Adaptive Scheduling over a Wireless Channel under Constrained Jamming*

Antonio Fernández Anta[1], Chryssis Georgiou[2], and Elli Zavou[1,3]

[1] Institute IMDEA Networks, Madrid, Spain,
[2] University of Cyprus, Nicosia, Cyprus,
[3] Universidad Carlos III de Madrid, Madrid, Spain

Abstract. We consider a wireless channel between a single pair of stations (sender and receiver) that is being "watched" and disrupted by a malicious, adversarial jammer. The sender's objective is to transmit as much useful data as possible, over the channel, despite the jams that are caused by the adversary. The data is transmitted as the payload of packets, and becomes useless if the packet is jammed. In this work, we develop deterministic scheduling algorithms that decide the lengths of the packets to be sent, in order to maximize the total payload successfully transmitted over period T in the presence of up to f packet jams, *useful payload*.
We first consider the case where all packets must be of the same length and compute the optimal packet length that leads to the best possible useful payload. Then, we consider adaptive algorithms; ones that change the packet length based on the feedback on jammed packets received. We propose an optimal scheduling algorithm that is essentially a recursive algorithm that calculates the length of the next packet to transmit based on the packet errors that have occurred up to that point. We make a thorough non trivial analysis for the algorithm and discuss how our solutions could be used to solve a more general problem than the one we consider.

Keywords: Packet scheduling, Wireless Channel, Unreliable communication, Adversarial jamming

1 Introduction

Motivation and prior work. Transmitting data over wireless media is becoming increasingly popular, especially with the dramatic increase of the use of mobile devices (e.g., smart phones). A major challenge that needs to be addressed is to cope with disruptions of the communication over such media, especially when these disruptions are caused intentionally, e.g., by jamming devices. Several research efforts have been made in addressing this challenge under different assumptions and constraints (e.g.,[1–6, 9–12]).

In a recent work [2], we have initiated the investigation of the following problem: We consider a wireless channel between a single pair of stations (sender and receiver) that is being "watched" and disrupted by a malicious, adversarial jammer. The sender's objective is to transmit as much useful data as possible, over the channel, despite the jams that are caused

* This research was supported in part by Ministerio de Economía y Competitividad grant TEC2014-55713-R, Regional Government of Madrid (CM) grant Cloud4BigData (S2013/ICE-2894, co-funded by FSE & FEDER), and grant FPU12/00505 from MECD.

by the adversary. The data is transmitted as the payload of *packets*, and becomes useless if the packet is jammed. The adversary has complete knowledge of the packet scheduling algorithm and it decides on how to jam the channel dynamically. However, the jamming power of the adversary is constrained by two parameters, ρ and σ, whose values depend on technological aspects. The parameter σ represents the maximum number of "error tokens" available for the adversary to use at any point in time, and ρ represents the rate at which new error tokens become available (one at a time). Each error token models the ability of the adversary to jam one packet. This adversarial model could represent a jamming entity with limited resource of rechargeable energy, e.g., malicious mobile devices or battery-operated military drones. In these cases, σ represents the capacity of the battery (in packets that can be jammed) and ρ the rate at which the battery can be recharged (for instance, with solar cells). To evaluate scheduling algorithms, two efficiency measures are used: the *transmission time*, to completely send a fixed pre-defined amount of data, and the *goodput ratio* (successful transmission rate) achieved to do so, which intuitively are reversely proportional.

Under this model, we first showed in [2] upper and lower bounds on the transmission time and goodput when the sender sends packets of the same length throughout the execution (uniform case); in this case the scheduling policy does not take into account the history of jams. Then, considering the case $\sigma = 1$, we proposed an adaptive scheduling algorithm that changes the packet length based on the feedback on jammed packets received, and showed that it can achieve better goodput and transmission time with respect to the uniform case, for most values of ρ. However, the analysis technique used for the case $\sigma = 1$ turned out not to be easily generalized for cases where $\sigma > 1$.

