IQU: Practical Queue-Based User Association Management for WLANs

Amit P. Jardosh, Kimaya Mittal, Krishna N. Ramachandran
Elizabeth M. Belding, and Kevin C. Almeroth
Dept. of Computer Science, University of California
Santa Barbara, CA, USA
{amitj,kimaya,krishna,ebelding,almeroth}@cs.ucsb.edu

ABSTRACT

Flash crowds and high concentrations of users in wireless LANs (WLANs) cause significant interference problems and unsustainable load at access points. This leads to poor connectivity for users, severe performance degradation, and possible WLAN collapse. To validate this claim, we present two case studies of large, heavily loaded operational WLANs. These studies provide significant insight into the degraded performance and collapse of a WLAN during heavy use. To address these problems, we propose IQU, a practical queue-based user association management system for heavily loaded WLANs. IQU grants users fair opportunities to access the WLAN while maintaining high overall throughput, even when the WLAN is heavily loaded. The basic premise of IQU is to control user associations with the WLAN through request queues and work period allocations. We implement a prototype of IQU and evaluate it on a wireless testbed. Our evaluation demonstrates that IQU significantly improves network throughput under heavy load; the tradeoff is that users have to wait for network access. We explore the impact of IQU parameters on system performance, and validate the robustness of IQU under heavy load conditions. Through IQU, WLANs can be utilized efficiently and network collapse prevented.

Categories and Subject Descriptors: C.2.1 [Computer - Communication Networks]: Network Architecture and Design; C.2.3 [Computer - Communication Networks]: Network Operations

General Terms: Algorithms, Design, Experimentation, Management, Measurement, Performance.

Keywords: Association management, Wireless networks, Congestion, IEEE 802.11.

1. INTRODUCTION

WLANs are indispensable for providing Internet access to users at locations such as universities, corporate offices, conferences, airports, and coffee shops. Many of these environments often experience flash crowds, which we define to be a sudden surge in the number of users simultaneously attempting to access the WLAN. When flash crowds occur, WLANs are likely to suffer from destructive interference, excessive channel load, and unsustainable packet processing at access points (APs). These conditions lead to a plethora of problems, such as a deterioration in network throughput, heavy packet loss, intermittent connectivity, overwhelmed APs, and sometimes, a network collapse.

To verify these claims, we present two case studies of operational WLANs that experienced the aforementioned problems. The two WLANs each consisted of over 100 APs and more than 1000 simultaneous users, deployed at recently held Internet Engineering Task Force (IETF) meetings. In the first case study, a high concentration of users in adjacent rooms led to frequent packet collisions and detrimental interference. As a result, users experienced unusably low throughputs. In the second case study, users failed to establish associations with any APs due to either frequent packet collisions or excessive, unsustainable packet processing at the APs. The repeated association attempts made by users resulted in high control packet overhead, compounding the problem. The channels and the APs could not sustain such heavy workloads. The result was sparse or no connectivity for users in the network and eventual network collapse.

The connectivity and usage problems experienced by users at these events are not unique. Similar problems often occur in other scenarios, particularly those that are prone to high user concentrations, such as conferences and conventions. We predict that, as the popularity of WLANs continues to increase, these problems will become even more frequent and widespread and WLANs will have a greater need to handle flash crowds and large concentrations of users.

As a result, an effective solution to manage a large number of users in a WLAN is imperative. The solution should not only avoid network breakdown, but also ensure connectivity and high user throughput. Several approaches to manage heavily loaded WLANs have been presented and evaluated in previous work. These approaches can be classified into four categories: *over-provisioning*, *selective dropping* [8], *load balancing* [6, 9, 17] and *traffic shaping* [11, 19]. Each category has its benefits and can marginally improve performance during a flash crowd. However, they each have drawbacks as well. Over-provisioning is expensive, inefficient and limited by bandwidth availability, while selective dropping may lead to starvation of some users. Balancing load among neighboring APs is of limited help when the total load is high enough to overwhelm all APs in the vicinity. Traffic shaping limits individual throughput in order to accommodate all users, and therefore, when the number of simultaneous users is very high, traffic shaping alone may result in unacceptably poor performance for most users.

WLANs that need to support a large number of users are thus in critical need of a practical and effective system to handle heavy loads and flash crowds. The effectiveness of such a system and the viability of its deployment in an operational WLAN is driven by

MobiCom'06, September 23–26, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-286-0/06/0009 ...$5.00.

some important considerations: (1) fairness to users; (2) connectivity and high throughput for each user; (3) high network throughput; (4) resilience to increased load; (5) low user complexity and overhead; and (6) deployment feasibility. We view these considerations as the requirements for managing heavily loaded WLANs.

In this paper, we propose *IQU*, a practical queue-based user association management system for heavily loaded WLANs. The premise of user association management is to control the frequency and duration of user associations with the network when the number of users trying to access the network is greater than what the network can support. IQU maintains a queue of users requesting network access. Only as many users as can be simultaneously accommodated are granted access to the network. Any remaining users wait for admission in a queue. Admitted users are assigned periods of access, called *work-periods*, within which they can execute any network-related tasks. If the network is under-loaded, the user queue will be empty and users can continue to access the network even after their work-period expires. In a heavy load situation, the expiration of the work-period causes the user to be disassociated from the network and placed back into the queue. A different user from the head of the queue is then admitted into the network. Users with network access are updated with their remaining work period so that they can plan their network-related tasks accordingly[1]. Similarly, users waiting in the queue are given estimates of their wait time for network access and the duration of the work period they will be granted. Thus, unlike the scenarios presented in our case studies, there is no uncertainty about the availability and quality of network access. This information prevents users from making repeated unsuccessful association attempts, thereby both reducing network overhead and considerably improving user experience.

IQU is a simple and powerful system for managing heavily loaded WLANs. It has the ability to address all the requirements of managing heavily loaded WLANs previously listed. IQU changes the basic access model to which today's WLAN users are accustomed. In heavily-loaded WLANs, IQU requires users to wait in a queue for access. Moreover, when access is granted, users must complete their network-related tasks within an alloted work period. This is a significant change from the current model of obtaining immediate access for unlimited durations of time. However, we believe that this change is inevitable in order to maintain good performance in a heavily-loaded WLAN, while meeting the previously listed requirements. Moreover, we believe that the new model is intuitive and easy to understand. Users can be made aware of their assigned wait periods and work periods via a networking utility on the user's device. Note that the new access model may bring about unprecedented alterations in typical user behavior; for instance, users may generate traffic more quickly so that they can complete their tasks in the assigned work period.

To evaluate our system, we built a prototype of IQU and tested it on an 8-node wireless testbed. Our prototype demonstrates that IQU is a practical and viable solution for real-world deployments and can be easily implemented using currently available hardware. Through our experiments, we first establish a baseline for IQU evaluation, and then clearly show the benefits and highlight the trade-offs of IQU. The impact of various IQU parameters on system performance is explored, and appropriate values for the parameters are identified by comparing performance with the established baseline. We then demonstrate the robustness of IQU under challenging network conditions that resemble our case studies, which we emulate in our testbed.

Due to its elegance and effectiveness, user association management has far-ranging implications as a tool for managing limited resources in sophisticated WLANs. It coalesces the benefits of over-provisioning, selective dropping, load balancing and traffic shaping, while avoiding their drawbacks. We believe that our work creates new directions for further research in this area. Different strategies can be explored for managing the user queue. Although we use a simple FIFO queue in this paper, priority-based queues may also be used to support different network access policies. Determination of the optimal number of users that may be permitted to simultaneously access the network and accurate estimation of user wait periods are other parts of this system that have potential for further exploration and research.

The remainder of this paper is organized as follows: Section 2 presents our case studies, which motivate the need for a WLAN load management system. In Section 3, we briefly review previously proposed load management strategies and discuss the essential characteristics of an ideal solution. The design and operation of IQU is described in Section 4, while Section 5 presents our testbed evaluation. We discuss some implementation and deployment issues in Section 6 and conclude in Section 7. Note that the terms user and client are used interchangeably throughout this paper.

2. CASE STUDIES

To determine the effect of flash crowds and high concentrations of users on per-user throughput and control overhead in the network, we use two sets of packet traces in our analysis. The two packet traces were collected during the 62^{nd} and the 64^{th} IETF meetings, in which the WLAN, deployed by event organizers, was the primary source of Internet connectivity.[2] We use these packet traces to demonstrate the effect of a large population of users on network performance. The first set of traces is used to study user and network throughput when a high concentration of users access the WLAN simultaneously, while the second set illustrates the poor connectivity and large control overhead experienced during a flash crowd. The objective of our case studies is to motivate the critical need for a system that can address the detrimental effects of congestion in heavily loaded WLANs in a practical and effective manner.

