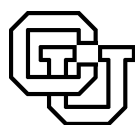


Wireless Networking Optimizations for Studying Algorithm Interactions

Gary Yee, Caleb Phillips, Dirk Grunwald, and Douglas Sicker

CU-CS-1070-10

July 2010



University of Colorado at Boulder

Technical Report CU-CS-1070-10
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Wireless Networking Optimizations for Studying Algorithm Interactions

Gary Yee, Caleb Phillips, Dirk Grunwald, and Douglas Sicker

July 2010

Abstract

Wireless networking optimizations are typically designed and evaluated independently of one another under the assumption that they can later be applied jointly. These works, however, do not provide sufficient information for future researchers and network administrators to determine which algorithms to combine, what order to combine them in, and how these joint optimizations will perform with one another.

In this paper we describe the optimization algorithms and testing system used to address these challenges. The algorithms fall into five categories: channel assignment, association control, beam form selection, transmit power control and rate adaptation. These algorithms have been specifically designed and chosen to make use of the observed inputs from the network. While none of these algorithms has been individually proven to be an optimal or efficient solution, they are representative of many *categories* of solutions. For example, some algorithms stress load balancing, while other algorithms stress increasing RSS values. Thus, these algorithms should be sufficient for revealing significant themes in the interactions among different channel assignment, association control, beam form selection, transmit power control and rate adaptation algorithms.

1 Introduction

There are a wide range of tunable parameters available to an administrator to optimize a wireless network. This set often includes channel assignment, association control between the stations and APs, transmit power control, bit-rate adaptation and more recently, beam forming / directionality. The common practice in the design and testing of wireless optimization algorithms is to do so across an explicit combination of environmental, traffic rate, network size, and topological contexts in order to show a given algorithm's robustness or scalability in comparison to one or more other algorithms from the same problem domain. This, however, provides only a limited understanding to a network administrator because the implicit optimization context (i.e., the configuration of the optimization algorithms assumed *a priori*) is usually a set of defaults that are not varied.

Since optimization algorithms provide solutions that inherently change the landscape of the network, they ultimately influence the inputs and thus the performance of any succeeding algorithms. This is especially problematic when multiple algorithms are evaluated under the assumption that their proposed optimization is the last one to execute. By understanding how algorithms interact with one another, we can not only draw stronger conclusions about their expected performance, but also discover real world flaws resulting from combinatorial / ordering assumptions made about an algorithm.

In this paper, we present a diverse set of wireless optimization strategies over five optimization domains, including channel assignment, association control, beam form selection, transmit power control, and bit-rate adaptation. We then propose a system for combining these algorithms, called COIN (Cooperatively Optimized Infrastructure Network).

2 Related Work

There exists a large body of research that optimizes channel assignment, association control, transmit power control, bit-rate adaptation and beam forming in an 802.11 WLAN. Many of these algorithm designs are approached independently due to the fact that many of these problems are NP-hard and designing a monolithic algorithm to address all of these problem domains would be increasingly difficult.

Many attempts have also been made at developing joint optimization algorithms across pairs of the aforementioned problem domains. In [1, 2, 3, 4, 5, 6, 7, 8, 9], the authors attempt to address the problems of transmit power control and bit-rate adaptation. In [10, 11, 12] the authors look at transmit power control and beam form selection. In [13, 14, 15] the authors look at channel assignment and association control algorithms. In [16, 17, 18] the authors design algorithms that solve association control and transmit power control. Das et al. look at channel assignment and beam form selection in [19]. Joint channel assignment and transmit power control are addressed in [20, 21].

Fewer works address three or more of the aforementioned problem domains. These include [22, 23], which focus on channel assignment, association control and transmit power control.

In 2007, Broustis et al. became the first to observe the problems with combining independently designed algorithms in [24]. In this work, the combined effectiveness of three optimization algorithms (one for channel assignment, association control and transmit power control algorithm, respectively) were studied by either enabling them or disabling them (i.e., allowing the default algorithm to run) in a series with a single ordering. The authors showed that applying all three optimization algorithms did not provide better throughput than applying a specific subset of their optimizations, showing that algorithms can negatively impact other algorithms.

3 Channel Assignment (C) Algorithms

There are effectively two channel assignment algorithms explored: one which is completely AP-centric, and another that is network-centric. Note that even though by default there is no channel assignment, the case where channel assignment is disabled is not considered due to previous work showing that channel assignment should always be enabled when possible.

3.1 Greedy Channel Assignment Algorithm (C.Greedy)

This channel assignment algorithm attempts to minimize the amount of interference between each set of APs. The network is first mapped into a complete directed graph, where nodes are APs and edges exist between all APs. Each edge that connects two APs α and β has a weight associated with it, $w_{\alpha,\beta}$, which is a function of the signal strength from β observed at α , $RSS_{\alpha,\beta}$ and the amount of traffic output by β , D_β .

