Postprint of: Konorski J., A study of HIPERLAN-type noncooperative medium access control setting in a wireless LAN, SYSTEMS SCIENCE, Vol. 28, Iss. 1 (2002), pp. 71 - 84

wireless LAN, MAC protocol, noncooperative setting

Jerzy KONORSKI[*]

A STUDY OF HIPERLAN-TYPE NONCOOPERATIVE MEDIUM ACCESS CONTROL SETTING IN A WIRELESS LAN

A single-hop wireless LAN is considered. Stations contend for access to the shared radio channel using an elective-type MAC protocol whose two components are: a common *scheduling policy* that provides a framework for taking elective actions and defines winning elective actions, and *contention strategies* that dictate actions in successive protocol cycles. In a noncooperative setting the two components are logically separate: besides cooperative stations that use a standard HIPERLAN-type contention strategy, there are noncooperative stations that use self-optimising strategies at the bandwidth cost of the former, yet in order to conceal their nature and maintain proper synchronisation adhere to the scheduling policy. This paper proposes a self-regulatory solution whereby the scheduling policy and the contention strategy used by c-stations are jointly designed so as to invoke a noncooperative inter-station game with a fair outcome. A family of policies, termed EB/ECD-Monotone(Δ), and contention strategies called Closed Shop and Best Response are proposed and found satisfactory upon evaluation via simulation with a focus on bandwidth distribution under heavy-load.

1. INTRODUCTION

We consider a single-channel, single-hop wireless LAN (WLAN) interconnecting a set of stations with full hearability. Stations with packets ready to transmit contend for access to the shared radio channel using a symmetric distributed MAC protocol, implying that there is no central management in any form and that MAC interfaces at the stations all execute the same protocol machine. Packet transmissions as well as the stations' actions dictated by the MAC protocol are assumed to be synchronised to a common slotted time axis. Each time slot is supposed to accommodate a basic protocol data unit, the maximum station-to-station propagation delay and the receive/transmit switchover time.

Many MAC protocols have been devised differing mostly by the way the time axis is structured into protocol cycles. A generic MAC protocol cycle consists of a scheduling phase followed by a transmission phase; in the former (also referred to as *scheduling penalty* and possibly further divided into sub-phases), stations take elective actions to

---

[*] Technical University of Gdańsk, ul. Narutowicza 11/12, 80-952 Gdańsk, Poland

produce one or more winners who subsequently transmit their packets. An important albeit seemingly artificial distinction can be made at this point between two MAC components at a station: the *scheduling policy* provides a procedural framework for taking elective actions and determines winning actions (i.e., defines the underlying protocol machine), whereas the *contention strategy* dictates the station what elective actions to take in successive protocol cycles.

Typical medium- and long-term goals of a scheduling policy include efficient bandwidth use (minimisation of the scheduling penalty) and fairness among stations (in terms of equal individual bandwidth shares i.e., cycle-wise winning rates). Scheduling policies can be categorised orthogonally considering: the effective memory length (e.g., single- or multiple-cycle, depending on whether winning actions are defined based on the outcome of the latest cycle or a number of preceding cycles), employed mechanisms (e.g., contention- or reservation-based – the latter entail a form of distributed queuing and typically attempt to schedule multiple packets per cycle), and required per-slot channel feedback (e.g., binary or ternary – in the latter, a station can distinguish a void slot, a successful transmission and a collision). Here we focus on single-cycle, contention-based scheduling policies. We only require binary channel feedback in that a successful transmission is to be distinguished at a station only if that station is the intended recipient.[1]

In a *cooperative MAC setting*, the above mentioned goals are global in nature and common to all stations, which therefore pursue a standard contention strategy and adhere to a scheduling policy oriented toward these goals. Thus the contention strategy and the scheduling policy constitute a monolithic MAC protocol. Current trends in networking point to several factors that are likely to redefine this setting in near future, among them: mobility (with a station's actions harder to enforce or even monitor), user preferences for anonymity and volatility (consequently, stations' identities tend to exist on a temporary basis only or to be unavailable at MAC level for security reasons), and ad-hoc design philosophy (whereby stations exhibit certain functional and administrative autonomy, combining the functionality of a user terminal and network node). There are two basic reasons to expect more wayward intelligence in WLAN stations: (i) they have to become increasingly self-programmable in order to adapt to changing and often uncontrollable traffic environment [1], and (ii) some of them are possibly of irregular origin and not guaranteed to conform to standard MAC; over time, indecent vendors may feel inclined to offer 'bogus' MAC chips (by analogy with airline industry) able to acquire a larger-than-fair bandwidth share by departure from standard MAC – required technology seems already within reach of small- and medium-size manufacturers. In the emerging *noncooperative MAC setting*, contention strategies are station specific and logically separate from the scheduling policy. Besides cooperative (c-) stations that use a *standard* contention strategy (usually based on a predefined probability distribution over the action space) and naturally adhere to a common scheduling policy, there are a number of noncooperative (nc-)stations that use various *self-optimising*

---

[1] Note that token-passing protocols like IEEE 802.4 or CRT [7] have a station keep track of the sender addresses of successive packets while waiting for its turn i.e., require ternary channel feedback. In some sensitive environments, however, stations may resort to whole-packet encryption; from a non-recipient's viewpoint, a successful transmission is then indistinguishable from a collision.

contention strategies at the cost of c-stations i.e., commit *bandwidth stealing*; yet to conceal their nature and maintain proper synchronisation they too adhere to the common scheduling policy[2].