In order to better understand the above problem and lay the groundwork for obtaining its optimal solutions, in this work we consider a simpler, more "static" version of the problem. In particular, we focus on a specific time interval of length T, and instead of assuming that new error tokens are continuously arriving we assume a fixed number of error tokens f. As before, the sender's objective is to correctly transmit the maximum amount of data, in the form of packets, under the jamming of the adversary. Now, the adversary is constrained only by parameter f, which is the maximum number of errors (packet jams) it can introduce in the corresponding interval T and are available from the very beginning of the interval. Therefore, given T and f as input, we would like to maximize the total *useful payload* transmitted within the interval of interest.[a] (Our modeling assumptions are detailed in Section 2.)

We plan to use the results obtained for this problem to derive solutions of the continuous version of [2], but we believe that this static problem is a fundamental and challenging problem and hence of interest by itself.

Contributions. We provide a comprehensive solution to the abovementioned problem (static version). Specifically:

– We first consider the case where the scheduling algorithm is restricted in sending packets of the same length (uniform case); this could be due to limitations in the communication protocol or the sender's specification (Section 3). Given a period of length T and up to f packet jams, we show that the optimal packet length is $p^* \approx \sqrt{T/f}$ that leads to optimal useful payload $T + f - 2\sqrt{T/f}$. In the case the adversary does not cause any packet error

[a] As we assume that the transmission time of each packet is equal to its length, it follows that T is an absolute upper bound on the useful payload transmitted.

in the interval of time T, we show that the useful payload achieved by uniform packets of length p^* is $T - \sqrt{T/f}$.

- Then, we devise adaptive scheduling algorithms, that is, algorithms that change the packet length based on the feedback on jammed packets received. We start by first considering the case of $f = 1$ (Section 4). We devise algorithm ADP$(T, 1)$ and prove its optimality. We show that the algorithm achieves optimal useful payload of $\frac{i-1}{i}T - \frac{i+1}{2} + \frac{1}{i}$, where i is the integer such that $T \in \left[\frac{(i-1)i}{2} + 1, \frac{i(i+1)}{2} + 1\right)$.

  Algorithm ADP$(T, 1)$ chooses the length p of the first packet to be transmitted as a function of T. If the packet is jammed then it transmits a second packet of length $T - p$ which is now guaranteed not to be jammed. If the first packet goes through, then the algorithm is invoked recursively as ADP$(T - p, 1)$.
- Next, we generalize algorithm ADP$(T, 1)$ into algorithm ADP(T, f) and show that it obtains optimal useful payload for any f (Section 5). Algorithm ADP(T, f) is essentially a recursive algorithm that also begins by choosing length p of the first packet to be transmitted as a function of T (a different function from that of ADP$(T, 1)$). If the packet is jammed, the adversary (unlike in the case of $f = 1$) still has error tokens that it can use. Therefore, instead of sending a packet that spans the rest of the interval, ADP(T, f) makes the recursive call ADP$(T - p, f - 1)$. If the packet is not jammed, then it makes a recursive call to ADP$(T - p, f)$.

  Although the above algorithmic approach is natural, the choice of the length p of the packet to be sent as well as the algorithm's analysis of optimality, are nontrivial.
- Finally, we discuss and compare the version of the problem considered in this work (static) with the one of [2] (continuous) and draw interesting conclusions (Section 6).

Related work. Several studies have investigated the effect of jamming in wireless channels. For example, Thuente et al. [12] studied the effects of different jamming techniques in wireless networks and the trade-off with their energy efficiency. Their study includes from trivial/continuous to periodic and intelligent jamming (taking into consideration the size of packets being transmitted). Pelechrinis et al. [6] present a detailed survey of the Denial of Service attacks in wireless networks. They present the various techniques used to achieve malicious behaviors and describe methodologies for their detection as well as for the network's protection from the jamming attacks. Dolev et al. [4] present a survey of several existing results in adversarial interference environments in the unlicensed bands of the radio spectrum, discussing their vulnerability.

