Price-based Channel Time Allocation in Wireless LANs

Samarth H. Shah and Klara Nahrstedt *
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801, USA
Email: {shshah, klara}@cs.uiuc.edu

Abstract

In access-point based wireless networks, employing the IEEE 802.11 protocol in DCF mode, without bandwidth management and rate control, users pump data into the network as fast as it is generated. This results in a loss of QoS for the user and performance degradation of the network. There is a need for bandwidth allocation and for users to co-operatively control their rates based on the allocated bandwidth. In this paper, we use price as a signal for bandwidth allocation in wireless hot spot networks. We allocate more bandwidth to users that pay more. At the same time, we also aim to maximize channel utilization and increase mean satisfaction across all the users.

1 Introduction and Motivation

Recent years have witnessed the emergence of the phenomenon of near-ubiquitous Internet access through wireless "hot spot" networks. A *hot spot* is a hotel, airport, cafe or restaurant providing access to the Internet via an access-point (AP) based wireless local-area network (WLAN) that usually employs the IEEE 802.11 protocol in DCF mode. Users can walk into the hotel or cafe with their 802.11 interface-equipped laptops or PDAs and, after some network configuration, can access the Internet. Such hot spot wireless networks are also available at most technical conferences for the use of the delegates.

One major challenge in hot spot networks is the distribution of the shared resource of *bandwidth* amongst the numerous users. Users can have different requirements, based on the applications they are running. In addition, they may also be willing to pay different sums of money for the same quantity of bandwidth. This is because the same amount of bandwidth is worth a lot to some users (e.g. a stock-broker who needs to make an important buying/selling decision based on the price of a certain stock) and not so much to others (e.g. a person who is checking his/her e-mail for the fourth time in an hour). In the absence of a bandwidth arbitration mechanism, each user will pump data into the network as fast as it is generated and if the sum of the rates exceeds the channel capacity, then there will be severe performance degradation. Bandwidth management and co-ordinated rate control enhances the QoS of the individual applications, as shown in [12]. In this paper, we address this problem of how to distribute bandwidth in a wireless LAN between users with different requirements and willing to pay different sums of money. We attempt to maximize channel utilization as well as system revenue, while keeping user satisfaction also in mind.

Each user, once allocated a certain portion of the total network bandwidth, must co-operate with the network provider and control his/her rate to conform to the bandwidth fraction allocated. While we do not address this problem of co-operative and collective rate-control, it is an active area of research at various layers of the OSI protocol stack. Fair scheduling at the MAC layer [13] addresses this problem of rate control. We address the problem from the point of view of the *policy* in allocating a portion of the overall network bandwidth to each user. The *mechanism* of how each user conforms to its allotted share falls under the domain of fair scheduling at the MAC layer and other rate control schemes.

The rest of this paper is organized as follows. Section 2 is a formal representation of the problem. Section 3 discusses the concept of channel time proportion which is integral to our scheme. In Section 4, we present the overall architecture of our scheme and our channel time allocation algorithm. Section 5 contains a summary of our experimental setup and results. Section 6 discusses the related work in the area and Section 7 concludes the paper.

2 Network Model

We assume a single IEEE 802.11 subnet with a single AP and a set S of users with wireless hosts. Each user $f \in S$ has a maximum bandwidth requirement $b(f, max)$ and a minimum bandwidth requirement $b(f, min)$, both in bits per second. The maximum and minimum bandwidth requirements include both uplink and downlink bandwidths. They can be estimated by the user based on the applications he/she intends to run and can be changed at any time during the user's activity. For best-effort flows, $b(f, min) = 0$. The user's utility curve is thus of the shape shown in Figure 1. Such a utility curve has also been used previously in literature [12, 10].

One of the major contributions of our scheme is that we convert bandwidth requirements into *channel time proportion (CTP)* requirements. The channel time proportion is the fraction of unit

*This work was supported by the DoD Multi-disciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant NAVY CU 37515-6281, and the NSF EIA 99-72884 grant. Any opinions, findings and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

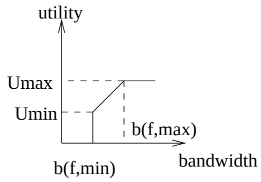


Figure 1. Utility curve of users.

time a user utilizes the channel for sending and receiving his/her data. This is directly related to the bandwidth because, larger this fraction, the more data in unit time a user can send/receive, and thus the larger his/her throughput. The details of the conversion of bandwidth requirements into CTP requirements are discussed in Section 3. The minimum bandwidth requirement of user f, $b(f, min)$ is converted into a minimum CTP requirement $c(f, min)$ and the maximum bandwidth requirement $b(f, max)$ is converted into a maximum CTP requirement $c(f, max)$. Obviously, $c(f, max) \leq 100\%$.

Each user f also has a maximum price $mp(f)$ that he/she is willing to pay per minute for 1% of the channel time [1]. $B(f) = mp(f) \cdot c(f, max)$ is thus the *maximum bid* (in cents per minute) that the user f is willing to make for the service it is requesting. The values of $b(f, min)$, $b(f, max)$ and $B(f)$ are valid for the entire session of user f, unless changed by the user in between.

