Smart Association Control in Wireless Mobile Environment Using Max-Flow

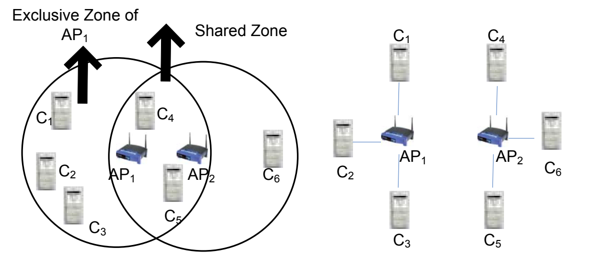
Sourav Kumar Dandapat, Bivas Mitra, Romit Roy Choudhury, and Niloy Ganguly

***Abstract*—WiFi clients must associate to a specific Access Point (AP) to communicate over the Internet. Current association methods are based on maximum Received Signal Strength Index (RSSI) implying that a client associates to the strongest AP around it. This is a simple scheme that has performed well in purely distributed settings. Modern wireless networks, however, are increasingly being connected by a wired backbone. The backbone allows for out-of-band communication among APs, opening up opportunities for improved protocol design. This paper takes advantage of this opportunity through a coordinated client association scheme where APs consider a global view of the network, and decide on the optimal client-AP association. We show that such an association outperforms RSSI based schemes in several scenarios, while remaining practical and scalable for wide-scale deployment. We also show that optimal association is a NP-Hard problem and our max-flow based heuristic is a promising solution.**

***Index Terms*—Load balancing, association control, wireless Internet, max-flow, fairness.**

I. INTRODUCTION

IEEE 802.11, commonly known as WiFi, has proliferated to a new dimension in the last decade. Several factors have contributed towards the popularity of WiFi network, such as connectivity on the fly, low cost and its user friendly characteristics like easy installation, automatic detection of WiFi network, displaying of signal strength from different APs for proper AP selection. Gradually, it is being deployed more and more in home wireless networks and enterprise networks. With the increasing number of users, especially in enterprise setting, WiFi technology faces severe challenges. Among these challenges one important challenge is to accommodate more users with limited infrastructure through *optimal resource allocation*. Study of mobility pattern suggests that the distribution of devices within WiFi network is not at all uniform; on the contrary this distribution follows power law [1], [2]. This implies that a large fraction of devices associate to a small fraction of WiFi APs. Fig 1(a) shows a small portion of WiFi network with two APs and distribution of wireless devices. In this figure, number of devices close to AP_1 is 4, while it is 2 for AP_2. Conventional association scheme based on Received Signal Strength Index (RSSI) [3], [4] would result in associating 4 devices (C_1, C_2, C_3, C_4) with AP_1 and 2 devices (C_5, C_6) with AP_2. With the limited capacity, every AP has a maximum limit on the number of devices that it can associate [5]. This limitation and the non-uniform device distribution may result in two undesirable outcomes: (1) inability to admit all clients, and (2) bandwidth wastage by some APs. The main drawback of the RSSI based association scheme is that it does not take the load on APs into consideration while associating a device. To overcome this problem Least Load First (LLF) [6], [7] based association scheme has been introduced, where a device associates with the least loaded AP. As this scheme only considers load, not the signal strength, devices may suffer from low bit-rate. These traditional association schemes (RSSI and LLF) also suffer from lack of fairness. By fairness we mean that all the devices in the network should get admitted with equal probability, irrespective of the fact that the device belongs to a dense or sparse area. In reality, admittance probability of devices in relatively dense area is much lower compared to the devices in sparse area, which is unfair. These scenarios motivate us to design a new association scheme.



(a) (b)

Fig. 1. Fig. 1(a): Small portion of WiFi network with two APs and a snapshot of clients distribution. Fig. 1(b): Probable association scenario using max-flow based association control algorithm.

One important issue that needs to be considered for the design of the new association scheme is whether the scheme would be centralized or distributed in nature; this decision depends on several factors such as the network size and the additional overhead incurred for implementing the scheme. Due to the NP-hard nature of the association problem [7], centralized methods require large execution-time and result in large overheads on the server implementing the scheme; hence centralized methods do not scale for large networks. Moreover, centralized methods also suffer from single point of failure. Considering the problems with centralized approach, some distributed association methods (such as highest RSSI

Manuscript received December 9, 2010; revised July 2, 2011 and October 31, 2011. The associate editor coordinating the review of this paper and approving it for publication was P. Bellavista.

S. K. Dandapat and N. Ganguly are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India, 703302 (e-mail: {sdandapat, niloy}@cse.iitkgp.ernet.in).

B. Mitra is with ISC-PIF & CREA-CNRS, Paris, France (e-mail: bivas.mitra@iscpif.fr).

R. R. Choudhury is with the Dept. of Electrical & Computer Engineering and the Dept. of Computer Science, Duke University, USA (e-mail: romit@ee.duke.edu).

S. K. Dandapat acknowledges the financial support from Vodafone Essar.

Digital Object Identifier 10.1109/TNSM.2011.113011.100098

or LLF) have been proposed, but they have several problems as discussed earlier. So to address all these issues, designing a distributed bandwidth allocation algorithm which can admit maximum number of devices as well as maintain fairness and high network throughput is an important research challenge.

Recent years are witnessing an architectural shift within WiFi systems. In enterprise settings, such as in airports, universities, and corporate campuses, multiple APs are getting connected to a central controller through high speed wired backbone [8], [9]. As a generalization, modern networks are becoming semi-centralized through hybrid wired-wireless architecture. This paper focuses on exploiting the hybrid nature of wireless networks to improve client association. Hybrid nature of wireless networks offers new opportunities to redesign protocols for future wireless networks. In our proposed Smart Association Control (SAC) protocol, we combine the information from multiple APs to augment load-balancing and fairness. Using the wired backbone, neighboring APs mutually exchange information about the number of clients that are within their communication ranges. Each AP assimilates this global information and divides the network into "exclusive" and "shared" zones (Fig. 1(a)), where clients located in the exclusive zones must associate to a specific AP, while clients in the shared zone have the choice among two or more APs. We model this system as a *max-flow problem* [10] where APs act as sources of bandwidth and zones act as sinks. Our goal is to distribute bandwidth fairly and to obtain the optimal client-AP assignment. Performance improves because the concentration of clients at specific hot-spot regions does not overwhelm a single AP. Instead, clients are distributed among different APs even if the signal strength of their links is relatively weaker. Fig. 1(b) shows a possible max-flow based association corresponding to the problem instance shown in Fig. 1(a). Performance results show that compared to highest-RSSI based association, the proposed max-flow based association accommodates greater number of clients, increases overall network throughput, while improving the fairness up to 50%. The rest of the paper is organized as follows. In section 2, we discuss some relevant works. In section 3, we define some terminologies, and metrics used to evaluate performance of SAC protocol. In section 4, we discuss our proposed protocol, its distributed implementation, and two benchmark algorithms used for comparison. In section 5 we discuss about experimental setup, different traffic and mobility model and simulation results. Section 6 concludes the paper with some future directions of work.

II. Related Work

Association control is a pressing problem; hence in recent times several attempts have been made from both academics and industry to address this issue. The problem has been studied on different kinds of network (e.g. WLAN, cellular).

Association control is done to optimize different network parameter. Among these network parameters fairness [7], [11], [12], [13], throughput [7], [13], interference [14], [15], percentage of blocking [16], handoff frequency [17], energy saving to transfer data packet [18] are important.

A wide range of methodologies have been tried to address association control problem. In [11], [19], [20] authors propose methods of load balancing by simultaneous association of a device with multiple APs. The idea is extremely interesting, however it is an energy hungry solution and hence not suitable for wireless devices. Another interesting approach is cell breathing technique which is investigated in [21]. It contracts (or expands) the coverage of under-loaded (or congested) cells by reducing (or rising) the power level, and therefore the load becomes more balanced. In another approach authors used to predict mobility and accordingly appropriate AP is chosen for handoff [17]. It reduces total number of handoff, hence delay and interruption. In [22], [23] authors tried to balance load of wireless LAN using proper channel management. In another approach, authors try to balance the load by adjusting the load in neighbors [6], [7], [24]. But in this case, adjustment is possible, only if its neighbor can share some load. Further in other proposals, load on AP is advertised in beacon, so that any device can consider signal strength along with load [3], [4]. But the general tendency of any mobile device is to get maximum signal strength instead of looking for overall bandwidth utilization; hence strength of received signal dominates as deciding factor.

In this perspective, our work closely resembles the approach defined in [7], however our approach takes advantage of the wired connection of APs and exploits the hybrid nature of wireless LAN. Our proposed protocol is able to adjust load with a distant AP (through a series of implicit adjustment), opposed to the earlier schemes which can just take advantage of less loaded immediate neighbors (if present). A poster describing our approach has appeared in [25].

III. Environmental Definition

In this section, we illustrate terminologies and technicalities of the mobile Internet used in this paper. We also introduce metrics used to evaluate performance of our proposed algorithm.

A. *Basic Terminology and Protocol*

In this section, we define the concepts of zone and beacon, and also discuss traditional RSSI based association control protocol as well as outline of our protocol.