Awerbuch et al. [3] designed a medium access (MAC) protocol for single-hop wireless networks that is robust against adaptive adversarial jamming (the adversary knows the protocol and its history and decides to jam the channel at any time) and requires only limited knowledge about the adversary (an estimate of the number of nodes, n, and an approximation of a time threshold T). One of the differences with our work is that the adversary they consider is allowed to jam $(1 - \varepsilon)$-fraction of the time steps. On a later work [10], Richa et al. explored the design of a robust MAC protocol that takes into consideration the signal to interference plus noise ratio (SINR) at the receiver end. In [11] they extended their work to the case of multiple co-existing networks; they proposed a randomized MAC protocol which guarantees fairness between the different networks and efficient use of the bandwidth. In [9], Richa et al. considered an adaptive adversarial jammer that is also reactive: it is allowed to make a jamming decision based on the actions of the nodes at the current step; this is similar to the adversary we consider in this work. However, they consider a different constrain on jamming, following the previously mentioned works: given a time period of length T, the adversary can jam at most $(1-\varepsilon)T$ of the time steps in that period. In our case, the adversary, within a time period T, can cause f channel jams, where f does not correspond to a fraction of time, but on the number of packets it can corrupt. Other differences is that they consider n nodes transmitting over the channel (hence, they deal with transmission collisions) and their objective is to optimize throughput over the non-jammed time periods.

Finally, Gilbert et al. [5] investigated the impact on the communication delay between two honest nodes that a third malicious, energy-constraint node can have. In particular, the three nodes share a time-slotted single-hop wireless radio channel and the two honest nodes begin with a value to communicate. The malicious node wishes to prevent them from communicating for as long as it can, by broadcasting messages. However, it is allowed to broadcast up to β messages. This is similar to the restriction we impose in our work, by allowing the adversary to cause up to f packet errors. The setting and objectives of the work [5], though, are different. First they show a tight bound on the number of rounds that the malicious node can delay the communication between the two honest entities: $2\beta + \Theta(\log|V|)$ rounds, where V is the set of possible values the two honest nodes may communicate. Then, they study the implication of this bound on more general n-node problems, such as reliable broadcast and leader election.

2 Model

We now present in detail the model considered in this paper.

Network setting. We consider an unreliable wireless channel that connects two end stations: a sender and a receiver. The sender has data to be transmitted to the receiver over a fixed time interval T, which is sent as the payload of the packets scheduled.[b] However, the channel is prone to instantaneous jams, that cause any packet in transit to be corrupted/destroyed. Hence, the sender needs to decide the length of the packets to be sent in the time interval, taking or not into consideration the history of transmissions and jams that already occurred. This is done by using an *online scheduling algorithm* [7, 8]. The objective of the sender is to provide the receiver with as much data as possible over the period of time T, despite the channel jams.

As in [2], each packet sent across the channel consists of a *header* of a fixed predefined size h and a *payload* of length l chosen by the algorithm; the total length of the packet is $p = h + l$. For simplicity we assume that $h = 1$, i.e., $p = l + 1$ (we assume l to be a real number). Recall that the payload corresponds to the useful data sent across the channel. In addition, we assume that the transmission time of each packet is equal to its length; the channel has a constant transmission rate. (Therefore, T is an absolute upper bound on the useful payload transmitted.)

Packet jamming. We assume that the jams occurring in the channel are orchestrated by an omniscient and adaptive adversary. That is, the adversary has complete knowledge of the packet scheduling and transmission algorithm it decides to cause the jams during the course of the computation. However, it has a constrained number of jams it can cause in a given period. Specifically, we consider adversary (T, f)-$\mathcal{A}$, that for a time interval of length T,

[b] We assume that the sender has data with total payload greater than T.

$T \geq 1$, it can cause up to f packet jams. As in [2], for a worst case analysis, we assume that the adversary jams some bit in the header of the packets in order to ensure their destruction. So given T, the adversary defines the *error pattern* E as a set of up to f jamming events on the channel over that period, each given by a time instant in the period. We will sometimes use the special error pattern $E = \emptyset$ that corresponds to the case in which the adversary causes no jamming. For a given T, we assume that f is known to the scheduling algorithm.