A centralized *Bandwidth Manager (BM)* situated at the AP takes as input the CTP requirements of all the active users as well as their maximum bids. It outputs the current price p (in cents per minute for 1% of the channel time) and the CTP $c(f, a)$ allocated to each user $f \in S$, which are calculated using the algorithm presented in Section 4. $c(f, min) \leq c(f, a) \leq c(f, max)$. The revenue from a particular user f, in cents per second, is $p \cdot c(f, a)$. If this is smaller than user f's bid $B(f)$, then the balance is refunded to the user. The refund is very important because it encourages the user to bid high. If the bid is excessive, no harm is done, the balance is refunded. The instantaneous revenue of the network provider, in cents/min, is $R = p \cdot \sum_{f \in S} c(f, a)$. The total revenue is R aggregated over the entire time of operation of the network. If the set S changes (users arrive or leave), or if a user changes his/her parameters, then the price p and the CTPs of all the users f have to be recomputed, and the value of R in cents/min also changes.

3 Channel Time Proportion (CTP)

3.1 Effective Channel Capacity

In the previous section, we mentioned that we converted bandwidth requirements of the users into channel time proportion (CTP) requirements. The motivation behind this was that each user's communication with the AP is affected by varying levels of medium contention, fading and interference. If these physical errors greatly affect the wireless link between the user and the AP, then a greater length of time must be expended towards sending a single IEEE 802.11 MAC frame, belonging to this user-AP connection, over the link. This is because the wireless terminal (user or AP) may have to wait longer for the medium to be unoccupied (backoffs are larger), and/or deal with collisions and RTS/DATA re-transmissions. If a greater length of time is expended in sending this connection's frames over the wireless link, then the effective bandwidth or channel capacity $b_e(f)$ perceived by this user-AP connection (reciprocal of the time expended) is smaller. Fewer frames of this user-AP connection can be sent over the wireless link in unit time. Since each user-AP wireless link is affected by a different level of physical errors, each link perceives a different effective network channel capacity. The physical errors also vary with time, so the effective channel capacity $b_e(f)$ also varies with time.

[1] We have arbitrarily picked minutes and cents, respectively, as the units of time and money in this paper. Similarly, the minimum resolution of CTP used in this paper (1%) has been arbitrarily chosen. Network providers can set these parameters to suit their network.

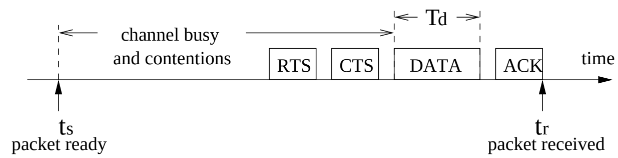


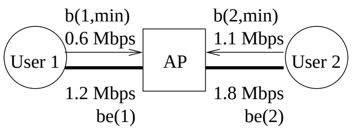
Figure 2. Measurement of $b_e(f)$.

Figure 2 describes the measurement of $b_e(f)$ at the MAC level. The effective channel capacity is estimated by averaging the value of $b_e(f) = \frac{1}{t_r - t_s}$ measured over several frame transmissions. This measurement takes into account medium contention effects due to other users, which would result in a longer backoff interval, and also fading and interference effects (at both sender and receiver), which may cause a loss and consequent re-transmission of RTS or DATA packets. The measurement can be done in the MAC layer 802.11 device driver at the AP alone, for each user-AP link. The BM program at the AP periodically probes its MAC layer API for the $b_e(f)$ estimate for the link to each user f. Details of the measurement mechanism, including dealing with initial conditions and normalization for different frame sizes, are available in [3]. The measurement mechanism has been tested for accuracy using the ns-2 simulator and also used in a real 802.11 network [12].

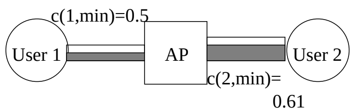
3.2 Converting Bandwidth to CTP

We use the effective channel capacity $b_e(f)$ perceived on a user-AP link, determined in the previous subsection, to convert the user f's bandwidth requirements to its CTP requirements. We can illustrate using an example the need for doing this conversion prior to admission control. Assume that the delay $t_r - t_s$ from Figure 2 in transmitting a single MAC frame is such that 10 frames of a particular size s can be transmitted in a second over the user-AP link for user f. Assume that user f's minimum bandwidth requirement is 3 frames of size s per second. Then, user f requires at least 30% of the channel capacity. It needs to be active on the channel for 30% of unit time to meet its minimum bandwidth requirement. This leaves only 70% of unit time available to other users, which directly affects their admission. Although our example is in terms of frames per second of frame size s, we can extend this logic to bits per second also. If k bits can be transmitted over a wireless link in a second, given a certain level of physical errors on it, and a user requires a minimum throughput of l bits per second, then in effect the user requires a fraction $\frac{l}{k}$ of unit time on the channel. The purpose behind the normalization operation in [3] is to convert the frames per second capacity of a wireless link, for different frame sizes, into a common bits per second capacity.

