

MAC Contention in a Wireless LAN with Noncooperative Anonymous Stations

Jerzy Konorski
 Gdansk University of Technology, Gdansk, Poland
 Email: jekon@eti.pg.gda.pl

Abstract—In ad hoc wireless LANs populated by mutually impenetrable groups of anonymous stations, honest stations are prone to "bandwidth stealing" by selfish stations. The problem is addressed at the MAC level by postulating that (i) honest stations use a carefully designed contention strategy teaching selfish stations that the best reply is to stick to the same strategy, and (ii) a verifiable winner policy be designed so that such a strategy can indeed be found and yields high bandwidth shares for honest stations. For a class of random token winner policies, a number of cycle-by-cycle reinforcement learning strategies are evaluated via simulation using an introduced notion formally akin to evolutionary stability.

Index Terms—ad hoc wireless LAN, MAC contention, noncooperative behavior, evolutionary stability

I. INTRODUCTION

Consider N stations exchanging data over a wireless local-area network (WLAN). We model the network as a *noncooperative setting*, where the stations may selfishly depart from standard communication protocols if it brings them a larger share of the network resources. It is well-known that the resulting network operation differs significantly from that in a cooperative setting. At the MAC layer, existing analyses of noncooperative settings focus on contention protocols: slotted ALOHA [1], [2] and CSMA/CA (as part of the IEEE 802.11 protocol [3]) [4], [5]. There is also a vast literature on selfish behavior of other communication mechanisms; some early papers deal with transmission power control [6], competitive routing [7], incentive compatible flow control [8], and fair queuing [9].

Studies of slotted ALOHA and CSMA/CA have so far assumed that the very principle of the protocol is observed by each station and only certain parameters may have been configured selfishly. For example, a selfish ALOHA station can only modify the retransmission probability, the slot by slot probabilistic scheduling of retransmissions being unchallenged; a CSMA/CA station

can decrease the minimum and maximum collision window, while still doubling it upon frame collisions, as the IEEE 802.11 standard prescribes. In this paper we generalize the contention model as follows: the MAC protocol proceeds in *cycles*, within each protocol cycle there is a finite *contention interval* to accommodate requests/attempts to transmit a data frame, and each station is free to select the instant within that interval where it makes its request/attempt.

A wide class of distributed contention protocols prescribe random deferment of requests to transmit upon commencement of a protocol cycle. Deferments are synchronized to a global slotted time axis, each deferment being a slot multiple. The generic term *Random Token* (RT) subsumes MAC mechanisms in which deferment lengths are drawn at random. RT mechanisms have been described in a pure form in [10]; they underlie some well-known standard solutions, including CSMA/CA (where the shortest deferment wins the contention) and HIPERLAN/1 [11] (where the longest deferment wins). Fig. 1 pictures a generic RT protocol cycle.

An RT-type MAC protocol breaks up into two components:

- a *selection configuration*, entirely within a station's discretion, dictates how deferments are selected within a contention interval, and
- a *winner policy*, common to all stations, defines the length of the contention interval, prescribes the rules of requesting to transmit, and determines a winning deferment in each protocol cycle (producing exactly one winner or none).

Two types of stations can be envisaged: *honest* and *selfish*. Honest stations use a predefined selection configuration e.g., a probability distribution over the contention interval, optimized with a view to improve

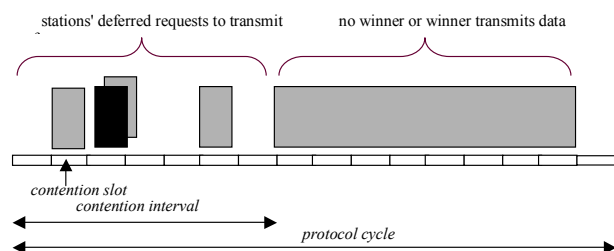


Figure 1. Generic RT protocol cycle.

Based on "Noncooperative Channel Contention in Ad-Hoc Wireless LANs with Anonymous Stations", by Jerzy Konorski, which appeared in the Proceedings of the 7th Int. Workshop on Distributed Computing, Kharagpur, India, December 2005. © Springer-Verlag.

global performance measures such as bandwidth utilization and fairness. Selfish stations are free to adopt any selection configuration and so self-optimize their bandwidth share to the detriment of honest stations. Given the growing volume of offered traffic, there is a strong motivation for stations to become bandwidth-greedy; enter advanced chip technology offering increasingly configurable station interfaces [12]. Note that the selection configuration and winner policy may be logically separate in a selfish station (unlike in an honest one, where they constitute a monolithic whole). Although it commits "bandwidth stealing" through deviation from a legally binding standard, a selfish station may hope to get away with it. This is particularly true in ad hoc WLANs, due to the inherent station mobility (stations' actions are hard to trace down) and anonymity (stations' identities may be unavailable at the MAC level). Nevertheless, most existing studies, unlike ours, assume that stations' identities are recoverable and trustworthy [4], [12], [13]; the few exceptions relate to slotted ALOHA [1], [2].

Selfish behavior calls for self-regulatory measures i.e., disincentives for "bandwidth stealing." To achieve flexibility at little cost, no administrative measures should be relied upon. One can take either one or a combination of the following approaches, so that a selfish configuration does not lead to "bandwidth stealing":

- design a winner policy given an honest selection configuration e.g., uniform probability distribution of deferments over the contention interval, referred to as *Honest Randomizer* (HR), and/or
- design an honest selection configuration given a winner policy.

In view of station anonymity, neither approach should make use of stations' identities. Within the first approach, the main task is to decide on a set of conceivable winner policies from which to choose. Any reasonable framework for a winner policy should preclude certain straightforward possibilities of "bandwidth stealing" by forgery of winning deferments; in this context, we discuss the issue of *verifiability*. We focus on a family of winner policies called *Random Token with Extraneous Collision Detection* (RT/ECD), where a request to transmit a data frame has the form of a short pilot frame sent in a slot, expected to be reacted to in the next slot by another short frame from the data frame's recipient. Thus the mechanism is not unlike IEEE 802.11 RTS/CTS access, although reaction frames need not be interpretable. It is easy to define a hash-based winner policy [14] whereby a randomly selected pilot wins the contention, hence HR cannot be outperformed by any other selection configuration. However, performance considerations dictate that the winning pilot be the earliest non-colliding one. Thus interesting winner policies only differ in how they account for the outcome of slots preceding the winning pilot. We compare two extreme cases referred to as RT/ECD-0 and RT/ECD- ∞ (indicating the maximum number of colliding pilots prior to a winning one).

The second approach can be dubbed "teach to learn": honest stations' behavior should induce a predictable learning process in selfish stations. Adaptive honest

behavior can be modeled by allowing a station to learn from the outcomes of past contentions, in which case it is appropriate to speak about *selection strategies*. Design of adaptive strategies able to withstand SR is a challenge. For example, a simple best-reply strategy has a station periodically update a histogram of winning deferments in recent contentions and use it to draw deferments in the next update period. Alternatively, various forms of responsive learning [15] may be used. The problem is that the set of conceivable selection strategies is infinite and probably cannot be parameterized. We suggest devising a set of heuristic strategies and within that set seek "evolutionarily stable" ones.¹ Obviously, how an honest strategy performs against a given selfish one also depends on the winner policy; this is confirmed by simulations reported in this paper. Thus the above two approaches may work well when combined. Care must be taken to control the level of adaptivity lest a more adaptive strategy be exploited by a less adaptive one. To give a flavor of the underlying issues we mention a scenario leading to a form of a Stackelberg game [9].