Efficiency measures. As in [2], we consider two efficiency measures, *useful payload* and *goodput rate*. The useful payload, is the sum of payloads of the successfully transmitted packets within a time interval of length T, under any f-size error pattern E. The goodput rate, is the corresponding ratio of the useful payload sent during the interval and is of interest mostly for the continuous version of the model presented in [2].

More formally, we denote by $\text{UP}_A(T, f, E)$ the useful payload (payload correctly received) when using scheduling algorithm A in an interval of length T against an adversary of power f that uses error pattern E. For a fixed algorithm A, its useful payload is then simply $\text{UP}_A(T, f) = \min_{E \in \mathcal{E}(f)} \text{UP}_A(T, f, E)$, where $\mathcal{E}(f)$ is the set of all possible error patterns with at most f jams. From this, we define the optimal useful payload as $\text{UP}^*(T, f) = \max_A \text{UP}_A(T, f)$.

The goodput metric is defined similarly, by simply dividing the useful payload by the length of the interval. More precisely, when using scheduling algorithm A in an interval of length T against an adversary of power f that uses error pattern E, the goodput rate is $G_A(T, f, E) = \text{UP}_A(T, f, E)/T$, the goodput of algorithm A is $G_A(T, f) = \text{UP}_A(T, f)/T$, and the optimal goodput is $G^*(T, f) = \text{UP}^*(T, f)/T$.

Feedback mechanism. Following [1, 2], we consider instantaneous feedback. In particular, at the time a packet is successfully received by the receiver, a notification/acknowledgement message is immediately received by the sender. If such a message is not received by the sender, then it considers the packet to be jammed. We assume that the notification packets cannot be jammed by the errors in the channel because of their relatively small size.
Remark: Observe that if $T \leq f$, then the adversary can jam all packets sent in the interval and no useful data will be received. Therefore, from this point onward we focus only in time periods that are initially of length $T > f$.

3 Uniform Packets

We first consider the case in which all the packets scheduled are of the same length. Having to use uniform packets may be a requirement due to limitations in the communication protocol, or the sender's specifications. In this case, the following result gives the uniform packet length that has to be used in order to maximize the minimum useful payload. (Note that the approximations below are due to floors and ceilings; these approximations get closer to equality as Tf grows.)

Theorem 1. *Let $U(p)$ denote the algorithm that only uses uniform packets of length p. In an interval of length T and maximum number of errors f, the optimal packet length for these algorithms is $p^* \approx \sqrt{T/f}$ that achieves useful payload $UP_{U(p^*)}(T, f) \approx T + f - 2\sqrt{Tf}$. When the adversary causes no jamming, the useful payload achieved by $U(p^*)$ is $UP^*_{U(p^*)}(T, f, \emptyset) \approx T - \sqrt{Tf}$.*

Proof. Let us denote by n the number of uniform packets of length $p = \frac{T}{n}$ sent in an interval of length T when the adversary has f error tokens available. Hence, we will be using $U(n)$ and $U(p)$ to denote the same algorithm. In the worst case, the adversary will use its error tokens to jam f packets in the interval, and hence there will be at least $n - f$ successfully received packets by the receiver by the end of the interval. The useful payload of the uniform algorithm using n (and hence p) will thus be $\text{UP}_{U(n)}(T, f) = (n - f)\,(\frac{T}{n} - 1)$ (recall that each packet consists of the payload and a unit-size header).

Deriving this expression with respect to n, we get

$$\frac{\partial \text{UP}_{U(n)}(T, f)}{\partial n} = \frac{fT}{n^2} - 1,$$

which implies that $\text{UP}_{U(n)}(T, f)$ is maximized for $n = \sqrt{Tf}$. Moreover, the derivative is positive for $n < \sqrt{Tf}$ and negative for $n > \sqrt{Tf}$, which implies that the useful payload is strictly increasing on the left of $n = \sqrt{Tf}$ and strictly decreasing on the right. From this, we get that (1) there is no other n that maximizes the useful payload, and (2) since the number of packets has to be an integer value, the only two candidates for the optimal number of packets n^* are $\lfloor \sqrt{Tf} \rfloor$ and $\lceil \sqrt{Tf} \rceil$. Hence the value of these two that maximizes the useful payload is the optimal number n^*. From this, and the fact that $p = \frac{T}{n}$, we get that $p^* \approx \sqrt{T/f}$, as claimed.