Zone: Let us consider an area (say an institute campus), which is entirely covered by wireless Internet. Every location of that area is either covered by a single AP or multiple APs. We denote an area as *exclusive zone* if the area is in the range of only one AP. Similarly, we denote an area as *shared zone* or *overlapping zone* if the area is in the range of more than one AP as shown in Fig. 1(a). The entire area covered by all the APs in Fig. 1(a) can be represented as set of three zones (exclusive zone of AP_1, exclusive zone of AP_2 and the shared zone formed by AP_1 and AP_2). Formally, zone is an area under the influence of specific set of APs.

Formation of Zone: Every device scans all available channels and receives beacons from all the APs in its range. On receiving beacon, device estimates the bit-rate from corresponding AP, from the RSSI field of the received beacon. If the bit-rate from a particular AP is not enough for connectivity, then device does not consider AP as a potential candidate for contributing bandwidth for it and hence a non-influencing AP for the device. Thus every device individually finds out the set of influencing AP/APs which can contribute bandwidth for it. Devices under same set of influencing AP/APs form a zone.

Beacon: An AP periodically broadcasts beacon packets after every *beacon interval period*. In response, mobile devices may either send a *probe request* (if they are interested for connection) or may ignore them.

RSSI Based Association: A mobile device receives beacon packets from all APs in its vicinity. Device, interested for association, checks RSSI field of beacon packet, and sends probe request to the AP with strongest RSSI. On receiving probe request, AP associates the device [3], [4].

AP-Client communication: As soon as any client joins the network it associates using RSSI based association and communicates with the APs as per the WiFi protocol standard. Clients which are associated with any AP have the flexibility to later re-associate to a different AP.

Inter AP communication: APs are connected through a wired backbone, and they communicate among themselves according to the CSMA/CD protocol. Such communication, being out-of-band, does not utilize the wireless bandwidth and is much faster than the wireless communication.

B. Metrics

In this section, we introduce the notations used in the paper, and illustrate the set of desirable metrics that an association control algorithm should optimize. We assume that there are *n* APs: AP_1, AP_2, ..., AP_n and *m* zones: z_1, z_2, ..., z_m in the entire network. We also assume that every AP has the same capacity C. The total number of devices requesting admittance in zone z_i is denoted as r_i, and the number of devices actually admitted (by all the APs) from that zone is s_i. We assume that a device is admitted only when an AP can offer it a minimum bandwidth, as configured by the system designer. For convenience, we assume that each device requires a unit amount of bandwidth to get admitted. Hence the maximum number of devices that can be admitted in the network is $n \cdot C$. When the total number of devices requesting admittance exceeds $n \cdot C$, the system needs to undergo admission control in order to optimize different metrics.

Jain's Fairness Index (JFI): JFI is used to measure fairness of bandwidth allocation over zones. This is an important metric in load balancing protocol [26]. To measure JFI of the network, first we measure Fairness Ratio (FR) for each zone. FR is measured as the ratio of admitted device and total number of device of a zone. Fairness ratio for j^{th} zone is defined as

$$FR_j = \frac{s_j}{r_j} \tag{1}$$

where s_j is the number of admitted devices from j^{th} zone and r_j is the total number of devices in j^{th} zone. To ensure higher fairness, FR of the zones across the network should be similar. The fairness is quantified using the following expression

$$\mathit{JFI} = \frac{(\sum_{j=1}^{m} FR_j)^2}{m \times \sum_{j=1}^{m} (FR_j)^2} \tag{2}$$

Higher value of JFI indicates higher fairness in bandwidth allocation. *JFI* value lies between 0 and 1. JFI obtains maximum value 1 when FR value of all zones across the network is same.

Percentage of Connection Admitted (PCA): The PCA is the percentage of devices admitted in the network. If the total number of devices requesting admittance is $M = \sum_{j=1}^{m} r_j$, and $D = \sum_{j=1}^{m} s_j$ is the total number of devices that are actually admitted by the APs, then PCA can be formally defined as

$$\mathit{PCA} = \frac{(\sum_{j=1}^{m} s_j) \cdot 100}{\sum_{j=1}^{m} r_j} = \frac{D \cdot 100}{M} \tag{3}$$

The proposed algorithm also aims to maximize the PCA in the network.

Throughput: Network throughput is the amount of data (bits) transmitted in the network per unit time (second). Throughput is computed by adding effective bit-rate of all the admitted devices in the network, where Bit-Rate (BR) of a device is defined as the number of bits that are transmitted per unit time by a device. If B bits of data are transmitted by a device in t time unit then bit-rate of that device can be defined as

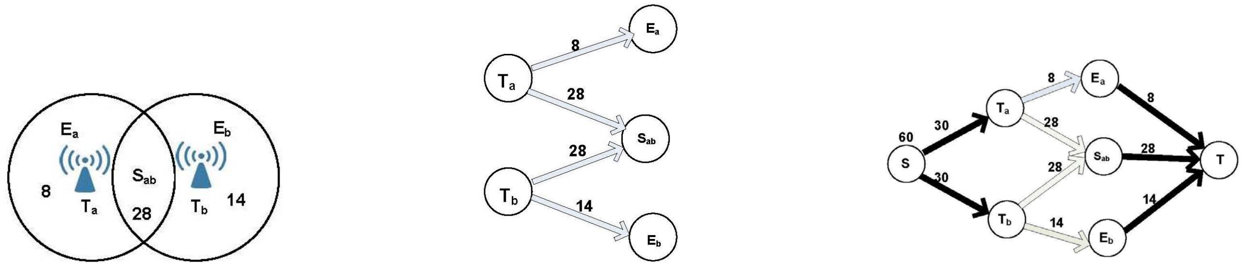
$$\mathit{BR} = \frac{B}{t} \tag{4}$$

If the average bit-rate of all admitted device (D) is assumed as BR_{avg} then throughput of the network can be defined as

$$\mathit{Throughput} = D \times BR_{avg} \tag{5}$$

IV. SMART ASSOCIATION CONTROL PROTOCOL

In this section, we propose Smart Association Control (SAC) protocol which consists of two algorithms: Fair Bandwidth Allocation (FBA) and Association for Maximum Throughput (AMT) algorithm. FBA is a modified max-flow algorithm, which determines number of devices from different zones that an AP should associate. To execute FBA, we map association control problem to a max-flow problem. We show that the pure max-flow algorithm lacks fairness; hence, we suitably modify the algorithm to incorporate fairness. We theoretically justify the applicability of FBA algorithm for association control problem. In AMT algorithm, we propose a protocol which precisely determines one-to-one association between an AP and a device. We present a distributed implementation of SAC (FBA and AMT) which is scalable and fault tolerant. Our analysis reveals that SAC is a very efficient algorithm and it executes in O($NlogN$) time (where N is the number of devices within a zone).



(a) Snapshot of small WiFi Network (b) Mapping (c) Complete max-flow graph

Fig. 2. Reduction of association control problem to a flow problem. Fig. 2(a): An instance of association control problem. Fig. 2(b): Mapping of the problem instance shown in Fig 2(a) to a multiple source multiple sink flow problem. Fig. 2(c): Logical graph at initial stage of flow.

A. *Fair Bandwidth Allocation Algorithm (FBA)*

In this section, we discuss the details of the FBA algorithm.

*1) **Mapping of Association Control Problem to Max-Flow Problem**:* Max-Flow is a classical graph theoretical approach for solving resource allocation problem. Objective of this algorithm is to achieve maximum flow in a given graph from source node to sink node without violating edge capacity constraint. Max-flow can produce many alternative solutions depending on the path selection strategy for flow. In the following, we represent a WiFi network as a flow graph.

Fig. 2(a) shows a small WiFi network where T_a and T_b are the APs and E_a, E_b and S_{ab} are the zones formed by those APs. Fig. 2(b) shows the flow-graph corresponding to the network in Fig. 2(a); both zones and APs are represented as nodes in the flow graph. We introduce an edge between an AP-node (e.g. T_a in Fig. 2(b)) and a zone-node (e.g. E_a in Fig. 2(b)) when the AP is able to contribute bandwidth for the zone. Every zone can avail bandwidth from the set of influencing APs up to the bandwidth requirement of the zone. According to the problem instance of Fig. 2(a), requirement of zone E_a is 8 units, requirement of zone S_{ab} is 28 units and the requirement of zone E_b is 14 units. As zone E_a is under influence of only AP T_a, there is only one directed edge from AP node T_a to zone node E_a and the edge capacity is 8. Similarly, there is an edge between AP node T_b and zone node E_b with capacity 14. Zone S_{ab} has the influence from both AP T_a and T_b. So, edge exists between T_a and S_{ab} and also between T_b and S_{ab}, and both having the capacity 28. Fig. 2(b) shows a connected bipartite graph where APs act as sources of bandwidth and zones act as sinks. Maximum number of devices across all the zones will be satisfied when there is a maximum flow of bandwidth from all the sources (APs) to all the sinks (Zones). Hence, max-flow algorithm is a natural choice for this scenario.

