Adversarial Contention Resolution Games

Giorgos Chionas[1] , **Bogdan S. Chlebus**[2] , **Dariusz R. Kowalski**[2] and **Piotr Krysta**[1]
[1]Department of Computer Science, University of Liverpool, UK
[2]School of Computer and Cyber Sciences, Augusta University, GA, USA
{gchionas, pkrysta}@liverpool.ac.uk, {bchlebus, dkowalski}@augusta.edu

Abstract

We study contention resolution (CR) on a shared channel modelled as a game with selfish players. There are n agents and the adversary chooses some $k \leq n$ of them as players. Each participating player in a CR game has a packet to transmit. A transmission is successful if it is performed as the only one at a round. Each player aims to minimize its packet latency. We introduce the notion of adversarial equilibrium (AE), which incorporates adversarial selection of players. We develop efficient deterministic communication algorithms that are also AE. We characterize the price of anarchy in the CR games with respect to AE.

1 Introduction

Agents using a shared channel concurrently need to resolve contention for access to the shared medium if they want to broadcast on the channel as quickly as possible. There are n agents and the adversary chooses some $k \leq n$ of them as players, each having a packet to transmit. A packet is transmitted successfully if there are no other simultaneous transmissions. The challenge is to resolve contention while not knowing the selection of the k participants in advance. Performance of algorithms scheduling transmissions is measured by latency defined as the delay of the last transmitted packet. Contention resolution on a shared channel has been defined in context of Ethernet by [Metcalfe and Boggs, 1976], and has won Metcalfe the 2023 Turing Award. It has been extensively studied in cooperative settings, see, e.g., [Chlebus, 2001; De Marco *et al.*, 2019].

[Fiat *et al.*, 2007] initiated the work on contention resolution algorithms for a shared channel that provide the number of contenders as feedback and selfish agents seek to minimize their individual latency costs. They designed randomized algorithms that are Nash equilibria with bounded latency. [Christodoulou *et al.*, 2014] studied games on a shared channel where the cost of each agent depends on the number of attempted transmissions before success. They designed randomized algorithms that are Nash equilibria with bounded expected cost.

1.1 Summary of Contributions

We study contention resolution (CR) games where selfish agents use only deterministic strategies. We assume minimal feedback from the channel in that once a player successfully transmits, it receives an acknowledgment of this fact and may no longer contend for access to the medium. To provide a framework facilitating design of equilibria, we incorporate the adversary who chooses a configuration of k players from among all n agents. We introduce a concept of *adversarial equilibrium (AE)* for such games. AE models a situation where the players are risk averse so they would not deviate even if there exist one configuration in the game that could be chosen by the adversary and make them worse-off. This concept combines a notion of classical worst-case adversary from distributed computing [Greenberg and Winograd, 1985; Komlós and Greenberg, 1985], with that of Nash equilibrium and Pareto optimal solution concepts from game theory [Osborne and Rubinstein, 1994]. AE is Pareto optimal in a sense that for any player there is no alternative strategy that is as good for each configuration, and strictly better for at least one configuration.

We design efficient deterministic communication algorithms that are also AE, with sublinear maximum latency. A summary of our communication algorithms along their performance is given in Table 1. The existence of such algorithms allows us to consider the price of anarchy and the price of stability [Nisan *et al.*, 2007; Maschler *et al.*, 2020] in CR games with respect to AE.

Our equilibria with deterministic strategies imply an upper bound of $O(1)$ on the price of stability (PoS) in the CR game for any n and $k = 2$ and $k = 3$. To compare, [Fiat *et al.*, 2007] design Nash equilibria in randomized strategies in their (different CR) model, achieving $O(n)$ latency with overwhelming probability.

We characterize the price of anarchy (PoA) by showing it is in the interval $[\frac{n}{\Theta(\log n)}, \frac{n+1}{\Theta(\log n)}]$ when $k = 2$ and it is unbounded when $k \geq 3$. In comparison, [Fiat *et al.*, 2007] prove that PoA is always unbounded in their model of shared channel, and PoS is $O(1)$ with high probability.

A CR game can be modeled as a simultaneous play of an extensive game, see, e.g., [Nisan *et al.*, 2007, Sec. 3.7]. A special kind of an extensive game is a repeated game, where each stage is the same game. Our game is quite different because the games played in each stage (time) might be dif-

Class	Active Players	Algorithm	Max-Latency	**PoS**, Theorem 4	**PoA**, Theorem 5
All	$k \geq 3$	Persistent_RR(n)	$O(n)$, [Full version]	$\frac{n}{\Theta(k\log(n/k))}$	unbounded
GT	$k \geq 2$	Noisy_GTRR(n)	$O(n)$, Theorem 1	$\frac{n}{\Theta(k\log(n/k))}$	unbounded
GT	$k = 2$	Half_Size(n)	$O(\log n)$, Theorem 2	$O(1)$	$\left[\frac{n}{\Theta(\log n)}, \frac{n+1}{\Theta(\log n)}\right]$
GT	$k = 3$	Alt_Rec(n)	$O(\log n)$, Theorem 3	$O(1)$	unbounded

Table 1: Our results. *Grim-Trigger* protocols (GT), see Section 2, is a subclass of all protocols (All). Bounds on PoS in the table are derived by dividing the latency of each algorithm by the known lower bound $\Omega(k\log(n/k))$ on latency of (n,k)-selectors, which is the optimal outcome without selfish agents.

ferent. A player who deviates and transmits successfully, no longer participates in the game and cannot be "punished" later for its misbehavior. For this reason "folk theorems" [Maschler *et al.*, 2020; Osborne and Rubinstein, 1994] about cooperation and punishment in repeated games do not apply, see [Fiat *et al.*, 2007]. On the other hand, designing equilibria is impossible without using an appropriate "punishment" mechanism. [Fiat *et al.*, 2007] used deadlines as such a mechanism, and using it showed existence of equilibria. We use a *grim trigger* punishment mechanism to design our equilibria.

