Design and Implementation of A MAC Scheme for Wireless Ad-hoc Networks Based on a Cooperative Game Framework

Zuyuan Fang and Brahim Bensaou
Computer Science Department, Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong, PRC.
E-mail: {zyfang,brahim}@cs.ust.hk

Abstract— **Due to their dynamic topologies, providing Quality of Service (QoS) in wireless/mobile ad-hoc networks introduces major new challenges to the research community. Today, the only commercially available ad-hoc network products are those based on the widely deployed IEEE802.11 Distributed Coordination Function (DCF). However DCF is a random access scheme, which not only can not provide any guarantees, but is also well known to suffer from a problem of fairness. In this paper, the bandwidth allocation problem in the medium access control (MAC) layer of ad-hoc networks is modelled as a constrained maximization problem. Based on duality, the problem is further modelled as a cooperative game and an algorithm to solve this problem is provided, and the discussion is centered around the design and implementation issues of the algorithm.**

I. Introduction

Of all the layers in the protocol stack, the MAC layer plays a very important role in maintaining QoS in ad-hoc networks, as the capability of the other layers relies on the services it provides. Today, the widely deployed MAC protocol IEEE802.11 DCF suffers from a fairness problem, which is believed to be caused by the conjunction of the inherent hidden terminal problem and the adopted binary exponential backoff algorithm. Many schemes have been proposed in the literature to overcome this problem and provide better bandwidth sharing between competing nodes. Some of these schemes depend on the sharing of link information between nodes in the network, while others try to solve the issue of fairness while maintaining topology-transparency.

Designing an efficient and fair MAC protocol can be modelled as a bandwidth allocation problem at the link layer. When considering link layer flows, contention relations between links in a wireless ad-hoc network can be represented by a link conflict graph. In such a graph, vertices represent link flows and edges between vertices denote contention between links, which means that there is interference between either the sender or the receiver of one link with either the sender or the receiver of the other link. Furthermore, in such a graph, a maximal clique (a maximal independent set in the complementary graph) represents a competition context [1], [2], in which one link can successfully carry data if and only if no other link in the clique is carrying data. Therefore, each clique represents a "channel resource", which has a given fixed capacity. The basic requirement for feasibility of a schedule or bandwidth assignment is that the total flow rate in each clique does not exceed the clique's capacity, subject to the conflict constraints. In addition, the bandwidth allocation should satisfy some performance requirement such as fairness.

In [3] the bandwidth allocation problem is modelled as a constrained maximization problem. Both a non-cooperative and a cooperative game frameworks are proposed to solve the problem, and the corresponding theoretical algorithms are derived. The present paper draws on the theoretical results of [3], and pushes further to consider the difficult problem of implementation. More precisely, while the theoretical algorithms achieve the fairness objective well, they rely strongly on the accurate availability of contention information at the network nodes. This paper addresses such issues in the practical implementation of the algorithms. However, for completeness, the model and the algorithm are briefly discussed herein (refer to [3] for more details). The remainder of the paper is organized as follows: Section II introduces the system model. Section III presents the cooperative game framework based algorithm. Section IV discusses the implementation details. Section V gives simulation results. Finally, Section VI concludes the paper.

II. Model

Assuming all nodes of the ad-hoc network use omnidirectional antennas with the same power level to transmit packets in the same shared wireless channel. A link conflict graph can be used to describe the contention relations between link flows, and each maximal clique is treated as a "channel resource" with a given fixed capacity. The capacity of a clique depends on the topology of the network, and the fairness principle under consideration. For example under the max-min fairness principle, if the topology of the network is such that the induced link conflict graph is a perfect graph[1] then the capacity of each clique can be normalized to 1. However, this is not

[1] A graph is perfect if for all of its induced subgraphs the size of the maximum clique is equal to the chromatic number.

This work is in part supported under Grant DAG03/04.EG26

true in general graphs. We can prove based on [4] that it is sufficient to systematically reduce the capacity of each clique to 2/3 in order to ensure feasibility of a bandwidth assignment.