Then, the optimal number of packets n^* gives optimal useful payload $\text{UP}_{U(n^*)}(T, f) = (n^* - f)(\frac{T}{n^*} - 1) \approx T + f - 2\sqrt{Tf}$, as claimed. Finally, when n^* packets are used, and no packet is jammed by the adversary, the useful payload is maximized reaching $\text{UP}_{U(n^*)}(T, f, \emptyset) = n^*(\frac{T}{n^*} - 1) \approx T - \sqrt{Tf}$, as desired. □

Corollary 1. *The optimal achievable goodput rate is* $G_{U(p^*)}(T, f) \approx \left(1 - \sqrt{f/T}\right)^2$.

4 Optimal Algorithm for $f = 1$

In this section, we turn our focus on the case of a single error token available to the adversary for an interval of length T. We give an adaptive algorithm, named ADP$(T, 1)$, and prove its optimality. By doing so, we hope to give an intuition to the reader for how the general optimal algorithm, for any number of error tokens, works.

Algorithm ADP$(T, 1)$ is used in a time recursive fashion, with respect to the length of the interval of interest, T. Its scheduling policy is as follows: It chooses the length p of the first packet to be transmitted as a function of T. If the packet is jammed then it transmits a second packet of length $T - p$ which is guaranteed not to be jammed. If the first packet goes through, then the algorithm is invoked recursively as ADP$(T - p, 1)$.

The detailed description of the algorithm is given as Algorithm 1. Let us fix the interval length $T \geq 1$, and let i be the integer such that $T \in \left[\frac{(i-1)i}{2} + 1, \frac{i(i+1)}{2} + 1\right)$, as described in the above pseudocode. Let us also define parameters $\alpha = i - 2$ and $\beta = \frac{(i-1)i}{2} - 1$, packet length $p = \frac{T+\beta}{\alpha+2}$, and interval length $T' = T - p$. We first present the following two lemmas that are used to show the optimality of Algorithm ADP$(T, 1)$.

Algorithm 1 **ADP**$(T, 1)$

If $T \in [1, 2)$ then
  Send packet with length $p = T$
else
  Let i be the integer such that $T \in \left[\frac{(i-1)i}{2} + 1, \frac{i(i+1)}{2} + 1\right)$
  Let $\alpha = i - 2$, and $\beta = \frac{(i-1)i}{2} - 1$
  Send packet π with length $p = \frac{T+\beta}{\alpha+2} = \frac{T-1}{i} + \frac{i-1}{2}$
  If packet π is jammed then
    Send packet with length $p' = T - p$
  else
    Call ADP$(T - p, 1)$

Lemma 1. *Interval length $T' = T - p$ is such that $T' \in \left[\frac{(j-1)j}{2} + 1, \frac{j(j+1)}{2} + 1\right)$ for $j = i - 1$, where i is an integer such that $i \geq 1$.*

Proof. Replacing the values of α and β in the calculation of $T' = T - p$,

$$T' = \frac{(\alpha+1)T - \beta}{\alpha + 2} = \frac{(i-2+1)T - \left(\frac{(i-1)i}{2} - 1\right)}{i - 2 + 2} = \frac{(i-1)T - \frac{(i-1)i}{2} + 1}{i}.$$

Now, using the fact that $T \geq \frac{(i-1)i}{2} + 1$, we have

$$T' \geq \frac{(i-1)\left(1 + \frac{(i-1)i}{2}\right) - \frac{(i-1)i}{2} + 1}{i} = \cdots = \frac{(i-1)(i-2)}{2} + 1.$$

