Game Theory and Learning at the Medium Access Control Layer for Distributed Radio Resource Sharing in Random Access Wireless Networks

Eric Ayienga[1], Elisha Opiyo[1], Bernard Manderick[2] and Okelo Odongo[1]
[1]*School of Computing and Informatics, University of Nairobi, Nairobi, Kenya*
[2]*Computational Modeling (COMO) Lab, Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium*
{*ayienga, opiyo, wokelo*}*@uonbi.ac.ke, bmanderi@vub.ac.be*

Keywords: Distributed Channel Sharing: Random Access: Medium Access Control: Game Theory: Strategy: Exploration: Reinforcement Learning: Q-learning.

Abstract: Game theory is not only useful in understanding the performance of human and autonomous game players, but it is also widely employed in solving resource allocation problems in distributed decision-making systems. Reinforcement learning is a promising technique that can be used by agents to learn and adapt their strategies in such systems. We have enhanced the carrier sense multiple access with collision avoidance mechanism used in random access networks by using concepts from the two fields so that nodes using different strategies can adapt to the current state of the wireless environment. Simulation results show that the enhanced mechanism outperforms the existing mechanism in terms of throughput, dropped packets and fairness. This is especially noticeable as the network size increases. However the existing mechanism performs better in terms of delay which can be attributed to increased processing.

1 INTRODUCTION

The fundamental access mechanism adopted by the IEEE 802.11 MAC (medium access control) is a *distributed coordination function (DCF)* known as *carrier sense multiple access with collision avoidance (CSMA/CA)* (IEEE, 1999; Committee, 2007). This function makes coordinated wireless medium access possible when multiple nodes are vying for access in an uncoordinated manner. This mechanism is a distributed strategy to access and share the wireless channel among contending nodes in random access wireless networks (Forouzan, 2007). It consists of two components: a contention resolution mechanism that is used for contention before the start of a transmission, and a feedback mechanism (known as backoff) that updates a contention measure and sends it back to the wireless nodes when a collision occurs.

The current CSMA/CA mechanism still has some shortcomings that lead to inefficiency and unfairness. These include: The inherent weaknesses of the mechanism and parameter manipulation by self-interested nodes (Cagalj et al., 2005; Kyasanur and Vaidya, 2005; MacKenzie and DaSilva, 2006; Raya et al., 2006; Buttyan and Hubbaux, 2007). Approaches to optimizing it involves tuning its parameters i.e. the *interframe space (IFS), contention window (cw), slot length, backoff time (BO)* and *control message lengths*. We try to address these shortcomings through the adaptive tuning of the parameters using concepts borrowed from the fields of game theory and reinforcement learning.

The characteristics of random access networks give rise to a strategic setting in which the physical network is the environment and the nodes are the agents. Past research shows that game theory has been applied to the four bottom layers of the OSI reference model in wireless networks as summarized in (Felegyhazi and Hubaux, 2006; Chen et al., 2007; Cui and Chen, 2008). In particular we concentrate on the MAC layer under which the CSMA/CA mechanism falls.

In idealistic models of non-stationary environments, nodes should continually adapt their strategies based on the state of the environment. Because of non-stationarity, the nodes should be more sensitive to the trade-off between *exploitation*, which uses the best strategy so far, and *exploration*, which tries to find better strategies. Exploration is especially important in non-stationary environments e.g. ad hoc wire-

less networks or sensor networks (Zhu and Ballard, 2002; Zhu, 2003) and hence the use of reinforcement learning. Most existing reinforcement learning (Sutton and Barto, 1998) algorithms are designed from a single-agent's perspective and for simplicity assume the environment is stationary. The predominant approaches to game playing in those settings assume that opponents' behaviors are stationary.

As opposed to the current approach in which fairness and efficiency drop as the tunable parameters are manipulated, the main contribution of this work was the use of game theory and reinforcement learning to design and improve the existing mechanism such that the random access network stabilizes around a steady state with increased fairness and efficiency. Results from the experiments performed through simulations are quite encouraging. The modified mechanism outperforms the existing mechanism in throughput, dropped packets and fairness as the network gets heavily loaded. However, the existing mechanism does better in terms of delay when the network gets heavily loaded.