1.2 Related Work on Cooperative Contention Resolution

Shared channels model contention occurring in local-area networks, see [Metcalfe and Boggs, 1976; Chlebus, 2001]. [Bender *et al.*, 2005] proposed to study broadcasting in shared channels with queue-free stations in the framework of adversarial queuing. [Chlebus *et al.*, 2012] introduced deterministic distributed broadcast performed by stations with queues in adversarial shared channels. [Anantharamu *et al.*, 2019] studied latency of broadcasting by deterministic algorithms in shared channels with adversarial packet injection. The meaning of "adversarial" in these papers refers to the way of how players are generated and is different to our adversary, who decides which $k \leq n$ players are present in a configuration of the game. These papers do not study game theoretic settings.

Contention resolution (CR) algorithms have been studied to help multiple users use efficiently a shared channel. Initially it was assumed that agents would respect the given algorithm. [Greenberg and Winograd, 1985] proved a lower bound $\Omega(k\log_k n)$ on the length of CR protocols, later improved in [Clementi *et al.*, 2001], who showed a lower bound $\Omega(k\log(n/k))$. [Komlós and Greenberg, 1985] proved that there exist CR protocols of length $O(k\log(n/k))$, matching the lower bound in [Clementi *et al.*, 2001], but their proof was only existential. Later, [Kowalski, 2005] showed a polynomial time CR protocol of length $O(k\log^c n)$, for a constant $c > 1$, which employs partial selectors by [Indyk, 2002; Chlebus and Kowalski, 2005], efficient dispersers ([Ta-Shma *et al.*, 2007]) and superimposed codes ([Kautz and Singleton, 1964] and [Porat and Rothschild, 2011]).

Acknowledgement-based shared-channel algorithms, considered in this work, have been extensively studied in various communication problems, both deterministic and randomized [Abramson, 1970; Chlebus *et al.*, 2012; De Marco and Stachowiak, 2017; Hradovich *et al.*, 2021; Komlós and Greenberg, 1985; Kowalski, 2005].

2 Technical Preliminaries

There are n selfish agents (players), each having a single packet to be broadcast on a shared channel. We will use the term agent and player interchangeably. Each agent has a unique *name (id)* which is an integer in the range $[n]$. The communication is in synchronous time slots (also called rounds or steps, interchangeably).

A shared channel. If only one agent transmits a message in a round then the transmission is successful. If two or more agents transmit their packets at the same time slot then there is a collision on the channel and none of them is successful. Agents attached to the channel receive feedback in each round. In the model of *acknowledgements* we use, a successful transmission results in the transmitting agent receiving an acknowledgement (ack) while the feedback for other agents is undetermined. If an agent transmits a dummy message to increase contention, then this is a *noisy transmission*.

2.1 Contention Resolution Games

A distributed algorithm executed by an agent serves as its strategy. We consider only deterministic algorithms. An algorithm determines for each round if the agent transmits, pauses, or possibly halts and exits. An algorithm is *oblivious* if the sequence of attempts to transmit and pauses for each agent is determined in advance and encoded as a sequence of zeros and ones. An oblivious algorithm can be represented as a binary array with n rows, row i representing agent i's schedule of transmissions. The number of columns in such array is an upper bound on the algorithm's running time. An array representing an oblivious algorithm, with the property that if k agents only participate in an execution and each of them succeeds in its transmission, is an (n,k)-*selector*.

Each player executes the same deterministic algorithm determined by the parameters n, k. Player's actions are determined by its unique id. In the course of a game, the algorithm executed by a player could use feedback in a round to define its action in the next round(s). In the model with acknowledgements, a player can only hear acknowledgement of its own successful transmission, i.e., it follows a hardwired schedule of transmissions until it hears acknowledgement, and then it switches off.

A *game configuration* is as a subset of $k \leq n$ agents designated as players and none of the players knows who other $k-1$, players are. Let $\mathcal{F}_n^k$ be the set of all k-element subsets of $[n] = \{1,\ldots,n\}$, that is, the set of all possible game configurations. Initially an adversary chooses a configuration $K \in \mathcal{F}_n^k$ and informs every player of their status with respect to the configuration. We refer to our game as a *CR game with acknowledgements*. Any algorithm (strategy) for player i, usually denoted $s_i \in \{0, 1, \not{1}\}^*$, could be modelled using two constructs:

- $0/1$ sequence for player i, where 1, "transmission one", in position t means that i transmits in step t (unless it has already transmitted successfully and switched off), and 0 means that it is idle in step t. Players can also transmit a "noisy one", denoted $\not{1}$. If player i transmits $\not{1}$ in step t, their packet is not transmitted even if no one else transmits. If any j ($j \neq i$), transmits 1 in step t, they are unsuccessful.
- The adaptive switch-off rule: once a player receives an acknowledgement of its successful transmission, it stops following the activities encoded in the sequence (it could also be viewed as the agent swapping all next positions of the sequence to zeros).

Adversarial equilibria. An oblivious algorithm $(s_1, s_2, \ldots, s_n)$ for n players is called an (n, k)-*adversarial equilibrium*, (n, k)-AE, iff for each player $i \in [n]$ and any change of his strategy (a.k.a. a deviation) from s_i to some other strategy $s'_i \neq s_i$ (and all other players $j \neq i$ following their strategies s_j), if there is a configuration K of k players for which the change strictly decreases the latency of player i, then there is a configuration K' of k players for which the change strictly increases the latency of player i. K (K', resp.) is called an *improving* (*worsening*, resp.) configuration for player i under deviation s'_i. More formally, $s = (s_1, s_2, \ldots, s_n)$ is an (n, k)-AE, if and only if

$$\forall i \in [n]\, \forall s'_i, s'_i \neq s_i :$$

$$\big(\exists K \in \mathcal{F}_n^k : lat_i((s'_i, s_{-i}), K) < lat_i((s_i, s_{-i}), K)\big) \Rightarrow$$

$$\big(\exists K' \in \mathcal{F}_n^k : lat_i((s'_i, s_{-i}), K') > lat_i((s_i, s_{-i}), K')\big),$$

where $lat_i(s, K)$ is the latency of player i under s and K, $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ and $(s'_i, s_{-i}) = (s_1, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_n)$. Given $(s_1, s_2, \ldots, s_n)$, *maximum latency* is the latest successful transmission time of any player under any configuration without deviation.