The CTP requirements of user f, $c(f, min)$ and $c(f, max)$, can thus be obtained by simply dividing its respective bandwidth requirements $b(f, min)$ and $b(f, max)$ by $b_e(f)$. $c(f, min) = \frac{b(f,min)}{b_e(f)}$ and $c(f, max) = \frac{b(f,max)}{b_e(f)}$.



(a) Using bandwidth in admission control.



(b) Using CTP in admission control.

Figure 3. Difference between using bandwidth and CTP for admission control.

Figure 3 shows the difference between using CTP for admission control as opposed using pure bandwidth as in [10]. In Figure 3(a), both users are admitted because admission is done purely on bandwidth and the minimum requirements are less than the effective channel capacity measured for the respective users. In Figure 3(b), both users cannot be admitted because, in sum, they require more than 100% of unit time on the channel, which is obviously not possible. If admission control based purely on bandwidth requirements and not CTP (Figure 3(a)) is done, then an overflow will occur at the interface queue.

4 System Architecture and Channel Time Allocation Algorithm

The components of the system and its overall architecture are very similar to that of the bandwidth management scheme described in [12]. The price-based channel time allocation algorithm, however, is completely different from the "fair" allocation algorithm in [12].

4.1 System Architecture

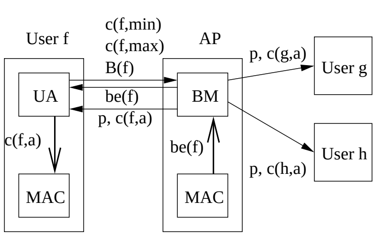


Figure 4. System architecture.

The overall architecture of the system is shown in Figure 4. The User Agent (UA), at the application layer, takes from the user f the values of $b(f, min)$ and $b(f, max)$. It obtains from the Bandwidth Manager (BM) at the AP the value of $b_e(f)$ for its link with the AP. The UA then computes CTP requirements $c(f, min)$ and $c(f, max)$, as described in the previous section, and sends them, along with user f's maximum bid $B(f)$ for these CTP requirements, to the BM at the AP. The BM computes, using the algorithm in the next subsection, the system price p and CTP allocated to each user. It then conveys these values to all UAs of all the users through a subnet broadcast. The UA of each user passes the CTP allocated to it to the scheduler at the MAC level so that it can use it as the "weight" of this user in the fair scheduling algorithm. A performance evaluation of this basic scheme via ns-2 simulations as well as in a real 802.11 network testbed, coupled with a "fair" channel time allocation algorithm, is available in [12].

If at any stage, the user's requirements or bid change or the value of $b_e(f)$ varies significantly, the UA has to re-negotiate with the BM with these new parameters. (The BM informs the UA of user f the $b_e(f)$ value it periodically retrieves from the AP's MAC layer API.) There is incentive for the user to re-negotiate even when requesting a smaller CTP, because the smaller the CTP requested, the lower the price he/she has to pay. If a user is within range of multiple APs, then his/her UA can just pick whichever one gives the user the best deal, in terms of CTP allocated and price charged. At any time, a user can query a BM for system information such as the current price p, reserve price $reservep$, or channel utilization $\sum_{f \in S} c(f, a)$, and use this information in his/her bidding decisions. The BM may also provide a new user all the parameters of all the existing users, without actually naming them, so that the new user can *locally* execute the BM's channel time allocation algorithm, and obtain a resultant price and CTP allocation. Depending on this result, the new user may decide to change some of its parameters, or may decide to not bid at all, at this time.

4.2 Channel Time Allocation Algorithm

The pseudo-code of the channel time allocation algorithm is shown in Figure 5. This algorithm runs at the BM and is responsible for determining the current channel price p and the individual users' CTP allocation.

The network provider sets a fixed *reserve* price which is derived from the cost of maintenance of the network. The price per second of 1% of CTP is never lower than the reserve price. In the trivial case where $\sum_{f \in S} c(f, max) \leq 100\%$, the price is just set to the maximum of the reserve price and $\min_{f \in S}\{mp(f)\}$. A reserve price ensures that users do not get away with setting their $mp(f)$ very small, and thus yielding very low revenue to the system. By setting the price to $\min_{f \in S}\{mp(f)\}$, in the case where it is greater than than $reservep$, our algorithm finds the largest price for which average user satisfaction is 100% and channel utilization is maximum. (See Appendix for proof.) Alternatively, in this trivial case, the network provider may opt to just set the price equal to the reserve price, irrespective of the $mp(f)$ values.

proc $CTAlloc(V := S sorted\ in\ non-decreasing\ order\ of\ mp(f)) \equiv$

do

$redo := false;$

if $(\sum_{f \in V} c(f, max) \leq 100\%)$ **then**

$p := max\{reservep, \min_{f \in V}\{mp(f)\}\};$

foreach $f \in V$ **do**

$c(f, a) := min\{c(f, max), B(f)/p\};$

od

else

$W := Move\ users\ from\ top\ of\ V\ until\ \sum_{f \in V} c(f, max) < 100\%\ or\ V = \{\};$

while $\left(\left[p := max\{reservep, \frac{\sum_{g \in W} B(g)}{100\% - \sum_{f \in V} c(f, max)}\}\right] > \min_{f \in V}\{mp(f)\}\right)$ **do**

if $(V = \{\})$ **then**

$break;$

else

$W := W + First\ user\ \in V;$

fi

od

foreach $f \in V$ **do**

$c(f, a) := c(f, max);$

od

foreach $g \in W$ **do**

$c(g, a) := B(g)/p;$

od

fi

if $(\exists f \in V \mid c(f, a) < c(f, min))$ **then**

$delete(f, V);$

$redo := true;$

fi

while $(redo = true);$

.