This paper is organized as follows: Section 2 gives the preliminaries of the CSMA/CA random access mechanism, game theory and reinforcement learning. Section 3 presents a model for applying game theory and learning at the MAC layer. Section 4 presents some simulation results and analysis. Section 5 concludes the research and gives some direction for further work.

2 PRELIMINARIES

2.1 The Distributed Coordination Function (DCF)

The basic principle of the CSMA/CA mechanism is listen before talking (carrier sensing) and contention (collision avoidance) as illustrated in Figure1.

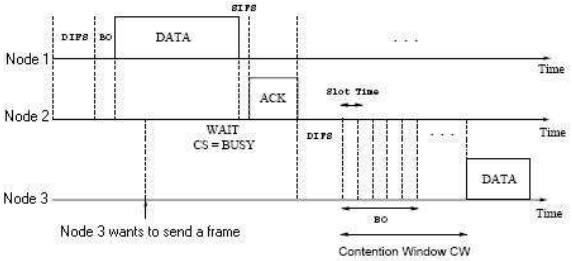

Figure 1: The CSMA/CA Basic Access Mechanism. This timing diagram show 3 nodes competing for the same radio channel. Node 1 transmits to Node 2 while Node 3 waits to transmit.

From the Figure, the following things should be noted.

$$DIFS = SIFS + 2(slot\,times) \tag{1}$$

1. If the channel is sensed "idle" for a DIFS (distributed inter-frame space) time, then the node begins a random backoff (BO), where

$$Random\,backoff = Random[0, cw] \times slot\,time \tag{2}$$

   - *Random* is a random number generator function which randomly selects a number from a uniform distribution [0, *cw*].
   - *Contention Window (*cw*)*: A number computed using the equation:

$$cw = 2^{BE} - 1 \tag{3}$$

     The *Backoff Exponent (BE)* enables the computation of the *cw* value. Some of the values used include 3, 5 etc. to give *cw* values of 7, 31 etc. The *cw* is bounded by $cw_{min} \leq cw \leq cw_{max}$.
   - *Slot time*, is defined under the physical layer: It is equal to the time needed by any node to detect the transmission of a packet from any other node.
2. If the backoff expires and the channel is still free, the node transmits the frame
3. If the channel is sensed "busy" during the backoff, the backoff is paused until the channel is sensed idle again, and then resumed.

If on transmission the frame is received correctly, the receiver sends back an acknowledgment (ACK) frame after a short inter-frame space (SIFS) period. The random backoff process before sending a data frame is used for mitigating collisions. After each successful transmission, *cw* is reset to the minimum contention window size cw_{min}. If a frame is lost (An ACK frame is not received), the *cw* is doubled if the maximum allowable contention window size cw_{max} has not been reached yet. Otherwise, cw_{max} is used as the new *cw* up to a maximum of 15 times. If this is exceeded, the transmission is aborted.

2.1.1 Intrinsic imperfection of the CSMA/CA mechanism

Based on the parameters used by the CSMA/CA mechanism, the following issues arise:

1. *Efficiency*: Based on the size of the *cw* and *BO* there may be a lot of collisions. With this increase in the probability of collisions, the number of *lost packets* increases as well. This reduces the *throughput* and increases the *delay*. Additionally,

the medium may also remain idle for long periods of time while some nodes still have data to transmit . This makes the mechanism probabilistic and unsuitable for time-sensitive applications.

2. *Fairness*:
   - Under contention, unlucky nodes will use a larger *cw* and *BO* than lucky nodes. This is based purely on the random manner in which the two values are obtained.
   - Similarly, a node which is successful in transmission sets its *cw* to the minimum size (cw_{min}) while other nodes are still counting down. This has a possibility of giving the successful node a higher probability of accessing the channel with its next attempt.

2.1.2 Protocol manipulation by nodes

Since a decision made by any node effects other nodes, it is implicitly assumed that the nodes follow the prescribed protocol without any deviation when performing network functions. In general, if nodes are owned by autonomous entities and their objective is to maximize their individual goals through strategic behaviour, then such nodes are said to exhibit *self interest* (Narahari et al., 2009). Nodes with self-interest could manipulate the operation of the mechanism in order to maximize their utility in the following manner (Cagalj et al., 2005; Raya et al., 2006; Konorski, 2006; Zhao, 2006):

1. Selectively scrambling frames sent by other nodes forcing them to increase their *cw*s. Targeted frames include CTS, RTS and data frames.
2. Manipulating the protocol parameters;
   - *Using shorter DIFS*
   - *BO manipulation*; The nodes choose a small or fixed *cw*, thus, the backoff interval is always short.