Each $\alpha \in AP$ solves for the total interference it would experience on channel λ_α (denoted as $I(\alpha)$) by summing the weight of each incident edge as follows:

$$I(\alpha) = \sum_{\beta \in AP} w_{\alpha,\beta} * \text{Overlap}(\lambda_\beta, \lambda_\alpha) \quad (1)$$

$\text{Overlap}(\lambda_\beta, \lambda_\alpha)$ is a function that weights the negative influence of channel λ_β on channel λ_α . It is assumed that if $\lambda_\alpha = \lambda_\beta$ then $\text{Overlap}(\lambda_\beta, \lambda_\alpha) = 1$, and is otherwise 0.

Assuming all nodes start on channel 1 and only three channels exist, a solution is found using a greedy approach as follows.

Algorithm 1 AP-Centric channel assignment algorithm

```

1:  $AP \leftarrow$  a list of APs sorted by the total interference values of the current topology.
2:  $\Lambda \leftarrow$  The set of all channels.
3: for all  $\alpha \in AP$  do
4:    $\text{minInterference} = I(\alpha)$ 
5:    $\text{bestChannel} = \lambda_\alpha$ 
6:   for all  $\lambda \in \Lambda$  do
7:      $\lambda_\alpha \leftarrow \lambda$ 
8:     if  $\text{minInterference} < I(\alpha)$  then
9:        $\text{bestChannel} = \lambda_\alpha$ 
10:     $\text{minInterference} = I(\alpha)$ 
11:   end if
12:    $\lambda_\alpha = \text{bestChannel}$ 
13: end for
14: end for

```

This process can be repeated any number of times until a suitable solution is found.

3.2 HSum Algorithm (C.Hsum)

The Hsum algorithm developed by Mishra et al. differs from the previous channel assignment algorithm in that it utilizes signal strength measurements from stations and knowledge of station-AP associations to determine a better channel assignment. For a detailed description of the Hsum algorithm, see [25].

4 Association Control (A) Algorithms

Association control can attempt to optimize a wide range of attributes. The following algorithms attempt to arrange STAs such that RSS is maximized, interference is minimized, load balancing is encouraged, or hidden terminals are removed.

4.1 Default: Highest Beacon RSSI (A.None)

Stations scan each channel and observe the RSSI values that come from AP beacons. They then select the AP that has the highest RSSI value.

4.2 Quadratic Program (A.QP2)

The algorithm used to determine the APs that each station is associated with is formulated as a quadratic program. Each AP and station can be thought of as vertices in a complete directed graph with edges that are weighted by signal strength and the traffic load. Channel utilization time is the cost of a given solution and is thus minimized by the quadratic program given below:

- Objective:

$$\begin{aligned} \min f(x) &= \frac{1}{2}x^T Q x \\ Q &: \text{Our link-pair cost matrix} \\ x &: \text{The 'active link' binary solution vector} \end{aligned}$$

- Constraints:

$$\begin{aligned} Ax &\leq b \\ A &: A[i, j] = 1 \text{ if STA } i \text{ and link } j \text{ have same STA, else } 0 \\ b &= 1 \end{aligned}$$

- Definitions:

$$\begin{aligned} \alpha &: \text{Link } \alpha \\ Data_\alpha &: \text{the load generated by } \alpha \\ Interference_{\alpha, \beta} &: \text{the interference between the four endpoints of } \alpha \text{ and } \beta \\ RSS_{\alpha, \beta} &: \text{RSS of } \alpha + \text{RSS of } \beta \\ Penalty_{\alpha, \beta} &= Data_\alpha * Interference_{\alpha, \beta} \\ Utility_{\alpha, \beta} &= Data_\alpha * RSS_{\alpha, \beta} \\ w_{LB} &: \text{weights the utility of two STAs that share AP.} \\ w_{IA} &: \text{weights the penalty of RSS from interferers} \\ w_P &: w_{LB} \text{ if two STAs share the same AP, else } w_{IA} \\ w_U &: \text{weights the utility of maximizing RSS of active links} \\ Q_{\alpha, \beta} &= Penalty_{\alpha, \beta} * w_P + Utility_{\alpha, \beta} * w_U \end{aligned}$$

The outcome of the association control algorithm can be determined by adjusting the weights w_{LB} , w_{IA} and w_U . By increasing w_{LB} , links that share the same AP will be additionally penalized, causing STAs to associate with an even distribution. Increasing w_{IA} will emphasize *interference avoidance*. Finally, changing w_U will result in AP-STA links that have higher RSS values.