In Section II we formulate a WLAN model and make station anonymity more precise. In Section III the RT/ECD family of winner policies is described in detail, the notion of verifiability is introduced, and performance considerations are presented to narrow the interest down to RT/ECD-0 and RT/ECD- ∞ . For these two policies, in Section IV we examine a number of heuristic selection strategies and discuss evolutionary stability. Section V concludes the paper and outlines future research.

II. WLAN MODEL

An ad hoc WLAN consists of N stations equipped with transceivers and omnidirectional antennae to exchange data frames over a single wireless channel. They remain within the reception range of one another at all times. There is no fixed communication infrastructure or central administration. To simplify the considerations we take the single-channel assumption to mean that any two colliding (overlapping) frame transmissions are entirely incomprehensible. Moreover, the channel and the stations operate perfectly, so that all the stations receive the same channel feedback. Throughout the period of network operation, each station

- synchronizes to a common slotted time axis and employs a common winner policy,
- always has a data frame ready to transmit (operates at saturation load), and
- is free to use a selfish selection configuration but does not behave maliciously i.e., is not interested in degrading the other stations' bandwidth shares if it does not help enlarging its own.

We model anonymity as follows: each station

- conceals its MAC-layer identity from non-recipients of a transmitted frame (e.g., writes a random bit

¹ We call a strategy evolutionarily stable if it is a single best invader to all its best invaders – cf. [19], [20]; this conjures up an evolutionary process, which we do not define formally.

string into the source MAC address field, while encrypting the true address in the data field),

- relies on binary per-slot channel feedback i.e., can only distinguish an empty slot from a busy one; recipients of a non-colliding transmission are also able to interpret the slot's content.

We thus envisage a WLAN populated by multiple *mutually impenetrable groups* (Fig. 2). Stations within each group know one another, may use full data frame encryption and need not share the same data format with other groups. Neither do they exchange any control data regarding inter-group transmission scheduling, statistics etc. The presence of other groups is only perceived as bursts of carrier reducing the available bandwidth.

It might seem artificial that a greedy station should strive to seize an unfairly large bandwidth share for its data frame transmissions, given that typically it is willing to *exchange* (both transmit and receive) data frames. Firstly, however, a station might be transferring a large file of UDP traffic (file sizes on order of gigabytes are not uncommon), for which it needs no transport-layer acknowledgments. Secondly, and more importantly, in our model of mutually impenetrable groups a transmitting station is indistinguishable from – and thus an adequate model of – a group of stations exchanging data. What the outsiders perceive as a sequence of transmissions by a station can in fact be produced by a group of stations that have reached an intra-group agreement as to how to take turns at acquiring the medium. Thus more *transmission* opportunity for a selfish station is equivalent to more *communication* opportunity for a selfish group.

III. FRAMEWORK FOR A WINNER POLICY

Contention-based winner policies can be categorized considering their *time span* (e.g., single- or multiple-cycle, depending on whether winning deferments are defined independently in each cycle or based on the outcome of recent protocol cycles). Here we focus on single-cycle winner policies, since in an ad hoc network a consistent view of recent outcomes may be difficult to maintain across multiple stations. The specifics of the winner policy are public knowledge and are assumed to be commonly adhered to. These are: the number of *contention slots* per protocol cycle, the way of acquiring the medium, and the *win function* naming a single winner based on the channel feedback. All of them should be designed with the winner policy performance in mind.

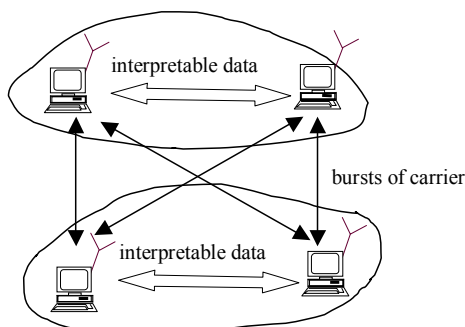


Figure 2. Perception of transmission in mutually impenetrable groups.

A. Requests to Transmit

Let a contention interval consist of E contention slots. Binary per-slot channel feedback permits to suppress data frame collisions analogously to RTS/CTS access in IEEE 802.11. Prior to the start of a protocol cycle an empty slot is allowed for synchronization. A station willing to transmit a data frame transmits a *pilot* frame in a selected contention slot $1, \dots, E$ (i.e., upon a selected deferment $0, \dots, E-1$). A non-colliding pilot is only interpretable to the recipient(s) of the data frame; to any other station it signals a request to transmit and otherwise appears merely as a burst of carrier. A station is only allowed to transmit one pilot in a contention interval; any deviations from this rule are easily detectable. A contention slot occupied by at least one pilot is immediately followed by a *reaction slot*, whereas the slot following an empty contention slot counts as the next contention slot. On sensing a non-colliding pilot, a recipient issues a *reaction* frame, a non-interpretable burst of carrier, and refrains from reaction if a collision of pilots is sensed. Reaction frames implement positive clear-to-send semantics: stations whose pilots were not reacted to perceive themselves as non-winners and back off until the next protocol cycle, while the rest (if any) are candidates, from which a winner is elected by the win function.

In Fig. 3 there are $N = 6$ stations and $E = 8$. Two stations select contention slot 2 for their pilots and the other stations select contention slots 4, 5, 7, and 8. Thus slots 2, 4, and 5 are followed by reaction slots. Reaction frames are transmitted following contention slots 4 and 5. Upon the latter, the pilot in slot 4 (say) is elected as the winner and its sender is prompted to start a data frame transmission in the next slot. Note that once the winner is elected, stations that have selected subsequent contention slots have no chance to transmit pilots and must defer until the next protocol cycle. Also note that even though a contention slot where two or more pilots collide is certain to produce no reaction frame, it must be followed by a reaction slot – otherwise an ambiguity might arise when a non-colliding pilot is perceived as such at a recipient station and as a pilot collision at non-recipients.

In practice, the slot size should be larger than a pilot or reaction frame to accommodate propagation delays (also transceiver turnaround times). Three requirements must be met: a sensed pilot or reaction frame must entirely fit in a slot, a pilot collision must not be confused with two consecutive non-colliding pilots, and a station must be unable to postpone its pilot until it makes sure no other station is transmitting one in the present slot. We translate these requirements respectively into

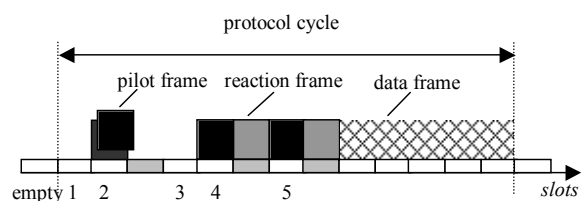


Figure 3. Example of an RT/ECD protocol cycle.

$$\begin{aligned}\tau_{\text{slot}} &> \tau_{\text{pilot}} + \max_propag + \max_postpone, \\ \tau_{\text{pilot}} &> \max_propag + \max_postpone, \\ \max_postpone &< \max_propag,\end{aligned}\quad (1)$$

where $\tau_{(\cdot)}$ is the duration of a specified interval or frame transmission, and \max_propag and $\max_postpone$ are the maximum inter-station signal delay and the time within which a station must start a pilot frame transmission after a slot start. All the inequalities in (1) must hold with enough "safety margins." To get the idea of the involved magnitudes, assume a WLAN 1 km in diameter, 54 Mb/s transmission rate, and $\max_postpone = \max_propag/3$; one may then suggest a 40-byte pilot and reaction frame size, and a 75-byte slot size. We neglect some obvious optimizations (e.g., empty slots may be shorter than slots containing pilot or reaction frames).