Given conflict graph G, let the flow rate of a specific link i be x_i. According to [5], maximizing a strictly concave utility function in terms of active link flow rates can achieve system wide fairness. Following this principle, define $f(x_i)$ to be a strictly concave utility function for link i when its flow rate is x_i, let $Q(i)$ be the set of cliques that include link i, and $S(j)$ be the set of links that form clique j. The fair bandwidth allocation problem can be formulated as a constrained maximization (primal) problem $\boldsymbol{P}$ as follows:

$$\begin{aligned} \boldsymbol{P}: \quad & \max_{x_i} \quad \sum_i w_i f_i(x_i) \\ & \text{subject to:} \quad \sum_{i \in S(j)} x_i \le c_j, \quad j = 1, \cdots, M \\ & \qquad\qquad\quad x_i \ge 0, \quad i = 1, \cdots, N \end{aligned} \tag{1}$$

where c_j denotes the capacity of clique j, N is the number of links, and M is the number of maximal cliques in the link conflict graph. The constraints simply state that the total flow rate in one clique can not exceed the capacity of the clique. Since the constraints of problem $\boldsymbol{P}$ are linear inequalities, and the flow rates x_i are positive and upper bounded by the capacity of the clique, it can be shown that the feasible set is nonempty, convex and compact. In addition, the objective function is strictly concave in x_i. Therefore there exists a unique maximizer for this problem [6]. However, although the objective function is separable in x_i, solving this problem requires coordination of possibly all links in the network, which is not practical in ad-hoc networks.

III. Distributed Algorithm based on Cooperative Game Framework

By considering the dual of problem $\boldsymbol{P}$, a distributed algorithm based on the Cooperative Game Framework (CGF) is derived in [3]. The CGF algorithm is a price based algorithm. That is, in clique j, when the total flow rate is $\sum_{i \in S(j)} x_i$, the price rate is λ_j. From the viewpoint of link i, for a fixed value of λ_j, the optimal flow rate can be computed by solving the following problem [3]:

$$\max_{x_i} \left(w_i f_i(x_i) - x_i \sum_{j \in Q(i)} \lambda_j \right). \tag{2}$$

Since $f_i(x_i)$ is strictly concave, the unique maximizer for problem (2) exists and is given by:

$$x_i^* = f_i^{\prime -1} \left(\frac{\sum_{j \in Q(i)} \lambda_j}{w_i} \right). \tag{3}$$

The details of the CGF algorithm is shown in Algorithm 1.

Intuitively, (4) implies the basic requirement of supply and demand: if the total offered flow rate is less (respect. more) than the capacity of the clique, the price decreases (respect. increases).

Algorithm 1 Cooperative Game Framework based Algorithm

The algorithm is executed by each link i (the sender node in a link) round by round:

1) Initially, choose a feasible flow rate $x_i(0)$.
2) Collect local conflict information and construct local set of cliques $Q(i)$.
3) Set initial price $\lambda_j(0)$ (global parameter) for each clique j in $Q(i)$.
4) In round k, calculate a new link flow rate $x_i(k)$ according to (3).
5) Disseminate the new flow rate information to all links in one hop.
6) In round $k+1$, calculate a new price $\lambda_j(k+1)$ for clique j as
$$\lambda_j(k+1) = max \left(0, \lambda_j(k) + \gamma(\sum_{i \in S(j)} x_i^* - c_j) \right), \tag{4}$$
where γ is a step size, c_j is the capacity of clique j, $\sum_{i \in S(j)} x_i^*$ is the total flow rate in clique j in the previous round k.
7) If the local conflict graph has changed (e.g., due to mobility), go back to (2), otherwise go back to step (4).

