Research on throughput-Optimal Control in Multi-radio Multi-Carrier Wireless Sensor Networks

Weixu Zhuang, Xin Huang, Qi Shi, Hang Qiu
Dept. of Electronic Engineering
Shanghai Jiao Tong University, China

Jun. 10th, 2012

Abstract

Wireless sensor networks are widely used in different scenarios. One of the major concerns in most of these applications involves the availability of communication channels to satisfy the channel requirements for all users of a specific network or application type. Nowdays, lots of measures have been developed for sensors to choose the most appropriate channel especially to optimize the capacity of the whole network through channel allocation. However, in reality, it is very hard to extend the network capacity without increasing sink nodes and the upbound of the whole network capacity in real application still keeps unknown. In this report, as Game Theory is widely used in many fields of application, such as flow control and routing, which has a significant potential in future. We specify a real measurement algorithm to estimate the optimal up-bound of the whole network capacity for different certain wireless sensor network scenarios.

At the same time, we figure out an algorithm based on the Game Theory to estimate the application efficiency in increasing the network capacity. Several simulation comparison will be presented for the evaluation of the algorithms. traditional cases of

1 Channel Allocation Methods Survey

1.1 Existent Methods of Channel Allocation

Wireless communication constitutes one of the fastest growing industry segments in recent years, encompassing a number of application domains such as the cellular networks, ad hoc networks, ubiquitous and pervasive computing, sensor networks, and so on. One of the major concerns in most of these applications involves the availability of communication channels to satisfy the channel requirements for all users of a specific network or application type. In this section, we present a discussion on the various challenges and approaches that have been used to solve the problem of channel allocation (CA).

In general, all the channel allocation approaches are classified into two categories: centralized or distributed algorithms. For the centralized approaches, a central control is assumed and it has the complete knowledge about the networks. Thus, the formulated channel allocation problems can be solved at a single place. After the result is calculated, it is distributed to the nodes to accomplish. Altogether, we have seen three types of such problems being formulated: the graph-based problem, the network flows problem, and the network partition- ing problem. Accordingly, the centralized approaches are classified into three categories using the above problem formulations respectively.

For the distributed approaches, no central control is assumed and each node runs its own copy of the algorithm to assign the channels. They are further classified into two categories according to the traffic pattern being considered: gateway-oriented CA approaches and peer-oriented CA approaches. The former assumes that the main network traffic is to or from the gateways, so the CA can exploit the heuristic that the near-gateway links should be given a relatively high bandwidth. The latter assumes that the network traffic can occur between any pair of nodes with no fixed pattern, so the CA approaches have to be as general as possible to accommodate various kinds of network traffic.

1.2 Centralized CA Approaches

The centralized approaches are classified into three categories according to their problem formulations: the graph-based approaches, the network flows approaches, and the network partitioning approaches. To solve these problems, only a centralized algorithm and no communication protocol is needed. Thus, our classification on the centralized approaches is actually a classification on the centralized algorithms. We observe that the inputs, the objectives of outputs, and the heuristic or ap- proximation methods are the three characterizing aspects within each category of algorithms. For the inputs, there are many parameters from the networks to consider. Specifically, all the approaches discussed in this section consider some, if not all, of the following parameters as the inputs.

- Node deployment: the geometric position of each node in the network.
- Numbers of radios and channels: the number of radios at each node and the number of non-overlapping channels available at each radio.
- Interference model: all the centralized approaches surveyed in this paper adopt the Protocol Model, with some of them claiming that their schemes can also be generalized to support the Physical Model.
- Traffic profile: the bandwidth of each link and the end-to-end traffic rate of each flow.
- Connectivity constraint: the level of network connectivity to be achieved. All the approaches at least require the network topology to be connected after the channel assignment, while some approaches have stronger requirements.
- Fairness constraint: specifying a certain fairness requirement to be satisfied.

For the objectives of outputs, some approaches aim to maximize the overall throughput, while others aim to minimize the overall interference, etc. Despite different problem formulations, different sets of inputs, and different objectives of outputs, most centralized approaches surveyed in this paper state or formally prove that their formulated problems are NP-hard. Consequently, they resort to heuristic or approximation algorithms to get suboptimal results such that the computation complexity is polynomial. Thus, we consider the heuristic or approximation methods as a third characterizing aspect within each category of approaches.

Therefore, when these three categories of approaches are examined in the following subsections, special attention is paid to the three aspects (inputs, objectives of outputs, and heuristic or approximation methods) in contrasting them within each category. The choice of these aspects is actually natural since a well-accepted definition of 'algorithm' is 'a sequence of computational steps to transform inputs into outputs', which exactly includes these three aspects.

(1)Graph-based approaches

In graph-based approaches, the network is modeled by a graph with a vertex set and an edge set, and the CA problem is formulated into the problem of assigning channels to vertices or edges in this graph. The basic ideas of the conflict graph concept are as follows: (1) it is derived from the network topology $T(V, E)$ and models whether two links in E interfere with each other; (2) any link in E is represented by a vertex in V_C ; and (3) if two links interfere, an edge connecting the two corresponding vertices in V_C is included in E_C to show this conflict.Three types of graph theoretic concept are generally exploited by this category of approaches: unit disk graph, network topology, and conflict graph.

As an aside, in most of the literature, the elements in V and V_C are called nodes and vertices respectively, and the elements in E and EC are called links and edges respectively. For clarity, this naming convention is followed throughout this paper. Next, four approaches belonging to this category are discussed: (1) Connected Low Interference Channel Assignment (CLICA), (2) minimum Interference Survivable Topology Control (INSTC) [40], Centralized Tabu-based Algorithm (CTA), and (4) Breadth First Search Channel Assignment (BFS-CA), all of which base their algorithms on the above graph concepts.

We also point out the advantages and limitations of the graph-based approaches as a category below.

- Advantage: graph models are intuitive and convenient for designing the CA algorithms.
- Limitation: by solely modeling the network by vertices and edges, it will be difficult to consider traffic load, an important factor for a CA algorithm to improve the network performance.

(2) Network flows approaches

In network flows approaches, the network is modeled by a flow network, thus overcoming the aforementioned limitation of the graph-based approaches. Network flows is a discipline originated from the 19th century that has applications in many fields. Here we only describe two basic definitions in this discipline: flow network and flow, which are essential for understanding later discussions.

Moreover, the advantages and limitations of the network flows approaches as a category are summarized as follows.

- Advantages: the traffic load information, which is neglected by graph-based approaches, is inherently included in the network flows formulation.

- Limitations: all the three surveyed approaches assume constant traffic rates, which is not realistic in most practical networks where the traffic pattern is usually bursty and characterized by random on/off sources.

(3) Network partitioning approaches

In network partitioning approaches, the CA problem is viewed as the problem of partitioning the radios and links in network into disjoint subnetworks such that each subnetwork uses a single channel and different channels are assigned to these subnetworks to reduce interference. Of all the surveyed centralized approaches, two approaches fall in this category: Matroid Cardinality Intersection Channel Assignment (MCI-CA) and Maxflow-based Centralized Channel Assignment (MCCA).

The advantages and limitations of the network partitioning approaches as a category are summarized as follows.

- Advantage: the network partitioning methodology is straight- forward and simple, resulting in polynomial-time problems in- stead of NP-hard problems.
- Limitation: the approaches in this category are not flexible, since all links in a partition are fixed to a common channel; consequently, they cannot optimally achieve certain objectives such as maximizing the throughput or minimizing the interference in the network.