2.1 Degraded Network Performance

We collected packet traces from the WLAN deployed during the 62^{nd} IETF meeting [14]. The IEEE 802.11b-based WLAN consisted of 152 APs (38 physical APs, each supporting four virtual APs) placed on three adjacent floors of the event location and serviced more than 1000 users. Three laptops using Prism2 chipset cards in the RFMon mode were used to sniff traffic and record packet traces on three orthogonal channels 1, 6, and 11. The traces used in our analysis are from two sessions of the meeting.

We examine the throughput experienced by individual users and by the network as a whole. Per-user throughput and aggregate network throughput are computed based on the instantaneous number of users recorded in the data sets. In other words, to compute these metrics for a particular one-second interval, we consider all users who contribute at least one data frame during that interval. Figure 1 shows the throughput per user versus the number of simultaneous users in the network during the same one-second interval. We observe from the figure that as the number of active users increases from 1 per second to 80 per second, the per-user throughput decreases significantly. For instance, when there are less than

[1] Applications designed for disconnected operation [18] can also leverage this information.

[2] These packet traces are available on the CONAN Project webpage at http://moment.cs.ucsb.edu/conan/.

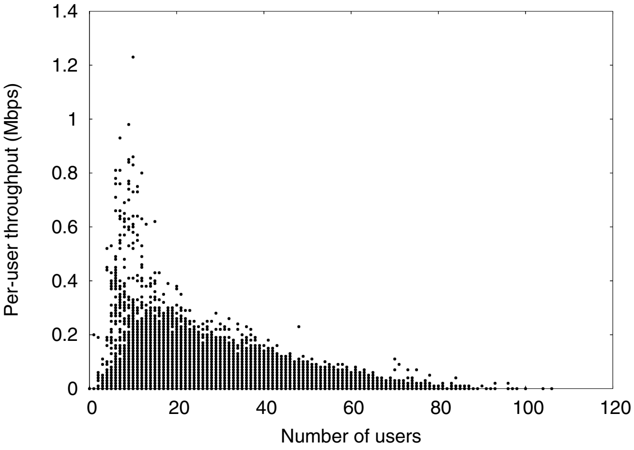


Figure 1: Per-user throughput.

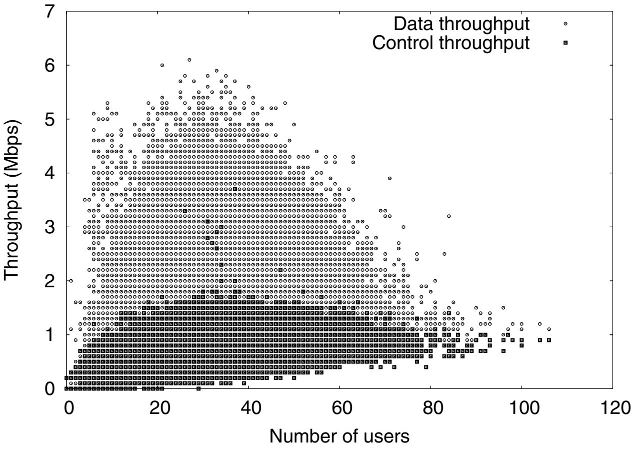


Figure 2: Aggregate data throughput and overhead.

15 simultaneous users, per-user throughputs of up to 1 Mbps are obtained. This value drops to 0.1 Mbps or less with more than 60 simultaneous users. This drop is due to increased contention in the medium leading to large back-offs and frequent packet collisions.

The reduction in per-user throughput with a larger population of simultaneous users is expected to degrade the performance of many user applications. In such situations, users will be forced to use only low bandwidth-consuming applications. Our hypothesis is that if the number of simultaneous users in the network is controlled through user association management, the limited resources can be time-shared among all users in a fair and effective manner. In this way, the users who are connected to the network will be better able to satisfactorily attend to their tasks.

Figure 2 shows the aggregate data throughput and control and management overhead versus the number of simultaneous users in the network during the same second. We observe that when the number of simultaneous users is between 1 and 40, the maximum data throughput obtained is high, approaching the theoretical maximum of 6 Mbps [15]. The control and management overhead, indicated by the denser area in the figure, increases significantly as the number of simultaneous users increases from 1 to 40. However, as the number of simultaneous users further increases from 40 to 80, the data throughput decreases, while the control and management overhead does not decrease as significantly.

We make two observations from this graph. First, an increase in the number of users may cause the network to be inefficiently utilized. Second, as the number of users increases from 40 to 80, there is a decrease in data throughput without any proportional increase in control overhead, which leads us to believe that the throughput decrease is due to higher MAC layer contention. Our hypothesis therefore is that to maintain network utilization at a desirable level and ensure efficient use of network resources, the number of simultaneous users in the network should be limited. This limit can be determined by intelligent and dynamic user association management systems deployed in the network.

Both the graphs make a convincing case that there is a need for a system to manage heavy loads in WLANs. In the next section, we show how the absence of such a system can result in network collapse during a flash crowd.

2.2 Network Collapse

Data was collected during the 64^{th} IETF meeting using the same sniffing technique as used for the previous meeting. The IEEE 802.11b WLAN installed during this event consisted of over 100 APs operating on channels 1, 6, and 11, and serviced over 1000 simultaneous users. Similar to the previously described data collection exercise, three laptops using Prism2 chipset cards in RF-Mon mode were used to capture packet traces from the operational WLAN. The laptops gathering the traces were placed at the center of one of the busiest meeting rooms. At this location, they could record traffic from a large percentage of users, though not all. The laptops started recording packet traces at 7:55 AM. By 9:15 AM, a large number of users assembled in the meeting rooms for the first event of the day. The WLAN could not sustain the heavy control, management, and data packet processing required by this flash crowd. As a result, users obtained only intermittent connectivity. This intermittent connectivity led to even more control and management traffic since user devices repeatedly tried to probe and associate with various APs. The resulting increase in management and control traffic further degraded connectivity in the WLAN. Within an hour, the WLAN collapsed, leaving all users without Internet connectivity.

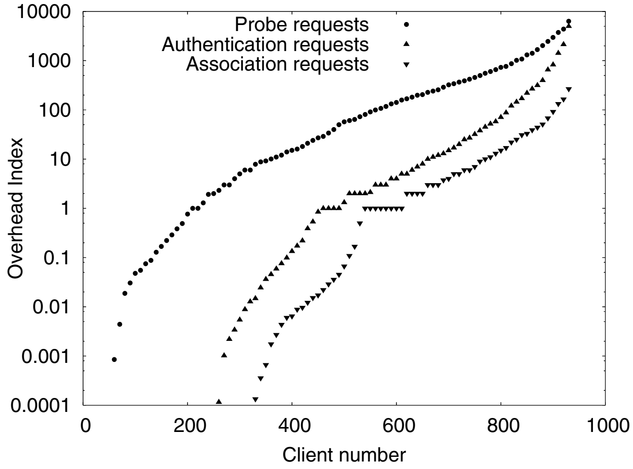


Figure 3: Overhead indices.

In order to quantify the proportion of control and management traffic versus the transmission of useful data traffic in the network, we introduce a metric called *overhead index*. The overhead index is defined as the ratio of the number of control/management frames transmitted per data byte sent by a user over a given period of time. We compute the overhead index with respect to probe requests, authentication requests, and association requests for all users recorded in the data sets. Figure 3 shows the overhead indices computed for the more than 1000 users recorded by the sniffers. For each index, clients on the x-axis are ordered such that the value of the index increases. Index values lower than 0.0001 are not represented in the graph. Note that the vast difference in the index values for different users is due to temporal and spatial heterogeneity, diverse traffic patterns, and differences in wireless device behavior. The figure indicates that the overhead index for a majority of users is greater than one. This means that the users had to transmit an average of more than one probe, authentication, or association request in order to transmit a *single byte of data*. In a healthy network, control and management frames are only occasionally transmitted, hence, the overhead index values are expected to be well below one. These results indicate that, in the WLAN under study, the majority of time and device power were utilized to repeatedly probe for APs and then authenticate and associate with an AP.

Our hypothesis is that if the WLAN employed a user association management system, the severe effects of the flash crowd could have been successfully curtailed. The users admitted into the WLAN could have used the network for allotted time intervals, and thus the WLAN could have served all the users while avoiding collapse. We believe that, during a flash crowd, a user association management system would significantly improve the usability of the WLAN, avoid unnecessary user device power consumption, and allow each user the opportunity to accomplish useful tasks when connectivity is granted. In the remainder of this paper, our goal is to verify this assertion.

