slot size is short enough so that the channel quality of a user stays approximately constant within one or even a few consecutive time slots. (To be more precise, this is true only for relatively low mobile user velocities [2]). In each time slot data can be transmitted to only one user. Each user i constantly reports to the base station its "instantaneous" channel capacity $r_i(t)$, the rate at which data can be transmitted to the user if it is scheduled for transmission in the current time slot t. The data rate can be chosen from a finite set corresponding to the specified set of data frames; for details see [1]. The channel capacities $r_i(t)$ change in time randomly and *asynchronously* for different users, as illustrated by Fig. 2.

In an HDR system, as well as in any other system where the multi-user variable channel model arises, a scheduling algorithm can take advantage of channel variations by giving some form of priority to users with (temporarily) better channels. Since channel capacities (service rates) of different users vary in time in an asynchronous manner, the QoS of all users can be improved over scheduling schemes that do not take channel conditions into account.

To illustrate the latter point, consider a very simple system with two users. The service rate (in a time slot) for user 1 is either 76.8 kb/s or 153.6 kb/s with equal probabilities 0.5. For user

0163-6804/01/$10.00 © 2001 IEEE

2 the rates are 153.6 kb/s or 307.2 kb/s, also with equal probabilities. (Thus, the channel quality of user 2 is better on average.) Assume that channels for both users are *independent*, and there is unlimited amount of data to transmit to each user. Then, a "naive" *round-robin* allocation of time slots to the users will result in users served at the average rates

$$R_1 = 0.5 \text{ x } (0.5 \text{ x } 76.8 + 0.5 \text{ x } 153.6) = 57.6 \text{ kb/s}$$

and

$$R_2 = 0.5 \text{ x } (0.5 \text{ x } 153.6 + 0.5 \text{ x } 307.2) = 115.2 \text{ kb/s},$$

respectively.

Consider another scheduling scheme where a user with a "relatively better" (in a current time slot) service rate (153.6 kb/s for user 1 and 307.2 kb/s for user 2) is scheduled. In case of a "tie" (i.e., if channels are relatively better or relatively worse for both users), the user to serve is chosen randomly with equal probabilities 0.5 . A straightforward calculation shows that with this scheduling, the average rates are $R_1 = 67.2$ and $R_2 = 134.4$ kb/s, which is 16 percent higher for each user than with the round-robin discipline. This is an example of *proportionally fair* scheduling, proposed and analyzed for HDR in [3]; its goal is to maximize long-term throughputs of the users *relative* to their average channel conditions.

The simple example above shows that, as expected, algorithms which do take current channel conditions into account (e.g., proportionally fair) have clear advantages over standard algorithms (like round-robin or first come first serve) which do not.

The proportionally fair algorithm described above is a viable option for scheduling best effort data traffic: it utilizes asynchronous channel variation to improve the overall system throughput, while giving each user a "fair share" of the throughput. However, this algorithm is less efficient for *real-time* users. For instance, suppose the two users in the simple system described above are real-time users, running an application (e.g., streaming audio) that requires a minimum throughput of, say, 85 kb/s. Is it possible to support this rate for both users? Suppose in addition it is required that packet delays for both users be kept under a certain threshold with high probability. How can this be achieved? In this article we discuss simple scheduling strategies that can be used to achieve such goals, that is, support (different forms of) QoS for multiple users sharing a wireless channel.

The rest of the article is organized as follows. In the next section we discuss two notions of QoS that are relevant to real-time data traffic: providing packet delay and minimum throughput guarantees. Then we define the notion of a throughput-optimal scheduling algorithm and introduce one such algorithm, Modified Largest Weighted Delay First (M-LWDF). Then we show how throughput-optimal scheduling, in particular M-LWDF, can be used to provide different forms of QoS. Finally, we discuss some of the implementation issues.