1.3 Distributed CA Approaches

Since the distributed CA approaches involve communication and coordination among multiple parties, they are more challenging to design than their centralized counterparts. In all the eight distributed CA approaches surveyed in this paper, each node measures local channel information and exchanges it with other nodes to calculate the channel assignment. However, these approaches differ in their choices of the local channel information. For instance, these choices can be the number of links sharing a common channel within the interference range, the traffic load on a channel, the signal to interference and noise ratio on a channel, or a combination of them. For convenience, we refer to these choices as CA metrics hereafter. Analogous to the routing metrics that serve as the basis for routing, the CA metrics also play an equivalent role in making the CA decisions. Therefore, when each CA approach is examined below, special attention is paid to its CA metrics.

The distributed CA approaches are classified into two categories: gateway-oriented approaches and peer-oriented approaches, which are in turn discussed below.

Gateway-oriented approaches assume a certain number of gateways as the central sinks of the major network traffic. There are three such approaches: Hyacinth, DMesh, and CoMTaC.

we highlight the advantages and limitations of the gateway-oriented CA approaches as follows.

- Advantage: the opportunity to utilize the gateway nodes to simplify the CA approach.
- Limitation: the incapability of accommodating other kinds of traffic patterns.

Unlike the gateway-oriented approaches, the peer-oriented approaches do not make any assumptions on the traffic pattern, so they are general to accommodate various kinds of network traffic. In this subsection, five such approaches are examined: (1) Probabilistic Channel Usage based Channel Assignment (PCU-CA), (2) Joint Optimal Channel Assignment and Congestion Control (JOCAC), (3) Superimposed Code based Channel Assignment (SC-CA), (4) Distributed Greedy Algorithm (DGA), and (5) Self-Stabilizing Channel Assignment (SS-CA). The advantages and limitations of the peer-oriented CA approaches as follows.

- Advantage: the capability of adapting to various kinds of traffic patterns, making it applicable for most of the current WMNs, where the peer-to-peer traffic is also a major part of the total network traffic.
- Limitation: the difficulty of dealing with CA issues such as fault tolerance, ripple effect, and channel oscillation without any assumption on the traffic pattern.

2 Game Theory

2.1 Background Introduction of Game Theory

Game theory is a tool for analyzing the interaction of decision makers with conflicting objectives. In recent years, it has seen some application by computer scientists to problems such as flow control and routing, but it is believed that game theory can be applied fruitfully to a much broader class of problems in communications systems.

Modern day communications systems are often built around standards. Some such standards are open, such as the TCP/IP standard on which the internet is based. Other standards, such as IS-95 (CDMA), contain intellectual property which must be licensed by the developer. In most cases, though, devices to access these systems are being built by a variety of different manufacturers. In many cases, these manufacturers may have an incentive to develop products which behave selfishly by seeking a performance advantage over other network users at the cost of overall network performance. In other cases, end users may have the capability to alter products to behave in a selfish manner. Given our reliance on standards, it seems that we should design and build systems that are prepared to cope with users who behave selfishly. If possible, such systems should make selfish behavior unprofitable, so that users will prefer to behave in a manner which is optimal for the system as a whole. When this is not possible, designers should at least be aware of the impact that selfish users would have on the operation of the specified system.

Note that while specifications can establish the rules of interaction, it is difficult for a specification to enforce a specific algorithm for the end user to execute. For instance, consider a slotted Aloha system. While the system designer can specify that users use slotted Aloha for access to a given channel, it may be impossible for the designer to ensure that every user uses the Pseudo-Bayesian algorithm to estimate the number of backlogged users and choose a retransmit probability. In this case, it is impossible for a central controller to know what retransmit probability the end user is using, making it difficult to enforce such a choice.

Another reason that game theory is an appropriate tool in the setting of communications networks is that game theory deals primarily with distributed optimization individual users, who are selfish, make their own decisions instead of being controlled by a central authority. Many of the problems which must be solved in a communications system are known to be NP-hard; as a result solving these optimization problems centrally becomes computationally infeasible as network size increases. Because game theory focuses on distributed solutions to system problems, we expect systems designed with game-theoretic concerns in mind to be highly scalable.

2.2 Channel Allocation Application of Game Theory

2.2.1 Cooperative Game Theory for Channel Allocation

Suris et al.develop a cooperative game theory model to analyze a scenario where nodes in a multi-hop wireless network need to agree on a fair allocation of spectrum. They show that in high interference environments, the utility space of the game is non-convex, which may make some optimal allocations unachievable with pure strategies. Also, as the number of channels available increases, the utility space becomes close to convex and thus optimal allocations become achievable with pure strategies. They propose the use of the Nash Bargaining Solution(NBS) and show that it achieves a good compromise between fairness and efficiency, using a small number of channels. Finally, they propose a distributed algorithm for spectrum sharing and show that it achieves allocations reasonably close to the Nash Bargaining Solution.

A. Game Model

The spectrum sharing problem can be modeled as follows. The available bandwidth is divided equally into multiple channels. Each wireless device (referred to as node) can transmit in any combination of channels at any time and can set its transmit power on each channel. Each transmitting node is only interested in communicating to a single receiver node. Receiver nodes do not transmit and thus are not considered players in the game (since they will act in coordination with the transmitter).

Let $\chi = \{1, \cdots, K\}$ be the set of available channels, B be the aggregate bandwidth, with each channel having bandwidth $\frac{B}{K}$, and N be the number of transmitter nodes in the network. We formulate the spectrum sharing game as follows: $M = \{1, \ldots, N\}$, $P_i^\chi = \{(P_i^k)_{k\in\chi} | p_i^k \geq 0, \sum_{k\in\chi} p_i^k \leq P_{max}\}$ and $P^\chi = P_1^\chi \times \cdots P_N^\chi$. Let $\boldsymbol{p} \in P^\chi$ and $u_i(\boldsymbol{p}) = C_i(\boldsymbol{p})$, where $C_i(\boldsymbol{p})$ is the Shannon capacity.

B. Distributed Algorithm

The goal is to design a distributed algorithm that achieves the NBS for the spectrum sharing game. We need the algorithm to operate only with local information and no centralized control. In this section it is shown that nodes can be aggregated into overlapping groups, which then leverage to distribute the computation of the NBS. Nodes within each group are in close proximity, which allows nodes to only use local information. Finally, they propose an algorithm for computing an approximation to the NBS and prove its convergence. They make the following assumptions:

1) There is an underlying method for information exchange such that nodes within two hops can communicate within a time scale shorter than the time scale for updates to channel allocation.

2) Nodes run the algorithm at random intervals such that the probability that two or more nodes (within two hops of each other) run the algorithm simultaneously is small.

3) The execution time of the algorithm is small relative to the interval between executions of the algorithm.

4) The initial agreement point is 0.

Algorithm: Distributed NBS Computation

$IZ = i \cup j | distance(R_x(j), i) < R, u_j > 0$
$oldNP = \prod_{j \in IZ} u_j(\boldsymbol{p})$
$\hat{\boldsymbol{p}} = MaximizeNP(i, IZ,)$
$newNP = IZuj(\hat{\boldsymbol{p}})$
if $newNP > (1 + tol) * oldNP$ then
$p_i^k = \hat{p_i^k}$
$endif$

2.2.2 Adaptive Channel Allocation for Cognitive Radio Networks

Nie Nie and Cristina Comaniciu propose a game theoretic framework to analyze the behavior of cognitive radios for distributed adaptive channel allocation. They define two different objective functions for the spectrum sharing games, which capture the utility of selfish users and cooperative users, respectively. Based on the utility definition for cooperative users, they show that the channel allocation problem can be formulated as a potential game, and thus converges to a deterministic channel allocation Nash equilibrium point. Alternatively, a no-regret learning implementation is proposed for both scenarios and it is shown to have similar performance with the potential game when cooperation is enforced, but with a higher variability across users.