B. Verifiability

Brute-force station behavior e.g., deviations from the prescribed way of requesting to transmit, cannot be ruled out. Straightforward deviations consist in making false claims as to having or not having transmitted a frame, or having or not having sensed carrier. Not all types of such behavior yield the perpetrator an unfairly large bandwidth share, hence can be classified as selfish. Instances of potentially selfish brute-force behavior include:

- transmitting several pilots in different slots of the contention interval, to increase the chances of one of them winning,
- postponing a pilot transmission in violation of (1),
- transmitting a pilot immediately upon sensing another station's pilot (in effect jamming that pilot and causing its sender to back off); subsequently, transmitting a pilot again with fewer contenders,
- transmitting a data frame without first transmitting a pilot,
- transmitting a data frame pretending to have sensed a nonexistent reaction frame following own pilot.

Instances of malicious and/or counterproductive brute-force station behavior include:

- transmitting a reaction frame following own pilot in order to be recognized as the winner (if the pilot has collided, the station's data frame will too),
- transmitting a reaction frame following a colliding contention slot (only strengthens a recipient's reaction frame if the slot actually contained a successful pilot, otherwise draws two or more stations into a data frame collision),
- transmitting a pilot in a reaction slot, pretending not to have sensed the preceding pilot or pilot collision (will be perceived by some other station as a reaction frame, leading to a data frame collision),
- refraining from transmitting a reaction frame following a successful pilot (prevents from reception of a data frame),
- pretending not to have sensed reaction frames following other stations' non-colliding pilots in order

to recognize itself as the winner (the station's data frame will collide with a legitimate winner's), and

- jamming other stations' data frames.

We postulate that the winner policy be *verifiable* i.e., that selfish brute-force behavior be detectable by some feasible detection mechanism. Thus verifiability of a winner policy not so much depends on the rules of requesting to transmit as it does on the agreed upon detection mechanism. A conceptual mechanism, called a *verifier* and meant as a deterrent but not necessarily actually deployed, is a tracking device complete with both a directional and an omnidirectional antenna. The directional antenna is able, which all the stations are aware of, to track any station for a while and check the timing of the pilot and reaction frames it transmits.² Upon detection of a deviation from the winner policy, the verifier may impose a predefined punishment upon the perpetrator e.g., jam its forthcoming data frame transmission using the omnidirectional antenna. Assume further that any WLAN station is allowed to do the same upon a deviation it has detected with its omnidirectional antenna.

Note that a verifier may be implemented as a separate trusted party or at any WLAN station, honest or selfish, willing to incur the extra cost of a directional antenna.³ This is because an abuse of the right to jam data frames only amounts to malicious, and not selfish behavior. Moreover, a verifier need not interpret the contents of the frames it receives and need not (in fact cannot) receive frames transmitted by other stations than the one it is tracking. The latter limitation prevents a verifier from detecting all the above types of malicious and/or counterproductive behavior except the last. Nevertheless, all the above types of selfish behavior are detectable, except the last one, which, however, is detectable with an omnidirectional antenna at any other WLAN station (Fig. 4). From a practical viewpoint, since the length of the contention interval is upper bounded by $2E$ slots, so is the required duration of tracking a station.

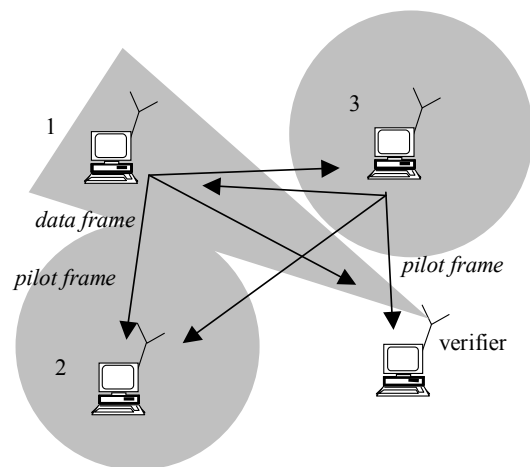


Figure 4. Detection of selfish behavior: station 1 transmits a data frame to station 2 pretending to have sensed a reaction frame following its colliding pilot, which the verifier cannot detect, but station 3 can.

² Thus a steered, rather than switched antenna system is required [16].

³ A distributed punishment scheme is proposed in [13].

Definition 1: A winner policy is called *verifiable* if any selfish brute-force station behavior is verifier detectable.

Verifiability imposes some limitations on the rules of requesting to transmit: they have to entail transmission of some physical signals, so that deviations from the winner policy will not go unnoticed. The benefits of extraneous collision detection are thus tangible. Basic access CSMA/CA does not qualify: frequently starting a data frame transmission in an early slot is selfish, but not detectable behavior (may pass as a lucky streak under HR). A winner policy must also provide enough rules to check against e.g., limiting the number of pilots to one per station per protocol cycle. Inappropriate rules might enable undetectable selfish behavior e.g., negative clear-to-send acknowledgment semantics (where the *lack* of a reaction frame indicates a successful pilot).

C. Performance

To formally describe how a winner is elected, let the vector $\mathbf{f} = (f_1, \dots, f_E)$ represent the channel feedback produced by the stations' selections in a protocol cycle, where f_i is the number of stations selecting slot i . For RT/ECD, only $f_i = 0, 1$, and >1 can be distinguished. Recall that \mathbf{f} may not be fully observable since there is no pilot transmission once a winner is elected. Still, an observed \mathbf{f} (or a prefix thereof) is the same at all the stations and provides enough information to elect a winner, if any. Define the win function as $u: \{1, \dots, E\} \times \{0, 1, >1\}^E \rightarrow \{0, 1\}$, with $u_i(\mathbf{f}) = 1$ if the pilot in slot i wins the contention and $u_i(\mathbf{f}) = 0$ otherwise. Obviously, $u_i(\mathbf{f}) = 1$ implies $f_i = 1$ and $u_j(\mathbf{f}) = 0$ for all $j \neq i$. If $u_i(\mathbf{f}) = 0$ for all $i = 1, \dots, E$ then \mathbf{f} produces no winner. For the protocol cycle depicted in Fig. 3, we have $\mathbf{f} = (0, >1, 0, 1, 1, 0, 1, 1)$, and $u_5(0, >1, 0, 1, 1, 0, 1, 1) = 1$.