Grim-trigger algorithms. Our particular focus is on a natural and broad subclass of protocols, called *Grim-trigger algorithms*, where a player (before deviation) is allowed to use noisy ones $\not{1}$, only after the last transmission one 1. The notion of Grim-trigger strategies were used in a similar context in repeated games, see [Axelrod and Hamilton, 1981; Nisan *et al.*, 2007, Sec. 27.2].

Price of Anarchy and Stability. [Koutsoupias and Papadimitriou, 2009] introduced the notion of the *Price of Anarchy* (PoA) to measure how far from the socially optimal outcome is the outcome of the worst (Nash) equilibria. More formally, PoA is the ratio of the outcome of the worst case Nash equlibrium and the socially optimal outcome without selfish agents. The other widely studied concept of *Price of Stability* (PoS) [Anshelevich *et al.*, 2004], is the ratio between the best-case (Nash) equilibrium and the socially optimal outcome.

Algorithm 1: Noisy_GTRR(n), player i

```
for t ∈ [n] do
    if t = i then
        transmit(packet)   (switch-off upon ack)
    else if t > i then transmit(noise) ;
```

Notation. Given a sequence ρ, $\rho[t_1, \ldots, t_a]$ denotes a subsequence of ρ consisting of values of a given sequence ρ in positions $t_1, \ldots, t_a$. Also, $[n] = \{1, \ldots, n\}$.

3 Algorithmic Equilibria

We first show simple adversarial equilibria (AE's) with maximum latency n, to be used as building blocks to construct more sophisticated AE's with sublinear latency.

3.1 Enhanced Round Robin Equilibria

Consider the following Persistent Round Robin-type algorithm Persistent_RR(n): given any deterministic permutation π of $(1, 2, \ldots, n)$, player $i \in [n]$ transmits in time slot i and if i is unsuccessful in this time slot i, i.e., i has not heard an acknowledgement (ack), then i transmits from that time point until it gets acknowledgement but no later than time slot n (inclusive). Algorithm Persistent_RR(n) is obviously an (n, k)-selector with maximum latency n. It can also be shown to be (n, k)-AE for all n, k such that $n \geq k > 2$, except $n \geq 3$ and $k = 2$ (these claims are relegated to the full version).

A small modification could turn Persistent_RR(n) into an $(n, 2)$-AE. Namely, if the first transmission of player i in round i is not successful, then the player continues transmitting a dummy message (noise) till the end of round n. The noisy message introduces a permanent channel blocking effect after unsuccessful unique transmission of an active player; hence, belongs to the class of grim-trigger protocols and we call it *Noisy_GTRR*(n). See Figure 1 for illustrations.

	1	2	3	4	5
pl1	1	1	1	1	1
pl2	0	1	1	1	1
pl3	0	0	1	1	1
pl4	0	0	0	1	1
pl5	0	0	0	0	1

(a) Persistent_RR(5) for parameter $k \geq 3$.

	1	2	3	4	5
pl1	1	1	1	1	1
pl2	0	1	1	1	1
pl3	0	0	1	1	1
pl4	0	0	0	1	1
pl5	0	0	0	0	1

(b) Noisy_GTRR(5) for $k \geq 2$. Orange are noisy $\not{1}$'s.

Figure 1: Round Robin algorithms examples.

Theorem 1 *For any $n \geq k \geq 2$, Noisy_GTRR(n) is a grim-trigger protocol that is an (n, k)-selector with maximum latency n and an (n, k)-AE for the CR game with acknowledgements.*

Proof. Being an (n, k)-selector with latency at most n follows from the fact that without any deviation, there are no

Algorithm 2: Half_Size(n), player i

$s_i \leftarrow Gen(n, i)$ /* half-size seq. for player i */
$t_{\text{last}} \leftarrow \max\{j : s_i[j] = 1\}$ /* last slot of 1 in s_i */
for $t \in \{1, 2, \ldots, t_{last}\}$ **do**
  if $s_i[t] = 1$ **then**
    $transmit$(packet) *(switch-off upon ack)*
  else $idle$;
if $t > t_{last}$ **then** $transmit$(noise) ;

	1	2	3	4
pl1	0	0	1	1
pl2	0	1	0	1
pl3	0	1	1	1
pl4	1	0	0	1
pl5	1	0	1	1
pl6	1	1	1	1

(a) Strategies of 6 players. Noisy $\cancel{1}$'s are in orange.

	1	2	3	4
pl1	0	0	1	1
pl2	0	1	0	1
pl3	0	1	1	0
pl4	1	0	0	1
pl5	1	0	1	1
pl6	1	1	1	1

(b) In configuration $\{2, 3\}$, pl3 is switched off in yellow.

	1	2	3	4
pl1	0	0	1	1
pl2	1	0	0	0
pl3	0	1	0	0
pl4	1	0	0	1
pl5	1	0	1	1
pl6	1	1	1	1

(c) $\{2, 3\}$-improving configuration for pl2.

	1	2	3	4
pl1	0	0	1	1
pl2	1	1	0	1
pl3	0	1	1	1
pl4	1	0	0	1
pl5	1	0	1	1
pl6	1	1	1	1

(d) $\{2, 6\}$-worsening conf. for pl2; pl6 blocks pl2 using $\cancel{1}$'s.

Figure 2: Half_size(6), $k = 2$. Improving and worsening configurations: pl2 deviates by putting 1 in first slot.

collisions – thus, every player transmits once and successfully in its exclusive time slot. In the remainder, we prove that it is also an (n, k)-AE.