A. Potential Game Formulation

The first utility function (U1) they propose accounts for the case of a selfish user, which values a channel based on the level of interference perceived on that particular channel:

$$U1_i\left(s_i, s_{-i}\right) = -\sum_{j \neq i, j=1}^{N} p_j G_{ij} f\left(s_i, s_{-i}\right)$$

For the above definition, we denoted $P = [p_1, p_2, \cdots, p_N]$ as the transmission powers for the N radios, $S = [s_1, s_2, \cdots, s_N]$ as the strategy profile and $f(s_i, s_j)$ as an interference function.

The second utility function we propose accounts for the interference seen by a user on a particular channel, as well as for the interference this particular choice will create to neighboring nodes. Mathematically we can define U2 as:

$$U2_i\left(s_i, s_j\right) = -\sum_{j \neq i, j=1}^{N} p_j G_{ij} f\left(s_j, s_i\right) - \sum_{j \neq i, j=1}^{N} p_j G_{ji} f\left(s_i, s_j\right)$$

The complexity of the algorithm implementation will increase for this particular case, as the algorithm will require probing packets on a common access channel for measuring and estimating the interference a user will create to neighboring radios.

In order to have good convergence properties for the adaptation algorithm we need to impose some mathematical properties on these functions. There are certain classes of games that have been shown to converge to a Nash equilibrium when a best response adaptive strategy is employed.

It is shown that for the U2 utility function, they can formulate an exact potential game, which converges to a pure strategy Nash equilibrium solution.

B. no-regret learning for dynamic channel allocation

While they showed in the previous section that the game with the U2 utility function fits the framework of an exact potential game, the U1 function lacks the necessary symmetry properties that will ensure the existence of a potential function. In order to analyze the behavior of the selfish users game, they resort to the implementation of adaptation protocols using regret minimization learning algorithms. No regret learning algorithms are probabilistic learning strategies that specify that players explore the space of actions by playing all actions with some non-zero probability, and exploit successful strategies by increasing their selection probability. While traditionally, these types of learning algorithms have been characterized using a regret measure (e.g., external regret is defined as the difference between the payoffs achieved by the strategies prescribed by the given algorithm, and the payoffs obtained by playing any other fixed sequence of decisions in the worst case), more recently, their performance have been related to game theoretic equilibria.

A general class of no-regret learning algorithms called ϕ-no-regret learning algorithm are shown in to relate to a class of equilibria named ϕ-equilibria. No-external-regret and no-internal regret learning algorithms are specific cases of ϕ-no-regret learning algorithm. ϕ describes the set of strategies to which the play of a learning algorithm is compared. A learning algorithm is said to be ϕ-no-regret if and only if no regret is experienced for playing as the algorithm prescribes, instead of playing according to any of the transformations of the algorithms play prescribed by elements of ϕ. It is shown in that the empirical distribution of play of ϕ-no-regret algorithms converges to a set of ϕ-equilibria. It is also shown that no-regret learning algorithms have the potential to learn mixed strategy (probabilistic) equilibria. We note that Nash equilibrium is not a necessary outcome of any ϕ-no regret learning algorithm.

2.2.3 Channel Allocation Using an Auction Algorithm

Jun et al.use the second-price auction mechanism whereby user bids for the channel, during each time slot, based on the fade state of the channel, and the user that makes the highest bid wins use of the channel by paying the second highest bid. Under the assumption that each user has a limited budget for bidding, we show the existence of a Nash equilibrium strategy, and the Nash equilibrium leads to a unique allocation for certain channel state distribution, such as the exponential distribution and the uniform distribution over [0, 1]. For uniformly distributed channel state, we establish that the aggregate throughput received by the users using the Nash equilibrium strategy is at least 3/4 of what can be obtained using an optimal centralized allocation that does not take fairness into account.

A. Problem Formulation

We now describe the second-price auction rule used in this paper. Let α_i be the average amount of money available to user i during each time slot. We assume that the values of α_is are known to all users. Moreover, users know the distribution of X_i for all i. Assuming that the exact value of the channel state X_i is revealed to user only at the beginning of each time slot. During each time slot, the following actions take place: 1) each user submits a bid according to the channel condition revealed to it; 2) the transmitter chooses the one with the highest bid to transmit; and 3) the price that the winning user pays is the second highest bidders bid. Users who lose the bid do not pay. In case of a tie, the winner is chosen among the equal bidders with equal probability. Formally, this N-players game can be written as which specifies $\gamma = [N, \{S_i\}, \{g_i(\cdot)\}]$ for each player a set of strategies, or bidding functions,S_i (with $s_i \in S_i$) and a payoff function $g_i(s_1, \cdots, s_N)$ giving the throughput associated with outcome of the auction arising from strategies $(s_1, \cdots, s_N)$.

The formulation of our auction is different from the type of auction used in economic theory in several ways. First, we look at a case where the number of object (time slots) in the auction goes to infinity (average cost criteria). While in the current auction research, the number of object is finite. Second, in our auction formulation, the money used for bidding does not have a direct connection with the value of the time slot. Money is merely a tool for users to compete for time slots, and it has no value after the auction. Therefore, it is desirable for each user to spend all of its money. However, in the traditional auction theory, an objects value is measured in the same unit as the money used in the bidding process, hence their objective is to maximize the difference between the objects value and its cost.

2.3 Scenario Introduction

The multi-channel (MC) wireless networks have become a really hot topic, which holds the attention of not only the academy, but industry as well. Intuitively, since the radio have more channel choices, the problem will become extremely complicated as the number of possible flow paths increases exponentially as the data hops from the source to its destination. In a multi-hop wireless network, the throughput-optimal scheduling scheme problem has been proved to be an NP-hard problem [4] [10]. However, relevant method and algorithm has been brought about continuously. In [20] and [2] by *Lin Gao et al*, the two-hop and multi-hop wireless network channel allocation problem is extensively discussed. The throughput maximization problem of multi-channel wireless networks is researched by *H Li,Yu Cheng et al* [3] who have developed an tuple-based method which can get an cross-layer optimal control over throughput.

In this paper, we study the throughput and data congestion in single-radio multi-channel (SR-MC) wireless networks. We introduce the single-radio single-channel (SR-SC) model at first, and then extend its approach to SR-MC wireless networks. Then, we form an approximation algorithm to get the upper-bound of the throughput of SR-MC wireless networks. Next, we introduce distributed algorithms using game theory to solve this channel allocation problem in SR-MC network. The simulation results show our algorithms perform well in both getting a high network throughput and avoiding data congestion. Then comparison between our algorithm and the upper bound help us to estimate the performance of our network.

The algorithms we introduce in this paper have following advantages:

- There wouldn't be any conflict in the channel at last.

- No channel will be wasted which means every channel will be occupied unless there are not enough nodes to use all channels in particular region.

- Nodes with low ability for the channel will give up the channel so that nodes with higher ability will be able to use the channel without conflict.

This advantages are well suited for our SR-MC network and the simulation results will show the performance of our algorithms to solve the channel allocation problem.