3. WLAN LOAD MANAGEMENT

In Section 1, we briefly described four categories of solutions that have been previously proposed to manage high load in WLANs. We also listed the requirements of an ideal solution. In this section, we discuss these topics in greater detail.

3.1 User Workload Management Strategies

In this section, we discuss four different solution categories that have been previously proposed for managing heavy workloads in WLANs. Similar solutions have also been researched in other areas, such as Internet services [10, 20] and supercomputing resource management [4, 16]. We explain why none of the four solution classes can successfully address the WLAN load problem.

Over-provisioning: Over-provisioning uses additional APs to accommodate the extra load generated in flash crowd situations. However, the solution is limited by bandwidth and spectrum availability in the WLAN. If the demand on the WLAN increases beyond this limit, over-provisioning ceases to provide better performance to the users. Moreover, over-provisioning is not cost-effective, particularly because the APs are likely to be largely under-utilized during normal network operation.

Selective dropping: With selective dropping, service is denied to some users when the network load increases beyond a threshold [8]. This solution is unfair to the users who remain indefinitely starved for network resources while other users consume more than their fair share. Furthermore, the starved users are likely to make repeated unsuccessful requests for network resources, leading to additional control overhead, wasted network bandwidth, and a poor user experience.

Load balancing: In this case, users are distributed among APs in the network based on various parameters such as user workload and the experienced load at accessible APs [6, 9, 17]. This solution is of limited help when the total load becomes high enough to overwhelm all APs in the system.

Traffic shaping: Traffic shaping limits individual throughput in order to accommodate all users in the network [11, 19]. Traffic shaping may be beneficial in a WLAN since it prevents users from consuming an inordinately large portion of the bandwidth at the expense of other users. However, the users whose throughput is limited may be prevented from accomplishing their desired tasks. Also, when the total number of users becomes very large, trying to accommodate all users through traffic shaping can result in unacceptably low throughput for most users.

3.2 Heavy Load Management Requirements

Our observations from the previous section bring us to the list of requirements for a practical and effective system for managing heavy user loads in WLANs. These requirements are as follows:

Fairness to users: Fairness is a much discussed term that is often defined differently. For the purposes of this paper, we consider long-term fairness, i.e. where every user has an equal opportunity to access the network over the long term (in the order of minutes).

Good connectivity and throughput: When users are allowed to access the network, they should be granted sufficient resources to accomplish useful tasks. Clearly, the definition of "sufficient resources" can vary from user to user, and all user requests may be impossible to satisfy simultaneously. However, the system should accommodate users' requirements as best as possible.

High overall network throughput: Efficient systems for managing heavily loaded WLANs should ensure that the utilization of network resources, such as bandwidth, is maximized. For instance, scenarios that cause users to generate excessive amounts of control traffic should be avoided.

Resilience to increased load: A load management system should be capable of handling a large number of users and events such as flash crowds, frequent requests, and large user turnover rates.

Low user complexity: Several previously proposed load management approaches place heavy demands on users. For instance, users may be asked to estimate local network interference [9, 17], to forecast their bandwidth requirement prior to admission [6, 8], or to change their geographic location in order to use a different AP [6]. Users may also be required to install complex applications, middleware, or driver code. These requirements significantly hinder the deployment of these techniques. For a system to be deployable in the real world, user complexity must be low.

Deployment feasibility: A system should be simple to implement, deploy, and configure so that it can be easily used in an operational WLAN. Further, the requirement for expensive equipment should be avoided.

In the next section, we present IQU, our practical queue-based user association management system that satisfies these requirements.

4. IQU: DESIGN AND OPERATION

IQU is a queue-based user association management system for WLANs. The design of IQU satisfies all the requirements listed in Section 3.2. In this section, we first describe the operation of IQU in a single-AP WLAN. We explain the algorithms and parameters used, and highlight the design tradeoffs. We follow with details on the implementation of IQU. Finally, we describe IQU operation in an enterprise WLAN involving multiple APs.

4.1 Operation Description

Central to the design and operation of IQU is the user queue implemented at the AP. As users enter the WLAN, they request network access from the AP and are placed in the AP's user queue. Depending on the observed network utilization levels, the AP computes a limit for the number of users that may simultaneously be granted access to the network. If the network is not heavily loaded, users are immediately granted network access and the queue remains empty. In a heavy load situation, only a limited number of users from the head of the queue are granted access. The remaining users wait for admission into the network. Admitted users are assigned a limited period of access, called a *work-period*, during which they may associate with the AP. When a user's work-period expires and other users are waiting in the queue to obtain access, the AP places the user back in the queue, and a different user from the head of the queue is allowed to associate with the AP. Connected users are informed about their allotted work-periods so that they can plan their network-related tasks accordingly. Users waiting in the queue are given an estimate of their wait time. Each user is guaranteed that network access will be granted when the estimated wait time elapses, and that access will not be pre-empted before the assigned work-period expires. Note that, if the network is not heavily loaded and the user queue is empty, the user can continue to access the network even after the work-period expires.

In the remainder of this section, we provide a more detailed description of the various components of IQU operation and discuss the available design choices.

Queue assignment and management: The user queue maintained by the AP is the central component of IQU. The network access policy dictates how the user queue is organized. If all users have an equal right to the medium, a FIFO queue is used, which grants access to users in the order of their arrival. A priority queue could be used to implement other access policies, such as prioritized access to high-paying users. In our implementation, we use a FIFO user queue.

Work-periods: A work-period, T_{work}, is the minimum time for which a user is granted network access. This period should be of sufficient duration to allow the user to accomplish common network-related tasks. This value may vary in different WLAN deployments. The work-period for users of a particular deployment is set by the network administrator based on the typical usage profile for that deployment. For example, conference users typically check email or browse Internet websites [7]. The administrator may hence configure the work-period for a conference WLAN to a duration appropriate for accomplishing these tasks.

When the network is heavily loaded, shorter work-periods allow users to complete fewer tasks; however, users also spend proportionally shorter intervals of time waiting in the queue. On the other hand, longer work-periods allow users to remain associated for longer durations, but cause longer wait times as well. Note that, although the work-period grants users an equal amount of time to access the WLAN during heavy loads, the throughput obtained by each user depends on the data rate of the user device and may therefore vary for different users.

Determination of the number of permissible users: In IQU, the AP computes a limit for the number of permissible users, N_{perm}. This value is the maximum number of users that can simultaneously access the network such that congestion and packet loss are avoided and high network throughput is maintained. In a previous publication, we demonstrated that channel utilization[3] correlates well with congestion and throughput [14]. We therefore use channel utilization at the AP as a metric to anticipate the onset of congestion. Congestion can also be estimated in other ways, for example, by monitoring the delay between the first attempted transmission of a packet and the receipt of the corresponding acknowledgment.

Channel utilization is low when the network is under-utilized. When channel utilization increases, traffic in the network has increased. If more users are allowed to access the network when the traffic level is high, the network may enter a congested state. Therefore, in order to prevent congestion, IQU monitors channel utilization and prevents additional users from associating with the network when channel utilization exceeds a threshold. This is accomplished by reducing the number of permissible users. Although the number of permissible users decreases, users who are already associated with the AP continue to access the network, at least until their assigned work-periods expire. When channel utilization decreases, IQU increments the number of permissible users, thereby allowing waiting users to access the network.

Thus, IQU dynamically adjusts the number of permissible users by monitoring channel utilization. A lower and an upper utilization threshold are maintained. We denote the observed utilization as U_{obs}, and the lower and upper utilization thresholds as U_{lower} and U_{upper}, respectively. Then,

if ($U_{obs} < U_{lower}$), then $\{N_{perm} = N_{perm} + 1\}$, while
if ($U_{obs} > U_{upper}$), then $\{N_{perm} = N_{perm} - 1\}$.

In other words, if the observed utilization value lies below the lower threshold, the number of permissible users is increased and more users are granted network access. On the other hand, if the observed utilization exceeds the upper threshold, the limit is decreased and no further users are granted access until the network traffic decreases, some users disassociate from the network, or the wait time of a user in the queue has expired. The observed utilization is compared to the utilization thresholds at every second and the number of permissible users is adjusted as necessary. The number of permissible users is never increased beyond an upper limit $N_{perm-max}$; this limit is set by the network administrator based on the anticipated maximum number of users that may associate with an AP.