Suppose first $k \geq 3$ and an agent i deviates so that his first transmission is in some earlier slot $j < i - 1$. Then we show existence of a worsening configuration of size k for agent i. Let us consider any configuration K of size k such that $j, j + 1, i \in K$. Both agents i and j will be unsuccessful in round j and, moreover, agent j starts transmitting dummy messages (noise). Then, in time step $j + 1$, agent $j + 1$ will collide with j, hence $j + 1$ will start transmitting noise. From this time unit on, all time steps will have agents $j, j + 1$ transmitting dummy noise messages, hence no agent will be successful till the end of the algorithm (time step n inclusive), including agent i. Therefore, the latency of agent i will be larger than n, which is worse than before the deviation ($i \leq n$).

The next possible case is when an agent i deviates and his first transmission is in slot $j = i - 1$. Let us consider any configuration K containing, among other players, i and $j = i - 1$. Similarly as in the previous case, i and j collide in time slot j and j starts noisy transmissions, which means that agent i cannot successfully transmit in slot $i = j + 1$. Hence, agent i worsens his latency in this configuration when doing this deviation.

Finally, note that if the first transmission of deviating agent i is in slot i or later, he does not have any configuration for which the deviation would improve its latency (originally i), hence in order to satisfy the definition of (n, k)-AE we do not have to show any configuration for which this deviation worsens the latency of agent i.

Let us now assume that $n \geq k = 2$. The player $i = 1$ cannot improve its latency by any deviation, as he has a unique slot 1 to transmit; so, we do not have to show any worsening configuration for him. If $i > 1$, then any transmission attempt by player i in round $j < i$ worsens his latency in configuration $K = \{j, i\}$, as after collision in round j player j triggers its grim of noise messages till the end of step n, blocking all transmissions. □

3.2 Low-Latency Solution for $k = 2$

We visualize the strategies as n $0/1$ sequences of length $\log n$, that is, as a $0/1$ matrix M of size $n \times \log n$. As the $(n, 2)$-selector, we choose as the rows of the matrix, all sequences of length $\log n$, each containing $(\log n)/2$ of 1's. The number of such sequences is $\binom{\log n}{(\log n)/2} = \Theta\left(\frac{2^{\log n}}{\sqrt{\log n}}\right) = \Theta\left(\frac{n}{\sqrt{\log n}}\right)$. We need to slightly increase the length of those sequences to say $x \log n$ for some $x \geq 1$, so that the number of the sequences is exactly n, because we have n agents. We consider the number of players n to be a *central binomial coefficient*, i.e. $n = \binom{2y}{y}$, for some integer y. The first few *central binomial coefficients* are $2, 6, 20, 70, 252, \ldots$. There are infinitely many such numbers, and $\binom{2(y+1)}{y+1} = \binom{2y}{y} \cdot 4 \cdot \left(1 - \frac{1}{2(y+1)}\right)$, meaning that the next central binomial coefficient is at least 3 times larger and less than 4 times smaller than the previous one. Hence we choose the appropriate $x_i = \frac{2i}{\log n}$, where $i \in \mathbb{N}$ such that $\binom{x_i \cdot \log n}{\frac{x_i \cdot \log n}{2}} = n$. After matrix M is defined, in each row of M we change all 0's, if any, into $\cancel{1}$'s, after the right-most 1 in this row. This construction, see Algorithm Half_Size(n), is an $(n, 2)$-AE. Given a player $i \in [n]$, procedure $Gen(n, i)$ generates (in any fixed order) all $0/1$ sequences of length $x \log n$ with exactly $(x \log n)/2$ 1's and returns i'th such sequence, called a *half-size sequence*. Figure 2 shows an example of sequences, deviation and configurations.

Theorem 2 *Half_Size(n) is a grim-trigger protocol that is an $(n, 2)$-selector with maximum latency $O(\log n)$ and an $(n, 2)$-AE for the CR game with acknowledgements.*

Proof. We first argue that for any $k = 2$ players, both players successfully transmit. This follows from the fact that given any pair of rows of M, these are two different sequences of length $\log n$, each with exactly $(\log n)/2$ 1's. Their symmetric difference has at least two positions that are $0 - 1$ and $1 - 0$, implying that both players will successfully transmit at least in these time units, so this is a $(n, 2)$-selector with maximum latency $O(\log n)$.

Observe that Half_size(n) is a grim-trigger protocol, because any player i (without deviations) transmits noisy $\cancel{1}$'s only after its last transmission 1 – see Algorithm 2.

We now argue that Half_size(n) is an $(n, 2)$-AE. For each player i let t_i be the time of the left-most (first) 1. Notice that $t_i \leq (x \log n)/2 + 1$ for each player i. Let us denote by s_i the strategy Half_size$(n)_i$ of player i. Observe that there exists a configuration under which player i transmits at time t_i. We will denote by s'_i the deviation of player i from her strategy s_i. We consider three cases.

Firstly, consider that player i deviates by modifying some bits in the time interval $[1, \ldots, t_i]$. Assume that now there exists an improving configuration in which i transmits strictly earlier than t_i. By the construction of these n sequences we observe that there exists another player j such that both i (after deviation) and j have the same subsequence until t_i, i.e., $s'_i[1, \ldots, t_i] = s_j[1, \ldots, t_i]$. Hence under the configuration $K = \{i, j\}$, after deviation, i will transmit strictly later than t_i. However, before the deviation, i would have transmitted at time t_i, since the respective subsequences of players i and j differ at least at one bit and t_i is the first 1 of player i. Thus $K = \{i, j\}$ is a worsening configuration.