2.3.1 Background and related work

The problem of the capacity of wireless network was first proposed by *Kumar et al* in their inspiring literature [19], which not only present the fundamental model for calculation but also suggest a basic methodology on both the upper-bound and lower-bound. As the model became more and more complicated, the problem of the capacity of multi-channel wireless network draws huge attention. This problem has been proposed and extensively discussed by P. Kyasanur and N. H. Vaidya, in their work [5]. After that, a hot discussion has been triggered and much more excellent work has been done such as [7]. And in [6], three general techniques for upper-bounds: hop-count method, Shaded Area of Transmissions and Data Copies has been summarized. Also, the classic back pressure algorithm has been developed in [15] to achieve the optimal capacity region, however it is proved as NP-hard problem. The greedy algorithm for low-complexity has been comprehensively studied in terms of throughput [12], [13] and delay [16], [14]. The tradeoff between throughput and delay and their balancing algorithms are also discussed in [8], [9].

As for study on the throughput-optimal control and practical scheduling design in the multi-radio multi-channel (MR-MC) networks.A optimization framework is developed in [4] to jointly solve the resource allocation problem. However, it relies on the link based model and does not provide any practical scheduling algorithm. And [3] proposed tuple-based network model allows a decomposable cross-layer framework to enhance the delay performance of the throughput-optimal scheduling in MR-MC networks. In their experiment, a fully effectiveness comparison between their tuple-based greedy maximal scheduling (TGMS) , tuple-based distributed maximal scheduling (TDMS), and the aggregated maximal scheduling (AMS) and the single path (SP) algorithm proposed in [3] has been carried out.

2.3.2 Problem formulation

In this section, we describe our system model and assumptions used throughout the paper in detail. We consider a multi-hop network as our model. There is only one SINK which is the destination of all data and only the data being transmitted to the SINK can be counted as the throughput of our network. Every node has the same interference radius R which means when the node use the channel to transmit data, other nodes lay in the interference zone of this node cannot use the same channel simultaneously. We call the data transmitter and the data receiver as a data link. We can easily find that in data transmitters interference zone, no other nodes can receive data while data transmitter is transmitting. In addition, in data receivers interference zone, no other nodes can transmit data at the same time. Therefore, we can regard the data-link as a whole which has its own interference zone and we only need to consider the interference between data-links. To further clarify our problem, one node in our model can only transmit its data to one other node and the topological structure of our model is stationary. As a result, we can use the transit node in a data-link to represent this data link. The data-link's interference zone can also to be regarded as the transit nodes interference zone. To make our problem more general, we assume that there are three channels in our network. Figure 1 illustrates one possible topology of this model.

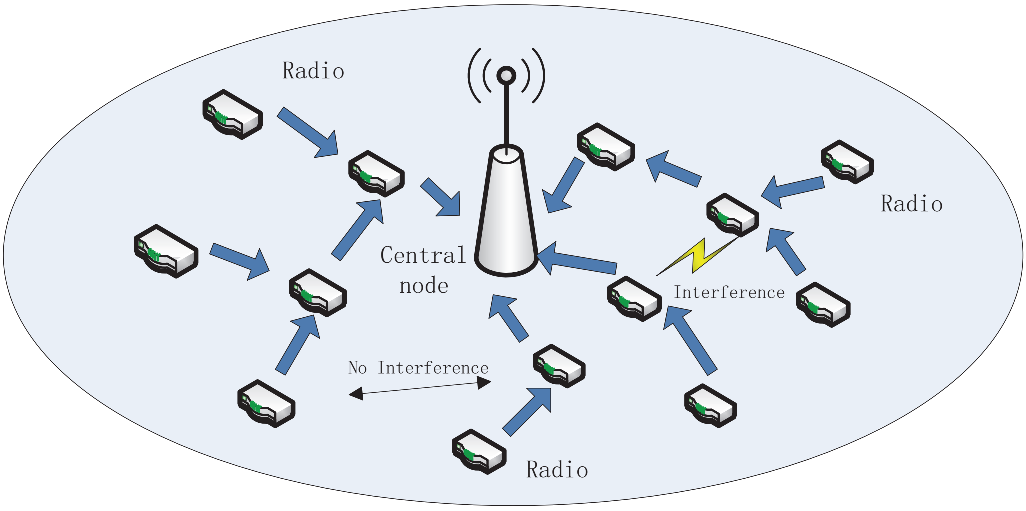

Figure 1: Possible topology illustration

Moreover, we use game theory to solve this channel allocation problem and use an approximate algorithm to calculate the upper bound of the throughput of our network. This upper bound of our network will help us to analysis the performance of our algorithms using game theory.

Then, we introduce a set of notations used to define a truthful spectrum game and efficient algorithms we propose in this paper.

Channel bid (x_{ik}) - It represents the ability of node i for channel k, we will show its definition in next section when we introduce it.

Channel bid (T_i) - It is the set of nodes interfering with node i.

Channel bid (U_{ik}) - This is the utility function of node i for channel k, we will show its definition in next section when we introduce it.

2.4 Upper-bound of throughput research and capacity analysis in WSNs

In this section, we discuss the problem of capacity upper-bound using the system model described before.

Once the system topology is given, the link-based model [17] is established. Consider a network topology G consists node set N and link set L. Each node n_i is equipped with one radio and available channel set C. Further, $s(n_i)$ is the source node of node n_i if n_i can receive information from $s(n_i)$, and $d(n_i)$ is the corresponding destination of node n_i if node n_i can transmit information to node $d(n_i)$. Since we are applying a link-based model, we define a successful data flow on one link as the functioning of the source node of that link which is a simplification for simulation. For each topology, the interference is considered with a scenario check 2.4.1, which is to avoid the interferential conflict between links. If a scenario satisfy the constraints in a scenario check, we call it a *practical scenario.*

To calculate the throughput in a *practical scenario*, we bring in the *quality of link* (QoL) which is defined as Q_i for link l_i in link set L, according to *kumar et al* and their work [19], we define capacity C in a *practical scenario* as equation (1).

$$C = \sum_{j=1}^{Nc} \sum_{i=1}^{Nc^j} Qi \times \tau \tag{1}$$

Q_i indicates the quality of link i and τ is the time in one time-slot. N_c and N_c^j is the number of channels and the number of links that are using channel j. And equation 2 defines the throughput λ of a practical scenario, in which N_n indicates the number of nodes and h_i means the number of hops from node i to the sink node.

$$\lambda = \frac{C}{\sum_{i=1}^{Nn} h_i} \tag{2}$$

2.4.1 Algorithm formulation

In this section, we develop the exhaust algorithm which can strictly reach the optimal throughput under SR-SC model discussed above, and its natural extension to SR-MC model. Then, due to its time complexity soars exponentially as the number of available channels increases, (i.e. the expansion of the set C), we further bring about an improved approximation method to decline the complexity to tolerable computing level.

- Interferential data setup

  To setup any particular scenario, we should first get the certain topology like Figure 1, and accordingly search the entire map for every interference that should be considered. The searching algorithm goes like this: on one hand for each transmitting node, we search for every node within its interference range and ensure that they cannot receive any information from other than this one node at the same time, on the other hand, for each receiving node, we also check every node within its interference range and ensure that they cannot transmit any information to other than this one node simultaneously. We present the pseudo-code in Table 1 and Table 2.

  The setup data of *TintTable* and *RintTable* are essentially important basis for further algorithm. Any relationship/link causing conflicts which is reflected in those two tables will not be established strictly. They are referenced for check in upper-bound calculation.