In this work, four combinations of weights are used to stress different association control strategies. A summary of these weight combinations is given in table 1. The first two algorithms do not perform load balancing across APs and differ between whether they stress interference avoidance or not. The final two algorithms test the same weight combinations with an added weight for load balancing across APs.

Name	Load Balance Across APs	Interference Avoidance	Maximize RSS
A.QP2:1.0e20,1.0e00,1.0e20			X
A.QP2:1.0e20,1.0e20,1.0e20		X	X
A.QP2:1.0e00,1.0e00,1.0e20	X		X
A.QP2:1.0e00,1.0e20,1.0e20	X	X	X

Table 1: Four association control strategies based off the quadratic program.

4.3 Minimize Hidden Terminals (A.MinHT)

MinHT leverages association control to reduce the hidden terminals effect by assigning STAs to APs on different channels.

The **hidden terminal set** of a single node is defined as the set of links that node N associates with and imparts some hidden terminal effect. First the set of nodes that N cannot hear, N_d , is derived. Then the set of nodes that N 's associated destination (an AP if N is an STA and all STAs if N is an AP) can hear is derived and designated as N_s . The total set of nodes that N is a hidden terminal of is therefore the intersection of N_d and N_s .

Predicting the likelihood of negative effects caused by hidden terminals is a difficult problem. In the case that all hidden terminals are independent, one can simply assume that transmissions are independent as well and evaluate the amount of generated interference at the destination as the product of the loads from each transmitter. If, however, there exist nodes that can hear two hidden-terminal causing transmitters, the probability of a transmission is no longer independent due to CSMA effects.

For simplicity, this metric pessimistically assumes that the transmissions of the nodes causing the hidden terminal are independent. By aggregating the hidden terminal effect of each node, we can get the hidden terminal effect of the entire network.

This algorithm iteratively assigns each node to different APs in an attempt to minimize the current lowest value for the network's hidden terminal effect, and only keeps the assignment that brings about the best change. The only constraint is that each node must maintain an RSS value above a certain threshold with each of its destinations (see section 6.2 for the specific thresholds). This process continues until the network's hidden terminal effect can no longer be reduced. MinHT will also attempt to perform some load balancing across APs if it is possible to lower the standard deviation of STAs assigned to APs without increasing the hidden terminal effect or assigning a STA to a link with worse RSS.

5 Beam Form Selection (B) Algorithms

In this work we assume that we have a set of 17 beam patterns available to us. The default beam pattern is the standard omnidirectional beam. The next 16 beam patterns are all modelled from measurements of the cophasal patterns provided by Fidelity Comtech. These patterns attempt to provide the tightest beam-width (42 degrees) for the main lobe of an 8 element circular array antenna along different azimuths. The azimuths differ in increments of 22.5 degrees.

5.1 Default: Omnidirectionality (B.Omni)

By default all APs will select an omni-directional beam form. This represents the case of most currently deployed networks.

5.2 Linear Program (B.LP)

The beam form selection model is formulated as an integer linear program and is intended to find an approximately optimal solution for each AP. The cost of selecting a given beam pattern for a given AP is defined as a function of the signal strength of interference sources (i.e., other APs or non-associated STAs) and the signal strength from STAs that are associated.

- Objective:

$$\begin{aligned} \min f(x) &= c^T x \\ c &: \text{Our cost vector for each (AP, beam) configuration.} \\ x &: \text{A binary solution vector for all (AP, beam) combinations.} \end{aligned}$$

- Constraints:

$$\begin{aligned} Ax &\leq b \\ A &: A[i, j] = 1 \text{ if AP } i \text{ and the AP in configuration } j \text{ are the same, else } 0. \\ b &= 1 \end{aligned}$$

- Definitions:

$$\begin{aligned} AP_\alpha &: \text{access point } \alpha \\ P_\alpha &: \text{set of all beam patterns for AP } \alpha \\ STA_\alpha &: \text{set of all STAs associated with AP } \alpha \\ Utility_{\alpha,p} &: \sum_{s \in STA_\alpha} (RSS_{dBm}(s, p)) \\ AP.STA_{\alpha,p} &: \sum_{\beta \in AP: \beta \neq \alpha, s \in STA_\beta} (RSS_{dBm}(s, p)) \\ AP.AP_{\alpha,p} &: \sum_{\beta \in AP, p_\beta \in P_\beta: \beta \neq \alpha} MAX(RSS_{dBm}(\beta, p_\beta, \alpha, p)) \\ w_{AA} &: \text{weight of an AP's interference on another AP} \\ w_{AS} &: \text{weight of interference on an AP from all non-associated STAs} \\ w_{BU} &: \text{weight of RSS from all STAs associated with an AP} \\ c_{\alpha,p} &= (AP.AP_{\alpha,p} * w_{AA} + AP.STA_{\alpha,p} * w_{AS}) - Utility_{\alpha,p} * w_{BU} \end{aligned}$$

Solutions that stress interference avoidance can be attained by weighting w_{AA} and w_{AS} more heavily. For example, given an AP with a sectored antenna in which each pattern has one main lobe, emphasizing interference avoidance will cause each AP to point away from other APs and stations outside its own network. The opposing weight, w_{BU} , can be used to encourage solutions that maximize signal strength without consideration for noise or interference. In the

previous example this would cause APs to point directly at their associated stations with their main lobe.