Secondly, assume that i deviates by modifying some bits after t_i, i.e., in the time interval $[t_i + 1, \ldots, x \log n]$. Assume that this results in an improving configuration. Apparently, in this improving configuration i transmits after t_i. We will now construct worsening configurations based on the total number of 1's in s'_i, denoted by $ones(s'_i)$:

- $ones(s'_i) \geq (x \log n)/2$. Let t_i^{last} be the time of the $(x \log n)/2$'th 1 (one) in s'_i. By the construction of these n sequences, there exists a player $j \neq i$ such that $s'_i[1, \ldots, t_i^{last}] = s_j[1, \ldots, t_i^{last}]$. Hence, in this configuration, $K = \{i, j\}$, they will be both blocked up until t_i^{last} and since j follows the protocol afterwards she will transmit noisy $\not{1}$'s and i will not be able to transmit. Thus $K = \{i, j\}$ is a worsening configuration.
- $ones(s'_i) < (x \log n)/2$. Let t_i^{last} be the time of the last 1 (one) in s'_i. We know by the construction of these n sequences that there exists a player $j \neq i$ such that $s'_i[1, \ldots, t_i^{last}] = s_j[1, \ldots, t_i^{last}]$. Before the deviation, under configuration $K = \{i, j\}$, since Half_Size(n) is an $(n, 2)$-selector, we know that player i would have transmitted with maximum latency $x \log n$. However, after this deviation she will not transmit at all. Thus $K = \{i, j\}$ is a worsening configuration.

Lastly, assume that i deviates before and after t_i. This falls into the first case as well and therefore we can find a worsening configuration in the same way as above.

Notice that the deviator i may choose to transmit noisy $\not{1}$'s, but this has the same effect on the congestion of the channel as transmitting a packet at this given time slot, and it prevents player i from transmit. Thus, the worsening configurations constructed in the same ways as above, remain worsening for player i after such deviations. □

Remark. In case when n is not equal to a binomial central coefficient, our construction can be made to work by using a recent construction in [Griggs *et al.*, 2023] of maximal antichains with an additional chronological property. We will include details in the full version of the paper.

Algorithm 3: Alt_Rec(n), player i

```
if n ≤ 5 then Noisy_GTRR(n)_i ;
if i ≤ n/2 then
    transmit(packet)   (switch-off upon ack)
    idle
    Alt_Rec(n/2)_i
else
    idle
    transmit(packet)   (switch-off upon ack)
    Alt_Rec(n/2)_{i-n/2}
```

3.3 Low-Latency Recursive Solution for $k = 3$

The "Alternating Recursion" algorithm Alt_Rec(n) for player $i \leq n$, called Alt_Rec$(n)_i$, executes the strategy of player i in Persistent_RR(n) for $n \leq 5$, otherwise it proceeds recursively as follows. If $i \leq n/2$, the player starts to transmit, followed by silence. If the transmission was unsuccessful, it continues by executing Alt_Rec$(n/2)_i$. If $i > n/2$, the player starts with a silence, followed by transmission. If the transmission was not successful, it continues by executing algorithm Alt_Rec$(n/2)_{i-n/2}$. See Table 2 for an illustration.

Players/rows	Rounds/columns		
1	1	0	Alt_Rec($n/2$)
2	1	0	
⋮	⋮	⋮	
$n/2$	1	0	
$n/2+1$	0	1	Alt_Rec($n/2$)
$n/2+2$	0	1	
⋮	⋮	⋮	
n	0	1	

Table 2: Schematic illustration of Alt_Rec(n).

We first prove some properties of Alt_Rec(n). Let $n' \in \{3, 4, 5\}$ be the last (and smallest) argument used in the recursive process initialized by Alt_Rec($n/2$). Let σ_i be the 0-1 transmission sequence of player i generated by Alt_Rec$(n)_i$, encoding a transmission of player i in step t as 1 in position t, and no transmission – as 0. We split σ_i into two, different length, subsequent parts: $\sigma_i^{(P)}$ and $\sigma_i^{(R)}$. $\sigma_i^{(P)}$ can be viewed as a sequence of $x = \log(n/n')$ pairs, $\sigma_i^{(P)}[1, 2], \ldots, \sigma_i^{(P)}[2x-1, 2x]$, while $\sigma_i^{(R)}$, as shown next, as a sequence of length n'.

Lemma 1 *The following holds for any player $i \leq n$:*
(1) Each $\sigma_i^{(P)}[2y-1, 2y]$ is $1, 0$ if the y-th digit of the binary representation of i is 1, and $0, 1$ otherwise, for any $1 \leq y \leq x$.
(2) Sequence $\sigma_i^{(R)}$ is equal to Noisy_GTRR$(n')_{i \bmod n'}$.
(3) For any 0-1 sequence ρ containing $y \leq x$ pairs, each being either 10 or 01, there exist $n/2^y$ different players j such

that $\sigma_j^{(R)}[1,\ldots,2y] = \rho[1,\ldots,2y]$ and each j has a unique schedule in Alt_Rec$(n/2^y)$.

Proof. The proof, by induction on $y = 1,\ldots,x$, is straightforward by Alt_Rec(n)'s recursive definition. □

Does Theorem 3 subsume Theorem 2, i.e., is $(n,3)$-AE also $(n,2)$-AE? This is not true, e.g., if player 1 plays against one other player in Alt_Rec (case $k = 2$), it can deviate and transmit in the second round. This way it improves against other player $\leq n/2$ and does not worsen against any player $> n/2$. Hence, it is not $(n,2)$-AE.

Theorem 3 *Alt_Rec(n) is a grim-trigger protocol that is an $(n,3)$-selector with maximum latency $O(\log n)$ and an $(n,3)$-AE for the CR game with acknowledgements.*

Proof. The algorithm's length $T(n)$ is given by the following recursive equations: $T(n) = n$ for $n \leq 5$, and $T(n) \leq 2 + T(n/2)$ otherwise, thus $T(n) \leq O(\log n)$.

Observe that for $n \leq 5$, the $(n,3)$-selector and $(n,3)$-AE follow from Theorem 1. Therefore, in the remainder we focus on the recursive case for $n > 5$. Assume that the theorem holds for any Alt_Rec(n^*), where $n^* < n$.