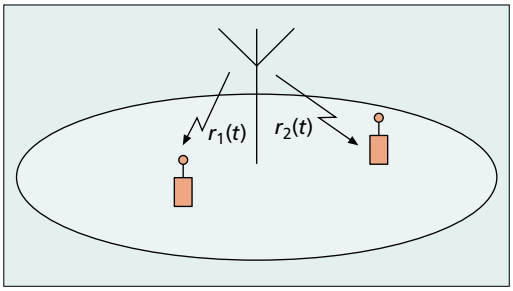

Figure 1. *The downlink of an HDR cell with two data users.*

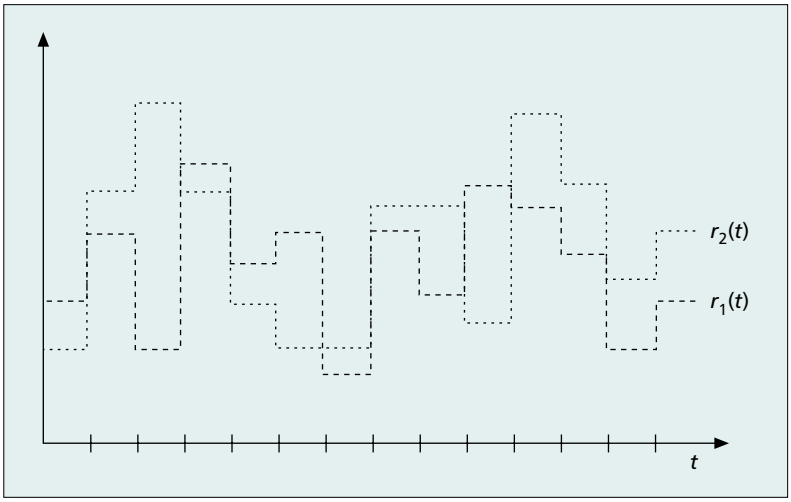

Figure 2. *Channel capacities change asynchronously in time.*

Different Notions of Quality of Service

The QoS of a data user can be defined in different ways. If the data user is a *real-time* user (e.g., it receives live audio or video streams), delays of most of the data packets need to be kept below a certain threshold. More formally, the QoS requirement of user i is

$$Pr\{W_i > T_i\} \le \delta_i, \quad (1)$$

where W_i is a packet delay for this user, and parameters T_i and δ_i are the delay threshold and the maximum probability of exceeding it, respectively.

A different notion of QoS is a requirement that the average throughput R_i provided to user i be not less than some predefined value r_i:

$$R_i \ge r_i. \quad (2)$$

We will show how QoS in either form (1) or (2) can be achieved with the use of *throughput* optimal scheduling schemes, in particular the M-LWDF scheme, which we describe in the next section.

Throughput Optimal Scheduling: The M-LWDF Scheme

Suppose we have N users in a system, and suppose each user (queue) receives a stream of data (i.e., there is no feedback to control input rate). If we want QoS requirement (1) to hold

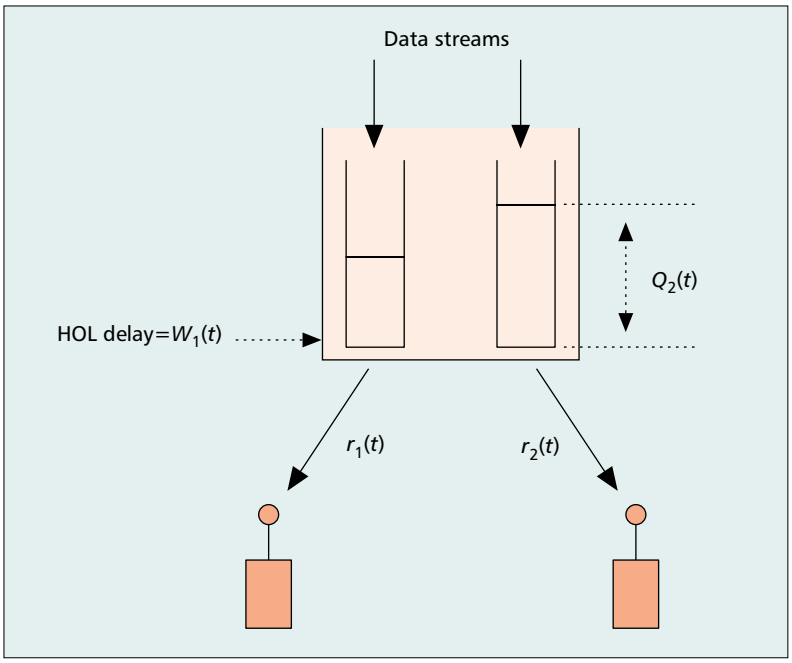

Figure 3. *A queuing model of a cell with two mobiles.*

for all users, obviously the scheduling algorithm must be able to keep all queues (of data intended for different users) *stable*, that is, be able to handle all the offered traffic without queues "blowing up."