Table 1:

Phase 1, for each transmitting node in SR-SC model.
for i = 2 to number of nodes do
for j = 2 to number of nodes do
if the distance between node i and the destination of node j is less than the interference radius && $i \neq j$ then
TintTable(i,j)=1; % record the existence of interference
end
end
end

Table 2:

Phase 2, for each receiving node in SR-SC model.
for i = 2 to number of nodes do
for j = 2 to number of nodes do
if the distance between node i and the destination of node j is less than the interference radius && $i \neq j$ and node i is not the destination of node j then
RintTable(i,j)=1; % record the existence of interference
end
end
end

- SR-SC model

  We first discomposed the multi-channel problem to single-radio single-channel problem, and then do an complete search in each scenario which is produced according to the binary form of the cycle variable i. Without loss of generality,considering a $N = 20$ nodes topology numbered from 1 to 20 (node 1 is the sink node) and map node 2 to node 20 to each bit of i in its binary form from LSB to MSB. Each bit indicates the establishment of a node link. For instance, if $i = 1_{(10)} = 0000000000000001_{(2)}$, then only node 2 would be functioning, if $i = 3_{(10)} = 0000000000000011_{(2)}$, then only node 2 and 3 would be both functioning. And by functioning, we mean it's transmitting information rather than receiving [2.3.2].We present the pseudo-code in Table 3 and Table 4.

- Extension to SR-MC model

  For a SR-MC model with L channels available, the strict extension would be that each link has L choices of channels. The interferential conflict searching algorithm will be: on one hand for each transmitting node, we search for every node within its interference range and ensure that they cannot receive any information from other than this one node at the same time in the same channel, on the other hand, for each receiving node, we also check every node within its interference range and ensure that they cannot transmit any information to other than this one node simultaneously using the same channel. Consequently, in the two phase data setup for *TintTable* and *RintTable* as mentioned before, the channel choice data of each link should be prepared for reference. We present the pseudo-code for data setup in SR-MC

Table 3:

```
SR-SC capacity upper-bound Algorithm (CUBA) Part 1
Initialization of the maximal capacity
MaxC = 0;
for i = 1 to 2^(thenumberofnodes − 1) % enumerate every possible scenario.
    Produce the scenario according to i.
    for i = 2 to the number of nodes
        if node i in this particular scenario is functioning then []
            check RintTable(i,:) and TintTable(i,:) for potential conflict.
            If there is interferential conflict then
                break; %Abandon this scenario
            end
        end
     end
    if this scenario is practical
        include part 2 for comparison and possible refreshment
    end
end
```

Table 4:

```
SR-SC capacity upper-bound Algorithm (CUBA) Part 2
Calculate the capacity of certain scenario [2.3.2]
if current capacity > MaxC then
    record the scenario.
    MaxC = current capacity; %refresh the capacity record
end
```

model in Table 5 and Table 6.

Table 5:

Phase 1, for each transmitting node in SR-MC model.
for i = 2 to number of nodes do
for j = 2 to number of nodes do
if the distance between node i and the destination of node j is less than the interference radius && $i \neq j$ then
TintTable(i,j)=1 only when they are on the same channel
% record the existence of interference
end
end
end

Table 6:

Phase 2, for each receiving node in SR-MC model.
for i = 2 to number of nodes do
for j = 2 to number of nodes do
if the distance between node i and the destination of node j is less than the interference radius && $i \neq j$ and node i is not the destination of node j then
RintTable(i,j)=1; only when they are on the same channel
% record the existence of interference
end
end
end

With the interferential data and channel allocation data, the *strict* capacity upper-bound algorithm for SR-MC scenario will need to add a channel allocation setup for interferential check at each scenario. The producing methodology is to change cycle variable to binary form, which is same with the methodology in scenario establishment for link setup [section 2.4.1]. We present the pseudo-code in Table 7 and Table 8

If there are L channels available, the scenario interference data amount would be L times the size of *TintTable* and *RintTable* and the complexity of CUBA would increase by the power of L which lead to much more complicated and time-consuming problem. However, if loosening the upper-bound, we develop an approximation approach to have the capacity upper-bound and decrease the complexity to the level of SR-SC model. The possible overlap for choosing the same node and different channel in calculation may compensate for the loss of throughput due to the decrease in interference by using multi-channel.

We present the pseudo-code in Table 9 and Table 10

2.5 Protocol research with game theory

In this section, we introduce the main algorithms to efficiently solve the channel allocation problem and discuss the effects and advantages of our algorithms. In our model, we use x_{ik} to describe the

Table 7:

```
SR-MC capacity upper-bound Algorithm (CUBA) Part 1
---------------------------------------------------------------
Initialization of the maximal capacity
MaxC = 0;
for i = 1 to 2^(thenumberofnodes − 1) % enumerate every possible scenario.
    Produce the scenario according to i.
    for count j = 1 to the number of channels then
        setup the channel allocation according to j.
        for i = 2 to the number of nodes
            if node i in this particular scenario is functioning then []
                check RintTable(i,:) and TintTable(i,:) for potential conflict.
                If there is interferential conflict then
                    break; %Abandon this scenario
                end
            end
         end
        if this scenario is practical
            include part 2 for comparison and possible refreshment
        end
    end
end
```

Table 8:

```
SR-MC capacity upper-bound Algorithm (CUBA) Part 2
---------------------------------------------------------------
Calculate the capacity of certain scenario [2.3.2]
if current capacity > MaxC then
    record the scenario.
    MaxC = current capacity; %refresh the capacity record
end
```

Table 9:

SR-MC capacity upper-bound Algorithm (CUBA) Part 1

```
for channelcount = 1 to number of channels do
    Initialization of the maximal capacity
    MaxC = 0;
    for i = 1 to 2^(thenumberofnodes − 1) % enumerate every possible scenario.
        Produce the scenario according to i.
        for count j = 1 to the number of channels then
            setup the channel allocation according to j.
            for i = 2 to the number of nodes
                if node i in this particular scenario is functioning then []
                    check RintTable(i,:) and TintTable(i,:) for potential conflict.
                    If there is interferential conflict then
                        break; %Abandon this scenario
                    end
                end
             end
            if this scenario is practical
                include part 2 for comparison and possible refreshment
            end
        end
    end
    C=C+MaxC;
end
```

Table 10:

SR-MC capacity upper-bound Algorithm (CUBA) Part 2

```
Calculate the capacity of certain scenario [2.3.2]
if current capacity > MaxC then
    record the scenario.
    MaxC = current capacity; %refresh the capacity record
end
```

ability of node i in channel k to determine the chance that node i can get the channel. We define x_{ik} as follow:

$$x_{ik} = Blog(1 + \frac{S}{N}) \cdot l_i^2 \tag{3}$$

We can find that this x_{ik} is determined by the capacity of node i when it uses channel k and the buffer length in node i. From the equation (1) we stated above, we can get that if the node i has a higher channel capacity or longer buffer length, It will get a better chance to get the channel to transmit the data. This effect is really suitable for our task because our aim for the network is to maximize the throughput of the multi-hop network. Because our network has only one SINK node and all the data has to be transmitted to the SINK node. If we only consider the channel capacity of node i, the multi-hop network may be congested because only the nodes with high channel capacity can get the channel to transmit their data. As a result, lots of data will be congested in the nodes which do not get the opportunity to use the channel and the throughput of our network may be very low. Now, we add the length of buffer of nodes to help determine the node selection. Node i having long buffer length means that it has not transmitted its data for a long time. And it will get a better chance to transmit its data because of the increase of x_i. As we have discussed above, we find that by using the utility function we design, the fairness of the network will also be ensured. Because everyone has opportunity to use the channel even they do not get a high SNR for the channel. Then, we will introduce the algorithms to solve the problem.