In a cooperative setting, two primary objectives of the winner policy are high bandwidth utilization and (at least long-term) fairness of the bandwidth distribution. The latter stems from the symmetry of RT/ECD, and the former depends on the win function. Let O denote the average scheduling overhead per protocol cycle, including the synchronization slot plus the contention and reaction slots (in Fig. 3 it is 9 slots); let ω_n be station n 's win rate (the long-term proportion of protocol cycles where it wins the contention), and let $\Omega = \sum_{m=1}^N \omega_m$. Under saturation load, station n 's bandwidth share and the overall bandwidth utilization are:

$$b_n = \frac{\tau_{\text{DATA}} \cdot \omega_n}{\tau_{\text{slot}} \cdot O + \tau_{\text{DATA}} \cdot \Omega}, \quad (2)$$

$$b_{\Sigma} = \sum_{m=1}^N b_m = \frac{\tau_{\text{DATA}} \cdot \Omega}{\tau_{\text{slot}} \cdot O + \tau_{\text{DATA}} \cdot \Omega}. \quad (3)$$

In a noncooperative setting, a win function should also discourage deviations from a standard selection configuration, say HR. The probability that HR selects slot i is $p_{\text{HR}}(i) = 1/E$. A selfish deviation i.e., SR, may consist in selecting slot i with a probability $p_{\text{SR}}(i)$ that decreases in i if the win function is to favor early slots, or

increases if the win function is to favor late slots. Let $b_{\text{HR}}(N, x)$ and $b_{\text{SR}}(N, x)$ be the corresponding bandwidth shares if x stations use SR and the rest use HR. The ratio

$$I(N) = b_{\text{SR}}(N, 1)/b_{\text{HR}}(N, 0) \quad (4)$$

measures the incentive to use SR given that the other stations use HR. Preferably, it should be kept around or below one. At the same time, adopting HR at all the stations should lead to a high bandwidth utilization $N \cdot b_{\text{HR}}(N, 0)$. Consider a few heuristic win functions:

- RT/ECD-0: $u_i(\mathbf{f}) = 1$ implies $f_j = 0$ for all $j < i$ (the first non-colliding pilot wins if not preceded by a pilot collision, otherwise no winner is elected),
- RT/ECD-∞: $u_i(\mathbf{f}) = 1$ for $i = \min\{j \mid f_j = 1\}$ (the first non-colliding pilot wins),
- RT/ECD-hash: the winner is elected from all non-colliding pilots using a deterministic hash function so that none of them is a priori favored or discriminated; formally, $u_i(\mathbf{f}) = 1$ for $i = \text{hash}(\mathbf{f})$, where $\text{hash}: \{0, 1, >1\}^E \rightarrow \{1, \dots, E\}$ and if the set $\{j \mid f_j = 1\}$ is nonempty then $\text{hash}(\mathbf{f}) \in \{j \mid f_j = 1\}$,
- RT/ECD-late: $u_i(\mathbf{f}) = 1$ for $i = \min\{j \geq i_0 \mid f_j = 1\}$ (the first non-colliding pilot in slots i_0, \dots, E wins),
- RT/ECD-second: $u_i(\mathbf{f}) = 1$ implies $f_j = 1$ for exactly one $j < i$ (the second non-colliding pilot wins), and
- RT/ECD-last: $u_i(\mathbf{f}) = 1$ implies $f_j \neq 1$ for all $j > i$ (the last non-colliding pilot wins).

RT/ECD-0 and RT/ECD-∞ favor pilot transmissions in early slots, which the latter three win functions discriminate, whereas RT/ECD-hash favors HR. Fig. 5 plots (3) (equal to $N \cdot b_{\text{HR}}(N, 0)$) and $I(N)$, assuming $E = 8$, $\tau_{\text{DATA}}/\tau_{\text{slot}} = 20$ (corresponding to 1500-byte data frames), and $p_{\text{SR}}(i) = \text{const.}/\psi^{i-1}$ with a moderate $\psi = 1.3$ for the first three win functions and $i = 1/1.3 = 0.77$ for the latter three. MAC-layer acknowledgments, if prescribed, are incorporated in τ_{DATA} . The plots have been obtained via Monte Carlo simulation (with 95% confidence intervals narrowed down to 5% of the sample averages), although exact values might also have been obtained as a tedious exercise in probability.

Clearly, with regard to O and Ω , RT/ECD-∞ is Pareto superior to RT/ECD-hash, RT/ECD-late, RT/ECD-second, and RT/ECD-last: if a given \mathbf{f} produces a winner under either one of the last four win functions, it does so under RT/ECD-∞ as well at the same or less overhead cost; an \mathbf{f} producing no winner under RT/ECD-∞ does not either under any other win function, but except for RT/ECD-0 results in the shortest contention interval. This explains why RT/ECD-∞ outperforms the other win functions in terms of bandwidth utilization (Fig. 5a). Note that RT/ECD-0 performs only slightly worse. From the performance viewpoint it is therefore desirable that

$$\text{if } u_i(\mathbf{f}) = 1 \text{ then } f_j \neq 1 \text{ for all } j < i \quad (5)$$

i.e., that the winner of a contention can start its data frame transmission immediately upon sensing a reaction frame, analogously to the uninterrupted RTS+CTS+DATA exchange in IEEE 802.11. Comparing Fig. 5a and Fig. 5b we see a tradeoff between (3) and (4).

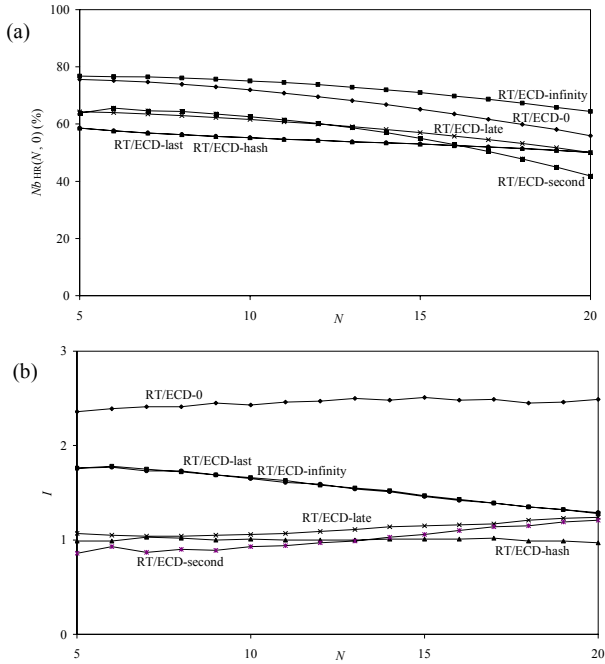


Figure 5. Performance measures of win functions; (a) (3), (b) (4).

Not surprisingly, $I(N)$ hovers around one for RT/ECD-hash,⁴ and is identical for RT/ECD- ∞ and RT/ECD-last, different as these two win functions are with regard to bandwidth utilization. RT/ECD-hash and RT/ECD-last, though they differ with regard to $I(N)$, show the same bandwidth utilization: a complete f vector must be observed, implying the same overhead, and the protocol cycles where a winner is elected coincide. RT/ECD-late (with $i_0 = 4$) and RT/ECD-second are satisfactory with regard to $I(N)$, but fail to achieve a high bandwidth utilization. RT/ECD- ∞ seems a good compromise with regard to the two performance objectives, hence will be focused upon in the sequel; RT/ECD-0 will be used for comparison. Examples of protocol cycles under RT/ECD-0 and RT/ECD- ∞ are shown in Fig. 6.