We will call a scheduling algorithm *throughput optimal* if it is able to keep all queues stable if this is at all feasible to do with any algorithm.

Consider the following simple "online" scheduling algorithm.

Modified Largest Weighted Delay First (M-LWDF) — In each time slot t, serve the queue j for which

$$\gamma_j W_j(t) r_j(t)$$

is maximal, where $W_j(t)$ is the head-of-the-line packet delay for queue j, $r_j(t)$ is the channel capacity with respect to flow j, and γ_j are arbitrary positive constants.

It has been proved analytically in [4] that

The M-LWDF scheduling algorithm is throughput optimal.

Moreover, the M-LWDF scheduling rule remains throughput optimal if for all or some users, the delay $W_j(t)$ is replaced by the queue length (amount of data) $Q_j(t)$ (Fig. 3).

The key feature of this algorithm is that a scheduling decision depends on both current channel conditions and the states of the queues. The M-LWDF scheme is very easy to implement. The scheduler only needs to time stamp arriving data packets of all users, or keep track of the current queue length. It is somewhat surprising that an algorithm this simple can be throughput optimal: it is able to handle all the offered traffic, unless it is not feasible at all. In addition, a choice of parameters γ_i allows one to control packet delay *distributions* for different users. Increasing the parameter γ_i for user i, while keeping γ_js of other users unchanged, reduces packet delays for this flow at the expense of a delay increase for other flows. Therefore, the delay distributions can be *shaped* [4]; we discuss this further in the next section.

QoS Scheduling

Controlling Flow Delays

The fact that the M-LWDF can handle all data flows (if feasible) does not guarantee that the QoS requirement of form (1) will also be satisfied for all users. This problem is addressed by setting "appropriate" values of the parameters γ_i. The simulations reported in [4] show that M-LWDF scheduling, with $\gamma_i = a_i/\overline{r_i}$, $a_i = -(\log \delta_i)/T_i$, and $\overline{r_i}$ being the average channel rate with respect to user i, performs very well. The rationale behind this parameter setting is as follows.

Parameter a_i embodies the QoS requirement, and provides QoS differentiation between the flows. For example, if users 1 and 2 have the same desired delay thresholds $T_1 = T_2$, but the desired maximum violation probability δ_i is four times less for user 2 than for user 1, then $a_2 = 2a_1$, and therefore user 2 is treated with some priority over user 1. (The basic reason for this setting of the parameter a_i is justified by the theoretical results of [5]). By its definition, the M-LWDF rule (with the above parameter setting) chooses for service a user with the maximal product $a_i W_i$ $(r_i(t)/\overline{r_i})$. Thus, the greater the user i current packet delay, channel quality *relative to its average level*, and the higher the QoS requirement, the greater the chance of this user being scheduled. As a result, this rule approximately "balances" different users' probabilities of deadline violation *relatively to their maximum allowed values* δ_i. Therefore, the rule allows support of *all* users with desired QoS of form (1), if this can be done at all with any other rule. This means that with M-LWDF scheduling, the maximal possible number of users can be supported. We refer the reader to [4] for details of simulation experiments.

Providing Minimum Throughput Guarantees

The problem of providing certain *minimum* throughput r_i for each user (i.e., QoS in form (2)) can also be solved by the M-LWDF scheduling algorithm, if it is used in conjunction with a token bucket control (Fig. 4). Indeed, suppose that, associated with each queue i, there is a *virtual* token bucket. Tokens in bucket i arrive at the *constant* rate r_i. In each time slot, the decision of which flow to serve is made according to the M-LWDF rule, although with $W_i(t)$ being not the delay of actual head-of-the-line user i packet, but the delay of a longest waiting token in token bucket i. After the service of the (actual) queue in the time slot is complete, the number of tokens in the corresponding bucket is reduced by the actual amount of data served. Note that since tokens arrive at a constant rate, $W_i(t)$ = [Number of tokens in the bucket i]/r_i. Thus, only the information on the number of tokens in the bucket is required, which is just a counter implemented in software.