2.5.1 Algorithm to initiate the network

The initial state of our network is really important because this directly influences how long the convergent time is. We will give out the initial algorithm to help the game of the network converge in relatively shot time. From the algorithm we give above, we know that there are at most half of

Algorithm: Initialization of the network

Each node i calculate its own ability for all three channels x_{i1}, x_{i2}, x_{i3}.
Each node i calculate the ability of other n nodes in its interference zone (denoted as T_i) for all three channels.
if $x_{ik} > \frac{\sum_{j \in T_i}(x_{jk})}{n}$ $k = 1, 2, 3$
i make the decision that it want to occupy the channel k
if node i want to occupy more than one channel
i choose the channel that it has the maximum x_{ik} to occupy
end if
end if

the nodes make the decision that they want to occupy the channels. This will help the system a lot in getting to a stable state in relatively short time.

2.5.2 Main algorithm for this problem

First, we will introduce the utility function of nodes in our network. This utility function will help the network to get stable in relatively short time and avoid any conflict in our network.

$$U_{ik} = x_{ik} - \max_{j \in T_i}(x_{jk}) \quad k = 1, 2, 3 \tag{4}$$

This is the utility function of node i for channel k. After initiate the network, every node get a different random number which determines the order of nodes' decision making. When it turns to node i to make decision, he will firstly examine the decision of surrounding nodes in last turn and get the ability $X = (x_{11}, x_{12}, x_{13}, x_{21}, \cdots, x_{n2}, x_{n3})$ of surrounding nodes. Then he uses the utility function shown above to calculate its utility of each channel. Next, he will choose the channel that he can get the maximum utility to "occupy" (this occupy means this node makes the decision that it wants to occupy the channel in next transmitting times lot and the decision may change between decision making turns). If its utility functions for all three channels are all negative, it will not "occupy" any channel. After several decision making turns (every node may make or change their decision for several times), it comes to a state that no nodes would like to change their decision any more. We called it Nash Equilibrium. When it comes to Nash Equilibrium, the decision making period ends and nodes which decide to use the channels begin to transmit their data for a period of time. The main algorithm is shown as follow: The critical interfere data links of node i means

Algorithm: Main algorithm to solve the problem

Input the initial condition of the situation
Each data-link get a random number let $B = (b_1, b_2, \cdots, b_n)$.
Sort the B in descending order B'.
do
for $j = 1$ to n
i calculate its utility function as
$U_{ik} = x_{ik} - max_{j \in T_i}(x_{jk}) \quad k = 1, 2, 3$
if there are critical interfere data links j
if $x_i \geq x_j$
Continue;
else
if
Continue;
else
Select the largest U_{ik}, and choose channel k to occupy
end if
end for
Until the condition of channel occupation are no longer changed.

some data links using the same transit node or receiving node with node i (or data link i). We have to check these critical interfere data links first because no critical interfere data links of node i can transmit their data while node i uses the channel even there are multiple channels in the network. Then, the block diagram of our algorithm is given as follow:

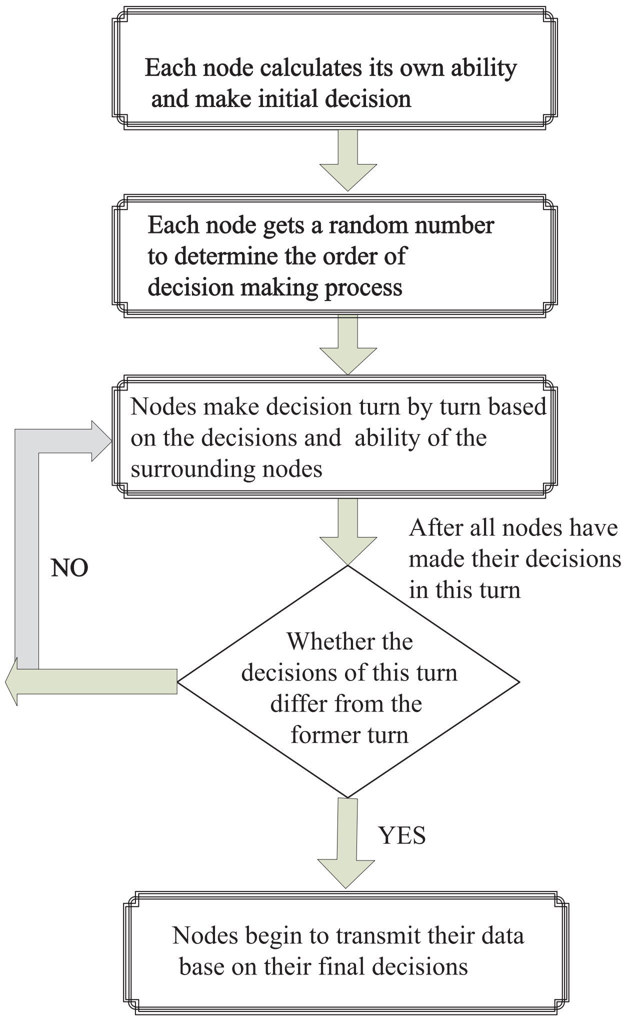

Figure 2: The block diagram of the main algorithm

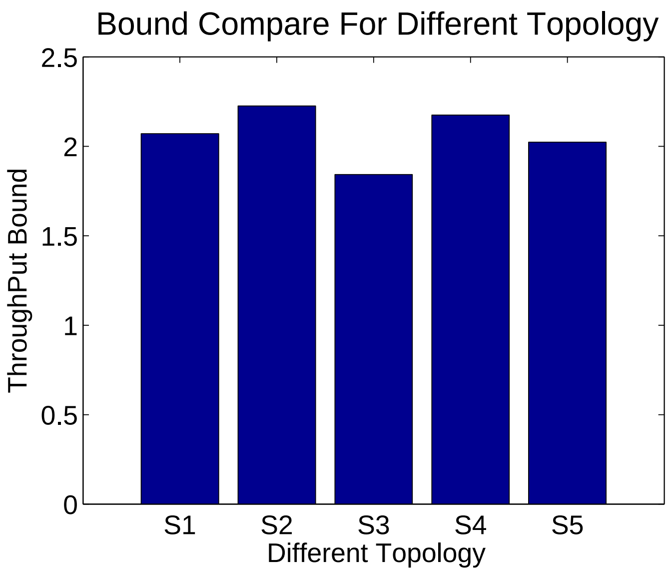

Figure 3: Throughput upper-bound

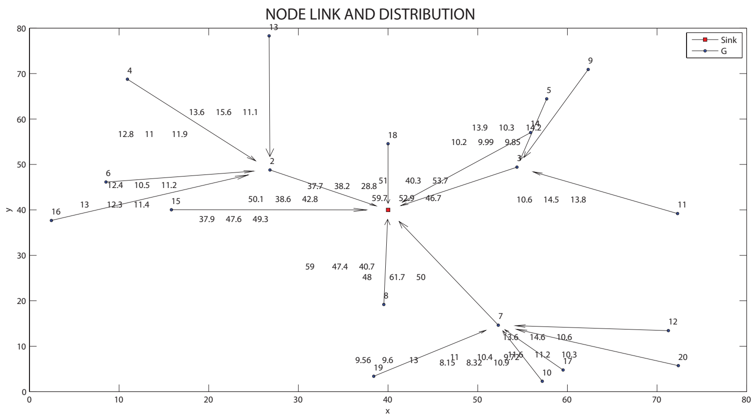

Figure 4: topology 1

3 Simulation result

3.1 Simulations and results analysis of throughput upper-bound

In this section, we run five times of simulation with the exactly same topology in section 2.3.2 which contains twenty nodes ($N = 20$) and three available channels ($L = 3$) for each node and we assign each link its transmission rate/data rate for each three channels. The QoL is produced by equation (1), and for each link l_i, Q_i may not be the same. According to each topology in five of them in section 2.3, we present five upper-bound with its maximal *practical scenario*.