In this work, three combinations of weights are used to stress different beam form selection strategies. A summary of these weight combinations is given in table 2. At a high level, the first algorithm leads APs to point at their STAs, the second algorithm leads APs to point away from non-associated nodes and the third algorithm attempts to combine these two strategies.

Name	Maximize RSS	Interference Avoidance to APs	Interference Avoidance to STAs
B.LP:1.0e20,1.0e00,1.0e00	X		
B.LP:1.0e00,1.0e20,1.0e20		X	X
B.LP:1.0e20,1.0e20,1.0e20	X	X	X

Table 2: Three beam form selection strategies based off the linear program.

6 Transmit Power Control (T) Algorithms

This work utilizes three simple transmit power control schemes: maximum transmit power, minimize hidden terminals, and minimize transmit power.

6.1 Default: Maximum Transmit Power (T.MaxPow)

Transmit power is fixed to the highest. For simulations this was set at 16 dBm.

6.2 Minimize Transmit Power (T.MinPow)

For each node this algorithm selects the lowest transmit power such that the expected SNR value (assuming gaussian white noise at -100.97 dBm) is above a threshold to its intended endpoint for a given bit-rate. For APs the intended endpoints are all of its STAs and for STAs the intended endpoint is its associated AP. These thresholds represent the SNR values needed by the simulator to obtain a 100% packet reception rate for a 1500 byte packet. Less conservative thresholds were briefly examined, but ultimately not used after they appeared to consistently perform worse. The thresholds for each bit-rate are given in table 3.

Bit-Rate	SNR Threshold (dBm)
1 Mbps	-79.270
2 Mbps	-74.270
5.5 Mbps	-74.870
11 Mbps	-71.570

Table 3: SNR thresholds to define a 'Strong Link' for each bit-rate

7 Bit-Rate Adaptation (R) Algorithms

This section describes the two rate adaptation algorithms tested.

7.1 Default: Maximum Bit-Rate (R.MaxRate)

The highest bit-rate is selected for each node, which in the case of an 802.11b network is 11 Mbps.

7.2 RSSI-Based Rate Adaptation (R.RSSI)

Given a bit-rate, an SNR value, and a frame size, and the expected bit-error rate, this algorithm determines the average amount of time (including retransmissions) needed to transmit a message. The expected bit-error rate used was derived from the simulator’s model. Thus, the best bit-rate is calculated for four different frame sizes.

8 Algorithm Inputs

Each of the above algorithms requires detailed knowledge about the network. Given the influence that the environment can have in a practical usage scenario, explicitly measuring inputs for various wireless networking optimization algorithms is a popular practice. There are some limitations to this approach, however, as the time and resources consumed by network measurements are finite and may interfere with actual data transmissions.

Below is a list of the metrics needed by the aforementioned algorithms:

1. the RSS of all (APs, beam patterns) combinations
2. the traffic transmitted in both directions
3. the RSS of a STA from an AP on a given beam pattern (and vice versa)
4. the RSS between all STAs

Metric 1 can potentially be time consuming, given the potentially large number of beam patterns and available APs. Assuming that there are P total patterns per AP and A total APs, transmitting and receiving between each pair of nodes one at a time could take up to $O(A^2 \cdot P^2)$ measurements. However, this can be cut down to $O(A \cdot P^2)$ because multiple APs can receive the same transmission. This “NxN scan” process is described in [26] and rarely needs to be updated, given the static locations of the APs.

Metrics 2, 3 and 4 are expected to be updated more frequently. Metric 2 requires that nodes periodically send the amount of traffic that was transmitted at the MAC layer to each host. Metrics 3 and 4 requires a minor change to typical AP behavior since APs and stations that lie on separate channels may be unable to hear one another when sending normal data. This is addressed by channel sounding, which is further discussed in section 9.