We first show that it is an $(n,3)$-selector. By inductive assumption, Alt_Rec$(n/2)$ is an $(n/2,3)$-selector. Consider any three players with ids $i_1 < i_2 < i_3$ in Alt_Rec(n). If all of them are in the same half-range, i.e., all in $[1,n/2]$ or all in $[n/2+1,n]$ respectively, the selection follows from the recursive part of Alt_Rec$(n/2)$, applied to ids i_1,i_2,i_3 or to $i_1-n/2, i_2-n/2, i_3-n/2$, respectively. Otherwise, there is only one player (i_1) in the first half-range or only one player (i_3) in the second half-rage of ids, and the two other players in the other half-range of ids. The one player transmits successfully during the first two steps (more precisely, in step 1 if it is i_1 alone in the half-range $[1,n/2]$ and in the second round if it is i_3 alone in half-range $[n/2+1,n]$). The other two players block themselves in the first two rounds (following the same transmission sequences), but then they execute Alt_Rec$(n/2)$ with different ids ($i_2-n/2, i_3-n/2$ in the former case and i_1,i_2 in the latter, resp.), guaranteeing selection for them by recursive assumption about $(n/2,3)$-selection of Alt_Rec$(n/2)$. In the latter argument, it is important that the other player transmitted successfully and thus switched off before the two players started Alt_Rec$(n/2)$, as otherwise it could have performed Alt_Rec$(n/2)$ with same id as one of the other two players and block him (e.g., player $i_1 \leq n/2$ and player $i_2 = i_1 + n/2 > n/2$ would both execute Alt_Rec$(n/2)_{i_1}$, with the same id i_1).

It remains to prove $(n,3)$-AE. First consider a deviation of a player i which changes only the recursive part of its schedule: Alt_Rec$(n/2)_i$ if $i \leq n/2$ and Alt_Rec$(n/2)_{i-n/2}$ otherwise. So any improvement of the latency of player i could happen only in that part, while the transmissions and their results in the first two steps are unchanged. Thus, i must be blocked in the first two steps by some other player j in the same range of ids as i, and if the third player is in the other half of ids then it would be successful in the first two rounds and switched off in the recursive part. Thus, there will be no players executing the same sequence Alternate_Recursive$(n/2)_{i'}$ in the recursive part, for any $i' \leq n/2$. Hence, by recursive assumption, if there is any improving configuration in the part Alt_Rec$(n/2)$, there will be also a worsening configuration K (for the latency of the deviating agent i) of size 3 in Alt_Rec$(n/2)$. Now we have to show such worsening configuration also in the original Alt_Rec(n): if $i \leq n/2$ then we just take K (all three players in the lower half-range of ids), otherwise we take $K' = \{i,j,j'\}$ such that $K = \{i-n/2, j-n/2, j'-n/2\}$.

Now consider a deviation that starts in the first two steps of Alt_Rec$(n)_i$. There are three possible flips: both steps flip, only 1 flips to 0, and finally, only 0 flips to 1. The first two deviations lead to a worsening configuration, despite of any further deviations that may happen later. Indeed, we can choose two other players from the other half-range of ids – after the double flip of player i or just flipping 1 into 0, it will have the first two steps identical to those players or full of 0's, resp. Thus they together block one of the steps and keep the other silent, while before the flip (deviation) player i would have transmitted successfully in one of the two first (the unblocked) steps.

It remains to consider the deviation, ρ_i, with two 1's in the first two steps. Analogously to the original σ_i obtained from Alt_Rec$(n)_i$, we partition ρ_i into part $\rho_i^{(P)}$ containing first $x = \log(n/n')$ pairs and the remaining sequence $\rho_i^{(R)}$ of the last n' digits. There are two cases, which we consider below.

Case 1: We assume that there is a pair $y \leq x$ such that $\rho_i^{(P)}[y]$ contains at least one 0 – in this case we apply analogous case analysis as above for the first pair, showing either a direct worsening configuration on steps $2y-1, 2y$ or by using a recursive assumption for Alt_Rec$(n/2^y)$, see Lemma 1 (3).

Case 2: All pairs $\rho_i^{(P)}[2y-1,2y]$ contain only 1's, for $1 \leq y \leq x$. We can block all these 1's of deviator i by choosing any players j,j' s.t. $\sigma_j^{(P)}$ contains only pairs 10, while $\sigma_{j'}^{(P)}$ contains only pairs 01, see Lemma 1(3). Let z be the location of first 1 in $\rho_i^{(R)}$ – it exists, as otherwise i would not have any transmission during the last n' steps while, as we have argued, it could be blocked along the first $2y$ steps by some players j,j' (so its latency would worsen by not having a successful transmission). As all i,j,j' have different first $2x$ positions, they will be in different copies of Noisy_GTRR(n') that are executed in the last n' steps. It means that each player j,j' can execute any of the schedules in Noisy_GTRR(n') – it follows by Lemma 1(3). We can choose schedule for j that has first 1 in position z in Noisy_GTRR(n'), while schedule for j' having first 1 on position $\min\{z+1,n'\}$. Together with deviator i, they block position z in $\rho_i^{(R)}$, and j,j' block position $\min\{z+1,n'\}$ and all the remaining ones in $\rho_i^{(R)}$. So, player i will be blocked by the end of the algorithm, hence worsening its latency. □

Remark. If n is not a power of 2, Alt_Rec(n) can be adjusted as follows. We put Alt_Rec$(\lfloor n/2 \rfloor)_i$ (line 5), Alt_Rec$(\lceil n/2 \rceil)_{i-\lfloor n/2 \rfloor}$ (line 9), and additional correction at the end: if some 5 players $i,\ldots,i+4$ of the same prefix $\sigma_i^{(P)}$ execute Noisy_GTRR(5) at the end, while other 3 players $i',\ldots,i'+2$ of the same prefix $\sigma_{i'}^{(P)}$ execute Noisy_GTRR(3), player $i+4$ replaces $\sigma_{i+4}^{(P)}$ by $\sigma_{i'}^{(P)}$ and "joins" $i', i'+1, i'+2$ to execute Noisy_GTRR(4) together. Lengths of the resulting schedules differ by at most 1 and the current analysis holds.

4 Price of Anarchy and Price of Stability

We study here the price of anarchy (PoA) and the price of stability (PoS). Theorem 4 follows from the lower bound $\Omega(k \log(n/k))$ on the latency of selectors by [Clementi *et al.*, 2001], and its part (1) follows by the properties of Persistent_RR; the part (2) is obtained by an application of Theorem 1; and, finally, (3) follows from Theorems 2 and 3.