Figure 5. Channel time allocation algorithm.

In the non-trivial case $\sum_{f \in S} c(f, max) > 100\%$, the BM allocates the CTP so that revenue is maximized while keeping channel utilization at 100%. The users $f \in S$ are sorted in non-decreasing order of $mp(f)$. Let V be this sorted set. Repeatedly, the user with the lowest value of $mp(f)$ is removed from V and added to a set W until $\sum_{f \in V} c(f, max) < 100\%$. The price is now set to $p = \frac{\sum_{g \in W} B(g)}{100\% - \sum_{f \in V} c(f, max)}$. If $p \leq \min_{f \in V}\{mp(f)\}$, then the algorithm terminates right there. If not, the current split of users between sets V and W is invalid. So, once again the user with the minimum $mp(f)$ is moved from V to W and the price is recomputed for the new values of set V and set W. The procedure continues until a value of p is found that is $\leq \min_{f \in V}\{mp(f)\}$ (i.e., a valid split of users between sets V and W) or until set V is empty. The last value of p computed is the price for the set of users S. The portion of the algorithm described so far runs in $O(\|S\|)$ time.

The CTP allocated to each user $f \in S$ is simply $c(f, a) = min\{c(f, max), B(f)/p\}$. If $c(f, a) < c(f, min)$ then user f is deleted from S. He/she is *blocked* because he/she is not paying enough to even have his/her minimum CTP requirement be satisfied. The algorithm must run all over again for the new value of set S. The worst-case overall running time of the algorithm is thus $O(\|S\|^2)$.

Basically, in the non-trivial case when $\sum_{f \in S} c(f, max) > 100\%$, the algorithm finds the maximum possible valid price for which channel utilization is 100%. The algorithm thus maximizes revenue and channel utilization simultaneously. As mentioned in the next subsection, if the users were to enter into an auction, the price they will naturally settle upon at equilibrium is the price our algorithm ultimately settles upon, in the case where $\sum_{f \in S} c(f, max) > 100\%$. This price thus reflects the true worth of channel time amongst the users in the network.

If the algorithm terminates with set V not empty, then it basically means that the CTP allocated to each user in set W is limited by its maximum bid, and the CTP allocated to each user in set V is limited by its maximum requirement. In other words, the users in set V have paid more than they require to satisfy their maximum CTP requirement and will hence receive a refund. Conversely, the users in set W have paid less than they require to satisfy their maximum CTP requirement and will have to settle for less CTP, while their entire bid is consumed by the system [2]. Incidentally, the algorithm ensures that users with the same value of $mp(f)$ are, ultimately, either placed all together in V or all together in W. Thus, when initially sorting the users in non-decreasing order of $mp(f)$ in V, ties can be broken arbitrarily.

At any instant, the price for 1% CTP is the same for all users [3]. As users arrive and leave, the price varies depending on the competing demand. The user is billed the instantaneous price times his/her instantaneous allotment, aggregated over his/her entire session, when he/she leaves the network. We adopt an "acceptable or nothing" QoS philosophy. A user gets acceptable QoS (between minimum and maximum requirements) or no service at all. When demand increases and CTP becomes dearer, users willing to pay too little for their minimum requirement are blocked [4]. Such users can either increase their maximum bid or decrease their minimum requirement and rejoin the network, after a time delay T. The algorithm works in such a way that, if $c(f, min) = 0$ and $c(f, max) = 100\%, \forall f \in S$, then the users will be allocated CTP in the ratio of their $mp(f)$ values. The price in this case will be $\sum_{f \in S} mp(f)$ (cents/sec)/%CTP.

4.3 Centralized Auction of Channel Time

For the case $\sum_{f \in S} c(f, max) > 100\%$, the algorithm described in the previous subsection is actually a centralized version of a distributed auction of channel time. It must be noted that an auction is a natural way of resource allotment when each bidder values the resource differently. This is the case with wireless channel time in a hot spot network, as we mentioned in Section 1.

Assume an ascending, multi-unit auction [6] of channel time between the users, such that each user is allotted a CTP proportional to its bid. (CTP allotted at any instant to a user is equal to the user's bid divided by the sum of all bids at that instant.) Now, users increase their bids in order to obtain more and more CTP. This bidding continues until all users either obtain their maximum CTP requirement $c(f, max)$ or reach the maximum they are willing to bid $B(f)$, i.e., the maximum sum of money they can afford. At this point, the system reaches equilibrium until a new user arrives, an existing user leaves, or some other parameter, such as a user's maximum bid, changes. The equilibrium price p is the true worth of CTP amongst the users.