We submit that a self-regulatory solution is possible whereby nc-stations are engaged in a noncooperative game in which, ideally, pursuing a bandwidth stealing strategy should leave each c-station a bandwidth share at least comparable to what it would get in a cooperative MAC setting. The approach we advocate is thus one that provides disincentives to commit bandwidth stealing instead of putting in more administration. Specifically, our solution involves

- a study of a number of heuristic self-optimising contention strategies that should be considered as a replacement of the standard contention strategy, and
- a number of modifications of a reference scheduling policy called EB/ECD, suitable for a cooperative MAC setting.

While the self-optimising contention strategies have been devised so as to perform well against the standard contention strategy under a wide range of scheduling policies, the modifications of EB/ECD are to enable these strategies to perform well against one another regardless of the number of nc-stations.

The idea of logical separation of station strategies and the scheduling policy was first presented in [8]. An example of an EB/ECD-like ternary-feedback scheduling policy for HIPERLAN/1-type MAC, called EB/ECD-Δ, was given in [9] and evaluated via simulation under heavy load. The policy was hoped to offer by itself enough protection against bandwidth stealing so that the standard contention strategy could be retained in c-stations. However, the self-optimising contention strategies it was to play against were only moderately aggressive. In the present paper, we improve thereupon by

- slightly changing the positive ACK semantics used in EB/ECD, which permits to relax the ternary feedback requirement[3],
- introducing more aggressive self-optimising contention strategies and a more general class of EB/ECD-like scheduling policies, and
- demonstrating that EB/ECD modifications oriented toward guaranteeing a high bandwidth share for each c-station should be evaluated jointly with suitable replacement strategies for c-stations.

Traces of the noncooperative paradigm are present throughout computer networks literature, cf. competitive routing [10], Fair Share queuing [15], incentive compatible flow control [14], auction-like resource allocation [11], and power control in wireless networks [4]. As for wireless multiple access, some authors have noted that adherence to standard MAC is vital for nc-stations lest they lose connectivity [1]; however, as argued below, contention protocols like CSMA/CA and HIPERLAN/1 prove quite the contrary. This is

---

[2] Another form of bandwidth stealing attempts to minimise energy consumption by refusing to forward other stations' packets [1]; this is irrelevant in a full-hearability environment.

[3] This change also provides a MAC-level ACK mechanism useful in a partial-hearability environment.

also true of S-ALOHA whose game interpretation in a (somewhat idealised) self-optimising environment was given in [12].

In Sec. 2, details of the communication model are stated with more precision and the binary-feedback version of EB/ECD is described. Sec. 3 outlines a framework for a noncooperative MAC setting with emphasis on the issue of *verifiability*, arising from the many ways an nc-station can forge winning actions. It is vital in this context that any 'profitable' departure from the correct operation of the underlying protocol machine be detectable as the presence or absence of carrier on the channel. EB/ECD is subsequently examined from this point of view. The limitations of verifiable EB/ECD-like scheduling policies and self-optimising contention strategies are pointed out in Sec. 4. Some heuristic examples thereof are described in Sec. 5 and evaluated via simulation in Sec. 6 against the backdrop of a cooperative MAC setting. Sec. 7 concludes the paper.

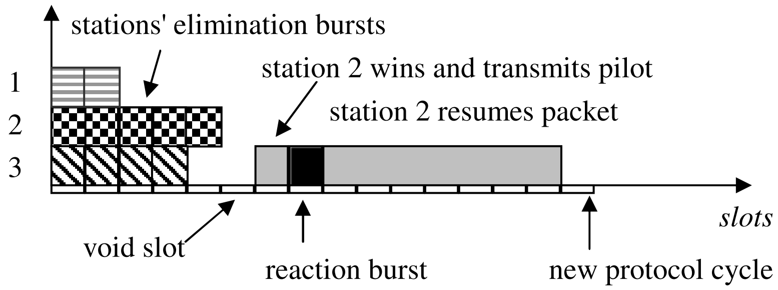
2. COMMUNICATION MODEL AND REFERENCE SCHEDULING POLICY

In a distributed ad-hoc environment it would be unrealistic to expect much prior knowledge about the number and identities of stations, their data encoding conventions or services they will use. We push this attitude to an extreme with the following set of non-assumptions:

1. Stations can plug in and out at will and no one attempts to monitor their current number.
2. Station identities are inaccessible at MAC level, nor can they be deduced from the contents of the packets they transmit, except by the intended recipients.
3. In the case of a successful transmission, the intended recipient (or recipients) recognises itself as such; no other station can extract any useful information except detecting the presence of carrier.
4. No particular traffic characteristics can be expected.