Theorem 4 *The PoS of the CR game with acknowledgements with n players, and k active players is at most:*
(1) $\frac{n}{\Theta(k \log(n/k))}$*, when* $n \geq k > 2$ *or* $n = k = 2$*,*
(2) $\frac{n}{\Theta(k \log(n/k))}$*, when* $n \geq k \geq 2$*,*
(3) $O(1)$*, in model with grim-trigger protocols and* $k = 2, 3$*.*

Here we will provide an almost complete characterisation of the price of anarchy.

Theorem 5 *Given the number of players n and active players* $k \leq n$*, the PoA of the CR game with acknowledgements and with grim-trigger protocols is*
(1) at least $\frac{n}{\Theta(\log n)}$*, and at most* $\frac{n+1}{\Theta(\log n)}$ *when* $k = 2$*,*
(2) unbounded when $k \geq 3$*.*

Proof. By Theorem 1, we know that Noisy_GTRR(n) is an (n, k)-AE with maximum latency n for $k = 2$. This implies that the price of anarchy for $k = 2$ is at least $\frac{n}{\Theta(\log n)}$, observing that $\Omega(\log n)$ is the latency of the shortest $(n, 2)$-selector, see [Clementi *et al.*, 2001].

We will now prove the upper bound of $\frac{n+1}{\Theta(\log n)}$ on the price of anarchy. Towards this goal we will prove that any $(n, 2)$-AE has maximum latency at most $n + 1$. Let $(s_1, \ldots, s_n)$ be any given $(n, 2)$-selector which is also $(n, 2)$-AE. Let s_1 be the player that has the maximum possible latency in this $(n, 2)$-AE and let this maximum latency be ℓ. This means that the left-most (i.e., last) transmission 1 in s_1 is at time slot ℓ, $s_1[\ell] = 1$. Because $(s_1, \ldots, s_n)$ is a $(n, 2)$-selector, there must exist other player, without loss of generality it could be s_2, such that player s_1 transmits at time ℓ under configuration $\{1, 2\}$ with no deviations. This means that player 2 blocks player 1 until time slot $\ell - 1$. More precisely, if player 1 has the but last 1 in time slot $i \leq \ell - 1$ (the last 1 in time slot ℓ could also be the only 1 in s_1), then player 2 must be such that: $s_1[1, 2, \ldots, i] = s_2[1, 2, \ldots, i]$, $i \leq \ell - 2$, $s_1[i + 1, \ldots, \ell - 1] = (0, \ldots, 0)$, and there must be at least one transmission 1 in $s_2[i + 1, \ldots, \ell - 1]$; let $i' \in \{i + 1, \ldots, \ell - 1\}$ be the time slot of the first 1 in the sequence $s_2[i + 1, \ldots, \ell - 1]$; i' is the time slot where player 2 transmits under configuration $\{1, 2\}$ with no deviations, because $(s_1, \ldots, s_n)$ is a $(n, 2)$-selector. Note that there is no noisy one $\not{1}$ in sequence $s_1[1, \ldots, \ell]$ because the players use grim-trigger protocols. To be precise, s_1 has the grim-trigger, i.e., sequence of $\not{1}$'s, after time slot ℓ, and the same is true for s_2 after the last 1 in the sequence $s_2[i + 1, \ldots, \ell - 1]$; that is, $s_2[\ell] = \not{1}$.

Let us consider the following deviations of player 1: $t_j = (\neg s_1[j], s_1[-j])$ for any $j = 1, 2, \ldots, \ell - 1$, where $\neg 0 = 1$ and $\neg 1 = 0$, $s_1[-j]$ is sequence s_1 except position j, and t_j is sequence s_1 with $\neg s_1[j]$ in time slot j. It is easy to check that configuration $\{1, 2\}$ is improving for player 1 under any deviation t_j, except for $j \in \{i, i'\}$. But $(s_1, \ldots, s_n)$ is an $(n, 2)$-AE, for each of these $\ell - 3$ deviations t_j of player 1, for $j \in [\ell - 1] \setminus \{i, i'\}$, so there must be a player $p_j \in \{3, 4, \ldots, n\}$ so that configuration $\{1, p_j\}$ is worsening for player 1 under deviation t_j.

We will call these players p_j "blockers", and define them as follows. For any $j \in [\ell - 1] \setminus \{i, i'\}$:

- If $s_1[j] = 0$, then blocker p_j has the same prefix as s_1 until time slot $j - 1$, it has $\neg s_1[j]$ in time slot j, and then follows sequence $s_1[j + 1, \ldots, \ell]$.
- If $s_1[j] = 1$, then blocker p_j has the same prefix as s_1 until time slot $j - 1$, it has $\neg s_1[j]$ in time slot j, and then it is arbitrary on time slots $j + 1, \ldots, \ell$.

It is easy to check that the configuration $\{1, p_j\}$ is worsening for player 1 under deviation t_j for any $j \in [\ell - 1] \setminus \{i, i'\}$. Players p_j have the same prefix as s_1 until time slot $j - 1$ and on time slot j they are flipped to $\neg s_1[j]$, so they all are pairwise distinct players. They are also all distinct from players 1 and 2. Thus, there are at least $\ell - 3 + 2 = \ell - 1$ distinct players, and $\ell - 1 \leq n$, implying $\ell \leq n + 1$. This concludes the proof for $k = 2$.