Nodes serving their self-interest inevitably contribute towards inefficiency and unfairness in network performance.

Two parameters that determine optimum network behaviour that can be manipulated are thus *cw* and *BO*. We can therefore argue that nodes using the CSMA/CA mechanism as currently implemented, are hard-wired to use a single strategy. This strategy uses moves based on *cw* and *BO* to ensure fairness and efficiency. Both cases depend on *randomization*. The moves made by the nodes are not premeditated i.e. as a result of or an anticipation of opposing nodes' behavior. They are simply a reaction to the state of the environment with no adaptivity. This can lend itself to abuse and/or suboptimal behavior. Optimization of the network performance involves seeking a stable operating point based on the parameters.

We used game theory to analyze this interaction, and based on the analysis we proposed an enhancement to make the mechanism more *adaptive* to network environment conditions.

2.2 Game Theory

A game comprises of at least two participants called *players*. A player may be an individual, a company, a nation, a wireless node, a biological specie etc. Each game consists of a set of *actions* or a sequence of moves. These are either decisions by the players or outcomes of chance events, which could be *sequential* or *simultaneous*. At the end of the game, each player receives a *payoff*, which is always assumed to be a real number. For the number to reflect the player's preferences, or all aspects of the outcome, it is represented numerically by its *utility value* obtained from a *utility function*. Our proposed random access game can be generally formalized and expressed as shown in the Definition below:

Definition .1. $G = \langle N, A, u_i \rangle$
Where

- *Game (G): A finite n-person game (N, A, u).*
- *Players (N): A finite set of n players indexed by i. They are all IEEE 802.11 nodes.*
- *Actions (A): $A = A_1 \times A_2 \times \cdots, \times A_n$ are action sets for each node i, where A_i is a choice of cw and BO parameters for each node i.*
- *Utility: $u_i = \{u_1, u_2, \cdots, u_n\}$, where u_i is the utility (throughput, delay, lost packets or fairness) that the players wish to maximize.*

2.2.1 Learning in Strategic Games

Game theory has traditionally been developed as a theory of strategic interaction among players who are perfectly rational, and who (consequently) exhibit equilibrium behaviour. This approach has been complemented by evolutionary game theory (EGT), which motivated by biological evolution, seeks to understand how equilibria could arise in the long term by selection among generations of players who need not be rational or even conscious decision-makers. Somewhere in between are models of learning, which apply to adaptive behaviour of goal-oriented players who may not be highly rational (in a game-theoretic sense), both to provide foundation for theories of equilibrium and to model empirically observed behaviour (Erev and Roth, 1998).

These models take many different forms, depending on the available information, the available feedback, and the way they are used to modify behaviour, giving rise to different models of adaptive learning in a continuum. In our settings there are two possible things to be learnt:

1. Nodes learn other nodes' strategy so that they can devise the best (or at least a good) response.
2. Nodes learn a strategy of their own that does well against the competing nodes without explicitly learning those nodes' strategies.

We concentrate on the second which is sometimes called a model-free learning an example of which is *reinforcement learning*.

2.3 Reinforcement Learning

In reinforcement learning, learners are assumed to have incomplete knowledge of the environment they are embedded in, and act in a simple stimulus-response way: the propensity to repeat a certain decision is positively related to the amount of satisfaction the learner obtained as a result of making the decision in the past.

Q-learning (Watkins and Dayan, 1992) is a form of model-free reinforcement learning. It provides agents with the capability of learning to act optimally in Markovian domains by experiencing the consequences of actions, without requiring them to build maps of the domains.

2.3.1 Multiplayer Environments

The traditional Q-learning is effective for a single learner in a stationary environment. It is problematic for multiple learners in a non-stationary environment because:

- The agent learns deterministic policies, whereas mixed strategies are generally needed;
- The environment is generally non-stationary due to adaptation of other agents.