It can be proven that under the appropriate value of step size γ, for any initial feasible flow rate x_0 and price λ_j^0, any accumulation point (x^*, λ^*) generated by the algorithm is primal-dual optimal. Therefore, the system is globally stable. Define $\bar{Q} = max_{i \in \mathcal{N}} |Q(i)|$ as the largest number of cliques that contain the same link. Denote $\bar{S} = max_{j \in \mathcal{M}} |S(j)|$ as the maximal size of a clique. Let $\bar{\alpha}$ be the upper bound of function $-f''(x)$, then the range of the step size can be defined as in [7]:

$$0 < \gamma < \frac{2}{\bar{\alpha} \bar{Q} \bar{S}}.$$

IV. Implementation

The CGF algorithm depends on the availability of local link contention relations and their flow rate information. Therefore, collecting such information in a dynamic environment is the key issue to the design and implementation of the algorithm. The design and implementation of the CGF algorithm based on modifying IEEE802.11 DCF are presented in this section, other approaches can be also possible by finding an approach to i) collecting conflict and rate information, ii) deploying an access mechanism that is capable of achieving the calculated rates.

A. Local link conflict graph construction and flow rate information collection

Control Channel: In wireless networks in general, the interference range of one link is determined by the distance between the sender and the receiver, the distance of the interfering node from either the sender or the receiver, as well as different power thresholds used to validate received data or dismissing it as noise. As shown in Fig 1, the distance between the sender A and the receiver B is d, the transmission range for link i, R_{tx} is determined by a threshold on the received SINR. According to [8], if the interference range for the receiver is denoted as R_i, then it can be calculated as $R_i = 1.78 \times d$ (based on the simplest propagation model and assuming interference comes from one single node). If $d \geq 0.56 \times R_{tx}$, then the interference range exceeds the transmission range. Therefore, the interference range of link i can be shown by the area enclosed by the dotted line in the figure. In such a configuration, nodes that are within the transmission range of each other can overhear and exchange information with each other (dubbed here packet sensing). For example, in Fig 1 node B can overhear link information of flow j. However, if two nodes interfere with each other while being out of the transmission range of each other, they can not exchange information, leading to incomplete information about the conflict graph. For example, links i and k interfere with each other but they can not overhear each other, they can only sense each other through the presence of energy on the channel (carrier sensing).

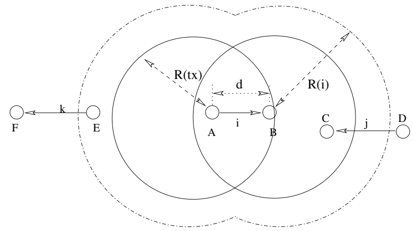


Fig. 1. The control channel and data channel span

In order to construct local link contention graph for a given link, all link information within the interference range should be collected by the sender of the link. To achieve this objective, we adopt an out of band control channel in our implementation. In the control channel, the transmission range is set to the interference range of the data channel. As an approximation of this, we adopt a scheme called Conservative CTS Reply (CCR) proposed in [8] to let the interference range equal the transmission range for both the control channel and data channel. In the CCR scheme, a node replies to a CTS packet only when the received power is larger than the CTS-REPLY-THRESHOLD, which is larger than the threshold adopted by IEEE802.11 DCF. The principle of the CCR scheme is to restrict the establishment of links such that the distance between the sender and the receiver is such that the interference range equals to the transmission range. Control messages that are used to construct the conflict graph and exchange flow rate information are transmitted in the control channel.

In order to create the two channels (control and data channels), a non-preemptive prioritizing has been adopted as a good approximation of a system with two separate physical channels. In the implementation, both the control channel and the data channel use the whole radio spectrum of the physical layer. They are however prioritized by assigning a smaller inter-frame space to the control channel, and its backoff is always done with the smallest possible contention window. Control messages also use the basic access method of IEEE 802.11 DCF (CSMA/CA). As such, control messages are always guaranteed to be sent out before data packets, and they are guaranteed a shorter IFS and a shorter backoff than data packets.

