WLAN Design: A Distributed Approach

Alan Mc Gibney, Martin Klepal, Dirk Pesch
The Centre for Adaptive Wireless Systems
Cork Institute of Technology
Cork, Ireland
Alan.McGibney@cit.ie, Martin.Klepal@cit.ie, Dirk.Pesch@cit.ie

Abstract—The complex nature of Wireless Local Area Network (WLAN) design, especially when designing large scale networks underlines the need for an automatic WLAN planning tool. The current approach to WLAN design is ad hoc and can lead to an adverse affect on network and service quality. There has been significant research into optimisation techniques and planning tools, many of which use a centralised optimisation approach that is not scalable especially when designing large scale WLAN. The research presented in this paper proposes an approach to WLAN design that is fully distributed based on a scalable optimisation technique. The proposed approach uses elements of Artificial Intelligence and Game Theory to design a viable WLAN regardless of the environment size.

Keywords: WLAN Design; Optimisation; Planning; Game Theory;

I. INTRODUCTION

The rapid roll out of IEEE802.11 based Wireless Local Area Networks (WLANs) has led to ad-hoc deployments without the support of appropriate planning tools. Many WLANs are set out by placing access points in the immediate vicinity of where coverage is required. The performance of such a network design can be significantly reduced compared to what can be achieved with the aid of a software design tool. Designing a reliable and cost effective WLAN is a complex task that requires the designer to consider factors such as environment configuration, technology, user requirements, signal coverage, co-channel interference and capacity. To aid the designer in this difficult task there is a need to have a planning tool that can automatically optimise access point positions within a user defined environment so that a viable design can be achieved.

There has been extensive research in the area of WLAN planning [1,2,3,4,5,6], most of which concentrates on accurate propagation modelling for efficient network design. This approach leads to a WLAN providing sufficient coverage it may not however, provide the quality of service expected by its users. Consequently, relying solely on signal coverage will not lend itself towards an effective WLAN design [9]. The core of a WLAN planning tool must be an accurate propagation model but it also must incorporate an effective optimisation technique to achieve the best design possible. The application of a number of optimisation techniques to WLAN design were investigated in [2,3,6,7,9,10] including, Linear Programming Models, Direct Search Methods, Evolution Strategies, Genetic Algorithms, Simulated Annealing, Tabu Search and Random Walks.

The application of these techniques highlights the issue of scalability when optimising large-scale WLANs. For example when ES was used it performed well for the majority of design problems, but became unstable for a large environment, with a considerable number of access points. The focus of the research presented in this paper is the development of a distributed optimisation algorithm based on game theoretical tools to yield a scalable optimisation technique for use in WLAN planning.

The remainder of the paper is structured as follows: Section II discusses the problem of WLAN design and pre-processing. Section III describes the implementation based on game theory including the agent and utility design, Section IV presents results and Section V reviews the research findings.

II. PROBLEM DESCRIPTION

The goal of a WLAN planning tool is to facilitate proficient network design based on the user requirements, i.e. maximise resources by minimising the number of access points used whilst optimally positioning them to satisfy user quality of service expectations. It is therefore necessary for the designer to describe the environment and specify some constraints on the network design through pre-processing.

This pre-processing is done using a planning tool we developed which includes drawing features to allow the designer to specify the environment where the WLAN is to be deployed. The environment definition is essentially a skeleton description of the building structure, which allows the designer to define it quickly with minimal effort. The floor plan is used as an input for a propagation model that estimates the electromagnetic propagation throughout the environment. Another feature of this planning tool is the user demand wizard. This wizard automatically groups walls to create potential demand areas and allows the designer to explicitly specify user demands in those zones.

The user demand specifications consist of a signal level threshold, number of users and their usage requirements. The final part of the pre-processing is the generation of a candidate access point mesh; this mesh is generated automatically by the planning tool, using a neural gas algorithm described in [11]. The algorithm was adapted to ensure access points are distributed evenly throughout a complex environment while maintaining desired positions such as along walls and ceilings. As a result, the mesh not only has candidate access point’s positions but also each access point has a signal coverage map, which is generated using the propagation model, and a list of
neighbouring access points. Figure 1 shows a snapshot of the candidate access point mesh being generated.