In Figure 3, the throughput upper-bound is indicated with 5 bars on the chart. The bars denote the bound value for topology 1 to topology 5 [Figure 4, 5, 6, 7, 8].

Figure 4,5,6,7,8 show the five topology.

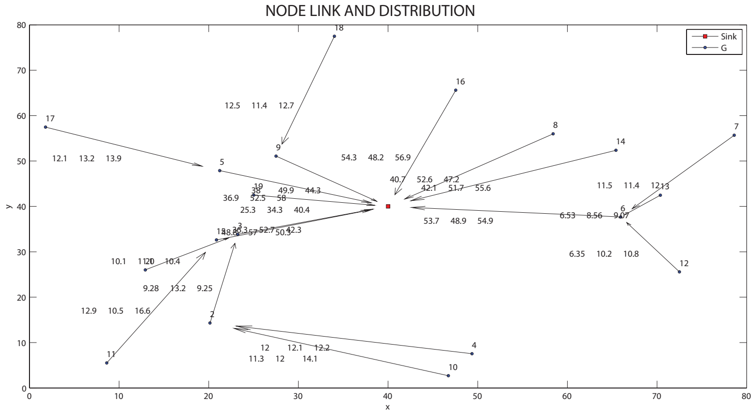

Figure 5: topology 2

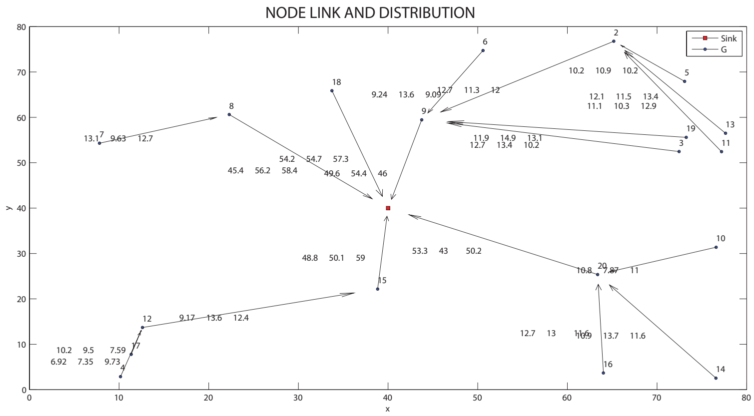

Figure 6: topology 3

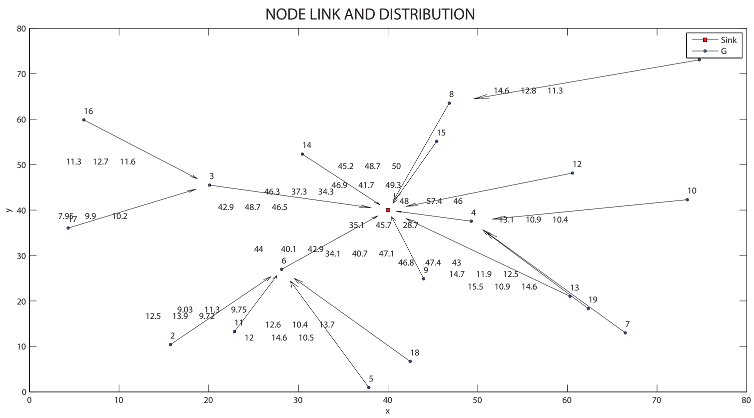

Figure 7: topology 4

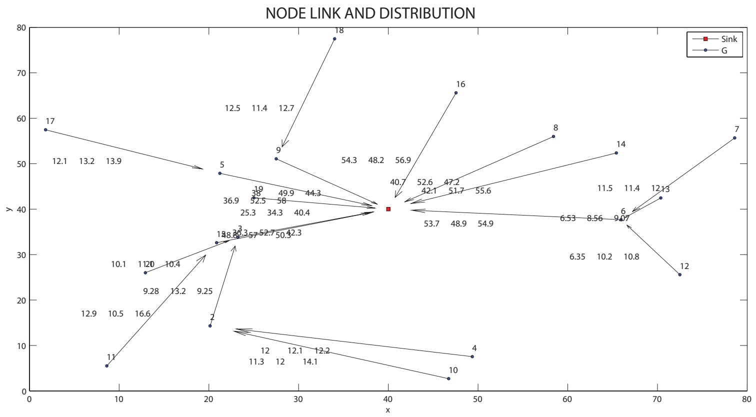

Figure 8: topology 5

3.2 Simulations and results of our game theory algorithm

In this section, we show the performs of our algorithm discussed above which uses game theory. Then, we compare our algorithm with greedy algorithms and the upper bound of the network. This will further illustrate the effects of our algorithm. First, we examine the length of data buffer in the node which has the maximum data buffer in the time slot through the 3000 time slots. Then, we explore the throughput of the network in 3000 time slots. Finally, the comparison among algorithms we use, greedy algorithm and the upper bound is revealed. In all of our simulation works, we randomly generate 20 nodes with different ability for all three channels in our multi-hop network. We let every nodes interference radius equals to its transmission radius. On the one hand, in transit nodes interference zone, no other node can receive data at the same time. On the other hand, in receiving nodes interference zone, no other node can transmit data at the same time. As a result, we can build up an interference table of the random topology. In addition, we assume that every node has the same generating rate of data, and this rate is comparable to the ability of nodes for the channel. From Fig. 9, we can find that our algorithms using game theory perform well in maintaining the fairness of our network. The maximum buffer size status in our network don't change much in 3000 time slots, which means there are no congestions occurring in our network. Therefore, the fairness of our network is ensured by using the algorithms we introduce above. In the other word, each node in our network can get some opportunities to transmit data. Fig. 10 represents the total throughput of our network. We can get that after about 300 time slots, the throughput of our network become stable. The algorithms we introduce above using game theory performs best in this aspect. By using this algorithm, we can get a relatively high network throughput and this is the most important element to evaluate the effects of an algorithm. In Fig. 11, we compare the performance of our algorithm with the upper bound of our network. The upper bound of our network is not the maximum throughput of our network. In the other word, it is much larger than the maximum throughput of our network. Getting the maximum throughput of our network is an NP-hard problem which means we have to calculate the result in an exponential time. Therefore, we use an approximate algorithms we introduce in section 2.4 to calculate the upper bound of our network throughput. And we can get the upper bound in a polynomial time.