Contention-based binary-feedback scheduling policies typically employ collision avoidance as in the CSMA/CA technique of IEEE 802.11 [5] or in the yield phase of ETSI HIPERLAN's EY-NPMA [2]. Binary feedback as described in non-assumption 3 above allows for a collision detection equivalent employed in our reference policy called *Elimination Burst with Extraneous Collision Detection* (EB/ECD). It works similarly as the elimination phase of EY-NPMA (Fig. 1a): at the start of a protocol cycle, each station contends for access by transmitting an *elimination burst* of arbitrary slot-multiple length between 1 and *E* slots, the longest burst(s) winning. Burst lengths are selected according to the station-specific contention strategy. Non-winner stations back off while the winner stations transmit 1-slot *pilot* packets and suspend transmission thereupon, awaiting reaction from the intended recipients. On sensing a successful pilot transmission, a recipient reacts by transmitting a 1-slot *reaction burst*. Lack of reaction causes the winner stations to synchronise to a new protocol cycle. Fig. 1b depicts the corresponding protocol machine at a station, assuming that there are always packets ready for transmission. The absence of a yield phase is advantageous since it simplifies the stations' actions and reduces the average scheduling penalty, as verified by a simple probabilistic argument [9]. However, if c-stations

(a)


stations' elimination bursts
1
2
3
station 2 wins and transmits pilot
station 2 resumes packet
slots
void slot
reaction burst
new protocol cycle

(b)

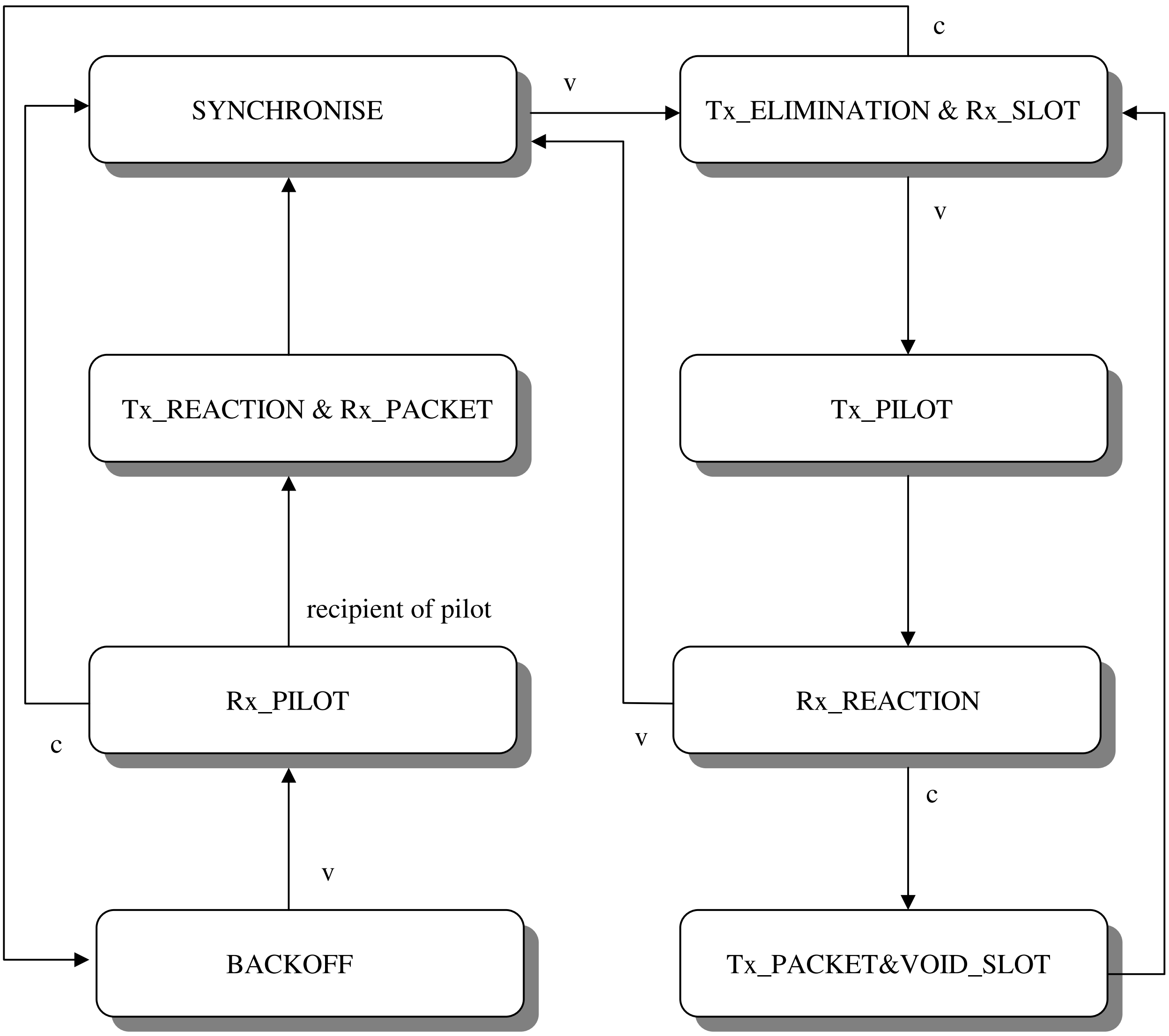

c
SYNCHRONISE
v
Tx_ELIMINATION & Rx_SLOT
v
Tx_REACTION & Rx_PACKET
Tx_PILOT
recipient of pilot
Rx_PILOT
c
v
Rx_REACTION
c
v
BACKOFF
Tx_PACKET&VOID_SLOT

Fig. 1. EB/ECD, (a) example of protocol cycle, (b) protocol machine (v: void slot, c: collision)

use the standard (simple randomisation) contention strategy, a straightforward bandwidth stealing strategy consists in systematic transmission of 'longer than random' elimination bursts. To prevent frequent collisions with other nc-stations using similar strategies, the selection of an elimination burst length might use a probability distribution biased toward *E*.