Traditional max-flow algorithm works on a graph with single-source and single-sink, so we add a super-source and super-sink to the flow graph to convert it to a single-source single-sink max-flow problem. The super-source node is connected to all the source nodes (APs) with edges and the super-sink node is connected to all the sink nodes (zones) with edges [27]. The capacity of an edge between the super-source-node and a source-node (AP) is equal to the actual capacity of the AP (C) and the capacity of an edge between a sink-node (zone) and the super-sink-node is equal to the bandwidth requirement of the zone (capacity of the edge from i^{th} zone to super-sink-node is the requirement of i^{th} zone which is r_i). This completes the formation of the flow graph (Fig. 2(c)).

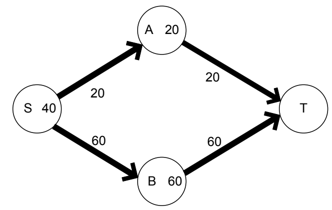
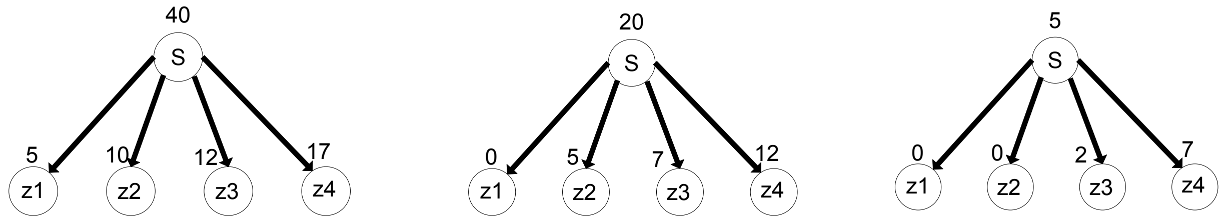


Fig. 3. Max-flow does not maintain fairness!

*2) **Introducing Fairness in Max-Flow**:* Max-flow algorithm ensures maximum flow of bandwidth through the flow graph. But it does not ensure equitable distribution of bandwidth over the zones. It may happen that some of the zones get 100% of their required bandwidth while several other zones get only a small percentage of the required bandwidth. In this section, we discuss the required modification of max-flow algorithm and integrate modified version of max-min fairness with our max-flow algorithm to improve fairness.

Max-Min Fairness and Its Limitation: Let us illustrate the scenario with an example. Fig. 3 depicts two bandwidth allocation scenarios. Zone A has bandwidth requirement of 20 units, and zone B has bandwidth requirement of 60 units and total available bandwidth of AP S is 40 units. Depending on the order of augmenting path selection there will be two kinds of allocation scenarios. In one case ($S-B-T$ is chosen) zone A does not get any bandwidth, in another case, zone B gets only 33% of its bandwidth requirement while zone A gets 100% of its requirement. So both the scenarios are unfair and hence undesirable. To counter this problem, we propose to incorporate the fairness features of the *Max-Min fairness* [28] algorithm in the max-flow algorithm.
Max-Min fairness algorithm is a well-known algorithm used to ensure fairness in resource allocation. Fig. 4 illustrates an example of Max-Min fairness with a single AP and four zones. Four zones z_1, z_2, z_3 and z_4 have bandwidth requirements 5 units, 10 units, 12 units and 17 units respectively (Fig. 4(a)). Fig. 4(b) and 4(c) represent the amount of bandwidth still required by the zones after 1^{st} and 2^{nd} stage of allocation. However, this kind of stage allocation policy penalizes zones having higher requirement and goes against the fairness principle of proportionate allocation. Moreover, single stage proportionate allocation in



(a) Initial Configuration (b) After Stage1 (c) After Stage2

Fig. 4. Three stages of bandwidth allocation according to Max-Min fairness.Fig. 4(a): Initial configuaration. Fig. 4(b): After 1^{st} stage of allocation. Fig. 4(c): After 2^{nd} stage of allocation.

a multiple source (multiple APs) scenario penalizes exclusive zones or zones with less number of influencing APs. Hence, bandwidth should be allocated in multiple stages instead of a single stage to reduce the unfair distribution of bandwidth. In the following, we discuss the modified stage allocation policy.

Modification: In every stage, each AP allocates a certain percentage of its total capacity among all its attached zones proportionately to their requirements. Let us assume that the total capacity of an AP is C, and p zones are attached with the AP with requirements req_1, req_2, ..., req_p respectively (this p is a small value due to the geometric distribution of APs in planned enterprise). In a particular stage, if an AP allocates $x\%$ (we refer this x as *allocation fraction*, which means fraction of the bandwidth to be allocated in current stage) of its capacity to all its attached zones, then the edge capacity between the AP and the i^{th} zone is $EdgeCap_i$, where $EdgeCap_i$ is formally defined as

$$EdgeCap_i = \frac{C \cdot x}{100} \times \frac{req_i}{\sum_{i=1}^{p} req_i} \tag{6}$$

An exclusive zone receives bandwidth from a single AP while shared zone receives bandwidth from multiple APs and every AP allocates bandwidth independently. It may result in over allocation of a shared zone and at the same time starvation of exclusive zone. This would reduce *PCA* of the network. To avoid this scenario, we should carefully choose x for every stage. Fig. 5 shows actual number of devices from different zones which are going to be associated with specific AP as outcome of 1^{st} stage of allocation on example WiFi network shown in Fig. 2(c) with $x = 90$. Although it was targeted to utilize 90% of capacity (27 units) of each AP, T_a could use only 53% of bandwidth (16 units). This choice of x impacts the allocation in 2 ways: (a) all the APs (here T_a) could not utilize their full quota, and (b) some zones (here E_b) would never be successful in acquiring their complete requirement (even after the last stage of allocation), because all the influencing APs will have imprudently exhausted their complete bandwidth. To counter these impacts, we should choose smaller value of x in the initial stages and progressively increase it. At the same time, we need to remember that smaller value of x increases the total number of stages in FBA, and hence the execution time. After a set of experiments, we have chosen a four-stage allocation policy; at the first stage, 50% of the capacity of each AP is targeted to be utilized, and the target is then successively set to 75%, 92% and 100%. We refer this as $targeted\ capacity$ for stages.

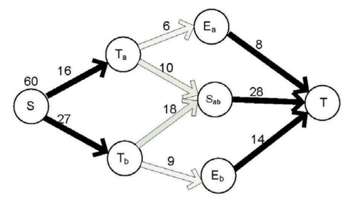


Fig. 5. Flow graph after 1^{st} stage of allocation when x is chosen to be 90.

*3) **FBA Algorithm**:* In this section, we integrate aforesaid stage allocation policy with max-flow algorithm to propose the FBA algorithm. This algorithm takes allocation fraction, bandwidth requirement of all the zones and capacity of all APs as input parameters and generates *association weight* for every AP-zone pair. *Association weight* of an AP-zone pair indicates the number of devices that the AP is going to associate from the zone. At the beginning of each max-flow stage, flow graph needs to be formed. To build the flow graph, remaining bandwidth requirement for all the zones as well as capacity of the APs to be allocated for the current stage is computed. Remaining bandwidth requirement for a zone in 1^{st} stage is equivalent to the original bandwidth requirement of the zone. For all other stages, remaining bandwidth requirement for a zone is computed as the difference between total bandwidth requirement of the zone and total *association weight* between this zone and all influencing AP of this zone. Capacity of an AP for the current stage is set by computing the difference between $targeted\ capacity$ (50%, 75%, 92% and 100% for consequent stages) and the capacity utilized by all previous stages. Then according to the procedure discussed in section IV-A1, a flow graph is formed and max-flow is executed on that flow graph. At the end of each stage, max-flow results in some flow between APs and zones which is *association weight* between AP-zone pairs for the current stage. This completes one stage of max-flow. At the end of four stages of max-flow, we get the cumulative *association weight* which is considered as final *association weight* between AP-zone pairs.

*4) **Distributed Implementation of the FBA**:* We have implemented a distributed algorithm which is scalable and fault tolerant [1].

[1]For any distributed system, fault tolerance is an important feature. When an AP fails, all the devices associated with it, will not receive beacon from it. The devices belong to the exclusive zone of the failed AP will experience network unavailability. However, the devices in shared zones will still get beacons from other APs in their range and they will associate with one of those APs.

Distributed implementation of max-flow needs active participation of every node in the flow graph. However, except AP nodes all other kinds of nodes in max-flow graph are logical nodes and hence do not have any physical processor. So the APs take the responsibility of executing processes on behalf of all zone nodes, the super-source node and the super-sink node. Procedure of selecting responsible AP for the execution of processes of other kinds of nodes (zone, super-source, and super-sink) is as follows. Responsibility of executing a zone process is taken by one of those APs having influence over the zone. As exclusive zone has influence of only one AP, zone process is executed in that AP. On the other hand, for a shared zone, one AP is elected from the set of influencing APs based on distributed election algorithm [29]. AP election to execute processes on behalf of super-source node and super-sink node is done in similar fashion as AP election is done for shared-zone.

For the problem instance shown in Fig. 2(a), a possible mapping of the processes and APs in which they are executed is shown in Fig. 6.