Determination of the utilization thresholds creates an important tradeoff. If the thresholds are set too high, the system allows more users into the network. This allows more users to obtain access and reduces user wait times; however, the likelihood of network congestion increases. Similarly, if the thresholds are set too low, fewer users are allowed to simultaneously access the network. Although this limit reduces the likelihood of congestion, it may result in an under-utilized network, wasted network resources, and higher user wait times. The appropriate channel utilization thresholds for a particular WLAN deployment depend on the traffic generated by users in that deployment.

Opportunistic user addition: The above mechanism for determining the number of permissible users increments the number of permissible users only until the observed utilization crosses the lower threshold. However, it is desirable to maintain the observed utilization as close to the upper utilization threshold as possible in order to utilize the network resources more efficiently. The impact of admitting a user on the observed utilization depends on the traffic generated by the user. If the traffic generated by the user is low, the user's admission may not impact the observed utilization significantly. Thus, even when the observed utilization lies between the upper and lower thresholds, it may be possible to accommodate more users in the network. To address such situations, IQU includes a mechanism to opportunistically increase the number of permissible users. If the observed channel utilization is found to consistently lie within the specified thresholds for a hold

[3] Channel utilization at a node is computed as the fraction of time consumed by the data, management and control frame transmissions that are received by that node, together with their corresponding delay components, such as DIFS and SIFS.

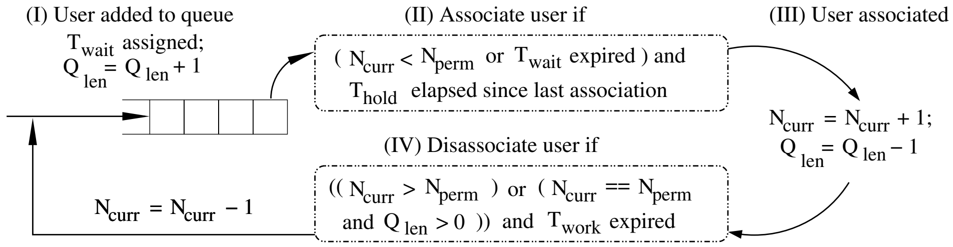

(I) User added to queue
T_{wait} assigned;
$Q_{len} = Q_{len} + 1$
(II) Associate user if
($N_{curr} < N_{perm}$ or T_{wait} expired) and
T_{hold} elapsed since last association
(III) User associated
$N_{curr} = N_{curr} + 1$;
$Q_{len} = Q_{len} - 1$
(IV) Disassociate user if
(($N_{curr} > N_{perm}$) or ($N_{curr} == N_{perm}$
and $Q_{len} > 0$)) and T_{work} expired
$N_{curr} = N_{curr} - 1$

Figure 4: User association management in IQU.

time, T_{hold}, and the number of currently associated users N_{curr} is equal to the limit N_{perm}, IQU increases the number of permissible users by one to allow the admission of one more user from its queue into the network:

if (($U_{lower} < U_{obs} < U_{upper}$ for duration T_{hold}) and
($N_{curr} == N_{perm}$)), then $\{N_{perm} = N_{perm} + 1\}$.

The resulting increase in channel utilization depends on the traffic generated by the newly admitted user. If the new admission causes channel utilization to increase beyond the target range, the number of permissible users is immediately decremented as per the mechanism described previously. On the other hand, if channel utilization continues to remain within the target range, the network is able to service more users. The duration of the hold time, T_{hold}, should be sufficiently long to allow the measured utilization to adapt to the additional traffic generated by the newly admitted user.

User association management: Depending on the number of permissible users, N_{perm}, and the number of currently associated users, N_{curr}, the AP may perform one or both of the following functions: (a) allow users waiting at the head of the queue to associate with the network and (b) pre-empt active users whose work-periods have expired and place them back in the queue. A user's network association cannot be pre-empted if the assigned work-period has not yet elapsed. In other words, we treat the work-period as a commitment to the user. This commitment assures the user that his network session will not be unexpectedly pre-empted and allows the user to confidently plan and execute network-related tasks. The downside of this choice is that congestion caused by the admission of a user, or a sudden rise in traffic from currently associated users, cannot be mitigated until the expiration of some users' work-periods. The likelihood of congestion can be reduced by appropriately configuring the utilization thresholds.

An increase in the number of associated users is likely to change the traffic level in the network, leading to a change in the observed utilization. In order to allow the changed utilization value to stabilize, IQU employs a hold time, T_{hold}, between subsequent user admissions. This mechanism prevents the system from adding too many users too quickly. Note that the same parameter T_{hold} is also used to regulate the opportunistic addition to users, as described previously. We investigate optimal values of T_{hold} in Section 5.5.

Wait time estimation: Users awaiting admission into the network are given an estimate of the maximum time they will have to wait. This estimate, T_{wait}, is a commitment and is strictly adhered to by IQU. In other words, when the estimated maximum wait time for a user elapses, the user is immediately granted access to the network. We believe that commitment to the estimated maximum wait time is necessary in order to make IQU acceptable to and trusted by users and to enable them to plan their network-related tasks. Users are less likely to be willing to wait for network access if the estimated wait period is unknown or subject to unexpected elongation.

The wait time for each user is estimated when the user is added to the queue and is based on the queue length, work period, and the number of permissible users at that time. Subsequent changes in traffic conditions may reduce the number of users that the network can simultaneously support. As a result, when the user's wait period expires, the network may not be able to adequately accommodate additional traffic, and the admission of the user may increase congestion in the network. This is a drawback of our policy of strict adherence to wait time estimates. In order to reduce the probability of such an event, wait times must be estimated accurately.

Accurate estimation of wait time is a non-trivial task because network conditions may change quickly, thereby invalidating a previous estimate. The challenge lies in the requirement to estimate a future network condition at a current instant. In our current implementation, we compute the wait time estimate for a user in the following manner. A user waiting in the user queue obtains network access only after the previous user in the queue has been granted access. The time interval between these two events can vary from zero (when both users are granted access simultaneously) to T_{work} (when the previous user completes his network access before the next user is allowed to associate). Therefore, on an average, the interval between the admissions of the two users is likely to be $T_{work}/2$. With this intuition, we estimate the time at which a user is likely to be admitted into the network by incrementing the estimated admission time of the previous user in the queue by $T_{work}/2$. The wait time is then derived from the estimated admission time. Obviously, this estimate of a wait time can be improved by using better techniques, particularly if the traffic patterns of users and the resulting changes in the number of permissible users can be modeled accurately. However, without that information, we believe that ours is a sufficiently accurate estimate. If traffic in the network decreases and additional users can be supported, network access may also be granted to a user before the estimated wait time expires.

Summary: Figure 4 summarizes the IQU user association management procedure. As shown in Step (I) of the figure, a user entering the WLAN is placed in the user queue and assigned a wait time estimate, T_{wait}. The user at the head of the queue is granted access to the network only if the current number of associated users is less than the permissible limit ($N_{curr} < N_{perm}$) or the user's wait time, T_{wait}, has elapsed. Additionally, the time elapsed since the last user admission should be greater than or equal to the hold time, T_{hold}. This requirement is indicated in Step (II) of the figure. If these conditions are satisfied, the user is admitted into the network, as Step (III) shows. The conditions for disassociating the user are indicated in Step (IV). Once T_{work} expires, the user is disassociated and placed back in the queue only if either the current number of associated users exceeds the limit on permissible users, $N_{curr} > N_{perm}$, or if the network has reached its capacity, $N_{curr} == N_{perm}$, and there are more users waiting in the queue, $Q_{len} > 0$.

4.2 Implementation Details

We now provide details regarding the implementation of IQU in an IEEE 802.11-based WLAN. All of the IQU functionality described in Section 4.1 is implemented at the APs, which is accomplished by modifying the AP device driver.

Since most of the processing required for IQU occurs at the AP, user devices that wish to connect to an IQU-enabled WLAN need minimal modifications to the device driver and no post-installation updates. The main requirement is that the user device must recognize and respond to IQU messages. The device must know whether it currently has network access or is waiting in the user queue, and it must know the duration of the corresponding work period or wait period. The device must also communicate this information to the user. This step can be accomplished by a simple networking utility in the user device. When the user is in the queue, the user device can choose to enter sleep mode for a portion of its wait time, thereby conserving energy.

The IQU message exchange is accomplished by using existing IEEE 802.11 messages as follows. When the user device wishes to request network access, it sends an unmodified IEEE 802.11 association request to the AP. On receiving the association request, the AP determines whether the user can be granted access immediately or must be placed in the user queue. In the former case, the AP sends an IEEE 802.11 association reply message to the user device. The *Status* field in this message is used to indicate the work period assigned to the user. If the user cannot immediately be granted network access and must wait in the queue, the AP replies to the association request with an IEEE 802.11 disassociation message. The *Reason* field in this message is used to specify the wait time estimate. IQU can thus be implemented by using existing IEEE 802.11 messages and no special packets are necessary.