3 APPLYING GAME THEORY AND LEARNING AT THE MAC LAYER

The strategy s_i of node i define its decisions, taking the decisions of other nodes into account. In the current CSMA/CA mechanism, the strategy defines how persistent a node i, is in contending for the available bandwidth on the channel by adjusting the *cw* and *BO* parameters. In our modified CSMA/CA mechanism, we denote the strategy space of node i by s_i. The increased strategy space of the players define the strategy profile:

$$s = s_1, \ldots, s_n \tag{4}$$

All nodes other than player i are denoted by '$-i$', and their strategy profile by:

$$s_{-i} = s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \tag{5}$$

Assuming the nodes to be rational, their objective is to maximize their payoffs or utility function in the network. We denote the payoff of node i by u_i. We assume that each node i wants to maximize its total throughput t_i, reduce delay d_i, reduce dropped packets p_i and increase network fairness f. Thus its payoff function is written as follows:

$$u_i = f(t_i, d_i, p_i, f) \tag{6}$$

The total utility is defined as the sum of the achieved utilities of all of the nodes on channel c, given by:

$$u_c = \sum_i u_i \tag{7}$$

This is a non increasing function of the number of nodes deployed on c. If the channel sensing is perfect, u_c is independent of the number of nodes on c for the CSMA/CA protocol. In practice, the *cw* and *BO* values used in the CSMA/CA protocol implementation are not optimal; and owing to packet collisions, u_c becomes a decreasing function of the number of nodes on c.

To characterize stability in the MAC game, we introduce the concept of the Nash Equilibrium. The strategy profile $s^* = s_1^*, \ldots, s_n^*$ defines a Nash Equilibrium (NE), if for each node i, and its strategy $s_i' \in S_i$ we have $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*)$. This means that in a Nash Equilibrium, none of the nodes can unilaterally change their strategy to increase their payoffs.

Two issues arise with using the basic Q-learning algorithm for nodes using the CSMA/CA protocol

- In Q-learning, the Q-matrix is only used once there is convergence. In this case, the end of the game can not be pre-determined and so is assumed to be infinite. The convergence of the Q-matrix for the nodes is therefore only theoretically possible. Nodes are therefore highly unlikely to benefit from a converged Q-matrix.
- Convergence is unlikely since the environment is dynamic.

We extend the Q-learning algorithm to the multi-agent stochastic game setting by having each agent simply ignore the other agents and pretend that the

environment is passive. Our modified algorithm is a combination of the Win-Stay, Lose-Shift (WSLS) strategy (Nowak and Sigmund, 1993) and Q-learning (Watkins, 1989). A simple learning paradigm for iterated normal form games in an evolutionary context. Following the decision theoretic concept of satisficing we design players with a certain aspiration level. If their payoff is below this level, they change their current action, otherwise they repeat it. The modified algorithm is as outlined below.

```
Begin:
For every contention
 1. Consult the Q-matrix
 2. Use the best action from the stored Q-values
 3. Evaluate the reward
    If positive
      Update the Q-matrix
    Else
      Randomly explore another action from the
      stored Q-values
End.
```

This algorithm has both:

- *Exploration*: When a node fails the contention
- *Exploitation*: When a node wins the contention

This modified algorithm will benefit the nodes by allowing them to make informed strategic decisions based on past history. This is because as the node learns and updates its Q-matrix, the matrix reflects the direction of convergence. During each episode, the Q-matrix remain the same or gets better compared to the previous one. Therefore the Q-values for each *state-action* pair represent how each move benefits a particular node at a particular time.

Our implemented Q-matrix is a six by six matrix where, the rows represent the previous state *s* and the columns represent the next state *s'* a node arrives at after taking an action *a*. The *Q(s, a)* values represent the appropriateness of taking such an action. The six states explored in our work are as shown in Table 1:

Table 1: Typical mapping of ad hoc network components to a game

State	Parameters
s_1	standard *cw* size and a slower rate of increase of the *BO* window
s_2	standard *cw* size and fixed *BO* window
s_3	standard *cw* size and standard rate of increase of the *BO* window
s_4	small *cw* size and a slower rate of increase of the *BO* window
s_5	small *cw* size and fixed *BO* window
s_6	small *cw* size and standard rate of increase of the *BO* window

4 RESULTS AND ANALYSIS

4.1 Simulation Characteristics

The proposed protocol was compared to the existing protocol by simulation through the use of the Opnet Modeler simulator. Different network sizes having similar topologies, running the two protocols were subjected to the same traffic types. The simulation characteristics were as follows.