3. FRAMEWORK FOR A NONCOOPERATIVE MAC SETTING

An nc-station will attempt to steal bandwidth in two ways: by pursuing a self-optimising contention strategy that promises a higher winning rate when played repeatedly (cycle-by-cycle) against the c-stations, and/or by forging winning actions i.e., forcing certain 'profitable' state transitions when executing the protocol machine; the latter can be done either by taking illegitimate actions or by deliberate misinterpretation of other stations' actions. While self-optimising contention strategies can be coped with using similar strategies at the c-stations, forgery of winning actions must be ruled out by the communication model. The underlying axioms of verifiability (as enforced by a potentially deployed device able to sense any particular station's transmissions) dictate that in any given slot a station cannot safely

(i) deny transmitting carrier having done so,
(ii) claim to have transmitted carrier having transmitted none,
(iii) claim to have sensed carrier while none was transmitted, or
(iv) deny sensing carrier transmitted by another station.

Let us return to the EB/ECD protocol machine in Fig. 1b and consider that both elimination and reaction bursts are just carrier sequences rather than meaningful bit patterns, as are pilot packets to non-recipients. Forcing any 'profitable' state transition would then violate one of the above axioms, namely:

- Tx_ELIMINATION & Rx_SLOT→Tx_PILOT involves forgery of own elimination burst length e.g., by interrupting transmission and resuming upon detection of other bursts still in progress (in violation of (i) and (ii)) or by deliberate misinterpretation of own burst as the longest (in violation of (iv)).
- Rx_REACTION→Tx_PACKET&VOID_SLOT may follow on the claim that a reaction to own pilot was sensed (in violation of (iii)).
- SYNCHRONISE→ Tx_ELIMINATION & Rx_SLOT may follow on the claim that a reaction to another station's pilot was not sensed (in violation of (iv)).

(Note that refraining from reaction to a pilot by its intended recipient, although leads to a 'profitable' state transition Rx_PILOT→SYNCHRONISE, would be counterproductive as it would result in a failure to receive a packet).

4. FRAMEWORK FOR A CONTENTION STRATEGY AND SCHEDULING POLICY

With verifiability measures in place, bandwidth stealing can only be accomplished using self-optimising contention strategies. These are subject to a number of limitations that follow from our communication model, namely:

- **limited prior knowledge**: neither the number of stations nor the number of nc-stations can be assumed known; moreover, an nc-station may not know the contention strategies used by other nc-stations (although it does know the standard contention strategy used by c-stations),
- **limited observability**: individual cycle-wise action trajectories (i.e., elimination burst lengths in successive protocol cycles) are obscured behind the longest bursts and cannot be learned from; the winning action trajectory may be observable, but without accessible station identities cannot be translated into individual winning rates,
- **isolation**: collusion with other nc-stations cannot be relied upon for self-optimisation,
- **rational behaviour**: self-optimisation of own bandwidth share should be the primary goal – this rules out diminishing other stations' winning rates at the price of self-damage.

The first two limitations make it impossible to determine a 'fair' or 'target' bandwidth share to strive for and leave a self-optimising contention strategy with the gradient-search 'learn by playing' option. The other two help specify the learning process. In particular, rational behaviour implies[4]

- the Law of Effect, whereby a course of play increasing own winning rate is more likely to be repeated in the future, and
- responsiveness i.e., no course of play is abandoned forever unless it results in a zero winning rate.

Given a set of actions (elimination burst lengths, of which not all are observable), the scheduling policy returns a unique winner station or a no-winner outcome. Any policy of interest should enable a c-station using a suitable contention strategy to obtain a fair bandwidth share. Moreover, can be soundly required to be

- **verifiable** i.e., state transitions in the underlying protocol machine can only be fired based on the presence/absence of carrier on the channel (as a counterexample, a station could flip a coin to decide whether to transmit a pilot or not, with the head probability dependent on own elimination burst length),
- **feedback compatible** i.e., each station should be able to determine the outcome upon binary feedback and subject to the limited observability stated above (as a counterexample, consider a modification of EB/ECD whereby the longest elimination burst wins only if it is not 'much longer' than the second-longest),
- **incentive compatible** i.e., not rely on a station's willingness to confirm another station's winner status (consider the same counterexample as above with the provision that

---

[4] For a general discussion of rational behaviour see e.g. [3, 6].

stations with second-longest elimination burst lengths announce their actions via a separate pilot – obviously they have no incentive to do so and a standstill may follow),
- **irreducible** i.e., no action should be dismissed a priori as non-winning, thereby reducing the actual action space (as a counterexample, consider a policy whereby the longest elimination burst wins only if it is 'sufficiently long'), and
- **protective** i.e., there should be no 'fail-safe' actions known a priori to either win or at least render other actions non-winning (to protect against irrational behaviour or 'playing for time'; EB/ECD is a counterexample, action E being 'fail-safe').