In our implementation, a user waiting in the queue periodically sends association requests to the AP in order to check whether network access can be obtained. Alternatively, the AP may send a probe to the client when it is ready to grant access to the client. The former strategy is preferable if a client wishes to shut down its network interface to conserve energy during its waiting time.

4.3 Operation in an Enterprise WLAN

The basic operation of IQU for a single AP network was described in Section 4.1. In this section, we explain how IQU operates in an enterprise WLAN involving multiple APs.

Enterprise WLANs consist of multiple APs that are all typically connected to a centralized controller through the wired backbone. This is known as a Distributed Access Point (DAP) architecture, and is commonly supported by leading wireless networking product vendors [1, 2, 5]. When IQU is deployed in such a network, the IQU-specific computations are all moved to the centralized controller. When the user first enters the network and transmits an association request, the centralized controller assigns the user to a particular AP. This assignment may depend on several factors such as the load, user queue lengths, and wait time estimates at various APs and the SNR values of the packets sent by the user as perceived by the AP. The procedure for determining the optimal AP is beyond the scope of this work. Previously proposed solutions for assigning users to APs [6, 9, 17] can be leveraged. Once the user is assigned to an AP, he remains in that AP's user queue until network access can be granted.

IQU can support user mobility between different APs under certain constraints. Suppose a user, U, moves from $AP1$ to $AP2$ within the WLAN. If the user was awaiting network access in the user queue at $AP1$, he is now placed in the user queue at $AP2$. The queue position at which U should be inserted is calculated by the centralized controller by considering the time previously spent in the queue at $AP1$. If the user was currently accessing the network at $AP1$, $AP2$ may or may not be able to immediately grant network access to the user depending on its own load conditions and user queue state. Thus, roaming users might not always obtain seamless session hand-off across APs. However, we argue that this is an acceptable tradeoff for the benefits that IQU provides.

5. EVALUATION

We now proceed to our evaluation of IQU. The objectives of our evaluation are to (1) study the behavior of IQU, demonstrate its effectiveness and examine the trade-offs involved; (2) explore the impact of IQU parameter settings on the trade-offs; and (3) demonstrate the robustness of the system under challenging load conditions such as flash crowds and constant heavy load.

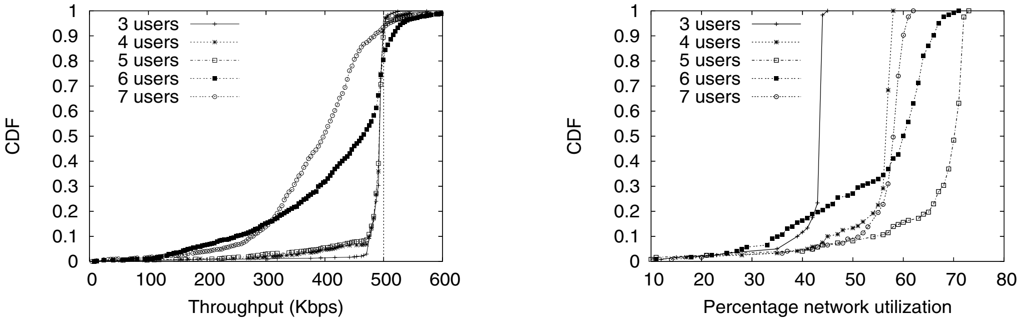
Simulation methods can be used to evaluate IQU. However, our goal is to convincingly demonstrate the practicality of an association management system such as IQU, as well as IQU's benefits in a real system. Therefore, we implement a prototype of IQU and test it on an 8-node wireless testbed. The objective of our evaluation is to study the impact of IQU on the performance of a heavily-loaded network. The 8 nodes in our testbed are sufficient to generate the necessary network load and allow us to understand the resulting trends and variations. Clearly, a testbed consisting of a greater number of nodes would facilitate more realistic network usage and traffic scenarios and would also better capture events such as hidden terminals. However, our testbed effectively allows us to create challenging load conditions such as a flash crowd and constant heavy load, and enables us to accomplish our current evaluation objectives. A more thorough evaluation of the impact of various WLAN environment parameters and traffic patterns on IQU is interesting future work.

In this section, we first describe our experiment methodology and define evaluation metrics. We then evaluate the performance of our testbed under increasing load conditions in the absence of IQU; this helps us establish a baseline for our evaluation. Next, we demonstrate the performance of IQU in a simple traffic scenario, and identify the trade-offs of this system by comparing its performance to our established baseline. We examine the impact of the hold time and utilization thresholds on system behavior, and identify appropriate values for these parameters. Finally we evaluate the performance and robustness of IQU under a challenging traffic scenario that is representative of the flash crowd observed in our case study in Section 2.

5.1 Experiment Methodology

Our testbed consists of eight laptops (three IBM Thinkpads and five Toshiba Satellite laptops) that run Linux and are equipped with Atheros chipset IEEE 802.11a/b/g wireless network cards. One laptop is configured to act as a wireless AP, while the remaining laptops act as WLAN clients. The AP and client laptops are placed within direct transmission range of each other.

The wireless network cards are managed by the MADWiFi driver, which is a Linux kernel device driver module for Atheros-based WLAN devices [3]. We implement the IQU prototype by appropriately modifying this driver. For our experiments, we configure the wireless cards to use the IEEE 802.11b protocol and fix the data rate at 11 Mbps. We disable the RTS/CTS collision avoidance mechanism and MAC layer retransmissions. Data rate adaptation, collision avoidance, and retransmissions can all affect the obtained throughput, especially when the medium is highly congested. In order to isolate IQU's impact on throughput, we disable these mechanisms in our experiments.



(a) Average user throughput without IQU.

(b) Observed utilization without IQU.

Figure 5: Network performance baseline.

We use UDP traffic for our experiments. As opposed to TCP, UDP allows us to control the traffic rate and generate the desired load in the testbed. On the other hand, TCP's congestion control and backoff mechanism prevents us from maintaining a comparable level of network load. Each client runs a UDP traffic generation program that bidirectionally exchanges UDP data with a peer program located at the AP. The intention is to create a testbed environment with both incoming traffic from the AP to the clients and outgoing traffic from the client to the AP. The bidirectional UDP traffic provides us with a mechanism to create high network utilization levels and congestion in the testbed.

We configure the traffic generation rate for each client to a specific value that depends on the test scenario. We also specify the duration of time for which the traffic is generated. When a client's network access is pre-empted, the traffic generation program pauses. The client resumes transmission when a new work-period is granted. For instance, if a client is configured to generate traffic for 10 minutes, it does so for 10 minutes of actual access time, not including the time spent waiting in queue.

5.2 Evaluation Metrics

We select evaluation metrics that help us clearly identify the benefits and tradeoffs of IQU. The following metrics are evaluated:

User throughput: User throughput is defined as the amount of data received per second by the peer traffic generation programs at the AP and the client. Clearly a high value for this metric is desirable. We examine both aggregate network throughput and individual user throughput in our experiments. The former metric indicates the overall performance of the system, while a comparison of the latter shows user fairness. Note that these are two requirements for heavy load management that we listed in Section 3.2.

User wait time: Making users wait for network access is the price paid by IQU to obtain improved performance in heavy load conditions. This metric quantifies this price. User wait time is defined as the interval of time that a user has to wait in the user queue before being allowed to associate with the AP. Note that during an experiment, depending on the network conditions and the load generated by the other clients, a client may have to wait more than once in order to complete its assigned tasks. We examine both average and individual wait times in order to understand the overall performance and fairness to users. The lower the wait time, the better the user experience and the more efficient the system.

Number of associated users: This metric is computed for every one second interval, and indicates the instantaneous number of users that are associated with the AP in that interval. In the case studies presented in Section 2, we observed that as the instantaneous number of users associated with the WLAN increased, the per-user throughput decreased significantly. The objective of IQU is to alleviate these problems by limiting the number of users associated with an AP such that the associated users can be offered good connectivity and throughput. We use this metric to evaluate how well IQU meets this objective.

Packet loss and the access time offered to users are other potentially interesting metrics that can be used to evaluate network performance. However, since the traffic generated by each user in our experiments is known, the measured throughput also indicates packet loss. Also, since the clients in our experiments continue to wait until they can complete the assigned network activity, each client eventually obtains the access time it needs. We therefore do not examine these metrics in our evaluation.