IV. CYCLE-BY-CYCLE SELECTION STRATEGIES

A station may select the contention slot in each protocol cycle based on own and other stations' selections in previous protocol cycles. With a fixed winner policy, RT/ECD-0 or RT/ECD- ∞ , we take the "teach to learn" approach: we seek a strategy discouraging other stations from playing a different strategy. For simplicity, the discouragement will be judged based on static long-term payoffs (2) rather than dynamic cycle-by-cycle scenarios.

A. Reinforcement Learning Strategies

We focus on two-type station game scenarios: for a pair of strategies, a fixed number of stations play one of them

⁴ The hash function is defined as proposed in [14]: it has a station multiply $pi = 3.14159265$ by the number whose ternary representation is f , round to the nearest integer, and use the result as an index into the set $\{j | f_j = 1\}$. It is shown via simulation not to favor any slot a priori, and the probability distribution of winning in a particular remaining slot, conditional on an observed l -slot prefix of f , is close to uniform for l up to five and N above five.

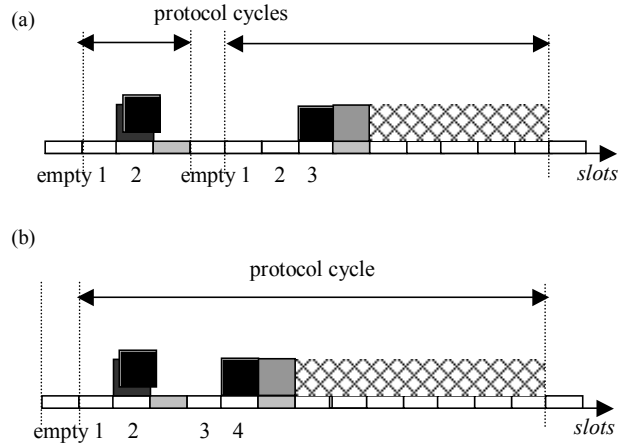


Figure 6. Example protocol cycles; (a) RT/ECD-0: slot 1 is not selected by any station, two stations select slot 2, lack of a reaction frame terminates the protocol cycle, (b) RT/ECD- ∞ : lack of a reaction frame after slot 2 causes the contention to continue, a reaction frame after slot 4 prompts the sender of the successful pilot to transmit a data frame.

and the rest play the other. One way of seeking a good candidate strategy is to devise a large number of strategies and have each possible pair of them confront each other in a two-type station scenario. The main difficulty lies in the fact that the set of conceivable strategies probably cannot be parameterized. We posit that a conceivable strategy is only constrained to be

- *anonymous* i.e., not relying on the knowledge of any other station's identity,⁵ and
- *rational* in the sense of [15], meaning that (i) short-term course of play rewarded by the win function becomes more likely in the future, and (ii) no course of play is abandoned forever (to maintain responsiveness to varying opponents' play).⁶

Numerical values of long-term payoffs under RT/ECD- ∞ were obtained via Monte Carlo simulation and compared for various station strategies against the backdrop of RT/ECD-0. Simulation runs were repeated until the 95% confidence intervals shrank to less than 5% of the sample averages. The following setting was assumed: saturation load, $E = 10$, $\tau_{DATA}/\tau_{slot} = 20$, and $N = 5, 10$, and 20 . Each simulation run generated a two-type station scenario with a fixed number of stations using a particular strategy. Eight strategies were examined, including HR, SR, and six heuristic strategies briefly described below; each of them was played by a number of stations varying from 0 to N against the other stations playing one of the other seven strategies. All the examined strategies are arguably anonymous and, except for HR and SR, rational. The six rational strategies define an *update period* spanning a number UP of consecutive protocol cycles and employ techniques similar to reinforcement learning [15]. Within an update period, a station sticks to a fixed selection configuration and

⁵ In particular, collusion with other selfish stations is ruled out. Merely knowing the number of selfish stations might enable token passing by the selfish stations with the honest stations all but cut off [14].

⁶ A selfish station might revert to honest play if its bandwidth share drops below fair; however, being oblivious of the other stations' identities, it cannot reliably determine a fair bandwidth share.

gathers the statistics of winning slots; at the end, the selection configuration may change as dictated by the gathered statistics. The considered strategies are:

- *Honest Randomizer* (HR) – static selection configuration with $p_{HR}(i) = 1/E$,
- *Selfish Randomizer* (SR) – static selection configuration with $p_{SR}(i) = const./\psi^{i-1}$ and $\psi = 2$ i.e., favoring early contention slots,
- *Adjusted Selfish Randomizer* (ASR) – a variety of SR with ψ adjusted as follows: at the end of an update period a station computes *tilt*, the difference between the number of protocol cycles where the winning slot preceded the selected slot and the number of those where the converse was true; if $tilt > 0$ (< 0), ψ is incremented (decremented) by 0.1 to fluctuate between the imposed bounds of 0.5 and 2,
- *Adjusted Range* (AR) – a uniform probability distribution on a subset of $\{1, \dots, E\}$ adjusted as follows: the station stores *left* and *right*, the left and right boundaries of the adjusted range, and at the end of an update period reads *newleft* and *newright*, the earliest and latest winning slot observed; subsequently the station sets $left := (1 - \kappa) \cdot left + \kappa \cdot newleft$ and $right := (1 - \kappa) \cdot right + \kappa \cdot newright$ with $\kappa = 0.3$; if no winning slots were observed, *left* and *right* expand by one to the left and right, respectively ($1 \leq left \leq right \leq E$ is maintained at all times),
- *Round Robin* (RR) – organizes token passing among a set of anonymous stations provided that some of them play along: a station selects slots $1, \dots, E, 1, \dots$ in successive protocol cycles and counts those in which it has won; if none have occurred within an update period (the probable cause of which is the inadvertent "synchronization" with another station) then a slot i is selected at random and in subsequent protocol cycles, slots $i + 1, \dots, E, 1, \dots$ are selected,
- *Fictitious SR* (FSR) – a variety of SR with the probability distribution on $\{1, \dots, E\}$ constructed based on the histogram of real and *fictitious* winning slots over the last update period; given the observed channel feedback $f \in \{0, 1, >1\}^E$ and the winning slot i_0 (by convention, $i_0 = E + 1$ if no winner is elected), slot $i < i_0$ is a fictitious winning slot if $f_i = 0$ and $f_j \neq 1$ for $j = 1, \dots, i - 1$ i.e., any one slot qualifies that would have won if it had been selected,⁷
- *Annealed Schedule* (AS) – the schedule of selected slots within an update period is adjusted using simulated annealing [17]: a slot yielding the fewest wins is tentatively replaced by another one, whose number of wins k in the next update period determines the probability of its final admittance into the schedule, equal to $1/(1 + \exp(-k))$, and
- *Modified Annealed Schedule* (MAS) – similar to AS except that k is the number of wins or no-winner outcomes over the next update period.