Table 1 illustrates an example of such a proportional, ascending, multi-unit auction. Three users, f_1, f_2 and f_3 compete for CTP. For the sake of simplicity, the minimum CTP requirements $c(f, min)$ of all users are kept at 0%. The maximum requirements $c(f, max)$ of the three users are 20%, 40% and 60% respectively. Their respective maximum budgets $B(f)$ are 6, 10 and 12 cents per minute, giving them the respective $mp(f)$ values of 0.3, 0.25 and 0.2 (c/min)/%CTP. The table shows how the auction plays out. The users each start bidding at the reserve price of 0.1 (c/min)/%CTP. At each stage, the %CTP available to a user is proportional to its instantaneous bid. We assume only one bid per iteration. The order of bidding has no bearing on the ultimate outcome of the auction, although the individual iterations may differ for different orders. Without any loss of generality, we assume a round-robin order of bidding between the three users. The only information required by a user to compute its next bid on its next turn is the current price. Using this alone, he/she can figure out what should be his/her next bid in order to obtain either the maximum requirement or exhaust the maximum budget.

Obviously, such a distributed auction is infeasible in our scenario. The repeated bids, each resulting in a different CTP allotment, constitute a very large overhead. The number of bids or iterations depends on the reserve price, the number of users, and the respective requirements and maximum bids of the users. The delay in attaining equilibrium can also be untenably long, and during this time, the users' CTP allotment will be continuously varying. We *centralize* this auction by having all the users provide the BM their limiting parameters: $c(f, max)$ and $B(f)$. The channel time allocation algorithm described earlier then *simulates* the ascending auction. It determines the same equilibrium price, i.e. 0.275 (c/min)/%CTP in the example, and CTP allotment as the distributed auction (see Appendix for proof) in $O(\|S\|^2)$ time, while the distributed auction would have required several iterations of bidding from each user. The auction in the example takes six iterations, while our algorithm takes only two for the same input. (Two users are successively moved from V to W.)

One possible problem with the auction-based channel time allocation policy is that of *collusion*. If all the bidders co-operate with each other and decide to make very small bids, then the system's revenue can be adversely affected. As is well known in auction literature [9, 4], having a reserve price mitigates this problem to an extent. Furthermore, in a hot spot network at a public place such as an airport or cafe, it is impractical for *all* the users of the network to meet and agree to collude. The collusion fails if there is even a single user from outside the group of colluders present in the network. Thus, if all the colluders have small maximum bids ($B(f)$) and only one user makes a normal maximum bid, that user will get most of the channel time and the others will all be adversely affected, causing them to increase their maximum bids. The promise of more channel time for a larger bid, as well as the assurance that any extra money will be refunded, should encourage users to set their maximum bids $B(f)$ high, and thus increase system revenue. The refund helps to avoid the "winner's curse" [2].

5 Results

The channel time allocation algorithm described in the previous section is a *variable price* algorithm because the price changes according to the "richness" of the users in set S. In this section, we evaluate the performance of this algorithm and also compare it with two *fixed price* schemes:

1. In the *fixed price proportional* (FPP) scheme, each user f is initially allotted CTP $c_r(f, a) = B(f)/p_r$ where p_r is the fixed price. If $\sum_{f \in S} c_r(f, a) > 100\%$, then the allocations

[2] It is not logical for a user with a larger $mp(f)$ to have its maximum requirement satisfied while one with a smaller $mp(f)$ value is not fully satisfied. This is the reason why users are moved from V to W only in non-decreasing order of $mp(f)$.

[3] Incidentally, the price for 1 bps throughput is different for users with different channel qualities. This is also the case in [8].

[4] Best-effort users, since they have $c(f, min) = 0$, are always admitted.

Iteration	Action	Bids (c/min)	Price ((c/min)/%CTP)	Allocations (%CTP)	Comments
0	Initial	2,4,6	Reserve price: 0.1	17,33,50	none satisfied
1	f_1 bids	2.5,4,6	0.125	20,32,48	f_1 satisfied
2	f_2 bids	2.5,5.67,6	0.142	18,40,42	f_2 satisfied
3	f_3 bids	2.5,5.67,12	0.202	12,28,60	f_3 satisfied/exhausted
4	f_1 bids	4.42,5.67,12	0.221	20,26,54	f_1 satisfied
5	f_2 bids	4.42,10,12	0.264	17,38,45	f_2 exhausted
6	f_1 bids	5.5,10,12	Eqm. price: 0.275	20,36,44	f_1 satisfied f_2 exhausted f_3 exhausted

Table 1. Example auction.

$c_r(f,a)$ are scaled down proportionately so that their sum equals 100%. If, for any user, $c_r(f,a) > c(f,max)$, then $c_r(f,a) = c(f,max)$. If, for any user, $c_r(f,a) < c(f,min)$, then this user is blocked on the grounds of insufficient budget.