5.3 Network Performance Baseline

Before evaluating the behavior of IQU, we first examine the performance of the network under increasing levels of load without IQU enabled. This examination allows us to identify the optimum level of load for this environment, thereby creating a baseline for the evaluation of IQU. The understanding gained from this exercise is later used to appropriately select IQU parameters.

For this evaluation we conduct five different experiments, each with between three and seven clients simultaneously associated with the AP. Each client is configured to send and receive UDP data packets at a uniform rate of 500 Kbps, both to and from the AP, for a period of 10 minutes. In each experiment, we record the per-second network utilization level and user throughput during the entire run of the experiment. This exercise enables us to observe network performance under increasingly higher levels of load.

Figure 5(a) shows the CDF of the average user throughput (averaged over both directions) for these tests. Each data point represents the fraction of time (shown on the y-axis) for which the system experienced equal or lower throughput than the corresponding value on the x-axis. The data points are averaged over three runs of the same configuration. When there are between three and five users in the network, the average user throughput is close to the sending rate of 500 Kbps over 90% of time. This result indicates that packet loss is low, which, in turn, implies that the network does not experience a high rate of packet collisions and congestion at these levels of load. Note that the average user throughput exceeds 500 Kbps for a small fraction of time. This can be attributed to the network buffering of packets at the clients. Figure 5(a) shows that the average user throughput decreases significantly when the number of simultaneously associated clients is six or greater. In other words, at this load, the network experiences frequent packet collisions and congestion, resulting in significant packet loss.

Figure 5(b) shows the CDF of the utilization for the same tests. As seen in the figure, when there are three clients in the network, the utilization is relatively constant at approximately 40%. The

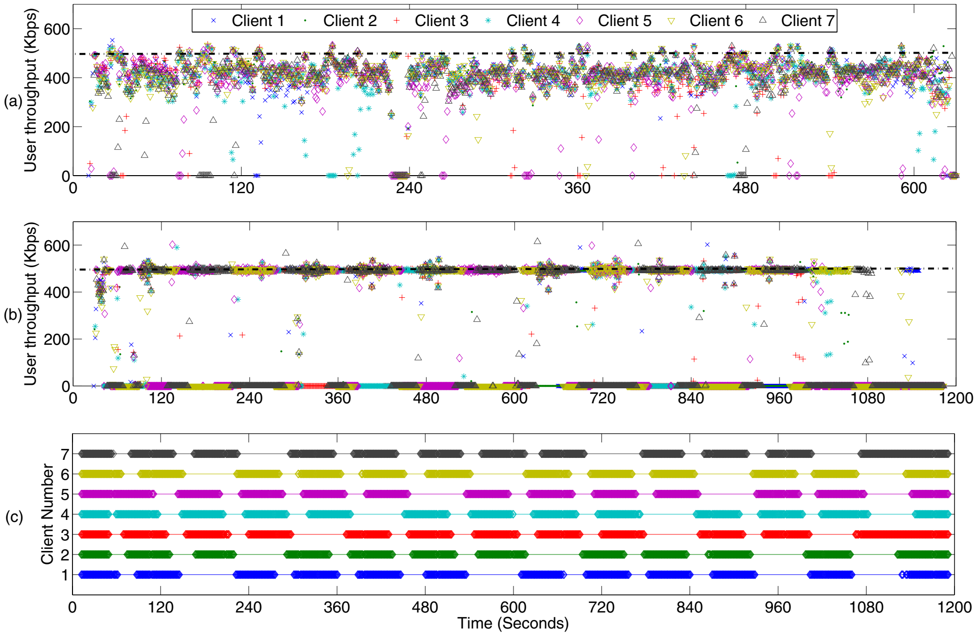


Figure 6: (a) Client throughput without IQU. (b) Client throughput with IQU. (c) User access and wait times.

maximum utilization increases to almost 70% with five clients in the network. However, when the number of clients exceeds five, the utilization decreses significantly. This decrease indicates that the AP receives fewer packets, although a greater number of packets are sent. These observations imply that the network experiences congestion with six or more clients under this traffic pattern.

The results obtained from these experiments indicate that, for the selected traffic pattern, the number of simultaneously associated clients should be five to avoid congestion and maintain optimum network throughput. With fewer than five clients, the network remains under-utilized, while with more than five clients, the network experiences congestion and suffers high packet loss. Based on the results of this experiment, we determine that the goal for IQU is to maintain the number of simultaneously associated clients close to five for this traffic pattern. This observation guides us in selection of the appropriate values for IQU parameters.

5.4 Demonstration of IQU Performance

In this section we discuss the operation of IQU in a simple traffic scenario. The purpose is to demonstrate the operation of IQU under constant heavy load and compare it with the performance of the network with IQU disabled.

We configure each of the seven clients to send and receive UDP data packets at a uniform rate of 500 Kbps in each direction, for a period of 10 minutes each. This high data rate results in saturation of the AP and network congestion. All clients enter the network simultaneously and send data for the same time duration. The utilization thresholds, U_{lower} and U_{upper}, are set to 30% and 55%, respectively, while the hold time T_{hold} is set to 5 seconds. We explore other values for these parameters in later sections. The work-period, i.e. the minimum duration for which a client is granted access, is one minute.

Figures 6(a) and 6(b) show time-series plots of the throughput experienced by each client when IQU was disabled and enabled, respectively. We observe that there is significant loss and variation in throughput when IQU is disabled. In comparison, with IQU enabled, whenever a client is granted network access, it experiences a throughput close to the 500 Kbps sending rate with only a few variations. This result demonstrates that IQU is able to successfully prevent congestion and achieve high user throughput.

Table 1: Aggregate access and wait times (in seconds).

Client Number	1	2	3	4	5	6	7
Access times	716	721	724	722	730	729	722
Wait times	324	350	323	339	335	337	329

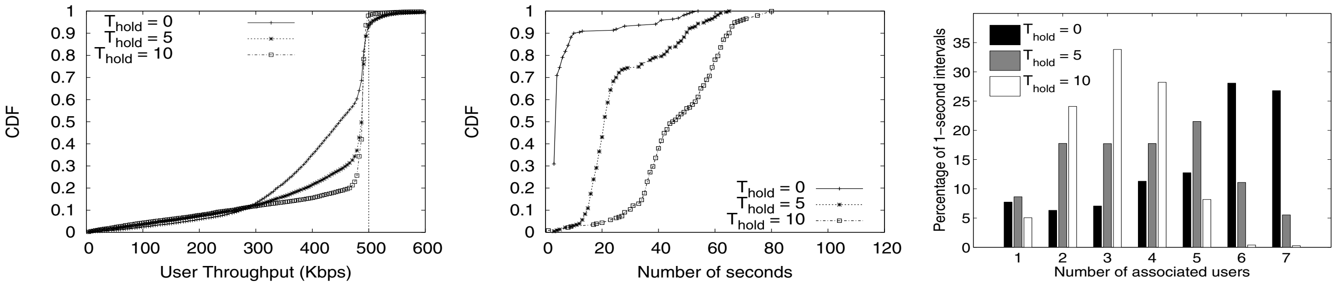
There is an important tradeoff to the increase in throughput shown in Figure 6(b). Since clients do not continuously have network access and take turns waiting in the queue, the total time required to complete the assigned network activity is higher with IQU enabled. Note that the scale on the x-axis is different in Figures 6(a) and 6(b), indicating an increase in the duration of the experiment when IQU is enabled. However, we argue that this tradeoff is acceptable to maintain high network utilization under heavy load.

Figure 6(c) shows the individual client access and wait times throughout the duration of the experiment. For each client, the solid blocks indicate access times while the thin lines indicate wait times. We see that the access periods of clients are staggered, indicating that clients take turns in accessing the network. Note that the work period offered to the clients is 60 seconds for these experiments. The aggregate access time and wait time for each client is shown in Table 1. We see that these values are fairly evenly distributed, demonstrating that the system is fair and satisfies an important requirement of load management as discussed in Section 3.2.

These experiments demonstrate the behavior of IQU and its ability to regulate network access and maintain high throughput. We now evaluate the impact of IQU parameters on system behavior.