5. PROPOSED SOLUTION

Within the framework of Sec. 4, a class of heuristic EB/ECD-like scheduling policies can be proposed with the following key elements:

- upon termination of own elimination burst of length l, a station senses other stations' bursts still in progress and counts successive slots until the longest, M-slot elimination burst terminates (as detected upon sensing a void slot); thus $r=M-l$ is obtained,
- the station next computes $priority(r)$ from a predefined function $priority{:}0..E-1 \rightarrow 1..\Delta+1$, where Δ is an integer parameter,
- knowing M, the station is aware of the set of possible other stations' actions in the current protocol cycle and computes $p\_min=\min\{priority(M-a) | a \leq M\}$,
- the void slot after the longest elimination burst is followed by a sequence of pilot and reaction slots in order for the stations to announce their priorities; the first pilot slot is reserved for stations with $priority=p\_min$ and is left void if there are no such stations, otherwise paired with a reaction slot; in the case of a reaction burst from the intended recipient, a unique winner is elected, otherwise subsequent pilot slots are reserved for stations with $priority=p\_min+1$ through Δ,
- stations with $priority=\Delta+1$ back off until the next protocol cycle.

Taking various Δ and functions *priority* yields a family of EB/ECD-like scheduling policies[5] with an extra scheduling penalty of up to 2Δ slots. Note that in general, winning priorities are observable rather than winning actions. Henceforth we shall focus on a subfamily of policies termed EB/ECD-Monotone(Δ) with $priority(r)\equiv\Delta-r$ if $r<\Delta$ and $priority(r)\equiv\Delta+1$ otherwise. This function is illustrated in Fig. 2a for $\Delta=4$ (EB/ECD-Monotone(1) coincides with EB/ECD). An example of a protocol cycle for EB/ECD-Monotone(3) is shown in Fig. 2b.

The following self-optimising contention strategies, fitting into the framework of Sec. 4, have been considered as possible replacements of the standard strategy in c-stations. Each defines an *update period* of *UP* consecutive protocol cycles, at the end of which relevant measures are calculated and adjustments made.

---

[5] A further generalisation might define $priority(l,M)$.

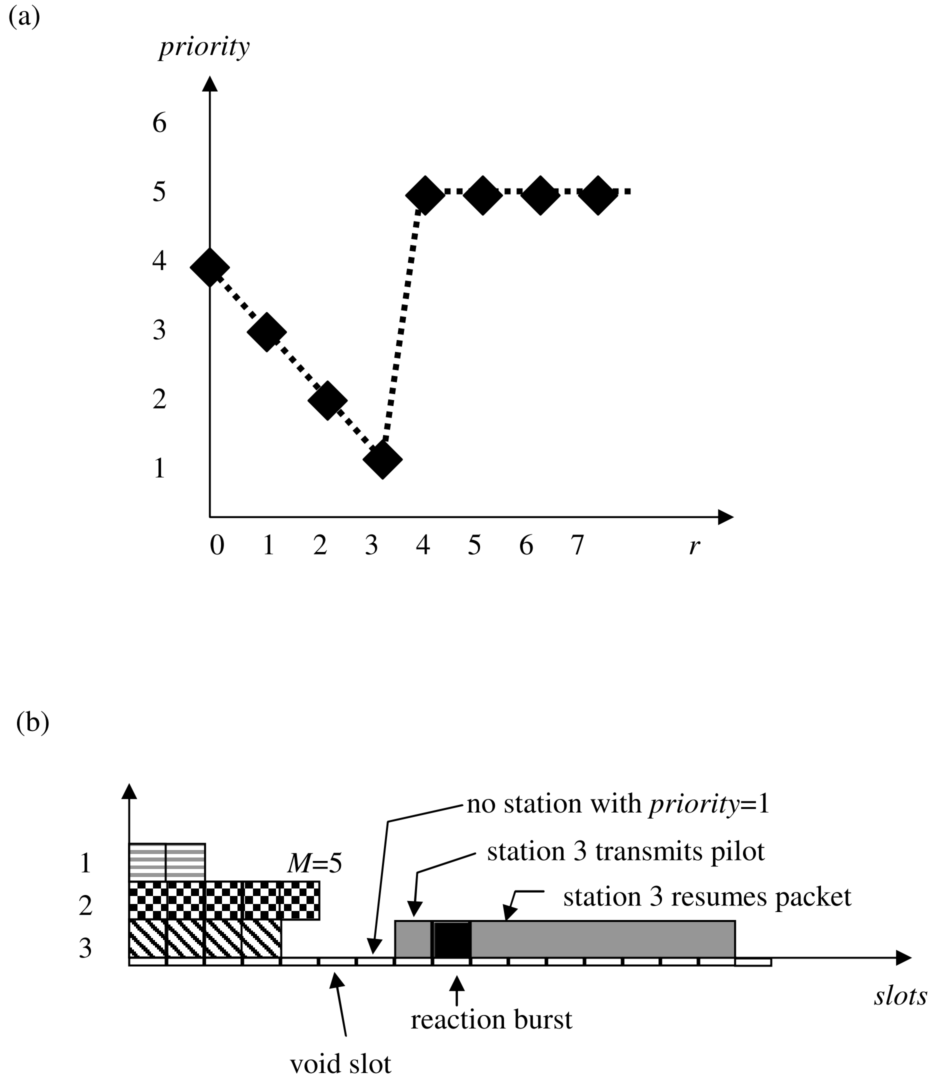


Fig. 2. EB/ECD-Monotone(Δ), (a) priority function (Δ=4), (b) example of protocol cycle (Δ=3); stations 1, 2 and 3 have priorities 4, 3 and 2, respectively