9 A COIN Optimization Stage

An optimization algorithm gets run once per optimization stage, which requires several steps needed to collect network metrics, distribute the solution, and allow the solution to reach a steady state. These steps include: **the NxN Scan, the Data Phase, the Assembly Phase, the Update Phase, the Algorithm Phase, the Commands Phase, and the Reconfiguration Phase**. This section describes each of these steps that make up a single optimization stage, shown in figure 1.

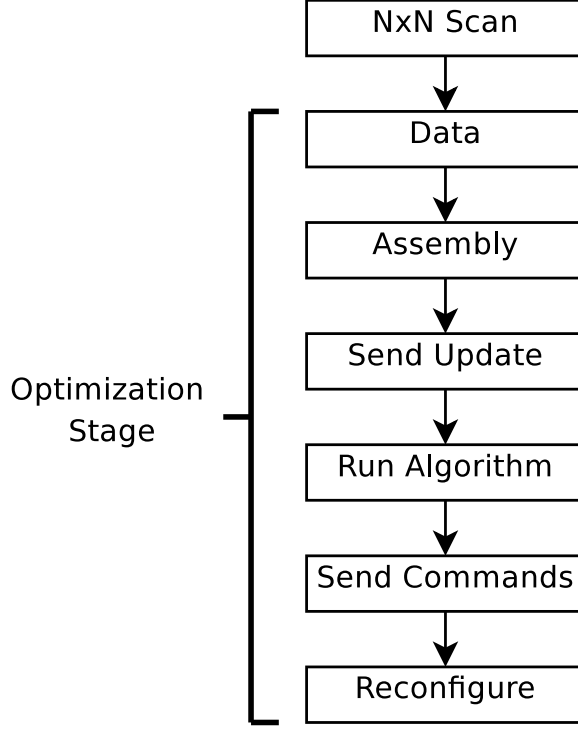


Figure 1: A single COIN optimization stage

9.1 NxN Scan

The NxN scan is responsible for measuring the signal strength of all possible links between the phased-array antenna APs. Given that a link is not only defined by the given end points but also the beam patterns selected, this measurement is challenging because the search space grows quickly.

In a network with N phased array antennas capable of using P beam patterns, there are P^N possible combinations. Multiple combinations can be tested in parallel, however, because multiple receivers can hear the same transmission. Thus the total number of measurement trials that need to be run is $P^2 * N$.

This approach has been shown to be the current state of the art in testing all possible beam pattern combinations between a series of nodes in [26]. While the time to conduct these measurements is expensive (on the order of one to two minutes for 3-4 nodes), in this work it is assumed that the environment between the APs is static and thus the NxN scan can be performed rarely or offline.

9.2 Data

In the Data phase, the network is allowed to operate normally, with APs and stations following the standard 802.11 protocol for network management and transmitting data packets. The goal of the Data phase is to take measurements from a representative sample of network behavior, given the current node configuration. Two metrics are collected in this phase: received signal

strength of every frame received at the MAC layer, the number of transmitted frames, and the size of each transmitted frame.

9.3 Assembly

The Assembly phase is a time scheduled event in which all nodes assemble on the same channel and take turns transmitting small beacon-like frames at the lowest bit-rate and the highest transmit power; this allows the rest of the network to collect more consistent signal strength measurements that are then used as optimization algorithm inputs.

The Assembly phase process is similar to an NxN scan with the exception being that all receiving is done in an omni-directional state. All access points transmit a handful of beacon-like frames in each of its beam patterns, while the stations transmit in an omni-directional state.

A full NxN scan, while more accurate, is not done due to the assumption that the network will have a far greater number of stations than access points, leading to a significantly greater amount of measurement time. Given A access points with P_A patterns and S stations with P_S patterns, a full NxN scan would take $O(MAX(P_A^2, P_S^2) * (A + S))$ transmissions. The assembly phase, on the other hand, would only take $O(P_A * A + P_S * S)$.

9.4 Send Update

During the Send Update phase COIN-Update messages are sent from all stations and APs to the Admin. At a high level, these messages indicate the current configuration of each node in the network and the observed noise, RSS, and transmitted bytes values for each link from the previous Data and Calibration phases.

All COIN messages initially begin with a simple preamble, shown in figure 3. These preambles allow COIN nodes to determine the context of the following header.

The header of a 19 byte long COIN-Update message is shown in figure 4. The source's 802.11 MAC address is followed by the node's current channel, beam pattern, transmit power, and bit-rate algorithm, which each take one byte. One bit is then used to indicate whether this node is an AP or an STA, followed by 7 bits to indicate how many beam patterns this node is capable of forming. Next is a 1 byte noise floor measurement that is then followed by a 1 byte count of how many other nodes are associated to this one (if it's an AP), and a 2 byte count of how many neighbors have been detected. These two fields indicate the number of COIN-Assoc and COIN-Link headers that follow the COIN-Update header.