Control messages: A link involves a sender and a receiver. In the CGF, the sender is responsible for controlling the behavior of the link flow. A link is represented by a link ID, which is the concatenation of the ID of the sender and the ID of the receiver. Two types of control messages are broadcast in the control channel. The first is the link "beacon" message. Each sender is required to periodically broadcast such beacon messages for each of its links, to announce existence of such a link and refresh them subsequently. The beacon message contains the flow ID, rate information (comprising two fields rate_0 and rate_1), and a round number for synchronization purpose. Beacons can be detected by neighboring nodes. The field rate_0 carries the value of the flow rate chosen for this link during the previous round of execution, while rate_1 carries the rate chosen in the current round. Initially these two fields are the same. They are needed to maintain coherent progression of the distributed algorithm: when the algorithm for a given link progresses to the next round, some of its neighbors may still be expecting rate information from the previous round. It can be shown easily that there cannot be local asynchrony of more than one round: in the worst case, the leading-most link among its neighbors completes round k and stops waiting for information to start round k+1, and the lagging-most link is just starting round k computations. The second type of control messages is the so called flow information message. Flow information messages consist of a collection of all overheard beacons including the link's own beacon, grouped into a packet with a header that contains notably the initiator node ID. Flow information messages are broadcast periodically.

To illustrate how the two types of messages are used we refer back to the example of Fig. 1. In order for sender A to announce to sender D that links i and j are in conflict, Sender A sends a beacon message for link i to receiver B who in turn recast back the beacon message to A. For short, we coin this peer-casting. Peer-casting works in principle similarly to the well know RTS/CST mechanism. It however fulfills a different purpose: any sender within the vicinity of A or B can overhear the beacon for link i, and thus knows that all its links conflict with link i, and also knows the rate information for link i. Flow information messages on the other hand are used to propagate contention information further one hop away, to cover the cases when the contention happens through the receiver(s). For example, in Fig 1, it is clear that peer-castings of sender A will only reach receiver C. If receiver C collects all the beacons it overheard into one information message and broadcast it, any node in its vicinity, including sender D, will be aware of all links that contend with node C and by extension, sender D can infer all hidden links that contend with all its links, including link j. Node D will know not only these contention relations but also the flow rate information of all such contenders. Flow

0-7803-8533-0/04/$20.00 (c) 2004 IEEE

information messages can be initiated by either the sender or the receiver. Under the assumption that the time needed for peer-casting and broadcasting is bounded[2] the sender of a flow can construct its local link conflict graph and exchange flow rate information with neighboring nodes within a bounded time.

B. Flow Rate Adjustment

The bandwidth allocated by the CGF algorithm is a normalized capacity. One link needs to adjust its flow rate according to the allocated capacity. If the physical capacity is known, the normalized capacity can be converted to the physical capacity and a link can then send a flow rate according to the physical value. In this paper, since the CGF algorithm is implemented by modifying IEEE802.11 DCF, the flow rate is adjusted by changing the contention window, guaranteeing thus the rate only statistically. Assume the bandwidth allocated to link i is r_i, then the contention window CW can be adjusted according to:

$$CW = min(\frac{\delta}{r_i} CW_{min}, CW_{max}), \quad (5)$$

where CW_{min} is the minimal contention window, CW_{max} is the maximal contention window, and where $\delta (\geq 1)$ is a scaling factor, which depends on the density of the network. If the density of a network is large, δ takes a large value and vice versa. In our implementation, δ is set as a constant. As we can see from (5), the smaller the value of r_i, the larger the contention window, and the smaller the flow sending rate. The flow rate is thus adjusted according to the requirement. The minimal flow rate is bounded by the maximal contention window.

C. Diagram of the Algorithm

When running the algorithm, in the control channel, the sender switches between four states: local link conflict graph construction, flow rate information exchange, flow rate calculation and idle, as shown in Fig 2(a). When a link becomes active, it starts from the flow construction state and sends a beacon message and sets a timer. After an overhearing period, it can start initiating flow information messages. Once the sender detects that the topology is static (no more new information received within a given time), it begins to construct its link conflict graph and decomposes the graph into a set of cliques. Then the system transits to the flow rate information exchange state. In this state, the nodes broadcast and collect flow rate information. If the sender has obtained all flow rate information of neighboring links, it enters the flow rate calculation state. In this state, the sender calculates a price for each clique, and calculates the flow rate for itself. Depending on whether the system converges (the rate changes by a very small fraction only) or not, the system will either enter the flow rate information exchange state or the idle state. If the link is inactive, the system also transits to the idle state. A threshold is set to indicate the convergence of the algorithm. After each round, the sender computes the difference of the new value of the flow rate to the old value, if the difference is less than the threshold, then the sender assumes that the algorithm has converged.