- **Parameter Adaptation** (PA). This is an adaptive version of the standard contention strategy. In each protocol cycle a station records l and M and computes the closest action l' such that $priority(M-l')$ equals the winning priority (if there is a winner). Actions are drawn from a truncated geometric probability distribution whose parameter is incremented or decremented at the end of an update period depending on whether the number of times when $l<l'$ is greater or smaller than the number of times when $l>l'$.
- **Range Adaptation** (RA). In each update period, a station records the lowest and highest action, a_L and a_H, corresponding to the set of winning priorities and at the end uses a linear autoregressive filter to obtain their smoothened counterparts, α_L and α_H. Actions in the next update period are drawn at random from the integers in the range $\alpha_L..\alpha_H$. If there have been no winners, this range expands by 1 at both edges.
- **Closed Shop** (CS). This strategy is supposed to mimic token passing among the nc-stations and thus cut off the other stations. An nc-station selects a starting action a_0 and in successive protocol cycles takes actions a_0, a_0+1, ..., E, 1, 2, ... a_0-1 etc. If there have been no winners in the past update period, a different a_0 is selected.
- **Best Response** (BR). Much in the spirit of fictitious play [13], this strategy seeks a best response to what the other stations have been playing. In each protocol cycle, a set of *shadow winning actions* (i.e., actions that either won or would have won if taken) is computed. Throughout the next update period actions are drawn at random with the relative frequencies of occurrence of shadow winning actions. Let w be the winning priority in the current protocol cycle (taken to be $\Delta+1$ if there is none). An arbitrarily selected action $x\le M$ whose priority equals $w\le\Delta$ is counted as a shadow winning action. To compute other shadow winning actions, the strategy calculates as follows. Let P be the set of announced priorities less than w (thus each corresponds to actions taken by more than one station), $A'=\{a|priority(M-a)\in P\}$ and $A''=\{a\le M|priority(M-a)\ge w\}$. The strategy looks at all $x\in 1..E$ and for each x computes $M'=\max\{M, x\}$. Then x is a shadow winning action if
    - $priority(M'-x)\ne priority(M'-a)$ for all $a\in A'$, and
    - $priority(M'-x)<priority(M'-a)$ for all $a\in A''$.

6. PERFORMANCE EVALUATION

In order to evaluate the performance of EB/ECD-Monotone(Δ) and the proposed self-optimising contention strategies in a noncooperative MAC setting, a simulation model was used that featured: N=10 stations, including a number of nc-stations, NC, varying between 0 and $N-1$, constant packet transmission times of L=50 slots and symmetric offered load of 1 packet arrival per L slots per station (i.e., the network operated at saturation). Burst lengths ranged from 1 to E=10. In the experiments, all c-stations used a selected contention strategy (PA, RA, CS or BR) against each of the other three strategies deployed in turn in all nc-stations. Owing to this homogeneity of strategy deployment, the study was confined to 12 strategy X vs. strategy Y scenarios, each repeated for Δ varying between 1 and $E-1=9$. Note that all these strategies 'learn by playing' only; to make the learning realistically asynchronous across the set of stations, update periods of constant length $UP=20$ protocol cycles were initialised at a random instant at each station.

The figure of merit was the c-station bandwidth share B_c (the ratio of the total packet slots, including pilots, and the number of elapsed slots), averaged over 100,000 protocol cycles. In a cooperative MAC setting under symmetric load, ideally $B_c=1/N$ (or 10%) of the available bandwidth. Scheduling penalties cause this figure to drop below the 10%, whereas non-ideal protection against bandwidth stealing in a noncooperative MAC setting may bring about a further decrease. For reference, a cooperative MAC setting was simulated with all stations using the standard contention strategy (a truncated geometric probability distribution over $1..E$ was employed with a Δ-specific optimum parameter).

Denoting $B_c(\mathrm{X},\mathrm{Y},NC,\Delta)$ our figure of merit for the strategy X vs. strategy Y scenario, one would suggest seeking X and Δ so as to maximise $\min_{\mathrm{Y}\neq\mathrm{X}}\min_{NC\in 0..N-1}B_c(\mathrm{X},\mathrm{Y},NC,\Delta)$ i.e., we look for a maximum bandwidth guarantee regardless of the number of nc-stations and the contention strategies they use[6]. In particular, $B_c(\mathrm{X},\mathrm{Y},0,\Delta)$ represents the efficiency of strategy X in a cooperative MAC setting.

The obtained results are shown in Figs. 3-6, where $\min_{NC\in 0..N-1} B_c(\mathrm{X},\mathrm{Y},NC,\Delta)$ (expressed as the percentage of the ideal $1/N$ share) is plotted against Δ separately for various X and Y. The dashed lines correspond to the cooperative MAC setting; they look particularly good at smaller Δ, which confirms that EB/ECD is the most suitable for this setting. Based on the above suggestion, one is inclined to regard BR or CS combined with $\Delta=5..7$ as satisfactory for the considered noncooperative MAC setting. Note that the guaranteed c-station bandwidth share then approaches that in the cooperative MAC setting.

7. CONCLUSION

In view of the noncooperative paradigm gaining ground in the computer networks design, noncooperative scheduling policies at MAC level seem of interest, in particular in a wireless LAN environment where user volatility and the lack of tight administration render c-stations prone to bandwidth stealing by nc-stations. It is advisable in such an environment to design jointly a self-optimising contention strategy to be used by c-stations instead of the standard randomisation strategy and a modified HIPERLAN-type scheduling policy so as to invoke a noncooperative game with a fair outcome from the c-stations' viewpoint. The presented EB/ECD-Monotone(Δ) scheduling policy provides a useful option though its significance is limited to the class of proposed self-optimising contention strategies. Further research should extend this class and answer some more fundamental questions as to the nature and asymptotic properties of the invoked noncooperative game.