Table 2: Parameters used in the simulation

Charateristic	**Type**
Network Size	4, 7 and 13 nodes
Bandwidth	11 Mbps
Radio type	DSSS
Applications	File Transfer: Heavy Database Access: Light Email: Heavy

4.2 Performance Metrics

The proposed model was evaluated against the existing MAC protocol based on

- *Efficiency*: The capacity of a MAC protocol is usually expressed in terms of its *efficiency* - the fraction of transmitted packets that escape collisions. In other words, the efficiency identifies the *maximum throughput* rate for a MAC protocol. This is the ratio of the successful transmission to the total number of transmissions.
- *Delay*: Average delay experienced per packet. This is influenced by the network load.
- *Dropped Packets*: Packets dropped when the load increases. This is an indication of the efficiency of the access mechanism.
- *Fairness*: How well the system shares bandwidth among multiple users. It is an important consideration in most performance studies especially in distributed systems where a set of resources is to be shared by a number of users. Assuming that fair implies equal and that all paths are of equal length, Raj Jain (Jain et al., 1984) proposed the following fairness index. Given a set of a set of flow throughputs, (x_1, x_2,..., x_n) the fairness index $f(x_i)$:

$$f(x_i) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{8}$$

The fairness index always results in a number between 0 and 1, with 1 representing greatest fairness.

4.2.1 Throughput

Figure 2 compares performance of the enhanced mechanism and the existing mechanism in terms of the global throughput of the network.

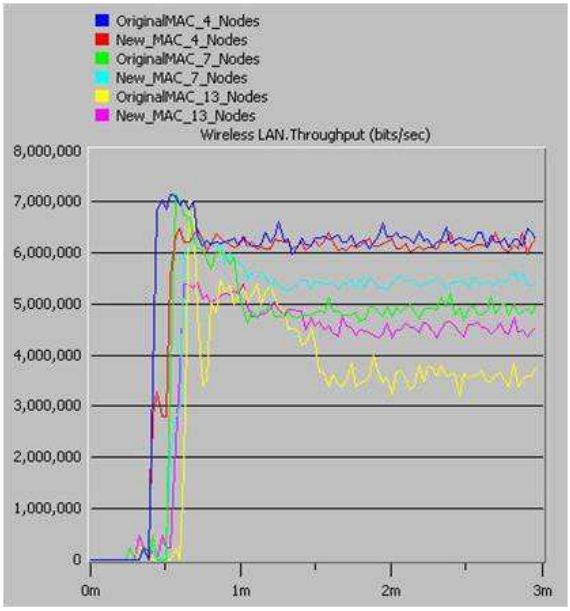

Figure 2: The global throughput comparing the enhanced protocol against the existing protocol using different network sizes

From the graph, one notices that the enhanced access mechanism performs better that the existing mechanism in all network sizes. As the network size increases, the difference is even much bigger.

4.2.2 Delay

Figure 3 compares performance of the enhanced mechanism and the existing mechanism in terms of the global delay of the network when transmitting packets. From the graph one notices that the existing mechanism perform better than the enhanced mechanism. This is expected because of the additional process of evaluating the best strategy and updating the Q matrix.

4.2.3 Dropped Packets

Figure 4 compares performance of the enhanced mechanism and the existing mechanism in terms of the global number of packets dropped by the network when nodes transmit. Here the enhanced mechanism performs far much better than the existing mechanism more so as the network size gets bigger. This indicates that the number of collisions has substantially

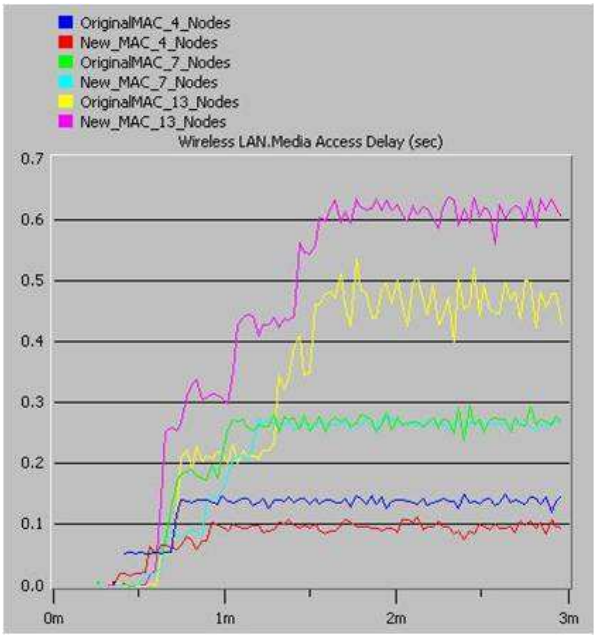

Figure 3: The global delay compares the delay of the enhanced protocol against the existing protocol using different network sizes.