2. The *fixed price greedy* (FPG) scheme greedily allocates CTP, using price p_g, to the users with the lowest maximum requirements. Users with larger maximum requirements get blocked, once 100% of the CTP has been allocated. Users who have too small a budget to even satisfy their minimum requirements, at the fixed price p_g, are also blocked.

Our simulation scenario consisted of 100 users with random arrival and departure times in a 5-hour time window. The minimum CTP requirements of the users ($c(f,min)$) were uniformly distributed in the range 0% to 2%, corresponding to 0 to 40kbps for a 2Mbps channel. The maximum CTP requirements were uniformly distributed in the range 2% to 10%, corresponding to 40kbps to 200kbps for a 2Mbps channel. The value of $mp(f)$, in (cents/min)/%CTP, for each user f was randomly chosen from the set $\{$0.1, 0.2, ..., 1.0$\}$. The reserve price for the network was set at 0.1 (cents/min)/%CTP.

Figure 6(b) is a plot of the number of users requesting service and the number of users admitted, using our variable price scheme. The remaining users are blocked. For the sake of simplicity, we assume that blocked users do not return and request service later. (Alternatively, we can assume that a user arriving later is actually a returning user, with different parameters.) Figure 6(a) shows the variation in price as users arrive and leave. The average price for the entire simulation run is 0.75 (cents/min)/%CTP. Figure 6(c) illustrates the mean satisfaction of the admitted users. We measure satisfaction, as a percentage, for a user f as: $\frac{c(f,a)}{c(f,max)} \cdot 100$. Figure 6(d) is a plot of the channel utilization, in percentage, over the course of the simulation. As more users are admitted, channel utilization increases but the individual users are allocated less CTP, so their satisfaction falls. The mean user satisfaction and channel utilization, averaged over the entire 5-hour simulation run, are shown under the "Variable Price" row of Table 2. The "Revenue" column of Table 2 shows the total revenue earned at the end of the 5-hour simulation run.

Table 2 presents the results of the comparison between our variable price channel time allocation algorithm and the fixed-price algorithms. For lower fixed prices, the revenue for the system is lower, but the mean user satisfaction and channel utilization are higher. This is because at lower prices, the users' budgets can buy them large CTPs. At higher prices, the revenue is higher, but mean user satisfaction, channel utilization and the blocking factor are worse. Our variable price scheme attempts to simultaneously optimize all the performance parameters. While there may be prices for which the fixed price schemes slightly outperform our scheme under one of the performance metrics, the corresponding penalty paid by the fixed price scheme in terms of the other metrics is large. For example, the FPP scheme, with a price of 1.5 (cents/min)/%CTP yields a 10% higher revenue than our scheme, but the mean user satisfaction is nearly halved, channel utilization falls from 83% to 51%, and blocking factor is also worse.

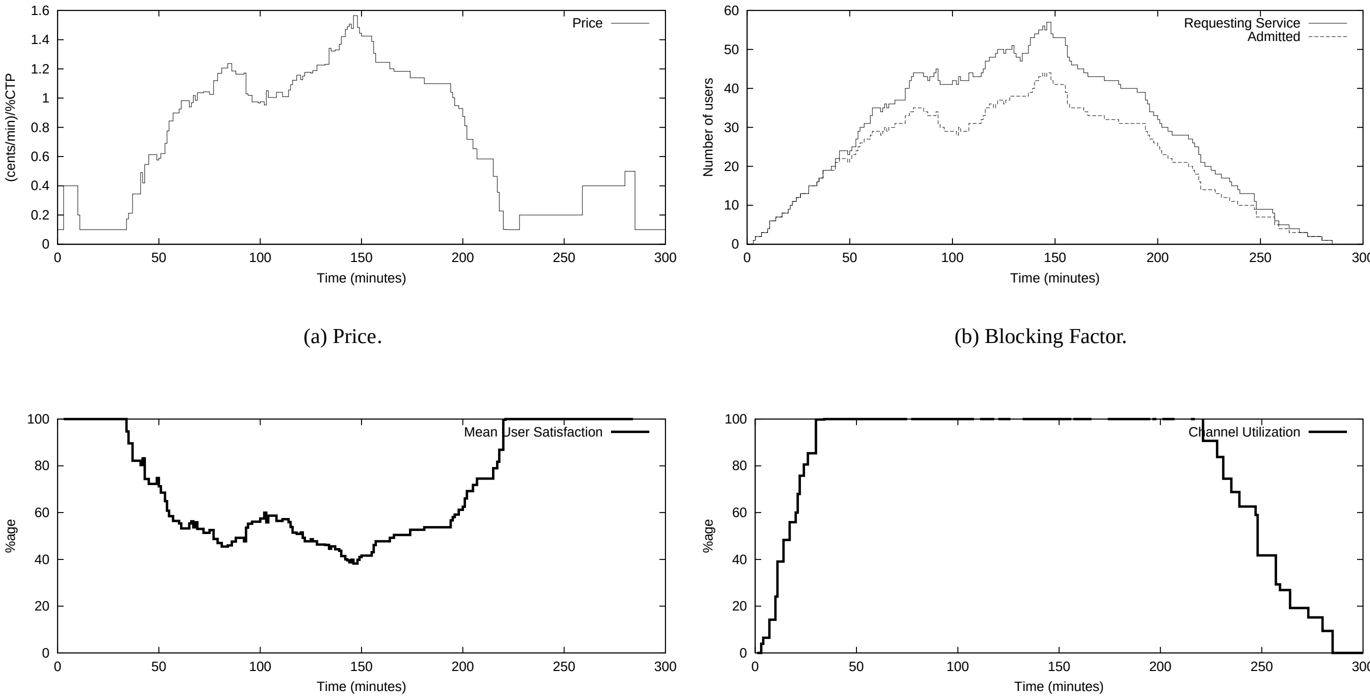
It should be noted that, even at the low fixed price of 0.2 (cents/min)/%CTP, the fixed price schemes do not necessarily perform better than our variable price algorithm in terms of channel utilization and mean user satisfaction. This is because our reserve price is lower than 0.2 (cents/min)/%CTP. Thus, when our scheme defaults to the reserve price, it results in much better utilization and mean user satisfaction at those instants, which affects the overall mean utilization and mean user satisfaction comparisons.