Similarly, using the fact that $T < \frac{i(i+1)}{2} + 1$, we have

$$T' < \frac{(i-1)\left(1 + \frac{i(i+1)}{2}\right) - \frac{(i-1)i}{2} + 1}{i} = \cdots = \frac{(i-1)i}{2} + 1.$$

Setting $j = i - 1$ in both cases, we have $T' \in \left[\frac{(j-1)j}{2} + 1, \frac{j(j+1)}{2} + 1\right)$ as claimed. □

Lemma 2. *Let $T \geq 2$ and assume that UP$_{ADP}(T', 1) = \frac{\alpha T' - \beta}{\alpha+1}$, where $T' = T - p$. Then, Algorithm ADP$(T, 1)$ achieves useful payload UP$_{ADP}(T, 1) = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2}$.*

Proof. Since $T \geq 2$, that Algorithm ADP$(T, 1)$ schedules first a packet π with length $p = \frac{T+\beta}{\alpha+2}$. If π is jammed, then a packet of length equal to the rest of the interval, i.e., $T' = T - p$, can be sent successfully, and hence the useful payload will be UP$_{\text{ADP}}(T, 1) = T - \frac{T+\beta}{\alpha+2} - 1 = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2}$.

Otherwise, if π is not jammed, the useful payload is obtained as UP$_{\text{ADP}}(T, 1) = p - 1 + \text{UP}_{\text{ADP}}(T', 1) = p - 1 + \frac{\alpha T' - \beta}{\alpha+1} = p - 1 + \frac{\alpha(T-p) - \beta}{\alpha+1} = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2}$. In both cases, the useful payload is as claimed, which completes the proof. □

Theorem 2. *Given an interval of length $T \geq 1$, Algorithm ADP$(T, 1)$ achieves optimal useful payload UP*$^*(T, 1) = \frac{i-1}{i}T - \frac{i+1}{2} + \frac{1}{i}$*, where i is the integer such that $T \in \left[\frac{(i-1)i}{2} + 1, \frac{i(i+1)}{2} + 1\right)$.*

Proof. The proof is by induction on T. The base case is when $T \in [1, 2)$, which implies that $i = 1$. In this case only one packet is sent by ADP$(T, 1)$, which spans the whole interval and can be jammed by the adversary. Observe that in this case at most one packet can in fact be sent in the interval. This matches the claim that ADP$(T, 1)$ achieves optimal useful payload UP$^*(T, 1) = 0$ in this case.

Let us now consider any interval length $T \geq 2$, which implies $i \geq 2$. Then, from Lemma 1, interval length $T' = T - p \in \left[\frac{(j-1)j}{2} + 1, \frac{j(j+1)}{2} + 1\right)$ for $j = i - 1$. By induction hypothesis, UP$_{\text{ADP}}(T', 1) = $ UP$^*(T', 1) = \frac{j-1}{j}T - \frac{j+1}{2} + \frac{1}{j} = \frac{\alpha T' - \beta}{\alpha+1}$, and from Lemma 2 we have that UP$_{\text{ADP}}(T, 1) = \frac{(\alpha+1)T-(\beta+\alpha+2)}{\alpha+2} = \frac{i-1}{i}T - \frac{i+1}{2} + \frac{1}{i}$.

To show that the useful payload achieved by ADP is optimal for this case $T \geq 2$, consider an algorithm A that follows one of the following approaches:
(a) First sends a packet π' of length $p' > \frac{T+\beta}{\alpha+2}$. We assume then that the adversary jams π'. The length of the rest of the interval is $T - p' < T - \frac{T+\beta}{\alpha+2}$. Hence the useful payload will be

$$\text{UP}_A(T, 1) < T - \frac{T + \beta}{\alpha + 2} - 1 = \frac{(\alpha + 1)T - (\beta + \alpha + 2)}{\alpha + 2} = \text{UP}_{\text{ADP}}(T, 1).$$