---

[6] In a more rigorous statement, one would define a threshold guarantee B_0 and for any Δ seek a set of *satisfactory* contention strategies, a satisfactory strategy being one that guarantees B_0 against any other satisfactory strategy.

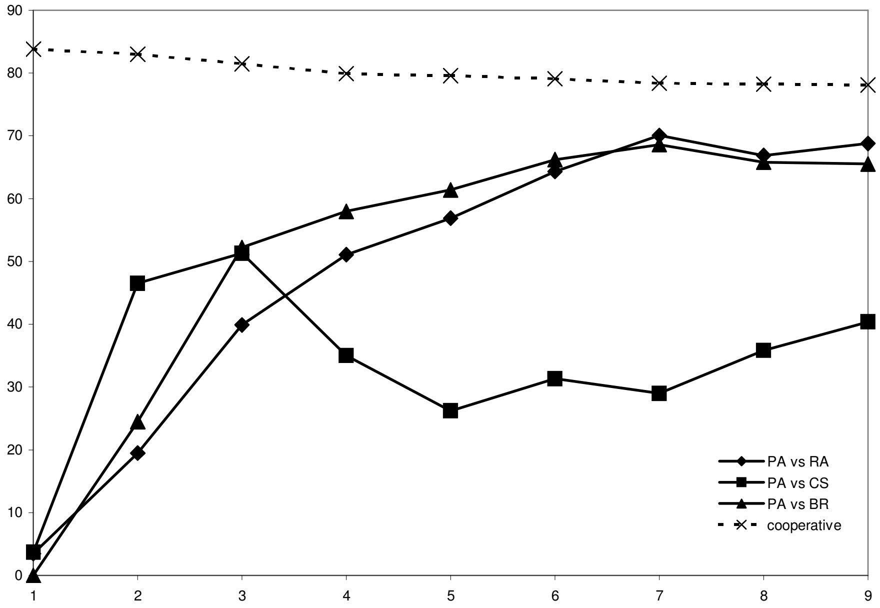


Fig. 3. Guaranteed c-station bandwidth share (% of 1/*N*) as a function of *Δ* for PA vs. Y

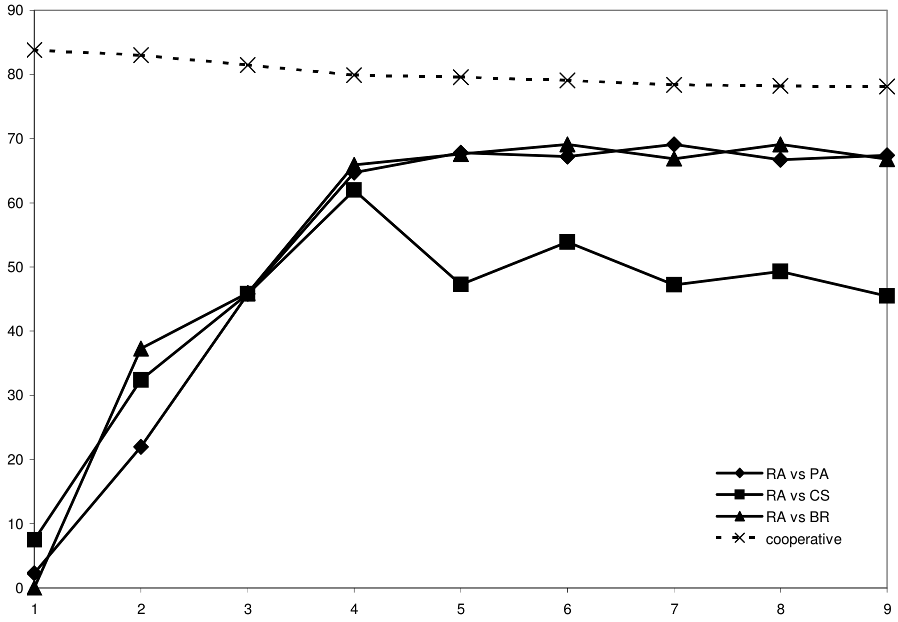


Fig. 4. Guaranteed c-station bandwidth share (% of 1/*N*) as a function of *Δ* for RA vs. Y

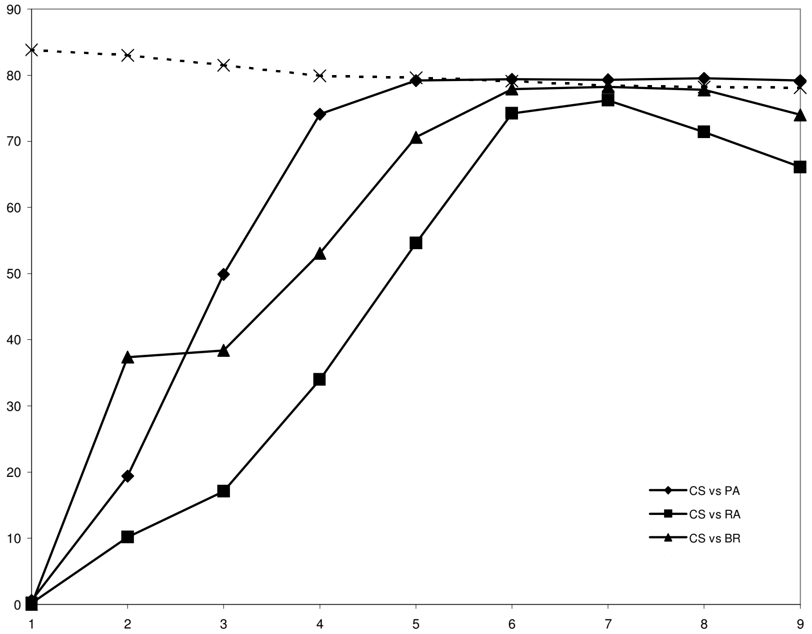


Fig. 5. Guaranteed c-station bandwidth share (% of 1/*N*) as a function of *Δ* for CS vs. Y