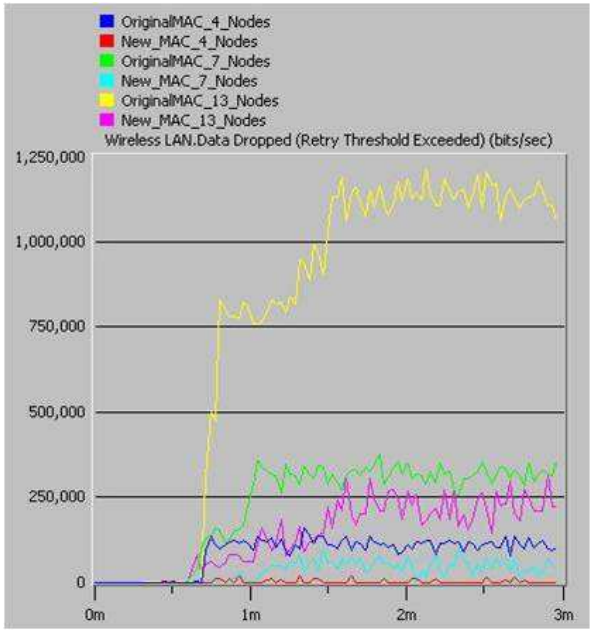

Figure 4: The global number of packets dropped compares the enhanced protocol against the existing protocol using different network sizes

reduced. This could be attributed to exploitation and learning where each node uses a better strategy as time progresses. This also contributes to the increase in throughput when the network uses the enhanced protocol.

4.2.4 Fairness

Figure 5 compares performance of the enhanced mechanism and the existing mechanism in terms of the global fairness among nodes as they compete for a shared resource. It can be observed that the original

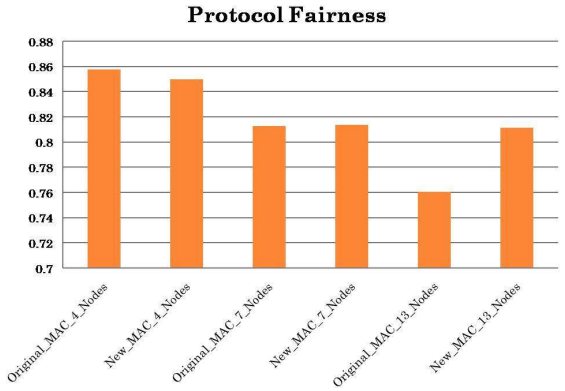

Figure 5: The global fainess comparing the enhanced protocol against the existing protocol using different network sizes.

mechanism performs better when the network is lightly loaded. But as the network becomes heavily loaded, the enhanced mechanism displays better fairness features. When the network is lightly loaded, the competition for the scarce radio resource is minimal and the resource is shared equally. As the load increases, the competition gets stiffer and there is now need to use better strategies against opponents. This calls for the use of more intelligent mechanism. The enhanced mechanism has these features inbuilt.

In all cases one notices that the graphs generated by the enhanced mechanism are smoother. This is an indication od stability as the network settles around a steady state.

5 CONCLUSIONS AND FUTURE WORK

Analysis of the simulation results indicates that the enhanced mechanism outperforms the existing mechanism in terms of throughput, dropped packets and fairness. This is more defined as the network size increases. The new mechanism additionally generated a more stable equilibrium candidate that produced different strategies in different network environments. However the existing mechanism does better in terms of delay. This is as a result of the additional processing required by the new mechanism.

There is still a lot of work to be done which would comprise of future work. This involves improving on the strategies and the strategy space. Additionally experiments need to be done to determine the optimal learning rate.