COIN-Assoc headers provide a list of nodes with which the message creator is associated. As seen in figure 5, this list consists of the associated node's MAC address and a flag indicating whether the list continues.

COIN-Link headers (shown in figure 6) provide link specific metrics and configuration settings. Following the observed node's MAC address, the packet gives an average of the RSS from that node. This is followed by four Tx Byte fields, which give the number of bytes transmitted for frames of a given upper-bound size (160, 330, 660, or more than 660 bytes). Next, there are four 1 byte fields indicating the bit-rate currently being used for each of these bucket sizes. The last field in a COIN-Link header indicates if a COIN-Pattern-RSS header immediately follows. These headers, shown in figure 7, give the average RSS of signals from different beam patterns that are measured during the Calibration phase.

Finally there is a 4-byte sequence number field that allows the administrator to know the age of the message contents. The process of creating a COIN-Update message is shown graphically in figure 2.

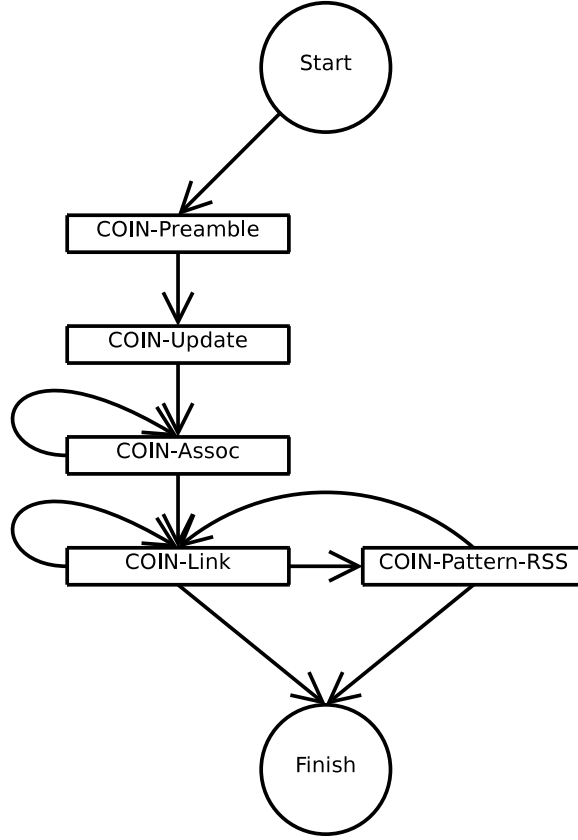


Figure 2: COIN-Update packet formation

Since all nodes attempt to transmit COIN-Update messages in an uncoordinated fashion, there is a good chance that nodes may interfere with one another during transmission. To improve the chances of a COIN-Update message being received, the COIN protocol supports random backoffs and retries if a COIN-Update-ACK (see figure 8) is not received in a short period of time. Only after four retries does the sender stop resending.



Figure 3: COIN-Preamble header

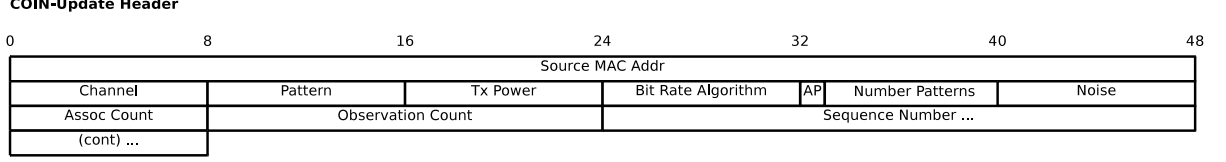


Figure 4: COIN-Update header

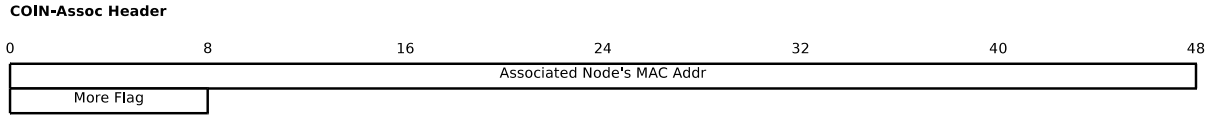


Figure 5: COIN-Assoc header

9.5 Run Algorithm

Using the network metrics from the previous phase, the admin will run an optimization algorithm and find a configuration for each wireless node.