Also note that the fixed price schemes cannot know the average worth of the channel time to the users before hand. But even if this knowledge *were* available, and the fixed price were set to this average worth (i.e., the average price of the variable price scheme 0.75 (cents/min)/%CTP), the performance still does not match the performance of the variable price scheme.

6 Related Work

In this section, we survey the literature in the field of price-based bandwidth allocation in different types of wireless networks. We point out the differences from our work, and finally highlight our contributions.

An early work on pricing and bandwidth adaptation in wireless networks was the TIMELY project [1]. Users benefitting from adaptation were charged and those suffering from adaptation were compensated. However, how the exact charges and credits were calculated, was not specified. In [8], the authors discuss price-based resource allocation on the downlink of a time-slotted or CDMA-based wireless LAN. They assume that users do not know each others' utility curves. In [7], the authors divide the network bandwidth



(a) Price.

(b) Blocking Factor.

(c) Mean User Satisfaction.

(d) Channel Utilization.

Figure 6. Performance evaluation of our variable price channel time allocation algorithm.

into stable (low bandwidth) and instantaneous (unstable, high bandwidth) classes, and broadcast a price-service menu for these classes periodically. The users are given an incentive to truthfully declare their required bandwidth and service class. As far as we know, the concept of *channel time proportion* is unique to our scheme. Our scheme also directly auctions channel time so that users who value it more obtain more of it.

There has also been research in the area of price-based resource allocation for wireless ad-hoc networks [14, 11]. In [14], the authors argue that the shared resource is not a link, as in the case of a wireline network, but a wireless neighborhood clique. On the basis of this, they adapt the concepts of Kelly et al's seminal paper [5] on price-based resource allocation in wireline networks to mobile ad hoc networks.

7 Conclusion

In this paper, we presented a price-based channel time allocation scheme for a wireless hot spot network, in order to alleviate the problem of congestion, and provide admitted users acceptable QoS. We made two main contributions in this work. First, we introduced the concept of *channel time proportion* (CTP), and converted users' throughput requirements to channel time requirements. Secondly, we presented a centralized auction algorithm, of quadratic time-complexity, to determine the instantaneous price of the channel and the users' respective CTP allocations. We compared the performance of our algorithm with two fixed-price channel allocation schemes and found that our algorithm does the best job of simultaneously maximizing revenue, user satisfaction and channel utilization.

8 Acknowledgement

The authors would like to thank Mr. Kai Chen for his valuable assistance.

References

[1] V. Bharghavan, K. Lee, S. Lu, S. Ha, J. Li, and D. Dwyer. The timely adaptive resource management architecture. *IEEE Personal Communication Magazine*, 5(4), August 1999.

[2] J. Bulow and P. Klemperer. Prices and the winner's curse. *RAND Journal of Economics*, 2002.

[3] K. Chen and K. Nahrstedt. Exact: An explicit rate-based flow control framework in manet. Tech. Report UIUCDCS-R-2002-2286/UILU-ENG-2001-1750, Dept. of Computer Science, UIUC, July 2002.

[4] K. Hendricks and R. Porter. Collusion in auctions. *Annales D'Economie et de Statistique*, 15/16, 1989.

[5] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of Operational Research Society*, 49, 1998.

[6] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 1999.

[7] R. Liao, R. Wouhaybi, and A. Campbell. Incentive engineering in wireless lan based access networks. In *IEEE ICNP*, Paris, France, November 2002.

Algorithm	Price ((c/min)/%CTP)	Revenue (c)	Avg. Satisfaction	Channel Utilization	Users blocked
Variable Price		19617	71%	83%	24
FPP	0.2	4496	68%	80%	24
	0.75	15718	59%	74%	24
	1.5	21410	38%	51%	26
FPG	0.2	3750	94%	69%	50
	0.75	13245	66%	64%	40
	1.5	20766	38%	50%	27

Table 2. Comparison of variable price and fixed price channel time allocation algorithms.