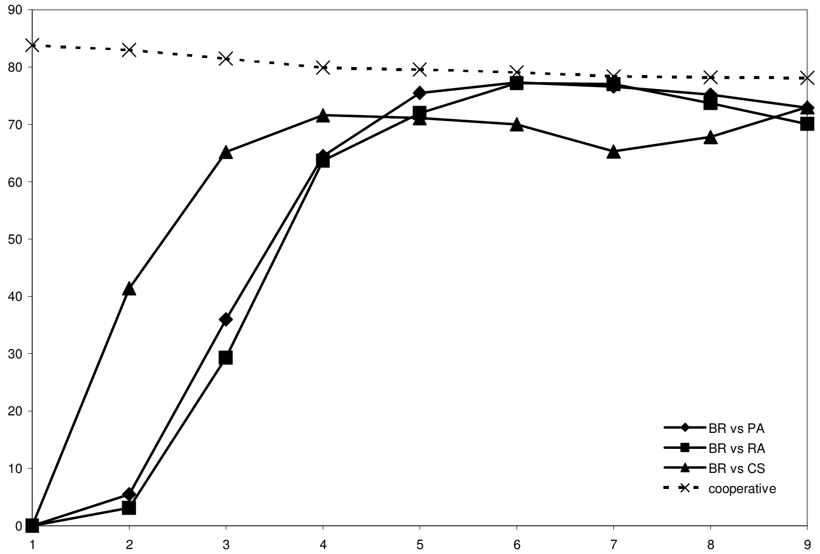


Fig. 6. Guaranteed c-station bandwidth share (% of 1/*N*) as a function of *Δ* for BR vs. Y

REFERENCES

[1] BUTTYAN L. and HUBAUX J.P., *Nuglets: A Virtual Currency to Stimulate Cooperation in Self-Organised Mobile Ad-hoc Networks*, Tech. Rep. DSC/2001/001, Swiss Federal Institute of Technology 2001.

[2] ETSI TC RES (Radio Equipment and Systems), *High Performance Radio Local Area Network (HIPERLAN); Services and Facilities; Version 1.1*, RES 10, Jan. 1995.

[3] FRIEDMAN E.J. and SHENKER S., *Synchronous and Asynchronous Learning by Responsive Learning Automata*, Mimeo 1996.

[4] HEIKKINEN T., *On Learning and the Quality of Service in a Wireless Network*, Proc. Networking 2000, Paris, LNCS vol. 1815, Springer-Verlag 2000, pp. 679-688.

[5] IEEE 802.11 Standard, *Wireless Media Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[6] KALAI E. and LEHRER E., *Rational Learning Leads to Nash Equilibrium*, Econometrica vol. 61 (1993), pp. 1019-1045.

[7] KONORSKI J., NOWICKI K. and J. WOZNIAK J., *The Cooperative Random Token Protocol for High-speed Radio LANs*, Proc. GLOBECOM'92, Orlando FL, Dec.1992.

[8] KONORSKI J., *Packet Scheduling in Wireless LANs – A Framework for a Non-cooperative Paradigm*, Proc. IFIP Int. Conf. on Personal Wireless Communications PWC'2000, Gdansk, Poland, Kluwer Academic Publishers 2000, pp. 29-42.

[9] KONORSKI J., *Protection of Fairness for Multimedia Traffic Streams in a Non-cooperative Wireless LAN Setting*, Proc. 6th Int. Conf. on Protocols for Multimedia Systems PROMS 2001, Enschede, The Netherlands, LNCS vol. 2213, Springer-Verlag 2001, pp. 116-129.

[10] KORILIS Y.A., LAZAR A.A. and ORDA A., *Architecting Noncooperative Networks*, IEEE J. Selected Areas Commun., vol. 13 (1995), pp. 1241-1251.

[11] LAZAR A.A. and SEMRET N., *Auctions for Network Resource Sharing*, Tech. Rep. CU/CTR/TR 468-97-02, Columbia Univ. 1997.

[12] MACKENZIE A.B. and WICKER S.B., *Game Theory and the Design of Self-Configuring, Adaptive Wireless Networks*, IEEE Comm. Magazine, vol. 39 (2001), pp. 126-131.

[13] MILGROM P. and ROBERTS J., *Adaptive and Sophisticated Learning in Normal Form Games*, Games and Economic Behaviour vol 3 (1991), pp. 82-100.

[14] SANDERS B.A., *An Incentive Compatible Flow Control Algorithm for Rate Allocation in Computer Networks*, IEEE Trans. Comput., vol. 37 (1988), pp. 1067-1072.

[15] SHENKER S., *Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines*, Proc. SIGCOMM'94, London, UK.