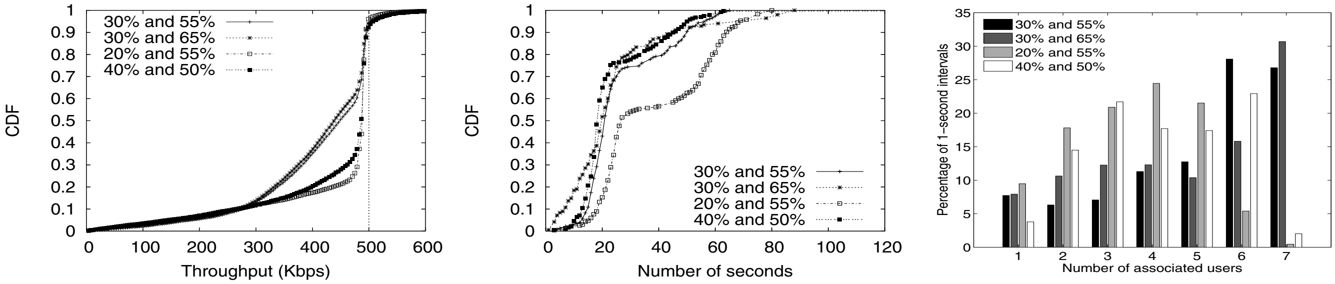
5.5 Impact of Hold Time

As explained in Section 4.1, the hold time, T_{hold}, is the minimum time between consecutive user admissions. The purpose of the hold time is to allow utilization measurements to stabilize based on the traffic generated by a newly admitted user, before more users



(a) Hold time and user throughput. (b) Hold time and user wait time. (c) Hold time and users.

Figure 7: Impact of hold time on user throughput, user wait times, and number of associated users.



(a) Utilization thresholds and user throughput. (b) Utilization thresholds and user wait time. (c) Utilization thresholds and users.

Figure 8: Impact of utilization thresholds on user throughput, user wait times, and number of associated users. The percentages for each plot indicate the U_{lower} and U_{upper} thresholds, respectively.

are admitted into the network. The purpose of the next set of experiments is to validate the need for this parameter and to examine its effect on the performance metrics. The traffic pattern for these experiments is the same as that in Section 5.4. We vary the hold time between 0, 5, and 10 seconds in the tests. Figure 7 shows the results of these experiments. Each data point is averaged over three runs.

In Figure 7(a), we plot a CDF of the average throughput over the duration of the experiment, similar to Figure 5(a). We observe that when T_{hold} is zero, the average user throughput is significantly lower than when T_{hold} is 5 or 10. This loss can be attributed to the high frequency at which clients are admitted into the network, resulting in congestion and packet loss. Note that once a client is admitted into the network, it is not pre-empted until the work-period elapses. An increase in hold time to 5 or 10 seconds results in more conservative admission of users. Therefore, even if the AP computes an N_{perm} greater than N_{curr}, it waits 5 or 10 seconds before admitting a client from the head of the user queue. This conservative behavior results in fewer associated users on average, thereby increasing average user throughput.

The downside of larger hold times is that there is a corresponding increase in user wait time, as depicted in Figure 7(b). Figure 7(b) shows a CDF of user wait times. Each data point represents the fraction of instances (shown on the y-axis) that the wait time of a client was less than or equal to the value on the x-axis. As seen in the figure, when the hold time is zero, users have to wait less than 10 seconds before obtaining network access in 90% of the cases. As hold time increases, users wait for longer periods. This result demonstrates that IQU should avoid very large hold times.

Figure 7(c) shows the percentage of time for which IQU granted simultaneous access to the number of users indicated on the x-axis. When the hold time is zero, the system often allows six or seven clients simultaneous admission. We have already seen that congestion occurs when the number of associated clients exceeds five. This result validates the need for a hold time parameter. As hold time increases, the percentage of time for which the system allows six or more users in the system decreases. In other words, congestion is reduced as hold time increases.

The results of this experiment highlight the impact of hold time on IQU. As hold time increases, the likelihood of congestion decreases and throughput improves, but the user wait times correspondingly increase. Among the values tested, a hold time of 5 seconds offers the best tradeoff between throughput and wait time for our particular usage scenario. We therefore use a hold time of 5 seconds for subsequent experiments. Network administrators using IQU can set an appropriate value for hold time based on their preferred trade-off and the characteristics of the target scenario.

5.6 Impact of Utilization Thresholds

We now examine the impact of the utilization thresholds on IQU behavior. We use the same traffic model as the previous experiments, and set the hold time to 5 seconds. In these experiments, we explore different combinations of values for the lower and upper utilization thresholds. Figure 8 shows the results of our experiments. Figure 8(a) shows a CDF of average user throughput. The figure illustrates that there is significant loss in throughput when (U_{lower}, U_{upper}) are set to (30%, 55%) or (30%, 65%). When U_{upper} is set to a high threshold (55% or 65%), a larger number of users is likely to be allowed to associate with the AP, thereby increasing the probability of congestion and lower throughput. On the other hand, if U_{lower} is set to a low value (20% or 30%), fewer users are allowed to associate with the AP. The system is then under-utilized and the likelihood of higher throughput is increased.

Figure 8(b) shows a CDF of user wait times based on the utilization thresholds. We observe that smaller U_{lower} values can result

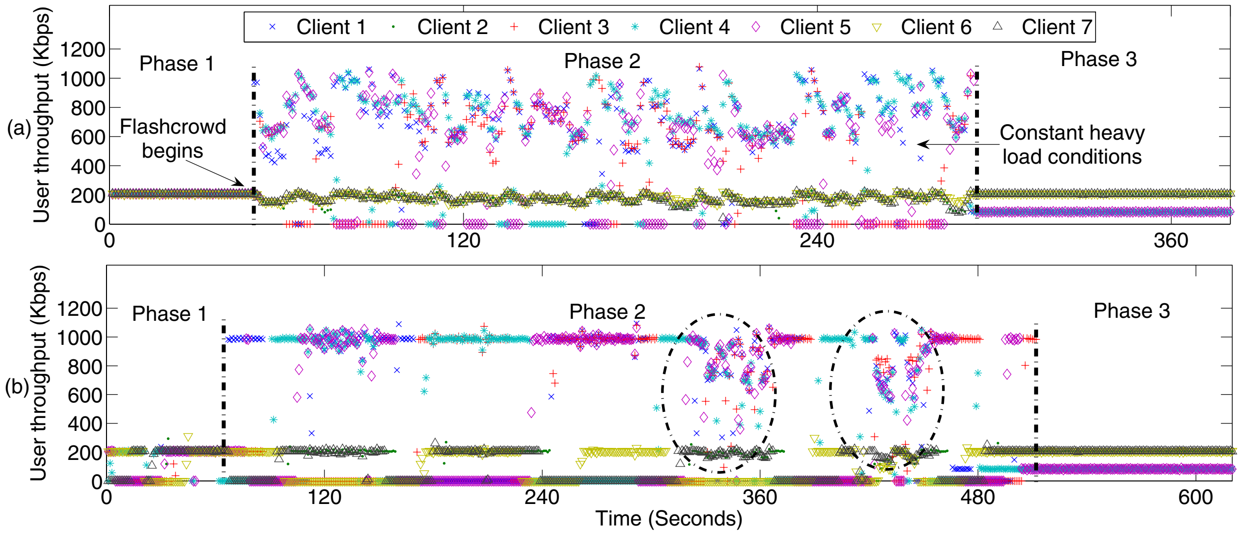


Figure 9: (a) Client throughput without IQU during a flash crowd. (b) Client throughput with IQU during a flash crowd.

in longer wait times for users, while larger values of U_{lower} allow shorter wait times. We identify a tradeoff between user throughput and wait-times in the choice of utilization thresholds. Our results show that selection of the appropriate utilization levels significantly impacts network performance. Further, we believe that this impact can vary under different traffic conditions. Network administrators can set the appropriate utilization thresholds based on their preferred trade-off and the characteristics of the target scenario. For our testbed network, we set U_{lower} and U_{upper} to 40% and 50% because it offers the best trade-off between throughput and wait time in our experiment scenario.

Figure 8(c) shows the percentage of the number of one second intervals during which between one and seven simultaneous users were associated with the AP. This figure validates our previous observation that higher utilization threshold values increase the possibility of a greater number of simultaneously associated users, which can result in congestion, while lower values cause the system to be under-utilized.

5.7 IQU Robustness Evaluation

We now demonstrate the ability of IQU to maintain network stability and high throughput in a scenario that emulates the flash crowd observed in the case studies presented in Section 2. We create a flash crowd through division of traffic generation into three distinct phases. During the first phase, which is one minute long, each of the seven clients maintains a UDP flow of 100 Kbps in each direction. During the second phase, four out of the seven clients instantaneously increase their flow level from 100 Kbps to 1 Mbps in both directions. Though our network is scaled in size, this creates a traffic environment similar to that observed in our case studies, in which a large number of users entered the event room at the same time. The second phase of the experiment lasts for four minutes. In the third phase, the four clients reduce their traffic level to 50 Kbps in each direction. This phase lasts for one minute. Note that the phase durations do not include the time spent waiting in the user queue. We evaluate three aspects of IQU in this scenario: (1) the performance of IQU during a flash crowd; (2) the performance of IQU under constant high load; and (3) the tradeoff of maintaining high throughput and adhering to work period and wait time commitments.