![Figure 1 - Generation of Candidate Access Point Mesh](image)

With the pre-processing complete the planning tool then continues to optimise the number of access points and their appropriate positions to provide an optimal design. The formulation of the optimisation problem using game theory is described in the following section.

### III. GAME THEORETIC DESIGN

Game Theory was developed by mathematician John von Neumann and economist Oskar Morgenstern when they were investigating economic behaviour [12]. In more recent years, it has received much attention from computer scientists and engineers for applications relating to wireless communications and networking [13]. The two main tools provided by game theory that are applicable to our problem are Agent Design, used to analyse the decisions of players and Utility Theory used to evaluate success of an action, rules associated with the environment where the game is played will also be described.

It is proposed to use an iterative game that begins with a single agent; this agent has an initial strategy of maximising coverage throughout the user defined target zones. The agent may introduce other agents into the game to help with this goal. Once this is achieved each agent should then optimise its position in order to meet user demands on the target zone. Figure 2 shows the architecture of the proposed game. The critical element of the game shown in Figure 2 is the agent.

![Figure 2 - Game Architecture for WLAN Design](image)

#### A. Agent Design

An Agent is an entity that can act, the features that distinguish an agent from a mere “program” include acting independently, having the ability to perceive its environment, persisting for a long period of time, adapting to change and being capable of taking on the goals of other agents [14]. For the WLAN optimisation we designed a simple reflex, utility-based agent. The main components of an agent are environment percepts, condition-action rules and utility function. In the current implementation of the optimisation tool, the agent is driven by maximising signal coverage in order to validate the proposed approach, but will be extended to consider user demands on throughput.

1) **Agent Percepts**

Percepts describe an agent’s perceptual input at a given time. The reflex based agent perceives its current environment state without a need for a history of previous precepts. Figure 3 shows the agents percept in a simple environment.

![Figure 3 - Percepts of Agent P1 in sample environment](image)

Where,

- **PC** ................. Perspective Coverage Cell
- **EC** ................. Edge of Coverage Cell
- **AC** ................. Actual Coverage Cell
- **P1** ................. Position of Player 1
- **P2** ................. Position of Player 2

The agent is aware of its current position within the candidate access point mesh, as described in Section II each position has a signal coverage map this is described as Perspective Coverage Cell (PC). This is the maximum area that
an agent can successfully cover (by meeting a user defined threshold) while in the current position. When agents are close to each other it is assumed that the agent with the strongest signal level covers that area. As a result the agent is required to evaluate the Actual Coverage Cell \(AC\). \(AC \subseteq PC\). The last element that is important for the agent to perceive from the environment is the Edge of Coverage Cell \(EC\). The \(EC\) gives the agent a view of just beyond its own \(PC\) and is used as part of the condition-action rules to decide whether another agent is required to extend coverage within the target area.

2) Condition-Action Rules

Once the agent knows what the world looks like it can then use condition-action rules to establish a connection between agent percept and which action should be considered by the agent. Let \(G(N)\) be an iterative game, \(G(N)\) has a set of agents or players \(N\), who will make decisions. Each player has a set of actions \(A_n\) that the players can choose. The possible actions available to the players when maximising signal coverage are defined as \(A = \{\text{Move, Split, No-Action}\}\). Table 1 describes the algorithm defining how the player \(n\) must play the game \(G(N)\) at iteration \(i\).

1. \text{Game\_Iteration}(i,n)
2. \text{static: } A = \{\text{MOVE, SPLIT, No-Action}\}
3. Evaluate Percept for \(n\)
4. \text{if } AC < PC \text{ then}
5. \text{if } u^n_x < \max_{x \in X^n} \{U^n_x\} \text{ then}
6. MOVE\((n \rightarrow x)\)
7. \text{if } EC < EC\_threshold \text{ then}
8. \(x_{max} = \max_{x \in X^n} \{U^n_x\}\)
9. SPLIT\((n, x_{max})\)
10. \text{else No-Action\()\)
11. return

Table 1- Game Iteration for Simple Reflex Agent

The game iteration begins when the current player \(n\) evaluates its percepts for the current state of the environment; the player’s dominant strategy is to move to a position that improves its utility. The condition-action rule used to indicate a move action is specified at Line 4. The player evaluates whether actual coverage \(AC\) is less than the perspective coverage area \(PC\) (Line 4). This encourages the agent to move to a position that maximises resource utilisation.