Algorithm:FBA
Input: Allocation fraction for the stages;
Bandwidth requirement of all the zones;
Capacity of all APs;
Output: *association weight* for every AP-zone pair
foreach *AP-zone pair* **do**
    initialize *association weight* to 0;
end
foreach *stage of allocation* **do**
    Consider the allocation fraction (x) ;
    Compute the capacity of all APs for current stage;
    Compute the remaining bandwidth requirement of all zones;
    Use modified max-min fairness to create flow graph;
    Execute max-flow on this flow graph;
    Update *association weight* for AP-zone pairs;
    foreach *zone* **do**
        if *total association weight by all APs> required bandwidth* **then**
            Push back the bandwidth to contributing APs in proportion to their contribution.
        end
    end
end

We adopt the distributed implementation of max-flow algorithm proposed by Pham et.al [30], [31]. In this implementation, all the internal nodes (except super-source node and super-sink node) in max-flow graph are similar. However, in our implementation, there are two types of internal nodes: AP nodes, zone nodes. Nodes exchange messages among themselves to build the flow graph and ultimately to push bandwidth towards super-sink.

*5) **Messages Exchanged in FBA**:* Two main jobs of FBA algorithm are flow graph formation and execution of max-flow on that flow graph. Flow graph formation requires knowledge of allocation fraction for the current stage, capacity left of APs and the current bandwidth requirement of zones. To know the bandwidth requirement of attached zones, every AP sends *bandwidth request message* (Fig. 7) to all its attached zones[2]. In response, every zone process sends *bandwidth response message* containing the current bandwidth requirement of the zone. Every AP waits for p such messages, where p is the number of zones under its influence. After receiving all p messages, AP computes the bandwidth (according to Eqn. 6) that needs to be allocated/pushed to each zone. Each AP then sends *pushed bandwidth message* containing the amount of bandwidth pushed and zone identifier to all its attached zones. Consequently, each zone process waits for q number of pushed bandwidth message, while it is under influence of q APs. After getting all pushed bandwidth message, a zone process updates the amount of bandwidth received from all the influencing APs and re-computes the current requirement. In response to pushed bandwidth message, zone sends *pushed acknowledgement message*. Pushed acknowledgement message contains a control bit which indicates the utilization of the pushed bandwidth. The values 0 and 1 of control bit respectively indicate full and partial utilization of pushed bandwidth. In case of over allocation of bandwidth for a zone (control bit is set to 1), excess amount of bandwidth is pushed back to the APs according to their contribution. The amount of bandwidth pushed back to AP is mentioned in the pushed acknowledgement message and receiving AP adjusts their capacity accordingly. From the pushed bandwidth messages of FBA stages, every zone computes the association weight between the influencing APs and the zone.

Fig. 6. Mapping of the processes and APs.

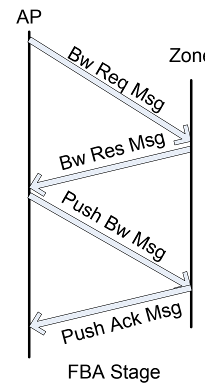


Fig. 7. Messages exchanged during FBA stage.

*6) **Theoretical Justification of FBA**:* We have shown that max-flow naturally fits into the bandwidth allocation algorithm. The effectiveness of max-flow can be shown by analyzing the complexity of the problem. In this section

[2]In this paper, zone is either a zone process or a geographical area corresponding to a zone. Meaning of zone will be clear from the context.

we reveal that finding the optimal solution of the allocation problem is NP-Hard, whereby max-flow is a good heuristic solution.

For the theoretical formulation of the problem we have defined another metric called Percentage of Flow (PF). This metric tries to estimate the performance of the system comparing the current association scenario with best possible association scenario at that load situation. At the low load (less than 100%) scenario this is similar as PCA; when the load is high, it compares the number of device associated with maximum number of possible association. Definition of PF is as follows.

Percentage of Flow (PF): We measure the performance of the system by the PF metric. It is measured as max (percentage of device satisfied, percentage of satisfied device with respect to number of devices that can be satisfied at max) which is formally defined as

$$\begin{aligned} PF &= \max\left\{\frac{\sum_{j=1}^{m} s_j \cdot 100}{\sum_{j=1}^{m} r_j}, \frac{\sum_{j=1}^{m} s_j \cdot 100}{n \cdot C}\right\} \\ &= \max\left\{\frac{D \cdot 100}{M}, \frac{D \cdot 100}{n \cdot C}\right\} \end{aligned} \tag{7}$$

Problem Formulation: We want to optimize two objective functions for our intended association control protocol. One objective is to minimize the variance of Fairness Ratio (FR) across the zones and other objective is to maximize the PF. So we want to minimize f_{obj}, where one way of representing f_{obj} is

$$f_{obj} = var(FR \text{ across zones}) - PF \tag{8}$$

Let us consider a WiFi network with m number of zones and n number of APs. Let us assume that in a particular moment, r_j is the requirement of j^{th} zone. We also assume that, in such traffic scenario the optimum number of device that i^{th} AP should associate from j^{th} zone is x_{ij}. If i^{th} AP does not have influence on j^{th} zone then x_{ij} is always 0, otherwise it is greater than or equal to 0. Formally,

$$x_{ij} = \begin{cases} 0 & \text{if } i^{th} \text{ AP does not have influence in } j^{th} \text{ zone.} \\ \geq 0 & \text{otherwise} \end{cases} \tag{9}$$

As the total amount of bandwidth that an AP can provide to its clients is less than or equal to its capacity (C), we formally represent the constraint as

$$\sum_{j=1}^{m} (x_{ij}) \leq C \qquad [1 \leq i \leq n] \tag{10}$$

Similarly, as j^{th} zone has requirement r_j, so contribution from all APs (s_j) for that zone should be less than or equal to r_j. Formally,

$$s_j = \sum_{i=1}^{n} (x_{ij}) \leq r_j \qquad [1 \leq j \leq m] \tag{11}$$

From equations (1) and (11) we can write

$$FR_j = \frac{s_j}{r_j} = \frac{\sum_{i=1}^{n} (x_{ij})}{r_j} \tag{12}$$

From equation (12) we can write

$$\mu = AVG(FR_j) = \frac{1}{m} \times \sum_{j=1}^{m} \frac{\sum_{i=1}^{n} (x_{ij})}{r_j} \tag{13}$$

Let us assume that X be a discrete random variable which takes the value from list of fairness ratio over the zones $[FR_1,\ FR_2,...,FR_m]$ with equal probability, var(X) can be mathematically expressed as

$$var(X) = \frac{1}{m} \times \sum_{j=1}^{m} (AVG(FR_j) - FR_j)^2 \tag{14}$$

From equations (7) and (11) we can write,

$$\begin{aligned} PF &= \max\left\{\frac{\sum_{j=1}^{m} s_j \cdot 100}{\sum_{j=1}^{m} r_j}, \frac{\sum_{j=1}^{m} s_j \cdot 100}{n \cdot C}\right\} \\ &= \max\left\{\frac{\sum_{j=1}^{m}\sum_{i=1}^{n} (x_{ij} \cdot 100)}{\sum_{j=1}^{m} r_j}, \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} (x_{ij} \cdot 100)}{n \cdot C}\right\} \end{aligned} \tag{15}$$

Hence our objective is to minimize value of f_{obj} where

$$f_{obj} = \frac{1}{m} \times \sum_{j=1}^{m} (\mu - FR_j)^2 - PF \tag{16}$$

Subject to:

$$\begin{aligned} &\sum_{j=1}^{m} (x_{ij}) \leq C \qquad [1 \leq i \leq n] \\ &s_j = \sum_{i=1}^{n} (x_{ij}) \leq r_j \qquad [1 \leq j \leq m] \end{aligned} \tag{17}$$

In the analysis of the problem, we show that FBA is suitable heuristic for this optimization problem.

Analysis of the Problem: As equation (16) contains quadratic term, instead of linear programming, quadratic programming is required to handle this optimization problem. Mathematically, quadratic programming can be represented as, Minimization of the objective function $f(x)$ where

$$f(x) = cx + \frac{1}{2} x^T Q x \tag{18}$$

Subject to:

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned} \tag{19}$$

In equation (18), c is a row vector describing the co-efficient of the linear terms and Qx is the symmetric matrix describing the co-efficient of the quadratic terms. In addition, x is a column vector of decision variables and x^T is transpose of matrix x. In equation (19), constraints are defined as matrix A.

Let us now compare equation (16) with equation (18). As there are n number of APs and m number of zones, we have $m \cdot n$ number of decision variables (x_{ij}) in our quadratic equation. So dimension of c vector for the equation (16) is $m \cdot n$ and dimension of Qx is $m \cdot n \times m \cdot n$. Moreover, derivation of equation (16) reveals that diagonal elements of Qx contain negative element. It implies that matrix is not positive definite matrix or it has negative eigenvalue. Quadratic programming with negative eigenvalue is a NP-Hard problem [32], [33]. Lots of heuristic solutions have been proposed to get solution of this kind of NP-Hard problem [34], [35], [36]. Out of which max-flow [34] is a possible method. Intuitively again, it is

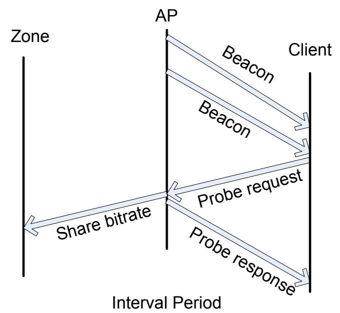


Fig. 8. Messages exchanged during the interval period.

easy to understand that we can satisfy maximum number of devices using max-flow and with a little modification, fairness can be ensured in max-flow. So, modified max-flow (FBA) is a natural choice for this NP-Hard problem.