If token queues are stable, the actual throughput of each flow i is at least r_i. Therefore, this scheme *guarantees the minimum throughput* r_i for each user if this is feasible at all. (See [6] for supporting simulation results.) For instance, in the simple two-user system described in the introduction, this "token-based" scheduling algorithm will indeed be able to provide the 85 kb/s throughput for both users, while it would not be possible with any algorithm "ignoring" current channel conditions.

Moreover, this scheme clearly provides *flow isolation*: since scheduling is based on token queues, a large burst of data for one of the users will not effect the minimum throughput provided to other users (see [6] for the simulation results).

We also note that parameters γ_i allow us to control the *time scale* on which throughput guarantees are provided. The greater the γ_i for flow i (relative to γ_j for other flows), the "tighter" the minimum rate assurance for flow i. This means that the desired minimum rate is provided on a finer time scale (i.e., as the average rate over shorter time intervals).

Providing Different Forms of QoS for Different Users

M-LWDF scheduling can still be used if among the users sharing the wireless link there are some which require QoS of form (1) and others QoS of form (2). In this case, the delay W_i used by the scheduling rule will be the actual head-of-line packet delay for the former users and the token delay for the latter ones.

Some Implementation Issues

Connection Admission Control

As described in previous sections, the M-LWDF scheduling guarantees a certain QoS if it is feasible at all. To make sure that the feasibility indeed holds, an efficient connection admission control (CAC) is required. With the M-LWDF rule, a natural *measurement-based* CAC is possible. Roughly speaking, a new data user arrival is blocked if the observed packet delays of existing users are already close to their deadlines. Another aspect of a CAC for a wireless link is that no *absolute* QoS guarantees are possible. Due to user mobility, a configuration of users which is feasible at one time may become infeasible at another. (In an extreme case, a user may move into a location with no radio coverage at all.) Therefore, just as in a conventional voice cellular system, any CAC must include a rule of downgrading QoS of some users (including a complete suspension of service, "dropping" a user), if it becomes infeasible to provide the desired QoS to all users.

Channel Information Inaccuracy

The channel rates $r_i(t)$ which are acceptable to the users at time t can only be *estimated* based on the (delayed in time) channel quality measurements reported by the users to the scheduler. The channel estimates cannot be perfect. This will lead to occasional loss of a data frame transmitted in a time slot (if the channel quality was overestimated) or underutilization of a time slot (in case of underestimated channel quality). This results in some degradation of the QoS experienced by the users. The degree of degradation will depend on the quality of the channel estimation.

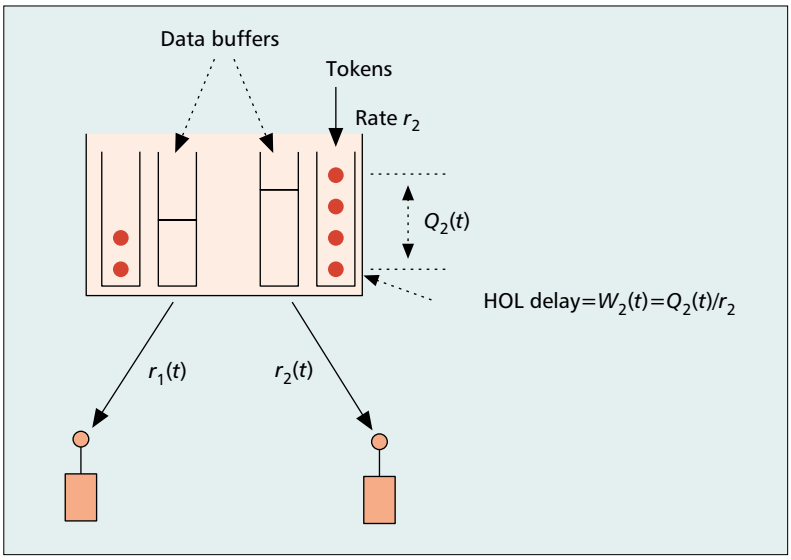

Figure 4. *M-LWDF scheduling applied to token buckets.*

Conclusions

The main conclusion of this article is that it is possible to support real-time data users over a shared wireless link, as in CDMA/HDR. An efficient throughput optimal scheduling algorithm, utilizing asynchronous channel quality variations, can be used to:

- Maximize the number of users that can be supported with the desired QoS
- Provide QoS differentiation between different users
- Provide minimum throughput guarantees and flow isolation

References

[1] P. Bender *et al.*, "A Bandwidth Efficient High Speed Wireless Data Service for Nomadic Users," *IEEE Commun. Mag.*, July 2000.

[2] A. J. Viterbi, *CDMA. Principles of Spread Spectrum Communication*, Addison-Wesley, 1995.

[3] D. Tse, "Forward Link Multiuser Diversity through Proportional Fair Scheduling," presentation at Bell Labs, Aug. 1999.

[5] A. L. Stolyar and K. Ramanan, "Largest Weighted Delay First Scheduling: Large Deviations and Optimality," to appear, *Annals of Appl. Prob.*, 2001, no. 1.

[4] M. Andrews *et al.*, CDMA Data QoS Scheduling on the Forward Link with Variable Channel Conditions," *Bell Labs Tech. Memo.*, Apr. 2000.

[6] S. Shakkottai and A. Stolyar, "A Study of Scheduling Algorithms for a Mixture of Real- and Non-Real-Time Data in HDR," *Bell Labs Tech. Memo.*, Aug. 2000.

Biographies

Matthew Andrews (andrews@research.bell-labs.com) received his B.A. in mathematics from Oxford University in the United Kingdom and his Ph.D. in theoretical computer science from the Massachusetts Institute of Technology. He is currently a member of technical staff in the Mathematics of Networks and Systems department at Bell Laboratories, Murray Hill, New Jersey.

The main conclusion of this article is that it is possible to support real-time data users over a shared wireless link, like in CDMA/HDR.

KRISHNAN KUMARAN (kumaran@research.bell-labs.com) is a member of technical staff in the Mathematics of Networks and Systems Research Department at Bell Laboratories, Murray Hill, New Jersey His interests are in modeling, analysis, and simulation of communication networks. He holds a Bachelor's degree in mechanical engineering from the Indian Institute of Technology, Madras, and a Ph.D in physics from Rutgers University, New Brunswick, New Jersey.

KAVITA RAMANAN (kavita@research.bell-labs.com) received her B.Tech. degree in 1992 from the Indian Institute of Technology, Bombay, and her Ph.D. degree in applied mathematics from Brown University in January 1997, and was a visiting scientist at the Technion, Haifa, Israel, in 1997. She has been at the Mathematical Sciences Research Center at Bell Laboratories as a member of technical staff since late 1997.

ALEXANDER STOLYAR (stolyar@research.bell-labs.com) is a member of technical staff in the Mathematics of Networks and Systems Research Department at Bell Laboratories, Murray Hill, New Jersey. He received a Ph.D. in mathematics from the Institute of Control Science, Moscow, Russia. His research interests are in stochastic processes, queuing theory, and mathematical models of communication systems, especially wireless.

RAJIV VIJAYAKUMAR (rvijayak@engin.umich.edu) received his B.Tech degree in electrical engineering from the Indian Institute of Technology, Bombay, and his M.S. degree in electrical engineering from Louisiana State University, Baton Rouge. He spent the summer of 1999 as an intern with the Mathematics of Networks and Systems Department at Bell Laboratories. He is currently a Ph.D. candidate at the University of Michigan, Ann Arbor.

PHIL WHITING (pwhiting@research.bell-labs.com) received his B.A. degree from the University of Oxford, his M.Sc. from the University of London, and his Ph. D. in queuing theory from the University of Strathclyde. After a post-doc at the University of Cambridge, his interests centered on wireless. In 1993 he participated in the Telstra trial of Qualcomm CDMA in southeastern Australia. He then joined the Mobile Research Centre at the University of South Australia Adelaide. Since 1997 he has been with Bell Laboratories, where his main interests are the mathematics of wireless networks, particularly stochastic models for resource allocation and information theory.