ACKNOWLEDGEMENTS

This research was supported in part by the University Cooperation for Development of the Flemish Interuniversity Council (VLIR-UOS-IUC) Project and the University of Nairobi. We thank Husna Hariz for her help in the coding and useful discusion and comments. We also thank the COMO lab, VUB for the support they accorded us.

REFERENCES

Buttyan, L. and Hubbaux, J.-P. (2007). *Security and Cooperation in Wireless Networks*. Cambridge University Press.

Cagalj, M., Ganeriwal, S., Aad, I., and Hubaux, J. P. (2005). On selfish behavior in csma/ca networks. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 4(C):2513–2524.

Chen, L., Low, S. H., and Doyle, J. C. (2007). Contention Control: A Game-Theoretic Approach. In *Proceedings of the IEEE Conference on Decision and Control*. IEEE.

Committee, L. S. (2007). IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184.

Cui, T. and Chen, L. (2008). A Game-Theoretic Framework for Medium Access Control. *IEEE Journal on Selected Areas in Communications*, 26(7):1116–1127.

Erev, I. and Roth, A. E. (1998). Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria. *American Economic Review*, 88(4):848–81.

Felegyhazi, M. and Hubaux, J.-P. (2006). Game Theory in Wireless Networks: A Tutorial. Technical Report LCA-REPORT-2006-002, EPFL. Available at http://www.crysys.hu/ mfelegyhazi/publications/FelegyhaziH06tutorial.pdf.

Forouzan, B. A. (2007). *Data Communications and Networking*. McGraw-Hill.

IEEE (1999). Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications. Technical Document IEEE Std 802.11/[ISO/IEC DIS 8802-11], Institute of Electrical and Electronic Engineers.

http://standards.ieee.org/getieee802/download/802.11-1999.pdf.

Jain, R. K., Chiu, D.-M. W., and Hawe, W. R. (1984). A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Technical Report DEC-TR-301, Digital Equipment Corporation.

Konorski, J. (2006). A game-theoretic study of CSMA/CA under a backoff attack. *IEEE/ACM Trans. Netw.*, 14(6):1167–1178.

Kyasanur, P. and Vaidya, N. H. (2005). Selfish mac layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing*, 4:502–516.

MacKenzie, A. and DaSilva, L. (2006). *Game theory for wireless engineers*. Synthesis lectures on communications. Morgan & Claypool Publishers.

Narahari, Garg, D., and Narayanam, R. (2009). *Game Theoretic Problems in Network Economics and Mechanism Design Solutions*. Springer-Verlag London Limited, London.

Nowak, M. and Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364.

Raya, M., Aad, I., Hubaux, J. P., and El Fawal, A. (2006). DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots. *Mobile Computing, IEEE Transactions on*, 5(12):1691–1705.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts London, England.

Watkins, C. J. and Dayan, P. (1992). Technical Note, Q-Learning. *Machine Learning*, 8:279–292.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, London. Available at http://www.cs.rhul.ac.uk/ *chrisw*/*new*thesis.pdf.

Zhao, D. (2006). Access control in ad hoc networks with selfish nodes. *Wireless Communications and Mobile Computing*, 6(6):761–772.

Zhu, S. (2003). *Learning to Cooperate*. PhD thesis, University of Rochester, Rochester, New York.

Zhu, S. and Ballard, D. H. (2002). Overcoming non-stationary in uncommunicative learning. Technical Report TR 762, University of Rochester, Computer Science Department. Available at http://www.cs.rochester.edu/ zsh/pub/.

APPENDIX

APPENDIX 1

The Q-learning algorithm goes as follows:

Algorithm 1 An Algorithm for Learning Q

1. For each s, a, **Initialize** table entry $Q(s,a) \leftarrow 0$
   Observe state s
2. Do forever:
   - **Select** an action a and execute it
   - **Receive** immediate reward r
   - **Observe** the new state s'
   - **Update** table entry for $Q(s, a)$ as follows:

$$Q(s,a) \leftarrow r + \gamma max Q(s',a')$$

   - $s \leftarrow s'$

APPENDIX 2

The algorithm to utilize the **Q** matrix is as follows:

Algorithm 2 An Algorithm to Utilize the Q-Matrix

- Input: Q matrix, initial state
  1. Set current state = initial state
  2. From current state, find action that produce maximum Q value
  3. Set current state = next state
- Go to 2 until current state = goal state