Throughout the experiments, *UP* was fixed at 20 protocol cycles, except for the initial update period,

⁷ For example, in Fig. 5a, the fictitious winning slots are 1 in the first protocol cycle and 1 and 2 in the second; in Fig. 5b they are 1 and 3.

which was of random length to make the learning process asynchronous across the stations. For three two-type station scenarios and $N = 10$, Fig. 7 through 9 plot the bandwidth shares (2) normalized with respect to $1/N = 10\%$ (the ideal fair bandwidth share with no scheduling overhead). As a reference, $b_{HR}(10, 0)/10\% = 0.73$ under RT/ECD-0, and $b_{HR}(N, 0)/10\% = 0.79$, under RT/ECD- ∞ . Let x stations use σ' and the other $N - x$ use σ ($x = 0, \dots, N$). Let $b_{\sigma}(N, x | \sigma, \sigma')$ and $b_{\sigma'}(N, x | \sigma, \sigma')$ be the respective bandwidth shares ($b_{\sigma'}(N, x | \sigma, \sigma) = b_{\sigma}(N, 0)$).

Fig. 7 shows $b_{RR}(N, x | SR, RR)$ and $b_{SR}(N, x | SR, RR)$ i.e., x RR stations play against $N - x$ SR stations. Under RT/ECD-0, SR stations fare much better; interestingly, playing RR creates a win-win situation (both long-term bandwidth shares increase with x). Under RT/ECD- ∞ , playing RR yields distinctly higher bandwidth shares than playing SR and guarantees a high bandwidth share regardless of x (at least 0.81 at $x = 9$). The dashed arrows in Fig. 7a illustrate a uniform incentive to switch from RR to SR: they originate at the RR payoffs and point northwest towards the higher SR payoffs with one fewer RR station. In Fig. 7b, the arrows pointing northeast indicate a uniform incentive to switch from SR to RR. Thus "all play SR" and "all play RR" are the unique *Nash equilibria* (NE) under RT/ECD-0 and RT/ECD- ∞ , respectively [18]. Under RT/ECD-0, there is an incentive for all the stations to switch one by one to SR i.e., RR can be "invaded" by SR, but not vice versa. Likewise, under RT/ECD- ∞ , SR can be "invaded" by RR.

In Fig. 8, FSR is played against SR (x is the number of FSR stations), yielding $b_{FSR}(N, x | SR, FSR)$ and $b_{SR}(N, x | SR, FSR)$. Under RT/ECD-0 the incentives are similar to the previous scenario, but under RT/ECD- ∞ either strategy can be "partly invaded" by the other, with a unique NE at $x = 7$. That is, starting with all-SR, there are incentives for seven stations to switch to FSR one by one; starting with all-FSR, three stations will find incentives to switch to SR. Also note that FSR guarantees lower long-term payoff against SR than does RR (0.68 at $x = 8$): by its nature, FSR is more inclined to "learn" the advantageous selection configuration than to "teach" an invader to avoid selfish play. One concludes that against SR, RR is a more attractive choice than FSR.

If RR and FSR play each other directly, the resulting $b_{FSR}(N, x | RR, FSR)$ and $b_{RR}(N, x | RR, FSR)$ are shown in Fig. 9 (x is the number of FSR stations). Under RT/ECD- ∞ , FSR yields lower bandwidth shares than RR and can be invaded by RR, so the conclusion does remain valid. Yet under RT/ECD-0 the reverse is true (Fig. 9a).

As more strategies enter the picture, the need for systematic evaluation of them arises. This is addressed below, with the emphasis on the methodology rather than on recommending specific strategies.

B. Evolutionary Stability

Imagine strategy σ' "invades" strategy σ , that is, starting from all- σ , a number of stations switch one by one to σ' until a NE is reached. This number, further denoted $x_{\sigma'/\sigma}^{NE}$, is the maximum x for which

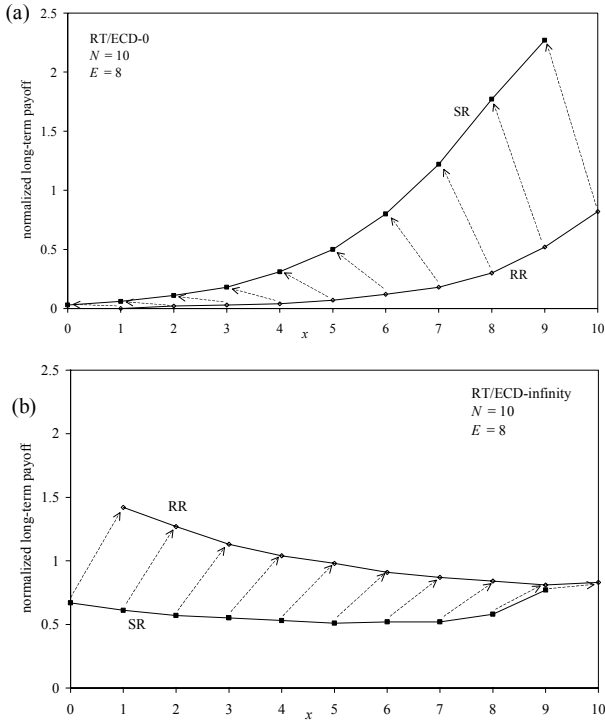


Figure 7. Two-type station scenario with RR and SR; (a) RT/ECD-0, (b) RT/ECD-∞.

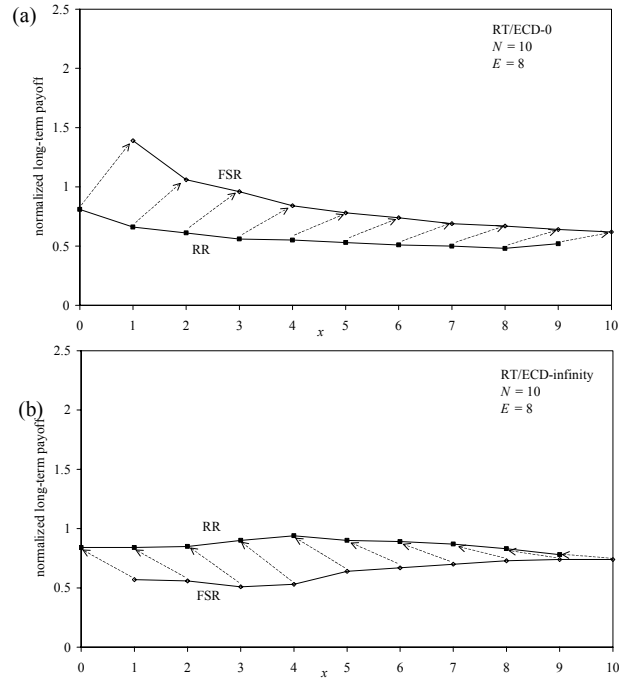


Figure 9. Two-type station scenario with FSR and RR; (a) RT/ECD-0, (b) RT/ECD-∞.