9.6 Send Commands

COIN-Command messages tell each node its new configuration and any link-specific configuration necessary to execute the optimized solution. After the COIN-Preamble (see figure 3), the COIN-Command header is given. As shown in figure 9, the admin tells the intended receiver its new channel, associated AP (not used for nodes that are APs), beam pattern, transmit power, and bit-rate algorithm. Next is a 4 byte deadline timestamp that indicates when the network should reconfigure and a 4 byte sequence number. The last field is a flag that indicates whether there is a list of COIN-Command-On-Link headers (see figure 10). These headers give the bit-rates that are to be used for frames of different bucket sizes for each link, but are optional. A graphical representation of how COIN-Command messages are built is shown in figure 12.

COIN-Command messages are sent using a stop-and-wait from the admin to each wireless node in the network; after each command is sent the admin waits for a COIN-Command-ACK and retries after a short timeout. COIN-Command messages are thus sent in a contention-free manner.

9.7 Reconfigure

Each wireless node that receives a COIN-Command message waits until the “Reconfiguration Deadline”. At that time, all nodes in the network attempt to make the recommended changes.

COIN-Link Header

8		16		24		32		40		48	
Observed Node's MAC Addr											
Avg RSS		Tx Bytes, Bucket 0								Tx Bytes, Bucket 1 ...	
(cont) ...						Tx Bytes, Size Bucket 2 ...					
(cont) ...				Tx Bytes, Size Bucket 3							
Bit-Rate, Bucket 0		Bit-Rate, Bucket 1		Bit-Rate, Bucket 2		Bit-Rate, Bucket 3		Has Pattern-RSS Flag			

Figure 6: COIN-Link header**COIN-Pattern-RSS Header**

0	8	16	24	32	40	48
RSS Pattern 0		RSS Pattern 1		RSS Pattern 2		RSS Pattern 3
RSS Pattern 4		RSS Pattern 5		RSS Pattern 6		RSS Pattern 7
RSS Pattern 8		RSS Pattern 9		RSS Pattern 10		RSS Pattern 11
RSS Pattern 12		RSS Pattern 13		RSS Pattern 14		RSS Pattern 15
						RSS Pattern 16

Figure 7: COIN-Pattern-RSS header**COIN-Update-ACK Header**

0	8	16	24	32	40	48
Sequence Number						

Figure 8: COIN-Update-ACK header**COIN-Command Header**

8		16		24		32		40		48	
Channel		AP MAC Address ...									
(cont) ...		Pattern		Tx Power		Bit Rate Algorithm		Deadline ...			
(cont) ...		Sequence Number									
More Cmds-Link Flag											

Figure 9: COIN-Command header**COIN-Command-On-Links Header**

0	8	16	24	32	40	48
Associated Node's MAC Address						Bit-Rate 0
Bit-Rate 1		Bit-Rate 2		Bit-Rate 3		More Cmds-Link Flag

Figure 10: COIN-Command-On-Link header**COIN-Command-ACK Header**

0	8	16	24	32	40	48
Sequence Number						

Figure 11: COIN-Command-ACK header

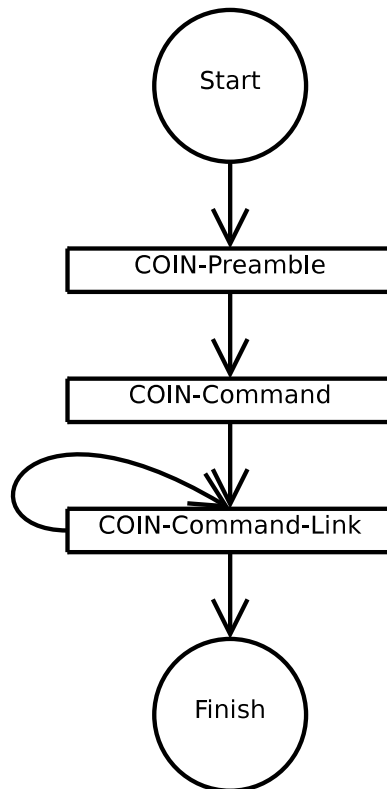


Figure 12: COIN-Command packet formation

Of all these, AP association is the only setting not guaranteed to take place if the COIN-Command message is successfully received. In this situation, the STA falls back to its default association algorithm and attempts to join with the AP that has the highest beacon RSS. After the reconfiguration stage, the nodes in the network will return to the Data phase.

10 Summary

This paper has described algorithms that can be used to study wireless optimization interactions. The scope of the algorithm classes covered channel assignment, association control, beam form selection, transmit power control, and rate adaptation.

This paper also provided a specification of the COIN system, which defines the protocols and architecture used to collect network measurements, execute optimization algorithms and reconfigure the network. Optimizations are carried out in a series of phases which define periods for normal usage, system calibration, optimization execution, and network reconfiguration. These phases as a whole define a single stage that is responsible for carrying out a single optimization algorithm. In this work, five optimization stages (one from each of the aforementioned algorithm classes) are studied.