To show that PoA is unbounded if $k \geq 3$, consider Noisy_GTRR(n), which by Theorem 1 is (n, k)-AE with maximum latency n. Let us add in front of Noisy_GTRR(n), ℓ columns containing only 1's, for any integer $\ell > 0$. We argue that the resulting $n \times (\ell + n)$ matrix, M', is (n, k)-AE. By Theorem 1, no player can profitably deviate in the Noisy_GTRR(n) part of M'. If any player $i \in [n]$ deviates in the first ℓ time slots, changing any of its 1's to 0's or $\not{1}$'s, such deviation has no effect of i's transmission for any chosen other $k - 1 \geq 2$ players in the configuration, because these players block player i in the first ℓ time slots. Thus, matrix M' is an (n, k)-AE with maximum latency $\ell + n$, which is arbitrarily large when $\ell \to \infty$, hence unbounded PoA. $\square$

5 Conclusion and Further Directions

We introduced contention resolution games, with adversarial configurations and deterministic algorithms as agents' strategies. We proposed adversarial equilibrium as a solution concept and efficient algorithms that are such equilibria. There are efficiency gaps for larger k, see Table 1, thus designing more efficient equilibria and/or proving negative results is a natural research direction.

We expect that our methodology of adversarial selection of players in equilibrium applicable to deterministic algorithmic strategies of selfish autonomous agents, could be used to study communication problems beyond transmission of a single packet on a channel. Natural generalizations include shared channels with stronger feedback than acknowledgements, adaptive algorithms, and contention resolution in radio networks with general topologies. Finally, it is intriguing if there are similar, natural definitions of AE for less risk-averse players, which admit efficient deterministic solutions.

Acknowledgements

The work of Dariusz R. Kowalski was partially supported by the NSF grant 2131538. The work of Piotr Krysta has been partially supported by the Network Sciences and Technologies (NeST) initiative of the University of Liverpool.

References

[Abramson, 1970] Norman M. Abramson. The ALOHA system: Another alternative for computer communications. In *Proceedings of the 1970 Fall Joint Computer Conference (AFIPS)*, pages 281–285. ACM, 1970.

[Anantharamu *et al.*, 2019] Lakshmi Anantharamu, Bogdan S. Chlebus, Dariusz R. Kowalski, and Mariusz A. Rokicki. Packet latency of deterministic broadcasting in adversarial multiple access channels. *Journal of Computer and System Sciences*, 99:27–52, 2019.

[Anshelevich *et al.*, 2004] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 295–304, 2004.

[Axelrod and Hamilton, 1981] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

[Bender *et al.*, 2005] Michael A. Bender, Martin Farach-Colton, Simai He, Bradley C. Kuszmaul, and Charles E. Leiserson. Adversarial contention resolution for simple channels. In *Proceedings of the* 17*th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 325–332, 2005.

[Chlebus and Kowalski, 2005] Bogdan S. Chlebus and Dariusz R. Kowalski. Almost optimal explicit selectors. In *Proceedings of the* 15*th International Symposium on Fundamentals of Computation Theory (FCT)*, volume 3623 of *Lecture Notes in Computer Science*, pages 270–280. Springer, 2005.

[Chlebus *et al.*, 2012] Bogdan S. Chlebus, Dariusz R. Kowalski, and Mariusz A. Rokicki. Adversarial queuing on the multiple access channel. *ACM Transactions on Algorithms*, 8(1):5:1–5:31, 2012.

[Chlebus, 2001] Bogdan S. Chlebus. Randomized communication in radio networks. In Panos M. Pardalos, Sanguthevar Rajasekaran, John H. Reif, and Jose D. P. Rolim, editors, *Handbook of Randomized Computing*, volume I, pages 401–456. Kluwer Academic Publishers, 2001.

[Christodoulou *et al.*, 2014] George Christodoulou, Katrina Ligett, and Evangelia Pyrga. Contention resolution under selfishness. *Algorithmica*, 70(4):675–693, 2014.

[Clementi *et al.*, 2001] Andrea E. F. Clementi, Angelo Monti, and Riccardo Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms (SODA), January 7-9, 2001, Washington, DC, USA*, pages 709–718. ACM/SIAM, 2001.

[De Marco and Stachowiak, 2017] Gianluca De Marco and Grzegorz Stachowiak. Asynchronous shared channel. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 391–400. ACM, 2017.

[De Marco *et al.*, 2019] Gianluca De Marco, Dariusz R. Kowalski, and Grzegorz Stachowiak. Deterministic contention resolution on a shared channel. In *Proceedings of the* 39*th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 472–482. IEEE, 2019.

[Fiat *et al.*, 2007] Amos Fiat, Yishay Mansour, and Uri Nadav. Efficient contention resolution protocols for selfish agents. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 179–188. SIAM, 2007.

[Greenberg and Winograd, 1985] Albert G. Greenberg and Shmuel Winograd. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *Journal of the ACM*, 32(3):589–596, 1985.

[Griggs *et al.*, 2023] Jerrold R. Griggs, Thomas Kalinowski, Uwe Leck, Ian T. Roberts, and Michael Schmitz. Sizes of flat maximal antichains of subsets. *CoRR*, abs/2302.07062, 2023.

[Hradovich *et al.*, 2021] Elijah Hradovich, Marek Klonowski, and Dariusz R. Kowalski. New view on adversarial queueing on MAC. *IEEE Communication Letters*, 25(4):1144–1148, 2021.

[Indyk, 2002] Piotr Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proceedings of the* 13*th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 697–704, 2002.

[Kautz and Singleton, 1964] William Kautz and Roy Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964.

[Komlós and Greenberg, 1985] János Komlós and Albert G. Greenberg. An asymptotically fast nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Trans. Inf. Theory*, 31(2):302–306, 1985.

[Koutsoupias and Papadimitriou, 2009] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

[Kowalski, 2005] Dariusz R. Kowalski. On selection problem in radio networks. In *Proceedings of the* 24*th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 158–166. ACM, 2005.

[Maschler *et al.*, 2020] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, second edition, 2020.

[Metcalfe and Boggs, 1976] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, 1976.

[Nisan *et al.*, 2007] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Osborne and Rubinstein, 1994] Martin Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[Porat and Rothschild, 2011] Ely Porat and Amir Rothschild. Explicit nonadaptive combinatorial group testing schemes. *IEEE Transactions on Information Theory*, 57(12):7982–7989, 2011.

[Ta-Shma *et al.*, 2007] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Lossless condensers, unbalanced expanders, and extractors. *Combinatorica*, 27(2):213–240, 2007.