[8] P. Marbach and R. Berry. Downlink resource allocation and pricing for wireless networks. In *IEEE InfoCom*, New York, NY, June 2002.

[9] R. McAfee and J. McMillan. Bidding rings. *American Economic Review*, 82, 1992.

[10] M. Mirhakkak, N. Schult, and D. Thomson. Dynamic bandwidth management and adaptive applications for a variable bandwidth wireless environment. *IEEE JSAC*, 19(10), October 2001.

[11] Y. Qiu and P. Marbach. Bandwidth allocation in ad-hoc networks: A price-based approach. In *IEEE InfoCom*, San Francisco, CA, March-April 2003.

[12] S. Shah, K. Chen, and K. Nahrstedt. Dynamic bandwidth management in single-hop ad hoc wireless networks. In *IEEE PerCom*, Fort Worth, TX, March 2003.

[13] N. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless lan. In *ACM MobiCom*, Boston, MA, August 2000.

[14] Y. Xue, B. Li, and K. Nahrstedt. Price-based resource allocation in wireless ad hoc networks. In *IEEE IWQoS*, Monterey, CA, June 2003.

Appendix

Theorem 1 *When* $\sum_{f \in S} c(f, max) > 100\%$*, the price and channel time allocation obtained by our algorithm is identical to the equilibrium price and equilibrium channel time allocation of the proportional, ascending, multi-unit auction described in Section 4.3.*

Proof: An equilibrium is reached in the auction when it is either infeasible or impossible for any user in the system to bid a larger amount. It is infeasible for users to do so when they already have their maximum requirement. It is impossible to do so when they have already exhausted their entire maximum bid. Until he/she reaches one of these two conditions, each user continues to bid higher and higher.

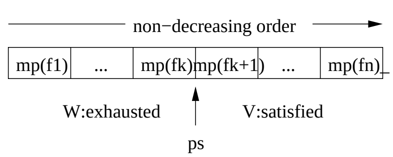


Figure 7. Split between exhausted and satisfied flows.

Ultimately, as shown in Figure 7, both schemes determine a price p_s that splits a set of users arranged in sequence of non-decreasing $mp(f)$ into two subsets:

1. The subset V of users with $mp(f) \geq p_s$. Each user in this subset obtains his/her maximum CTP requirement.

2. The subset W of users with $mp(f) < p_s$. Each user in this subset cannot afford his/her maximum CTP requirement but exhausts his/her maximum bid towards obtaining as much CTP as possible.

We have to prove that the splitting point of the sequence is the same for both schemes. We prove this by contradiction. The splitting point as determined by the auction at equilibrium cannot be lower in the $mp(f)$ sequence than that determined by our algorithm because our algorithm scrolls over all possible splitting points one by one from the lowest to highest value of $mp(f)$. If a valid split in the sequence existed earlier, such that all users were either satisfied or exhausted, our algorithm would have found it. Our algorithm finds the lowest possible valid splitting point in the non-decreasing sequence, if any. The auction can find a higher splitting point only if users by-pass the lower one found by our algorithm by continuing to bid even when fully satisfied or fully exhausted. This is not possible, so the auction does not find a higher splitting point than our algorithm, either. Thus both schemes split the set of users arranged in sequence of non-decreasing $mp(f)$ values at the exact same point.

Now, for a given valid split, there is only one possible price p_s, given that all 100% of the channel time is ultimately auctioned away. All the users in V get allotted their maximum requirements. The remaining CTP is shared proportionately by all the users in W who exhaust their respective maximum bids towards the share. Thus, the sum of the maximum bids of all flows in W divided by the balance CTP remaining after all users in V have received their maximum requirement is the price. For a given price p_s, the CTP allocation for each user f is the same, irrespective of what scheme is used: $min\{c(f, max), B(f)/p_s\}$.

Thus, both the auction at equilibrium as well as our algorithm return the same price and CTP allocation. □

Theorem 2 *When* $\sum_{f \in S} c(f, max) \leq 100\%$*, our algorithm finds the maximum price, no less than the reserve price, for which user satisfaction is 100% and channel utilization is maximum.*

Proof: We prove this theorem also by contradiction. If our algorithm used a larger price than the one currently used, then it would result in all users being allotted CTP lower than they are currently allotted. Thus, their respective levels of satisfaction would only decrease. Channel utilization would also consequently only decrease.

If our algorithm used a price lower than the one currently used, it may violate the reserve price. If the the price is set lower than the current value but no lower than the reserve price, then all flows are eligible to get CTP higher than their current allotment. However, since we are currently using $min\{mp(f)\}$ as the price, all flows can afford more than current price, so they are all *already* getting their maximum requirement, currently. The allotment cannot increase above this. So, the current price results in 100% user satisfaction and maximum possible channel utilization. A lower price does not change allotment and hence does not change user satisfaction and channel utilization.

Thus the current price, as determined by our scheme, is the maximum price, no less than the reserve price, for which user satisfaction is 100% and channel utilization is maximum. □