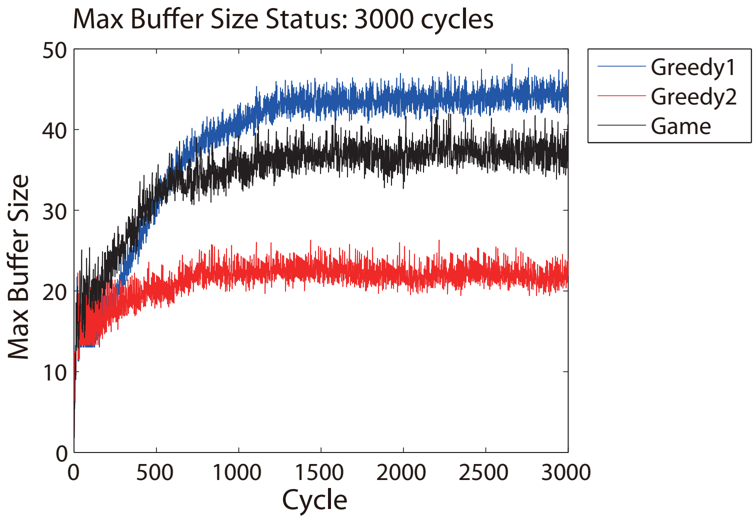

Figure 9: Maximum buffer size status

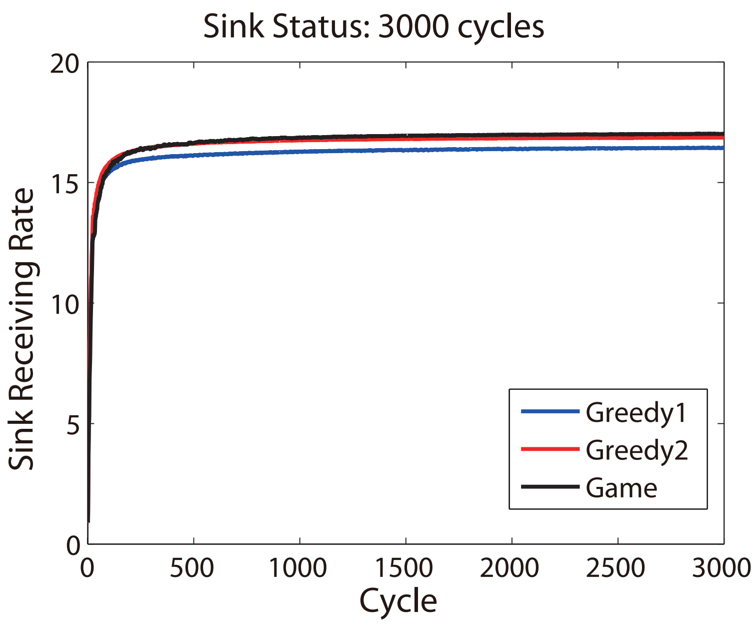

Figure 10: The throughput of our network

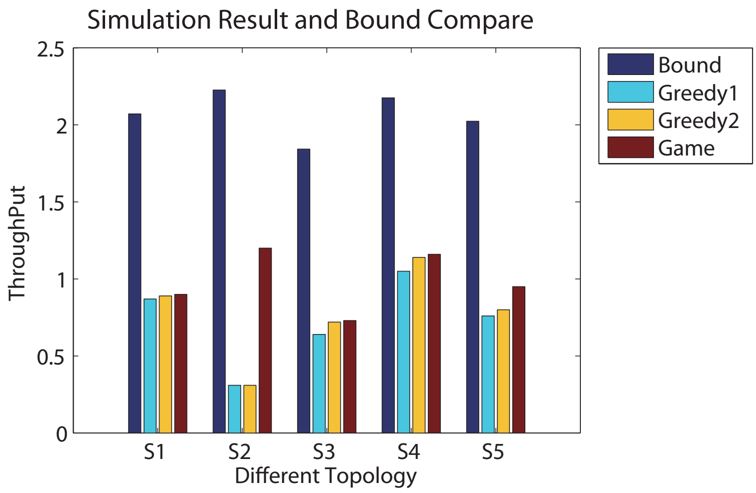

Figure 11: The comparison between algorithms and upper bound of our network

From this figure, we can find that our algorithm performs better than greedy algorithm in all five topologies. And the performance of our algorithm can achieve average up to 50 percents of the upper bound.

4 Conclusion

In our paper, we discuss the channel allocation problem in single radio multi channels wireless network to achieve high network throughput. We introduce an algorithm using game theory to solve this problem. In addition, the upper bound of the throughput of our network is also discussed in our paper. Finally, the simulation results show the comparison among our algorithm, the upper bound and the greedy algorithm. We can find that our algorithm performs better than greedy algorithm and can achieve average up to 50 percent of the upper bound result.

References

[1] Ravi Prakash, Niranjan G. Shivaratri and Mukesh Singhal,"*Distributed Dynamic Channel Allocation for Mobile Computing,*" Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing 95 Ottawa Ontario CA SESSION: Principles of Distributed Computing Pages 47-56 Year of Publication: 1995 ISBN:0-89791-710-3

[2] L. Gao, Xinbing Wang, Y. Xu, "*Multi-radio Channel Allocation in Multi-hop Wireless Networks*" in IEEE Transactions on Mobile Computing, vol 8, no. 11, pp: 1454-1468, Nov 2009.

[3] H. Li, Y. Cheng, X. Tian, and X. Wang, "*A generic framework for throughput-optimal control in MR-MC wireless networks*", accepted to IEEE INFOCOM, Orlando, Florida, Mar. 25-30, 2012

[4] Kamal Jain, Jitendra Padhye, Venkat Padmanabhan, Lili Qiu *Impact of Interference on Multi-hop Wireless Network Performance*," Microsoft Research One Microsoft Way, Redmond, WA 98052.

[5] P. Kyasanur and N. H. Vaidya, *Capacity of multi-channel wireless networks: impact of number of channels and interfaces*, in Proc. Of MobiCom 05. ACM Press, 2005, pp. 4357.

[6] Zheng Wang, Shirish S. Karande, Hamid R. Sadjadpour, and J. J. Garcia-Luna-Aceves "*on the multicast capacity of wireless ad hoc networks with network coding*," JOURNAL OF COMMUNICATIONS AND NETWORKS, VOL. 13, NO. 5, OCTOBER, 2011

[7] Pradeep Kyasanur, Member, IEEE, and Nitin H. Vaidya, Senior Member, IEEE, "*Capacity of Multichannel Wireless Networks Under the Protocol Model*," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 2, APRIL 2009

[8] Pan Li, Chi Zhang, Student Member, IEEE, and Yuguang Fang, Fellow, IEEE, "*Capacity and Delay of Hybrid Wireless Broadband Access Networks*,IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 27, NO. 2, FEBRUARY 2009

[9] Abbas El Gamal, James Mammen, Balaji Prabhakar, Devavrat Shah," *Throughput-Delay Trade-off in Wireless Networks*",IEEE INFOCOM 2004

[10] P. Chaporkar, K. Kar, and S. Sarkar, *Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks,*in Proceedings of Information Theory and Applications Inaugural Workshop, Feb. 2006.

[11] P. Gupta and P. R. Kumar, The capacity of wireless networks, IEEE

[12] M. J. Neely, *Optimal backpressure routing for wireless networks with multi-receiver diversity*, in Proc. Conf. on Information Sciences and Systems (CISS), 2006, pp. 18-25.

[13] X. Wu and R. Srikant, *Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks*, in Proc. IEEE Infocom, 2006.

[14] L. Ying, S. Shakkottai and A. Reddy, *On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks*, in Proc. IEEE Infocom, 2009.

[15] L. Tassiulas and A. Ephremides, *Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks*, IEEE Trans. Automat. Contr., vol. 4, pp. 1936- 1948, December 1992.

[16] M. J. Neely, *Delay analysis for maximal scheduling in wireless networks with bursty traffic*, in Proc. IEEE INFOCOM, Apr. 2008.

[17] T. S. Rappaport, Wireless Communications: Principles and Practice (2nd), Prentice Hall, 2002.

[18] M. Schwartz, Mobile Wireless Communications, Cambridge Univ. Press, 2005.

[19] P. Gupta and P. R. Kumar, The cpacity of wireless networks, IEEE

[20] L. Gao and Xinbing Wang, *A Game Approach for Multi-Channel Allocation in Multi-Hop Wireless Networks*, In Proceedings of the ACM Conference on Mobile Ad Hoc Networking and Computing (MobiHoc 08), pages 303-312, May 2008.