Based on our previous observations, we set T_{hold} to 5 seconds and U_{lower} and U_{upper} to 40% and 50%, respectively. Figures 9(a) and 9(b) show time-series plots of individual client throughputs with IQU disabled and enabled, respectively. In Figure 9(a) we observe that during the first phase of the experiment, each client receives a throughput of 200 Kbps, the offered load on the network. However, during the flash crowd phase, the throughput experienced by each of the seven clients notably degrades. Significant losses and variations are observed in the average individual throughput. This result clearly demonstrates the detrimental effect of a flash crowd on network performance when IQU is not enabled.

On the other hand, we see different individual throughput characteristics in Figure 9(b), where IQU has been enabled. During the first 30 seconds of the first phase of the experiment we observe that IQU allows only a few users to associate with the AP. IQU then adapts to the low load conditions and admits all the clients into the network. During the second phase, we observe that IQU successfully controls the number of associated users such that, when admitted, each associated user experiences throughput close to the offered load. The figure shows two brief intervals of time during the second phase when the individual user throughput decreases significantly. These intervals are circled in the figure for clarity. The throughput degradation occurs when the wait times of clients expire. IQU's policy of strict adherence to wait time estimates causes these clients to be admitted into the network even though the limit N_{perm} on the permissible number of clients is exceeded. This behavior is a limitation of the admission policy we selected for IQU. The likelihood of this situation can be reduced by more accurate wait time estimates.

Although IQU improves throughput during a flash crowd or high network load, the disadvantage is that it takes longer to service the clients in the network. This can be observed from the x-axis limits in Figures 9(a) and 9(b). The extent of increase in service time depends on the choice of parameter values and network traffic conditions. We argue that longer service times are an acceptable tradeoff for network administrators and users to avoid grossly unacceptable network performance or network collapse. Our results show that IQU enables stable and robust operation of a WLAN, especially under challenging load conditions.

6. DISCUSSION

In this section we discuss some issues related to the implementation and deployment of IQU.

Effect of external interference: The users and APs of a WLAN that employs IQU may experience interference from other WLANs and/or wireless devices in the vicinity. This interference decreases the maximum throughput and utilization that the WLAN can attain before becoming congested. To overcome this challenge IQU can be extended to be aware of interference by requiring APs to estimate external interference and determine appropriate utilization thresholds to be used. The procedures for estimating external interference and adjusting the utilization thresholds are open topics for further research.

Throttling large workloads: Some of the users associated with the AP may generate large traffic loads during their assigned work periods. These large workloads can have a detrimental impact on the performance of the network. IQU addresses this problem by appropriately controlling the number of associated users in the network, but is limited by its policy of commitment to work-period and wait time estimates. In addition to IQU, large traffic loads can also be curtailed through the use of throttling techniques.

Impact of client heterogeneity: Some clients in the network may use lower data rates. Data frames transmitted at lower rates occupy more channel time, thereby reducing the bandwidth available to other users and decreasing the overall system throughput [13]. However, it is in the interest of each client to use the highest data rate possible in order to complete its tasks in the allocated work-period.

Sybil attack on IQU-enabled WLANs: A Sybil attack [12] is the use of multiple identities by a single user for selfish or malicious gain. In an IQU-enabled WLAN, the use of multiple fake MAC addresses can enable a client to grab multiple slots in the user queue, thereby obtaining an unfair share of network access time as compared to other *well-behaved* clients using a single MAC address. The solution to such an attack is beyond the scope of this paper.

7. CONCLUSIONS

High concentrations of users in WLANs cause contention and interference problems and over-loading of access points, which in turn lead to poor connectivity and throughput for users, and possible network collapse. Based on our case studies from actual network deployments, we conclude that there is a critical need for a system that addresses these problems. Such a system should be fair to users, maintain high throughput for individual users and for the network as a whole, be resilient to an increase in load, have low complexity and overhead, and be feasible to deploy in an operational WLAN.

In this paper, we presented IQU, a practical queue-based user association management system that addresses the performance problems of heavily-loaded WLANs. By maintaining a queue of users that request network access and granting network access to only a limited number of users at a time, IQU successfully time-shares the network among users such that every user has a fair chance to access the network and high network throughput is maintained. The number of users that may simultaneously access the network is determined based on the observed channel utilization. Through our evaluation, we demonstrate the effectiveness of IQU and the robustness of the system to flash crowds and heavy load. Our evaluation verifies that IQU satisfies all the essential requirements of a WLAN load management system.

IQU is a user association management framework that can be extended and explored along multiple directions. Different queue disciplines can be used to implement diverse network access policies. Metrics other than utilization can be explored to identify the onset of congestion and compute the number of permissible users. Algorithms for accurate estimation of user wait time can be developed. In summary, IQU is simple, practical, and powerful, and creates a solid foundation upon which more complex user association management systems may be built.

Acknowledgments

This work was funded in part through NSF Career Award CNS-0347886 and NSF Network Research Testbeds (NRT) grant ANI-0335302 as part of the WHYNET project. We thank Jim Martin (Netzwert AG), Karen O Donoghue (Naval Surface Warfare Center), Bill Jensen (University of Wisconsin), Jesse Lindeman (Airespace), Gordon Moore, Nick Nikjoo, Phil Quinless (Nortel), and the IETF 62 and IETF 64 NOC teams for assisting us in the collection of the data used in this paper. We also thank the anonymous reviewers and our shepherd Suman Banerjee for their valuable feedback.

8. REFERENCES

[1] Aruba Wireless Networks. http://www.arubanetworks.com.

[2] Cisco Systems. http://www.cisco.com.

[3] Multiband Atheros Driver for Wireless Fidelity (MADWiFi). http://madwifi.org.

[4] San Diego Supercomputer Center. http://www.sdsc.edu.

[5] Trapeze Networks. http://www.trapezenetworks.com.

[6] A. Balachandran, P. Bahl, and G. Voelker. Hot-Spot Congestion Relief in Public Area Wireless Networks. In *IEEE WMCSA*, Monterey, CA, Oct 2002.

[7] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *ACM SIGMETRICS*, pages 195–205, Marina Del Rey, CA, Jun 2002.

[8] A. Barbaresi, S. Barberis, and P. Goria. Admission Control Policy for WLAN Systems based on the Capacity Region. In *IST Mobile Summit*, Dresden, Germany, Jun 2005.

[9] Y. Bejerano, S. Han, and L. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. In *ACM Mobicom*, Philadelphia, PA, Sep 2004.

[10] J. Blanquer, A. Batchelli, K. Schauser, and R. Wolski. Quorum: Flexible Quality of Service for Internet Services. In *USENIX NSDI*, Boston, MA, May 2005.

[11] C. Chiasserini and R. Rao. Performance of IEEE 802.11 WLANs in a Bluetooth Environment. In *IEEE WCNC*, Chicago, IL, Sep 2000.

[12] J. R. Douceur. The Sybil Attack. In *IPTPS*, Cambridge, MA, Mar 2002.

[13] M. Heusse, F. Rousseu, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *IEEE Infocom*, San Francisco, CA, Mar 2003.

[14] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding Congestion in IEEE 802.11b Wireless Networks. In *USENIX IMC*, Berkeley, CA, Oct 2005.

[15] J. Jun, P. Peddabachagari, and M. Sichitiu. Theoretical Maximum Throughput of IEEE 802.11 and its Applications. In *IEEE NCA*, pages 249–257, Cambridge, MA, Apr 2003.

[16] K. Measures, J. Martin, and R. McLatchie. Supercomputing Resource Management - Experience with the SGI Cray Origin 2000. In *WoTUG-22*, Keele, UK, Apr 1999.

[17] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *IEEE Infocom*, Barcelona, Spain, Apr 2006.

[18] L. Mummert, M. Ebling, and M. Satyanarayanan. Exploiting Weak Connectivity for Mobile File Access. In *ACM SOSP*, Copper Mountain, CO, Dec 1995.

[19] M. Portoles, Z. Zhong, and S. Choi. IEEE 802.11 Downlink Traffic Shaping Scheme for Multi-User Service Enhancement. In *IEEE PIMRC*, Beijing, China, Sep 2003.

[20] K. Shen, H. Tang, T. Yang, and L. Chu. Integrated Resource Management for Cluster-based Internet Services. In *USENIX OSDI*, Boston, MA, Dec 2002.