B. Association for Maximum Throughput (AMT)

In FBA stage, APs collaboratively decide the number of devices they are going to associate from each zone. We refer to this number as *association weight* between the AP and the zone. However, one-to-one association between device and AP is still left after FBA stage. A protocol needs to be devised for one-to-one association between devices and APs without violating FBA constraint (association weight) such that overall network throughput is maximized. We propose an *Association for Maximum Throughput (AMT)* algorithm to achieve aforesaid objective. This algorithm is being executed in each zone process in parallel.

Message Exchanged in Interval Phase: Before describing AMT in detail, we briefly describe the association mechanism in the interval period of two successive executions of SAC. As soon as a device enters in WiFi network, it receives *beacon message* from all the APs in its range. If the device is interested for association, it checks the RSSI field from all received beacon and sends *probe request message* to the AP from which it is getting strongest signal (Fig. 8) and the AP associates the device if possible. It is worth mentioning here that the probe request message contains some additional information like AP identifier of all the APs in device's range and corresponding bit rate from those APs. On receiving probe request message, AP takes the following actions: 1) sends *probe response message* and associates device if possible 2) extracts the extra information from probe request message and construct *share bit-rate message*.

Fig. 9 shows structure of share bit-rate message of a device D_1 where q is the number of influencing APs for the zone. AP identifier, bit-rate from that AP and the device identifier of client are kept as a 3-tuple in share bit-rate message. AP identifies the zone to which device belongs from the list of APs in share bit-rate message and sends share bit-rate message to the corresponding zone process. However, zone process for a zone may not exist (just after network setup) and in such case, a new zone process has to be formed. The AP receiving the first probe request message from any device of that zone initiates the process of electing a responsible AP for the new zone process; it sends a control message to all others APs in the list of share bit-rate message, indicating that a new zone has to be formed (i.e. an AP needs to be elected for handling the zone process for this new zone). All the APs getting this message generate a random number and exchange among themselves. The AP which generates the minimum number is elected as the responsible AP for the new zone. Once the zone process is active, share bit-rate message is forwarded to the zone process. Thus in interval period, zone process receives detail of all the devices that belong to it and make an aggregated list of messages.

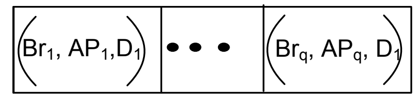


Fig. 9. Structure of share bit-rate message of a device.

Message Exchanged in AMT Phase: At the beginning of AMT stage, every zone has obtained the association weight of all influencing APs. Moreover, zone process has complete information of the devices belong to that zone and the bit-rate that each device can avail from all APs. This information is stored in a common share bit-rate message structure as shown in Fig. 9. The zone process sorts the (bit-rate, AP, D) tuples contained in all the share bit-rate messages in non-increasing order of the bit-rates. Let the tuple with the highest bit-rate among all share bit-rate messages in this zone be noted as (*peak.bit-rate, peak.AP, peak.device*). The devices are then associated as follows.

The zone process checks if *peak.AP* can still accommodate more devices from this zone according to current association weight. If *peak.AP* can still accommodate devices from this zone, the zone process sends *association message* (Fig. 10) containing the identifiers of *peak.device* and *peak.AP* to all the influencing APs for this zone. If *peak.device* is already associated with *peak.AP*, then no further action is required except for reducing the association weight for the AP-zone pair. However, if *peak.device* is associated with some other AP, that AP sends *disassociation notification message* to *peak.device*. Disassociation notification message contains the identifier of *peak.AP* to guide *peak.device* in re-association. Then *peak.device* sends *re-associate request message* to suggested AP (i.e. the *peak.AP*) and association is done. The above process is repeated until all the devices in this zone are associated or all APs influencing this zone finished their quota of association weight. Once a device is associated, all tuples of that device are ignored henceforth. When association weight reaches zero for some AP-zone pair, the zone process ignores all tuples corresponding to that AP.

C. Run Time Complexity

In this section, we analyze the run time complexity of SAC protocol. As SAC consists of two algorithms: FBA and AMT, we subsequently analyze the complexity of each of the algorithm separately.

Analysis of FBA Algorithm: Let us assume that the total number of nodes in flow graph is v and total number of APs in the network is n. If every vertex participates in the

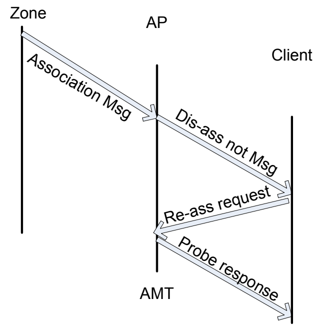


Fig. 10. Messages exchanged during the AMT phase.

Algorithm:AMT
Input: 1) *association weight* generated by FBA stage between this zone and all influencing APs;
2) List of tuples of share bit-rate message (li)
Output: One-to-one AP client association
while *li is not empty* **do**
  Find the tuple with *peak.bit-rate*;
  Denote the corresponding AP and device as *peak.AP* and *peak.device*;
  if *association weight for this zone and peak.AP* > 0 **then**
    Associate the *peak.device* with *peak.AP*;
    Remove all the tuples with *peak.device* id from li;
    Reduce *association weight* for the AP-zone pair;
  end
  else
    Remove all tuples with *peak.AP* id from li;
  end
end

computation, the distributed implementation ensures that the run time complexity of the max-flow algorithm is O(dv) where d is the length of the longest path between super-source and super-sink [31]. In our context, d always remains constant at 3 irrespective of network size (Fig. 2(c)); hence the complexity reduces to O(v). In our protocol, since only APs are the computing nodes, we need to establish a relationship between the number of AP's- n, and the total number of nodes- v. In planned enterprise settings, the AP deployment ensures that each AP influences a small number of overlapping zones. Hence the maximum number of zones is O(n). Therefore, the number of nodes in the flow graph is also O(n). This implies that each AP needs to execute the algorithm on behalf of a constant number of nodes in the flow graph. Therefore, the time complexity does not degrade, and each individual AP takes O(n) time. As max-flow takes O(n), FBA (4-stage max-flow) also runs in O(n).

Analysis of AMT Algorithm: Run time of *AMT* depends on number of devices within each zone. To achieve the goal of maximum throughput, every zone process needs to find out device with maximum bit-rate repeatedly. Let us assume that N is the number of devices in the zone. Using priority queue, we can perform the required operation in O($NlogN$) time. But, as N is not very big (in order of hundred) it does not take large time.

Now considering both FBA and AMT, SAC protocol executes in O($n + NlogN$) time where it is expected that $N > n$ so it reduces to O($NlogN$). This can allow operations in the order of packet transmission times. Though we have taken Indian Institute of Technology Kharagpur campus as a case study, proposed algorithm is scalable for a large network such as city network.

V. Results and Discussion

In this section, we state two schemes RSSI and LLF which are used as benchmark to evaluate performance of our algorithm. Next, we present a brief description about the experimental setup used in the simulation. We also discuss about the traffic distribution and mobility models used in our work. In simulation results, we compare performance of proposed SAC algorithm with RSSI and LLF based algorithms.

A. *Schemes for Comparison*

To evaluate the performance of our proposed algorithm, we compare its performance with one RSSI based association algorithm and one LLF based association algorithm. These approaches have been used for comparison in several recent papers: comparison with RSSI based algorithm (may be found in [11], [7]), comparison with LLF based algorithm (may be found in [7]).

In RSSI based algorithm, mobile devices associate with the AP having the strongest signal. Every device checks the signal strength from all the APs in its range and selects the AP with highest signal strength for association.

In our chosen LLF approach, there is a central server, which keeps track of the entire network. Every mobile device first associates with an AP with strongest signal. During this association, device provides 'QoS it expects' and 'list of APs' in its vicinity to the AP to which it is being associated. AP sends this information to central server. Central server chooses one AP with least load [6] which can fulfil the QoS requirement of device.

B. *Experimental Setup and Simulation*

We perform a simulation-based experiment in order to show the effectiveness of our proposed association control protocol over the RSSI and LLF. The simulator is developed using C. The detail of the simulator is presented in this section.

Physical Layer Model: We use a simple wireless channel model where the bit-rate depends only on the distance of the client from the AP. We have assumed that range of an AP is 100 meter. We have also assumed that a client will receive 36 Mbps speed when it is within 30 meter range from the AP, when the distance is more than 30 meter then bit-rate is measured according to the Shannon's law with constant (unit) noise. Transmission power of every AP is assumed as 100 mW. We assume that size of contention window is constant (32). It is also assumed that there is a limit on the number of devices that an AP can associate simultaneously and this limit is set to 60 [5].

Mac Layer Model: Presented network is a WiFi infrastructure network, where APs of every BSS's are connected through wired backbone and thus creates a distribution system. Devices use DCF for data transmission. We have assumed that, each device requires a unit amount of bandwidth to get admitted.