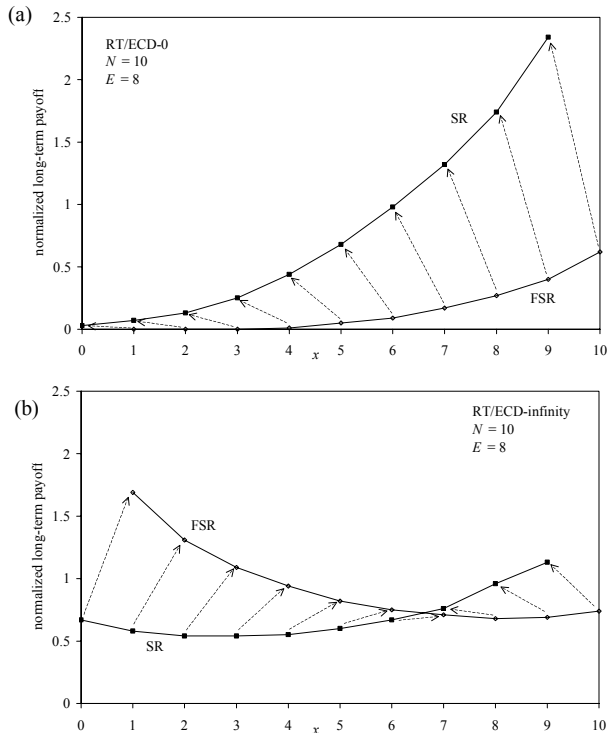


Figure 8. Two-type station scenario with FSR and SR; (a) RT/ECD-0, (b) RT/ECD-∞.

$$\forall_{y=1, \dots, x} b_{\sigma'}(N, y | \sigma, \sigma') > b_{\sigma}(N, y-1 | \sigma, \sigma'). \quad (6)$$

Because of imperfect observation of bandwidth shares, it is reasonable to consider the inequality in (6) up to some finite accuracy; in further calculations we set the accuracy to 0.1% of the total bandwidth. (Notice that in

general $x_{\sigma|\sigma}^{NE}$ and $x_{\sigma|\sigma'}^{NE}$ may differ, although in most cases they do not, cf. Fig. 8b.) Denote the bandwidth shares at $x_{\sigma|\sigma}^{NE}$ by $b_{\sigma', \sigma|\sigma}^{NE} = b_{\sigma'}(N, x_{\sigma|\sigma}^{NE})$ and $b_{\sigma, \sigma|\sigma}^{NE} = b_{\sigma}(N, x_{\sigma|\sigma}^{NE})$. Further imagine that the set \mathfrak{S} of the eight described strategies represents the whole universe of conceivable strategies for the cycle-by-cycle RT/ECD game.⁸ Consider $\sigma, \sigma' \in \mathfrak{S}$ and a two-type station scenario with strategy σ' "invading" strategy σ . This is only worthwhile if the invader stations' bandwidth shares at the resulting NE exceed the initial ones up to some finite accuracy. Hence, a desirable property of a candidate strategy σ is

$$\forall_{\sigma' \in \mathfrak{S}} b_{\sigma', \sigma|\sigma}^{NE} \leq b_{\sigma}(N, 0) + \varepsilon, \quad (7)$$

i.e., two bandwidth shares are considered almost equal if they are at most ε apart (ε is not to be confused with the accuracy mentioned in footnote 8). In essence, (7) states that no invader of σ is significantly better than σ itself.

A second desirable property states that σ is the single best invader of all strategies that are not significantly worse invaders of σ than σ itself. That is, for all $\sigma' \in \mathfrak{S}$, if $b_{\sigma', \sigma|\sigma}^{NE} > b_{\sigma}(N, 0) - \varepsilon$ then

$$\forall_{\sigma'' \in \mathfrak{S}, \sigma'' \neq \sigma} b_{\sigma'', \sigma|\sigma}^{NE} > b_{\sigma', \sigma|\sigma}^{NE} + \varepsilon. \quad (8)$$

A strategy fulfilling the "if" part might be considered "as good as σ ," were it not for the "then" part, whereby it could subsequently be invaded by another strategy, and the most effectively so by strategy σ itself. Thus σ can be called *stable* in that it resists being supplanted by another

⁸ This assumption is supported by examination of a large number of other heuristic strategies, none yielding qualitatively different results.

strategy in a process that models natural evolution [19], [20]. Because of this formal affinity to evolutionary stability we state the following definition.

Definition 2: A selection strategy σ fulfilling (7) and (8) is called *evolutionarily* $(\mathfrak{A}, \varepsilon)$ -stable.

For $N = 10$, $E = 8$, and $\tau_{DATA}/\tau_{slot} = 20$, Table I illustrates evolutionary $(\mathfrak{A}, 0.1)$ -stability of the considered strategies i.e., assuming $\varepsilon = 10\%$ of the ideal bandwidth share, or 1% of the total bandwidth. A blackened entry in row σ and column σ' indicates violation of (7). If the entry is shaded, there exist strategies σ' fulfilling the "if" part of (8) and the number of strategies σ'' violating the "then" part is indicated. An evolutionarily $(\mathfrak{A}, 0.1)$ -stable strategy has no blackened entries or nonzero numbers in its row. In Table I so are RR, FSR, and AS under RT/ECD-0, and RR and AS under RT/ECD- ∞ .

In general, a strategy σ may fare worse at the NE against a strategy σ' that is not among its best invaders than against one that is. Consider FSR under RT/ECD- ∞ for $N = 10$. As seen in the lower part of Table I, RR is among its best invaders, whereas MAS is not. However, simulation shows that $b_{FSR,RR|FSR}^{NE} \approx 4 \cdot b_{FSR,MAS|FSR}^{NE}$. Thus evolutionary stability alone (which focuses on best invaders) may therefore be misleading if some stations are unable to determine a best invader. A useful complementary measure is the minimum bandwidth share against any strategy i.e.,

$$b_{\sigma, \sigma'}^{NE} = \min_{\sigma' \in \mathfrak{A}} b_{\sigma, \sigma'}^{NE} \quad (9)$$

Under RT/ECD- ∞ , Fig. 10 depicts (9) assuming $N = 10$ and 20 , and the same E and τ_{DATA}/τ_{slot} as before. Both the

TABLE I.
EVOLUTIONARY $(\mathfrak{A}, 0.1)$ -STABILITY

		RT/ECD-0							
		HR	SR	ASR	AR	RR	FSR	AS	MAS
HR	x						2		
SR	3	x				4	2	2	
ASR	3		x			4	2	2	
AR	3				x	4	2	2	
RR						x			
FSR							x		
AS								x	
MAS	3					3	2		x

		RT/ECD- ∞							
		HR	SR	ASR	AR	RR	FSR	AS	MAS
HR	x				7		7	2	
SR		x	1	1					
ASR			x					2	2
AR					x				2
RR						x			
FSR						1	x	2	
AS								x	
MAS									x

evolutionarily stable strategies, RR and AS, along with MAS emerge as the prime recommendations. MAS is particularly attractive since it is the only one to remain evolutionarily $(\mathfrak{A}, 0.1)$ -stable for $N \neq 10$; AS remains so for $N > 10$ and is "nearly stable" for $N < 10$ (for $N = 5$, (8) is only violated for $\sigma' = MAS$). HR is the only strategy to yield higher bandwidth shares for $N = 20$ than for $N = 10$; this can be attributed to the NE probability distribution on $\{1, \dots, E\}$ becoming uniform as N increases.