[2]This is not the case in theory, due to the nature of the access method (CSMA), however in practice it is almost always the case since the control channel is designed such that it is under-loaded.

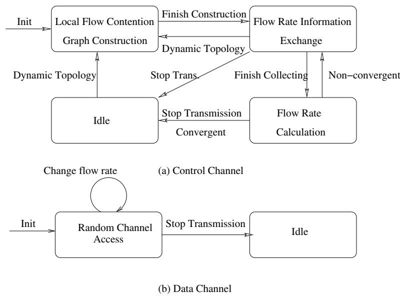


Fig. 2. The diagram of the algorithm

In the data channel, the sender switches between two states: random channel access and idle. Initially, the link uses the initial contention window (determined by the initial flow rate) to access the channel. If the link calculates a new flow rate, then the contention window is updated, and the link uses the new value to access the channel. If the link stops transmission, then the sender enters the idle state.

In the execution of the algorithm, changes in the topology of the network caused by node mobility or power on/off will cause the nodes to reconstruct their link conflict graphs.

V. Performance Evaluation

The CGF algorithm has been implemented in NS2. In this section, some simulation results are presented to illustrate the correctness and effectiveness of the algorithm.

A. Simulation setting and performance metrics

The network topology of the scenario is shown in Fig 3. In the following simulations, the transmission range is set to $250m$, and the channel bandwidth is set to $11Mbps$. Each sender initiates a Constant Bit Rate (CBR) traffic. In one simulation, the sources use the same flow rate. Different flow rates have been simulated. In our implementation, to ensure the feasibility of the rates, in random graphs, the capacity of the cliques has been normalized to $\frac{2}{3}$ instead of 1. The initial price is set to 0.1, the scaling factor $\delta = 1$ and the system parameter w_i is also set to 1. The utility function is defined as:

$$f_\alpha(x) = \begin{cases} logx, & \text{if } \alpha = 1 \\ (1-\alpha)^{-1} x^{1-\alpha}, & \text{otherwise} \end{cases} \quad (6)$$

The long term fairness property is measured on the average goodput, that is, only successfully received DATA packets are accounted for.

B. Comparison of numerical results and simulation results

Fig 4(a) compares the normalized bandwidth allocation obtained by numerical analysis to that obtained by simulation. Results show that under the control of the algorithm, the system converges, and simulation results match well the

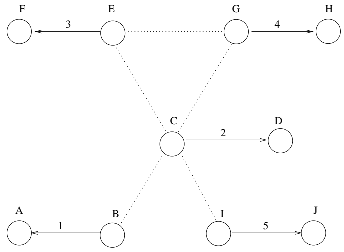
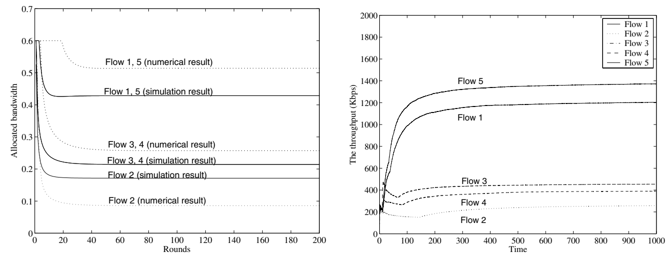
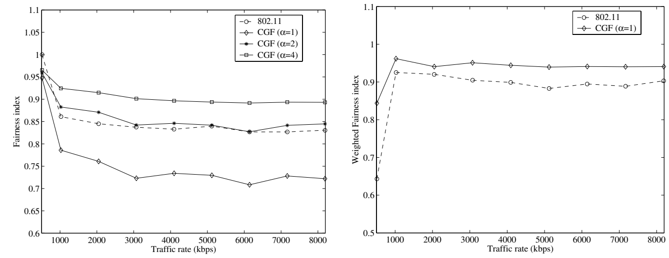