Topology: We have considered a part of the transportation network of Indian Institute of Technology (IIT) Kharagpur, India as a case study. Fig. 11 shows the map of the IIT Kharagpur campus with proper scaling [37]. The black solid circles on the map are considered as the APs. We consider that APs are placed at the junctions of the roads. If the distance between two nearest road-junctions is more than 150m, then we consider more APs are placed in between two road junctions, in order to make sure that the maximum distance between two APs is 150m. Proceeding in this way, we need 42 APs and 187 zones (42 exclusive zones and 145 shared zones).

Traffic Distribution: Next we consider the geographical distribution of mobile devices along the roads. We assume that the density of mobile devices on a given road is determined according to the importance of the road. To estimate the importance of a road segment, we compute its *betweenness* where, betweenness of an edge in a graph is defined as the fraction of shortest paths (out of the shortest paths between each pair of nodes) which pass through the given edge [38]. We consider important places in the campus (such as departments, hostels, cafes) as nodes and the roads connecting these places as the edges in a graph. We consider the shortest paths (along the roads) between each pair of important places, and compute the fraction of shortest paths which pass through a given road - this fraction (edge betweenness centrality) is considered as a metric of the importance of the road. Road segment between two adjacent APs (according to the transport network) is denoted as road fragment (an example is shown in Fig. 11 with symbol A). However, mobile devices are associated with human beings, so study on human mobility pattern can provide approximation and the pattern of mobile device distribution over an area. Studies of GPS trace have revealed that in a city/campus, traffic distribution is not uniform; some places are more popular than others. Many recent studies suggested that this nature of traffic distribution follows power law [1], [2]. Population of a zone is estimated considering all the portion of road fragments lying within that zone. Similarly, we measure the population of zone for uniform traffic distribution. It is assumed that devices are distributed randomly within a zone. However, we use both uniform and power law traffic distribution for our simulation.

Mobility Model: We consider a simple mobility model (random waypoint model [39]) to evaluate robustness of our algorithm in face of device movement. In this model, at each step devices, selected for movement, move from one zone to another. Destination zone is randomly selected and accordingly devices choose speed between 3km/hour to 25km/hour. However we made bit modification in the random waypoint model to make it more realistic. We also assume that 60% zones are participating in this movement at each step. Among these zones, half are originating points of devices while the rest are destination points of those devices. For convenience, we divide these chosen zones into pairs, whereby 20% of devices from the originating zones move to the destination zones. The originating zone and destination zone both are chosen randomly. This process is repeated 20 times.

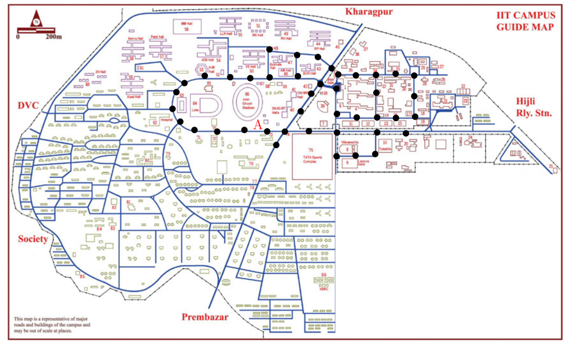


Fig. 11. Campus map of Indian Institute of Technology (IIT) Kharagpur and assumed deployment of APs across the campus.

Flash Crowd: Due to some temporal event like a fair, football match etc, few zones may get over crowded. We define this as flash crowd. To evaluate the performance of our association control protocol in such circumstances, we have simulated a special flash crowd environment. Three zones are randomly chosen as target zones and devices from other zones randomly move to either one of the target zones until total traffic of the targeted zones reach 10% of current population.

Simulation: As there are 42 APs and each of the AP can associate up to 60 devices, 100% load signifies 2520 (42×60) devices. In simulation, we vary the load from 5% to 250% with step size 5 and evaluate performance of proposed SAC algorithm, RSSI based algorithm and LLF based algorithm. To test performance of any algorithm on a particular load, we distribute the traffic according to power law and uniform distribution and subsequently execute the algorithm to compute Percentage of Connection Admitted (PCA), Jain's Fairness Index (JFI) and throughput. For every step, we compute different metrics and check the effect of movement and interference. We average performance over 200 runs and present the results.

C. Simulation Results

In this section, we present our simulation results to show that SAC outperforms traditional association control algorithms based on RSSI and LLF. First, we show that fairness of SAC is maintained even in very high load. Next, we show that SAC can accommodate the number of devices which is near to optimal and maintain a high network throughput. Finally, we show that SAC is robust in face of movement and the effect of interference on SAC is almost negligible in terms of PCA and throughput. However, SAC has some extra message overhead which is the additional cost incurred to achieve aforesaid performance. We show that this cost is tolerable.

Fairness Comparison: Fig. 12 shows that SAC protocol outperforms RSSI and LLF at higher load while it performs comparably with RSSI and LLF at low load scenario. Pro-

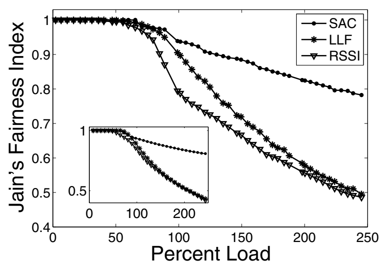


Fig. 12. Fairness (JFI) varying with load using 1.SAC 2.LLF 3.RSSI using power law traffic distribution and inset is the uniform traffic distribution.

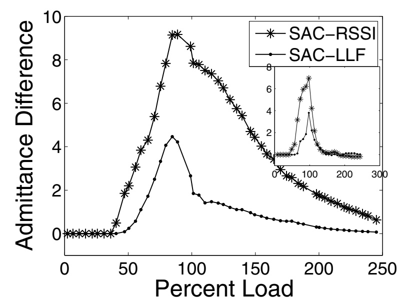


Fig. 13. Difference of PCA with varying load for 1.SAC and RSSI 2.SAC and LLF for power law distribution of traffic, and inset is the uniform distribution of traffic.

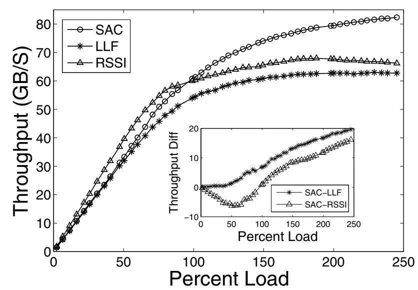


Fig. 14. Throughput (Gb/sec) against load of using 1.SAC 2.RSSI 3.LLF using power law traffic distribution. Difference of SAC-RSSI and SAC-LLF is shown in inset.

portionate allocation of bandwidth among zones makes the allocation fair. However, due to the skewness of the traffic movements, some zones become overcrowded and others remain less crowded. With increasing skewness, proportionate bandwidth allocation becomes difficult. Hence devices would experience non-admittance; and consequently JFI in the entire network decreases. The challenge lies in maximizing JFI in such arduous scenario. The ensured fairness of SAC stems from the fact that it can elegantly utilize the "global view" of all resources and distribute the load to suitably adjust the association. In effect the load of overcrowded area can be rippled out to distant (apparently not adjacent) APs.

Percentage of Connection Admitted: Fig. 13 confirms that SAC protocol performs better than RSSI and LLF also in terms of PCA. Inherent optimality of max-flow helps to maximize PCA. When total traffic is significantly less, all devices can be admitted irrespective of association strategy. Hence hardly any difference persists among strategies. As load increases, skewness of traffic distribution increases. In such scenario, SAC avoids bandwidth wastage of APs by multi hop adjustment using max-flow. Hence, SAC achieves PCA difference over other strategies and peaks around 100% load. Though admittance difference is not very high (difference is 5%-10%), but we achieve this difference in most viable load region (50%-250%). Inset of Fig. 13 shows the PCA difference when traffic distribution is uniform. In this case, difference of PCA becomes a bit lower (4%-8%). However, beyond 250% load, number of devices in each zone becomes much higher. In presence of skewness and without multi-hop adjustment, every AP gets enough devices around it so that no AP wastes bandwidth and associates maximum number of devices. So, PCA difference becomes insignificant beyond 250% load. It is worthwhile to comment here that PCA is best obtained when we run simple max-flow algorithm instead of FBA. However in simple max-flow, we observe huge fairness deviation.

Comparison in terms of Overall Throughput: Fig. 14 shows gain of SAC protocol over RSSI and LLF in terms of overall throughput. Total network throughput depends on the total number of devices admitted and availed bit-rate by each of the admitted device. But, PCA differences among strategies are not very high and hence availed bit-rate by each device plays the dominating role in throughput gain. Though the PCA obtained by the SAC algorithm is higher than that obtained by the RSSI algorithm in between 50% and 250%, the throughput obtained is insignificant in between 50% and 100%, which can be explained as follows. In this region of load, there are a moderate number of devices; to achieve higher PCA and fairness, some devices are associated sub-optimally (i.e. with an AP other than the one from which it would have obtained the highest bit-rate) in order to satisfy the constraint imposed by FBA, and this results in lower throughput than expected because of high admission. But when load is high, there are a large number of options for selecting devices to associate to each AP, hence the constraint imposed by FBA can be fulfilled without sub-optimal association. This possibility of sub-optimal association gradually reduces with load as number of devices becomes significant at higher load. At higher load, though there is no PCA difference, careful choice of devices for association by AMT algorithm helps to increase total throughput. So our scheme performs better than RSSI in higher load. In terms of throughput, SAC clearly outperforms LLF, as LLF never considers bit-rate during association. Inset of Fig. 14 is showing the difference of SAC with RSSI and LLF. However, we are not considering the possibility of wrong estimation of bit-rate which may happen in real life situation.