V. CONCLUSION AND FURTHER RESEARCH

In the context of ad hoc WLANs populated by mutually impenetrable groups of stations, we have identified two main components of a contention MAC protocol: a winner policy and a selection strategy, to conclude that honest stations are prone to "bandwidth stealing" by selfish stations. We have addressed this problem by postulating that (i) honest stations use a carefully designed strategy teaching selfish stations that the best reply is to stick to the same strategy, and (ii) the winner policy be designed so that such a strategy can indeed be found and yields high bandwidth shares for honest stations. For RT/ECD winner policies we have further argued that performance considerations narrow the choice down to RT/ECD- ∞ ; RT/ECD-0 was used as a backdrop. A number of cycle-by-cycle reinforcement learning strategies have been evaluated using an introduced notion formally akin to evolutionary stability.

Note that evolutionary stability is tied to a particular N (also to E and τ_{DATA}/τ_{slot} , which, however, can be assumed constant). This calls for checking (7) and (8) for a range of N . In particular, Fig. 10 shows that strategies like AS, MAS, or RR hold their own against other strategies across various N . There are two problems with our approach that deserve further research.

First, we have been unable to construct a universe of selection strategies that cover all possible outcomes in terms of the bandwidth share distribution. The very possibility of such a construction is an open problem.

Second, if a station is aware that the other stations use reinforcement learning, it may try sophisticated play [21]. That is, instead of using its strategy as is, it guesses the other stations' strategies and takes advantage of their learning capabilities. This possibility significantly widens the scope of conceivable strategies. For example, suppose

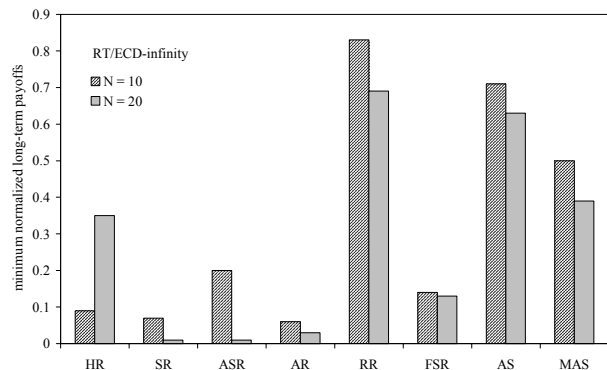


Figure 10. Minimum bandwidth shares under RT/ECD- ∞ .

that in the all-FSR scenario under RT/ECD- ∞ , one station lengthens its update periods whenever the current histogram of winning slots is tilted towards the early slots. Hence, it plays SR at times against $N - 1$ FSR stations. This is advantageous, cf. Fig. 8b at $x = N - 1$: the FSR stations will quickly have learned to avoid the early slots and so, instead of "teaching" the invader station a lesson will themselves be "taught." In game-theoretic parlance, a Stackelberg equilibrium will have been reached, the invader becoming the "leader" [9], [15]. As a form of protection, an FSR station might check if its own win count over the last update period is null and the other stations' remains positive, and if yes, temporarily switch to SR. When enough FSR stations have done so, the invader's bandwidth share drops even below that at $x = N$. However, the presence of such protection might trigger further sophistication on the part of the invader and so forth. Whether in the end "bandwidth stealing" still occurs is another open problem.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Education and Science, Poland, under Grant 1599/T11/2005/29.

REFERENCES

- [1] E. Altman, R. El Azouzi, and T. Jimenez, "Slotted Aloha as a game with partial information," *Computer Networks and ISDN Systems*, vol. 45, pp. 701–713, 2004.
- [2] A.B. MacKenzie and S.B. Wicker, "Stability of multipacket slotted ALOHA with selfish users and perfect information," *Proc. IEEE INFOCOM 2003*, San Francisco CA, March/April 2003.
- [3] IEEE standard for information technology, LAN/MAN - specific requirements - Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, ISO/IEC 8802-11:1999.
- [4] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On cheating in CSMA/CA ad hoc networks," *Proc. IEEE INFOCOM 2005*, Miami, Florida, March 2005.
- [5] J. Konorski, "Solvability of a Markovian model of an IEEE 802.11 LAN under a backoff attack," *Proc. IEEE MASCOTS 2005*, Atlanta GA, Sept. 2005, IEEE Comp. Soc. Press 2005.
- [6] T. Heikkinen, "On learning and the quality of service in a wireless network," *Proc. Networking 2000*, Paris, May 2000, LNCS 1815, Springer-Verlag, 2000, pp. 679–688.
- [7] Y.A. Korilis, A.A. Lazar, and A. Orda, "Architecting noncooperative networks," *IEEE J. Selected Areas Commun.*, vol. 13, pp. 1241–1251, July 1995.
- [8] B.A. Sanders, "An incentive compatible flow control algorithm for rate allocation in computer networks," *IEEE Trans. Comput.*, vol. 37, pp. 1067–1072, 1988.
- [9] S. Shenker, "Making greed work in networks: a game-theoretic analysis of switch service disciplines," *IEEE/ACM Trans. Networking*, vol. 3, pp. 819–831, 1995.
- [10] I. Chlamtac and A. Ganz, "Evaluation of the random token protocol for high-speed and radio networks," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 969–976, 1987.
- [11] ETSI TC Radio Equipment and Systems: High performance radio local area network (HIPERLAN); services and facilities; Version 1.1, RES 10, 1995.
- [12] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 Hotspots," *Proc. MobiSys 2004*, Boston MA, June 2004.
- [13] P. Kyasanur and N.H. Vaidya, "Detection and handling of MAC layer misbehavior in wireless networks," *Proc. Int. Conference on Dependable Systems and Networks*, San Francisco, June 2003.
- [14] J. Konorski and M. Kurant, "Application of a hash function to discourage MAC-layer misbehaviour in wireless LANs", *J. Telecomm. and Inf. Technology*, vol. 2, pp. 38–46, 2004.
- [15] E.J. Friedman and S. Shenker, "Synchronous and asynchronous learning by responsive learning automata," *Mimeo* 1996.
- [16] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: a complete system solution," *IEEE J. Selected Areas Commun.*, vol. 23, pp. 496–506, 2005.
- [17] L. Ingber, "Simulated annealing: practice versus theory," *Math'l. Comput. Modelling*, vol. 18, pp. 29–57, 1993.
- [18] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, London: MIT Press, 1991.
- [19] D. Fudenberg and D.K. Levine, *The Theory of Learning in Games*. Cambridge, MA, London: MIT Press, 1998.
- [20] X. Yao, "Evolutionary stability in the n-person iterated Prisoners' Dilemma," *BioSystems*, vol. 39, pp. 189–197, 1996.
- [21] P. Milgrom and J. Roberts, "Adaptive and sophisticated learning in normal form games," *Games and Economic Behaviour*, vol. 3, pp. 82–100, 1991.

Jerzy Konorski received the M.Sc. degree in electrical engineering from the Technical University of Gdansk, Poland, in 1976 and the Ph.D. degree in computer science from the Institute of Computer Science, Polish Academy of Sciences, Warsaw, in 1984. He is currently with the Department of Information Systems, Gdansk University of Technology, where he teaches probability theory, operational research, and network theory, and conducts research in computer networking, performance evaluation, information systems, data transmission, and distributed systems.

Dr. Konorski has worked on a number of government and European projects, and authored over 30 papers published in international journals and conference records, and co-authored another 10. His current work focuses on the application of game theory to medium access control in wireless networks.