If a player cannot improve its utility by moving and the edge coverage cell \(EC\) is greater than a defined threshold then there exists an uncovered \(EC\) (if there is no coverage from other players in \(EC\) then this indicates possible coverage gaps), then the player’s strategy changes to split (Line 7), i.e. if the player can not improve its utility then it should split in two, creating another player driven by the same goals. If the player utility cannot be improved by moving and there are no coverage gaps visible to the agent it maintains its current state within the environment.

Once the action has been selected based on the condition-action rules a utility value can be used to place precedence on different alternatives available, this is known as Ordinal Utility. In this case the utility values assigned encode a full behavioural ordering between members of a choice set, but nothing about the strength of preferences [14].

B. Utility Function

The utility function is a mathematical representation of preferred outcomes, to show the degree of success of selected actions. The goal of the agent is to maximise its utility \((U)\), this makes agents rational – the only rational choice is an action which increases the agent’s utility. The utility of an agent we consider here is evaluated as

\[
U = ED + RU + SC - B
\]

Where

- \(ED\)………………….Coverage Edge Demand
- \(RU\)………………….Resource Utilisation
- \(SC\)………………….Super Coverage
- \(B\)………………….Balance Factor

\(ED\) evaluated by equation (2) is the coverage at the fringe of the agents’ coverage cell; it is the difference between the elements of the coverage fringe and those covered by neighbouring agents, weighted by the size of the agents’ edge.

\[
ED = \frac{|EC| - NEC}{|EC|} \quad (2)
\]

Where

- \(NEC\)………………….Non Covered Edge Points

Resource Utilisation \((RU)\) (3) is calculated as the percentage of the agents \(PC\) that it dominantly covers \((AC)\). This element of the utility encourages the agent to move to a position that will maximise its coverage cell usage while meeting the user demand constraint on signal level.

\[
RU = \frac{|AC|}{|PC|} \quad (3)
\]

\[
SC = \frac{|PC|}{\max_{pce \in PC} |pc|} \quad (4)
\]
As described previously the candidate access point mesh provides possible positions of agents. Each position has a calculated \( PC \). The element of the utility \( SC \) calculated by (4) is advantageous to the agent at the beginning of the game, when the strategy is maximise coverage within the target area, as it is weighted by the maximum cardinality of \( PC \) achievable. \( B \) was introduced as part of the utility to ensure balanced signal coverage can be achieved once the demands on target zone are satisfied; it is calculated as the mean error of the signal level in \( AC \).

Based on the percepts each agent can evaluate its utility using (1). One of the rules of the game states that when a move action is being evaluated agents can only move to direct neighbours \( X \) as bound by the candidate access mesh. When a move action is selected, the agent evaluates the utility function of its direct neighbours where \( u^n_i(x) = U^i_z \) represents the utility achieved by the player \( n \) at iteration \( i \) if they move to a new position \( x \).

\[
U^i_\text{r} = \max_x \left\{ U^i_x \right\} \tag{5}
\]

If a player chooses to move to a new position then the equation specified in (5) states that the player must move to a position that maximises its utility. If the player can not improve its utility then it would be irrational to move therefore the player remains in its current position.

If the condition-action rule for split becomes true then the player evaluates each of it’s neighbours \( X \) to find the best position the new player should start. To maintain rationality when evaluating a neighbouring position as a potential player, the player must consider the environment change that will occur due to splitting, i.e. there is an extra player that influences the utility value.

To encourage faster convergence an exception to the game rules was introduced. When evaluating a split action, the player not only evaluates its direct neighbours but also closest candidate position to the player’s edge. This enhances the initial spread of players at the beginning of the game.