Effect of Movement: In between two executions of SAC protocol, association of new devices as well as re-association due to movement of devices is handled by RSSI. Fig. 15(a) shows variation of fairness with and without movement. Though JFI reduction due to movement is prominent, fairness after movement is still much higher than RSSI and LLF based algorithm. While the load on most of the APs is relatively low, it is expected that a device moving from one zone to another would get associated with a different AP. So at low load, hardly any difference exists in fairness due to movement. As load increases, fairness difference appears though maximum difference is 0.1. The difference arises because at the interval period devices get associated through RSSI, hence the inefficiency due to RSSI slightly creeps into the system. The difference gets eliminated with the next round execution of SAC. Fig. 15(b) shows the impact of load on PCA with and without device movement. Though PCA degrades due to movement, degradation is not significant. Fig. 15(c) shows the effect on throughput and JFI on different association control protocol due to flash crowd. In flash crowd scenario, SAC outperforms LLF and RSSI in terms of throughput and JFI, whereas PCA is comparable in all three.

Effect of Interference: In this paper, we consider two models of interference. In first model we consider that in a planned enterprise setting, interference can be ignored [40]. However, we try to estimate the performance degradation in presence of interference. We introduce an adversary in the proposed protocol to estimate the effect of interference. As there are only 3 non-overlapping channels [40], zones having influence from more than 3 APs suffer from interference (ignoring interference among devices). To model the adversary, we assume that when a zone is under the influence of $n > 3$ APs, the devices in that zone may not be able to associate with all APs due to interference; instead it may associate with maximum $n - k$ ($k = 1, 2$) APs, thus representing certain loss of association opportunity for a device. So, for every device in such zones, there are one or two prohibited APs, which are chosen randomly in simulation. Hence contribution of the corresponding APs in that zone gets reduced. However, capacity of these prohibited APs are being utilized in other zones where they have influence which helps to maintain the overall network PCA as shown in Fig. 16(b) as well as throughput as mentioned in Fig. 16(c). Only the fairness of the network goes down as shown in Fig. 16(a) as the PCA/throughput of the shared zones with $n > 3$ APs go down.

In another model, we have considered inter-node interference. As 802.11 protocols follow CSMA/CA for acquiring channel, inter-node interference increases with the increase in number of contending stations per channel. If more than one station sends their packet at the same time slot then there will be collision and in effect packets need to be transmitted again. Let us assume that P_c is the probability of collision and τ is the probability that a station transmits successfully in a randomly chosen slot. We assume that size of contention window w is constant (assumed 32 for our experiment) and there is no exponential backoff as an effect of unsuccessful transmission. Let us also assume that number of the contending stations for same channel is C_n. From equation 5 of [41] and equation 8 of [42] we can express P_c and τ as follows

$$P_c = 1 - 2^{(C_n - 1) \times log(1-\tau)} \tag{20}$$

where

$$\tau = \frac{2}{w+1} \tag{21}$$

Overall throughput will be reduced with the increment of P_c. Fig. 16(d) shows the effect of inter-node interference on throughput and JFI on SAC, RSSI and LLF. Though performance in terms of throughput deteriorates in all association control protocols (can be compared with Fig. 14), reduction is low in SAC (25% reduction) compared to RSSI (around 50%) and LLF (around 50%). This dramatic resiliency of SAC can be understood from the fact that it maintains JFI much better which implies that it distributes the load in balanced manner among the APs. Hence the number of contending devices per channel is low which ensures less effect on network throughput.

Message Overhead: In all the previous results we have shown that SAC outperforms RSSI and LLF based algorithms. However, SAC has some extra message overhead which is the additional cost incurred to achieve aforesaid performance. Here we discuss in detail about the extra cost incurred in SAC protocol in terms of message overhead.

Let us first discuss briefly about all the wireless messages used in SAC protocol. Apart from standard wireless messages used in any association control protocol, SAC does not use any new wireless message. Similar to traditional system, every AP sends beacon message and in response, device interested for association sends probe request message as shown in Fig. 8. Other than beacon and probe request messages, there are probe response message, disassociation notification message and re-association request message. Fig. 17 shows the fraction of devices getting re-associated against load. Although, probe request message contains some extra bytes, the important part of overhead arises from the re-association induced by SAC protocol. But, at low level of load ($< 50\%$), even association without sufficient care leads to an optimal performance in terms of PCA and fairness. Hence, re-association has no effect in terms of performance. The re-association phenomenon quickly diminishes beyond the 200% load where both RSSI and SAC have similar performance.

Inset of Fig. 17 shows the wired overhead of our proposed algorithm. For every probe request message, there is one share bit-rate message to communicate the details to zone process. So the number of share-bit rate message and the probe response message is in the order of number of device. Moreover, there is bandwidth request message and bandwidth response message. Both are in the order of the number of zones. Similarly, pushed bandwidth message and pushed acknowledgement message are also in order of number of zones. However, these messages are pure overhead for our association control algorithm. But as they are wired message (out of band communication) these messages do not incur much costs. Among these different wired messages, share-bit rate message is the dominating as it varies with number of devices. As the dominating factor of wired overhead varies proportionally with load, the overhead has increased with almost constant gradient with respect to load.

VI. Conclusion and Future Work

This paper combines results from traditional graph theory with the emerging opportunities in wireless setting to target the pressing problem of association control. SAC improves client association techniques in wireless networks by exploiting the wired backbone among WiFi APs. The key idea is to share local information from multiple APs, model it as a max-flow problem, and derive the optimal client-to-AP assignment. Simulation results demonstrate that such a technique can

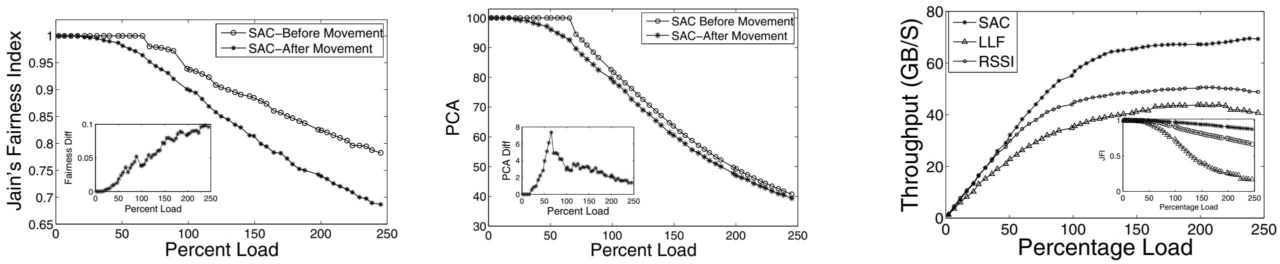


Fig. 15. Effect on performance due to movement. Fig. 15(a): Effect on fairness (JFI) due to movement on SAC and inset is the difference of JFI due to movement. Fig. 15(b): Effect on PCA due to movement using SAC and inset is the difference of PCA due to movement on SAC.Fig. 15(c): Effect of flash crowd on throughput. Inset shows the effect of flash crowd on JFI.

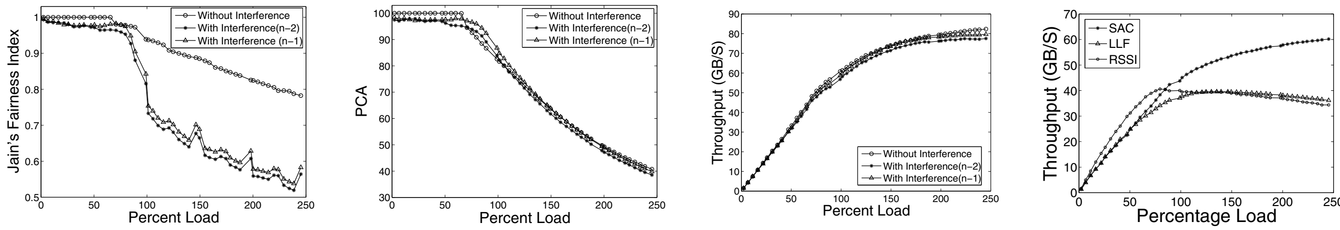


Fig. 16. Performance of proposed SAC protocol in presence of interference (using number of restricted AP=1 and 2). Fig. 16(a): Effect on fairness due to interference. Fig. 16(b): Effect on PCA due to interference and Fig. 16(c): Effect on overall throughput of the network.Fig. 16(d): Effect on throughput on different association control protocol due to inter-node interference.