Fig. 3. The link conflict graph of the network



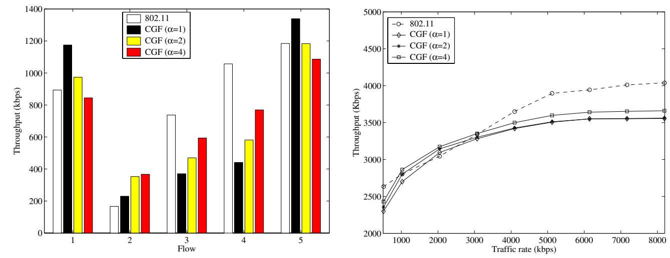
(a) Allocated normalized bandwidth (b) Individual throughput

Fig. 4. Theoretical result vs. simulation result



(a) Global (b) Proportional Weighted

Fig. 5. Fairness Index



(a) Individual throughput (8192K) (b) System throughout

Fig. 6. Throughput

numerical results (despite the coarse implementation with priority rather than the dual channel MAC). Fig 4(b) shows the achieved individual throughput under the control of the CGF algorithm, which also matches the theoretical results.

C. Fairness property

The simulation results shown in Fig. 5 and 6. Fig. 5(a) shows the global fairness index of the network $(\sum x_i)^2/N\sum x_i^2$, under the control of the DCF and CGF with different fairness objectives (by adjusting parameter α). It is shown that when the channel is not saturated (source rate is low), there is no fairness problem with the network. When the traffic is high, the network faces a fairness problem and the CGF algorithm can achieve the fairness objective. As seen from the figure, the larger the value of α, the globally fairer is the algorithm. When $\alpha = 1$, the global fairness index of the CGF is smaller than that of 802.11 DCF, which is natural since in this case the CGF targets proportional fairness (which is a local objective). If we examine the proportional weighted fairness index (obtained by replacing above the rate x_i by the weighted rate x_i/w_i) as shown in Fig. 5(b), the system is fairer under the control of the CGF algorithm. This shows that the CGF indeed achieves proportional fairness when $\alpha = 1$. Fig. 6(a) shows individual average throughput. It can be seen that flow 2, which faces the most competitive environment, achieves more throughput under the control of the CGF algorithm. Fig. 6(b) shows that while the CGF achieves the target fairness objectives, it does not sacrifice too much aggregate throughput.

VI. CONCLUSION

The bandwidth allocation problem in mobile ad-hoc networks is modelled as a constrained maximization problem. By considering the dual problem of this system problem, a distributed algorithm based on the gradient projection method is derived . It is shown that under some reasonable conditions, the distributed algorithm leads the system to the unique optimal point. In such a point, the bandwidth allocated to the links satisfies a specific fairness objective, which is determined by the adopted concave utility function. The design and implementation of the algorithm are discussed from the perspective of the difficulty of implementing such algorithms in real systems. Solutions to such practical problem are proposed and integrated in the NS2 implementation of IEEE 802.11 DCF. Such simulations confirm the fairness and stability of the algorithm.

REFERENCES

[1] A. Bar-noy, A. Mayer, B. Schieber, and M. Sudan, “Guaranteeing fair service to persistent dependent tasks,” *SIAM J. COMPUT*, vol. 27, no. 4, pp. 1168–1189, August 1998.

[2] X. L. Huang and B. Bensaou, “On Max-Min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation,” in *ACM MobiHOC*, 2001, pp. 221–231.

[3] Z. Fang and B. Bensaou, “Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks,” in *IEEE INFOCOM*, 2004.

[4] C. E. Shannon, “A theorem on coloring the lines of a network,” *J. Math. Phys.*, vol. 28, pp. 148–151, 1949.

[5] S. Shenker, “Fundamental design issues for the future internet,” *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1176–1188, 1995.

[6] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGRAW-HILL International Editions, 1996.

[7] S. H. Low and D. E. Lapsley, “Optimization Flow Control–I: Basic Algorithm and Convergence,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[8] K. Xu, M. Gerla, and S. Bae, “How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?,” in *IEEE Globecom*, 2002.