If an agent is operating in a fully observable environment where the agent has enough facts about it, then this implies a logical approach will enable the agent to derive a plan that is guaranteed to work. However our agent never has access to the whole truth about their environment; therefore the agent must act under uncertainty. In terms of the WLAN optimisation the uncertainty is the affect the players’ action has on other players in the game leading to the impact on the global optimum. The following describes a co-operative approach which was incorporated to help when dealing with this uncertainty.

C. Co-Operative Game

An advantage of using a utility function to decide on an action is that it can incorporate altruistic behaviour, simply by including the welfare of other agents as one of the factors contributing to the agents own utility. To ensure no player diminishes the utility of another a reward \( Ur \) was introduced.

\[
U^i_\text{r} = \sum_{k=0}^{k<K} U^i_k - U^i_i
\]

Equation (6) shows how the utility reward of the closest players is calculated. If the player moves to new candidate position \( x \) the reward is the difference between the neighbouring player \( k \)’s \( (K \) is all neighbouring players \( k \in K \)) utility in the current environment state \(^*\) indicates a change in the environment) and the utility of player \( k \) in the previous environment state. The reward, \( U^r_i \) is then added to the utility of the player \( n \) if the new position \( x \) is chosen,

\[
u^n_i(x) = U^i_z + U^i_\text{r}
\]

The game continues until it converges to a solution that is Pareto efficient, i.e. it is impossible for the player to increase its utility without decreasing the utility of another player.

IV. Initial Results

To evaluate the effectiveness of the described agent design and utility function some basic environments were investigated. The first is a simple building configuration that is 110 metres long and 36 metres wide. The goal of the optimisation is to achieve a signal level threshold of -55dBm throughout the building.

![Figure 4 - Optimisation Results showing signal level](image)

The results of the optimisation suggest three access points evenly distributed through out the building, are required to fully meet the user defined threshold. Figure 4 shows the results of the optimisation with the associated signal coverage map.

Figure 5 shows the progress of the iterative game. Each position that a player rests is marked by a label signifying the action taken resulting in that position and the iteration of that action, M is Move, S is Split. Therefore a circle marked M2 signifies player one took a move action at game iteration two.

At the game’s commencement there is only one player at 10 far right of the building; this represents a single access point in the environment. The path of the player and actions during the game can be tracked by the green arrow. The player continues to move to position M7 at the border of a wall, in this position the player cannot increase it’s utility function by moving, as there is an open edge to the left of the players coverage map it evaluates a split action, consequently generating another player (player two, identified by a square) to provide additional coverage. Both players continue to move apart from each other (red and green path) demonstrating the
A co-operative element of the game, each movement both improves personal utility and the utility of its closest neighbours. When player one rests at position marked as M9 the condition action rule for split becomes true. The splitting action leads to player two spawning a third player, represented by a triangle (S10 on graph), to the uncovered area. The game continues, with a total of 13 iterations before a Pareto Efficient solution was achieved, i.e. no player can take any action that can benefit itself without harming another player in the game.

The next environment is an example of a simple office configuration; the designer is required to provide seamless wireless access to workers throughout the office floor. The designer set a signal level threshold of -55 dBm throughout the environment to ensure the work areas are sufficiently covered. The floor layout and optimisations constraints let to a design that suggest ten access points be deployed throughout the floor, there optimal positioning is shown in Figure 6. It is clear from this figure that when the access points would provide required coverage if they were placed in those positions.

The results presented are indicative of the preliminary game theoretic implementation over basic environments. Currently this work is being evolved to optimise larger and more complex environments. We expect that these results will demonstrate an optimal design solution for large scale WLANs that addresses scalability in a distributed manner through the application of game theory.

V. CONCLUSION

Initial results of the current implementation show that the utility and strategy taken by the players does lead to an optimal solution, this approach will be extended to include user demand satisfaction to achieve a reliable and cost affective WLAN design. Future work will entail tuning the utility function and mechanism design, prove that a Nash Equilibrium can be achieved and show that when optimising large scale WLANs a distributed approach is the best option.

REFERENCES