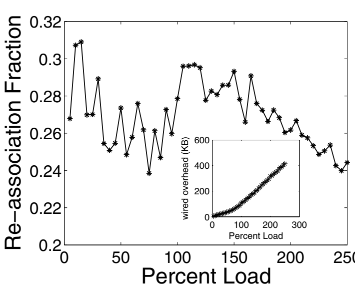


Fig. 17. Fraction of devices re-associate due to SAC protocol and inset is the wired messages overhead against load.

improve over purely distributed association schemes, resulting in higher fairness, better load balancing properties, and even some robustness to client mobility. Further the proposed algorithm also improves the overall throughput of the entire network and is also resilient to client mobility. The most important observation from the result is that our algorithm is better suited for higher load, in fact when the load is less than 75%, RSSI is a better choice. However, beyond 75% load SAC outperforms RSSI. Our future work is focused towards augmenting the system with sophisticated channel and traffic models and with support for multimedia traffic.

REFERENCES

[1] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: a mobility model for human walks," in *Proc. 2009 IEEE INFOCOM*, pp. 855–863.

[2] S. Gaito, G. Grossi, and F. Pedersini, "A two-level social mobility model for trace generation," in *Proc. 2008 ACM MobiHoc*, pp. 457–458.

[3] I. Papanikos and M. Logothetis, "A study on dynamic load balance for IEEE 802.11b wireless LAN," in *2001 COMCON*.

[4] "AP load balancing and AP fallback in unified wireless networks," Cisco Systems Inc., white paper, 2008.

[5] T. Iwanski, "Buyer's guide, 802.11b wireless devices," *WINDOWS 2000 MAGAZINE*, pp. 87–90, July 2001.

[6] A. Balachandran, P. Bahl, and G. M. Voelker, "Hot-spot congestion relief in public-area wireless networks," in *Proc. 2002 IEEE WMCSA*, p. 70.

[7] H. Gong and J. Kim, "Dynamic load balancing through association control of mobile users in WiFi networks," *IEEE Trans. Consumer Electron.*, vol. 54, no. 2, pp. 342–348, May 2008.

[8] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "CENTAUR: realizing the full potential of centralized WLANs through a hybrid data path," in *Proc. 2009 ACM MobiCom*, pp. 297–308.

[9] "http://en.wikipedia.org/wiki/wireless_access_point," *Wireless Access Point Wiki*.

[10] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[11] Y. Bejerano, S.-J. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," *IEEE/ACM Trans. Networking*, vol. 15, no. 3, pp. 560–573, June 2007.

[12] T. Bu, L. Li, Ramachandran, and Ramjee, "Generalized proportional fair scheduling in third generation wireless data networks," in *2006 IEEE INFOCOM*.

[13] A. P. Jardosh, K. Mittal, K. N. Ramach, E. M. Belding, and K. C. Almeroth, "IQU: practical queue-based user association management for WLANs," in *Proc. 2006 ACM MobiCom*, pp. 158–169.

[14] K. Son, S. Member, S. Chong, and G. D. Veciana, "Dynamic association for load balancing and interference avoidance in multi-cell networks," *IEEE Trans. Wireless Comm*, vol. 08, pp. 3566–3576, 2009.

[15] G. Li and H. Liu, "Downlink radio resource allocation for multi-cell OFDMA system," *IEEE TRans. Wireless Commun.*, vol. 5, no. 12, pp. 3451–3459, 2006.

[16] A. Jantaweetip and C. Prommak, "Bandwidth management for hybrid association control schemes and load balancing in WLANs," in *Proc. 2009 IEEE ECTI-CON*, pp. 1000–1003.

[17] M. Kim, Z. Liu, S. P. 0002, D. E. Pendarakis, and H. Yang, "Association control in mobile wireless networks," in *Proc. 2008 IEEE INFOCOM*, pp. 1256–1264.

[18] I. B. Dhaou, "A novel load-sharing algorithm for energy efficient MAC protocol compliant with 802.11 WLAN," in *Proc. 1999 IEEE VTC*, pp. 1238–1242.

[19] Y. Bejerano, S. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," in *Proc. 2004 ACM MobiCom*, pp. 315–329.

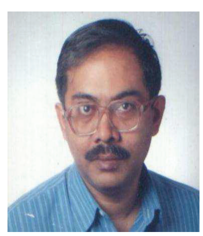
[20] D. Sarkar and U. K. Sarkar, "Balancing load of APs by concurrent

association of every wireless node with many APs," in *Proc. 2009 IEEE ICNS*, pp. 320–325.
[21] P. V. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in wireless LANs: algorithms and evaluation," *IEEE Trans. Mobile Computing*, vol. 6, pp. 164–178, Feb. 2007.
[22] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless LANs," in *2006 IEEE Infocom*.
[23] A. Mishra, V. Shrivastava, D. Agarwal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. 2006 ACM Mobicom*, pp. 170–181.
[24] H. Velayos, V. Aleo, and G. Karlsson, "Load balancing in overlapping wireless LAN cells," in *Proc. 2004 IEEE ICC*, pp. 3833–3836.
[25] S. K. Dandapat, B. Mitra, R. Roychoudhury, and N. Ganguly, "Fair bandwidth allocation in wireless network using max-flow," in *Proc. 2010 ACM SIGCOMM*, pp. 407–408.
[26] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation," technical report, Digital Equipment Corporation, Sep. 1984.
[27] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT Press, 1990.
[28] J. Jin, W.-H. Wang, and M. Palaniswami, "Utility max-min fair resource allocation for communication networks with multipath routing," *Comput. Commun.*, vol. 32, no. 17, pp. 1802–1809, Nov. 2009.
[29] E. Korach, S. Kutten, and S. Moran, "A modular technique for the design of efficient distributed leader finding algorithms," *ACM Trans. Program. Lang. Syst.*, vol. 12, no. 1, pp. 84–101, 1990.
[30] T. L. Pham, M. Bui, I. Lavallee, and S. H. Do, "An adaptive distributed algorithm for the maximum flow problem in the underlying asynchronous network," in *Proc. 2006 IEEE RIVF*, pp. 187–194.
[31] T. L. Pham, I. Lavallee, M. Bui, and S. H. Do, "A distributed algorithm for the maximum flow problem," in *Proc. 2005 IEEE ISPDC*, pp. 131–138.
[32] S. Sahni, "Computationally related problems," *SIAM J. Computing*, vol. 3, no. 4, pp. 262–279, 1974.
[33] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optimization*, 1991.
[34] E. Boros, P. L. Hammer, R. Sun, and G. Tavares, "A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (qubo)," *Discrete Optimization*, vol. 5, no. 2, pp. 501–529, 2008.
[35] D. Wang and R. Kleinberg, "Communication: analyzing quadratic unconstrained binary optimization problems via multicommodity flows," *Discrete Appl. Math.*, vol. 157, no. 18, pp. 3746–3753, 2009.
[36] H.-X. Huang, P. M. Pardalos, and O. A. Prokopyev, "Lower bound improvement and forcing rule for quadratic binary programming," *Comput. Optim. Appl.*, vol. 33, no. 2-3, pp. 187–208, 2006.
[37] "http://www.iitkgp.ac.in/map/mapfull.html," IIT campus map.
[38] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, June 2002.
[39] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," 2003.
[40] R. Musaloiu-elefteri and A. Terzis, "Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks," *International J. Sensor Networks*, vol. 3, pp. 43–54, 2008.
[41] I. Tinnirello and G. Bianchi, "Interference estimation in IEEE 802.11 networks," *IEEE Control Systems Mag.*, vol. 30, no. 2, pp. 30–43, Apr. 2010.
[42] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

Sourav Kumar Dandapat is currently pursuing his Ph.D. in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. He received his B.E degree from Jadvapur University, West Bengal, India in 2002 and M.Tech degree from Indian Institute of Technology Kharagpur in 2005 both in Computer Science and Engineering. He also worked as a Software Engineer in IBM ISL Bangalore during 2005-2007 and Magma Design Automation (India) Pvt. Ltd. during 2007-2009.His current research interest includes mobile wireless networks, peer-to-peer networks etc.

Bivas Mitra is currently working as a Postdoctoral Researcher at French National Centre for Scientific Research (CNRS), Paris, France. He received a Ph.D. in 2011 and a M.Tech in 2003 from Indian Institute of Technology Kharagpur after completing his B.Tech(Hons) from Haldia Institute of Technology in 2001 all in Computer Science and Engineering. During 2003 to 2006, he worked as a lecturer in the Department of Computer Science and Engineering at Haldia Institute of Technology. His research interests include complex networks, peer-to-peer networks, wireless Internet, optical networks etc.

Romit Roy Choudhury is an Associate Professor of ECE and CS at Duke University. He joined Duke in Fall 2006, after completing his Ph.D. from UIUC. His research interests are in wireless protocol design mainly at the PHY/MAC layer, and in distributed mobile computing at the application layer. He received the NSF CAREER Award in January 2008. Visit Romit's Systems Networking Research Group (SyNRG), at http://synrg.ee.duke.edu.



Niloy Ganguly is an Associate Professor in the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur. He has received his B.Tech from IIT Kharagpur in 1992 and his PhD from BESU, Kolkata, India in 2004. He has spent two years as Post-Doctoral Fellow in Technical University, Dresden, Germany before joining IIT, Kharagpur in 2005. His research interests are in peer-to-peer networks, on-line social networks and wireless networks. He is presently leading a research group comprising of several research scholars and masters student as well as various collaborators from industry and academia.