References

- [1] E. Altman, A. Kumar, D. Kumar, and R. Venkatesh, "Cooperative and non-cooperative control in ieee 802.11 wlans," in *19th International Teletraffic Congress (ITC-19)*, 2005.
- [2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self-management in chaotic wireless deployments," in *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 185–199, ACM, 2005.
- [3] A. Grilo and M. Nunes, "Link-adaptation and transmit power control for unicast and multicast in ieee 802.11 a/h/e wlans," *Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on*, pp. 334–345, 20–24 Oct. 2003.
- [4] J. Jelitto, A. Barreto, and H. L. Truong, "Power and rate adaptation in ieee 802.11a wireless lans," *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 1, pp. 413–417 vol.1, 22–25 April 2003.
- [5] T.-S. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 366–377, ACM, 2006.
- [6] T. Mo and C. Bostian, "A throughput optimization and transmitter power saving algorithm for ieee 802.11b links," *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 1, pp. 57–62 Vol. 1, 13–17 March 2005.
- [7] D. Qiao, S. Choi, A. Jain, and K. G. Shin, "Miser: an optimal low-energy transmission strategy for ieee 802.11a/h," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 161–175, ACM, 2003.

- [8] D. Qiao, S. Choi, and K. G. Shin, "Interference analysis and transmit power control in ieee 802.11a/h wireless lans," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1007–1020, 2007.
- [9] K. Wang, F. Yang, Q. Zhang, D. Wu, and Y. Xu, "Distributed cooperative rate adaptation for energy efficiency in ieee 802.11-based multihop networks," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 2, pp. 888–898, March 2007.
- [10] N. Fahmy, T. Todd, and V. Kezys, "Ad hoc networks with smart antennas using ieee 802.11-based protocols," *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, pp. 3144–3148 vol.5, 2002.
- [11] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 95–105, ACM, 2001.
- [12] R. Ramanathan, "Mobile ad hoc networking." www.ir.bbn.com/~ramanath/pdf/wiley-bookchap.pdf, 2004.
- [13] H. Al-Rizzo, M. Haidar, R. Akl, and Y. Chan, "Enhanced channel assignment and load distribution in ieee 802.11 wlans," in *IEEE International Conference on Signal Processing and Communication*, 2007.
- [14] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1451–1459, May 2007.
- [15] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless lans," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, April 2006.
- [16] Y. Bejerano and S.-J. Han, "Cell breathing techniques for load balancing in wireless lans," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.
- [17] O. Brickley, S. Rea, and D. Pesch, "Load balancing for qos optimisation in wireless lans utilising advanced cell breathing techniques," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 3, pp. 2105–2109 Vol. 3, 30 May-1 June 2005.
- [18] M. Haidar, R. Akl, H. Al-Rizzo, Y. Chan, and R. Adada, "Optimal load distribution in large scale wlan networks utilizing a power management algorithm," *IEEE Sarnoff Symposium*, pp. 768–771, 2007.
- [19] S. Das, H. Pucha, D. Koutsonikolas, Y. Hu, and D. Peroulis, "Dmesh: Incorporating practical directional antennas in multichannel wireless mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2028–2039, Nov. 2006.
- [20] A. Lozano and D. Cox, "Integrated dynamic channel assignment and power control in tdma mobile wireless communication systems," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 11, pp. 2031–2040, Nov 1999.

- [21] L. Cong, E. Gunawan, B. H. Soong, and C. B. Soh, "Power control supported dynamic channel assignment in cellular radio systems," *International Journal of Communication Systems*, vol. 10, no. 3, 1997.
- [22] M. Haidar, R. Akl, H. Al-Rizzo, and Y. Chan, "Channel assignment and load distribution in a power-managed wlan," *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pp. 1–5, 3–7 Sept. 2007.
- [23] E. Garcia, L. Faixó, R. Vidal, and J. Paradells, "Inter-access point communications for distributed resource management in 802.11 networks," in *WMASH '06: Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots*, (New York, NY, USA), pp. 11–19, ACM, 2006.
- [24] I. Broustis, J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, "Implications of power control in wireless networks: A quantitative study," in *Passive and Active Network Measurement*, vol. 4427, pp. 83–93, Springer Berlin / Heidelberg, 2007.
- [25] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for wlans," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 3, pp. 19–31, 2005.
- [26] M. Buettner, E. Anderson, G. Yee, D. Saha, A. Sheth, D. Sicker, and D. Grunwald, "A phased array antenna testbed for evaluating directionality in wireless networks," in *MobiEval '07: Proceedings of the 1st international workshop on System evaluation for mobile platforms*, (New York, NY, USA), pp. 7–12, ACM, 2007.