(b) First sends a packet π' of length $p' < \frac{T+\beta}{\alpha+2}$, $p' \geq 1$. Then the adversary does not jam π'. The rest of the interval has length $T - p' = T' + (p - p') > T'$. We consider two cases (from Lemma 1 no other case is possible):
Case (b).1: $T - p' = T' + (p - p') \in \left[\frac{(j-1)j}{2} + 1, \frac{j(j+1)}{2} + 1\right)$ for $j = i - 1$. Then, by induction hypothesis, UP$^*(T' + (p - p'), 1) = \frac{j-1}{j}(T' + (p - p')) - \frac{j+1}{2} + \frac{1}{j} < \frac{j-1}{j}T' - \frac{j+1}{2} + \frac{1}{j} + (p - p') = $ UP$^*(T', 1) + (p - p')$. Hence,

$$\begin{aligned}\text{UP}_A(T, 1) &\leq p' - 1 + \text{UP}^*(T' + (p - p'), 1) < p' - 1 + \text{UP}^*(T', 1) + (p - p') \\ &= p - 1 + \text{UP}^*(T', 1) = \text{UP}_{\text{ADP}}(T, 1).\end{aligned}$$

Case (b).2: $T - p' = T' + (p - p') \in \left[\frac{(i-1)i}{2} + 1, \frac{i(i+1)}{2} + 1\right)$. In this case,

$$\begin{aligned}\text{UP}_A(T, 1) &\leq p' - 1 + \text{UP}^*(T - p', 1) = p' - 1 + \frac{i - 1}{i}(T - p') - \frac{i + 1}{2} + \frac{1}{i} \\ &< \frac{i - 1}{i}T - \frac{i + 1}{2} + \frac{1}{i} = \text{UP}_{\text{ADP}}(T, 1),\end{aligned}$$

where the first equality follows from induction hypothesis, and the second inequality follows from the fact that $p' < i$ (derived from $p' < \frac{T+\beta}{\alpha+2}$, the definition of α and β, and the fact that $T < \frac{i(i+1)}{2} + 1$).

Hence, in none of the two cases, neither (a) nor (b), Algorithm A was able to achieve a higher useful payload than ADP, which implies that the latter achieves optimality. □

5 Optimal Algorithm for ANY $f > 1$

We now turn our focus on the case of any number of error tokens $f > 1$ available to the adversary for an interval of length T. We present the general adaptive algorithm ADP(T, f) for $f > 1$ as Algorithm 2, and prove its optimality in the rest of the section. The pseudocode of ADP(T, f) for $f > 1$ is similar to that of ADP$(T, 1)$, with a couple of differences. First, in this case it is not possible to explicitly give the length p of the first packet π sent (values of α, β, and γ) when $T \geq f + 1$ (see Theorem 3). Second, if π is jammed, the adversary still has some error tokens that it can use. Hence, instead of sending a packet that spans the rest of the interval, ADP(T, f) makes the call ADP$(T - p, f - 1)$, which could be recursive if $f > 2$, or a call to the algorithm ADP$(T - p, 1)$ (see Algorithm 1), if $f = 2$. It will not be surprising then that the proof of optimality of the algorithm ADP(T, f) will use induction on f.

Algorithm 2 ADP(T, f), for $f > 1$

```
If  T < f + 1 then
   Send packet with length  p = T
else
   Send packet π with length  p = (αT+β)/γ          // α, β and γ depend on T; see Theorem 3
   If  packet π is jammed then
      Call  ADP(T − p, f − 1)
   else
      Call  ADP(T − p, f)
```

Let us first prove some observations that hold for any optimal algorithm OPT, to be used later in the analysis of Algorithm ADP(T, f).

Observation 1 *The useful payload of an optimal algorithm OPT, follows a non-decreasing function with respect to the length of the interval of interest, T, when there are $f \geq 0$ available errors, i.e., UP*$^*(T, f) \leq$ *UP*$^*(T + \delta, f)$*, for* $\delta > 0$.

Proof. Let us consider an optimal algorithm OPT that achieves optimal useful payload UP$^*(T, f) = \alpha$, for an interval of length T and f error tokens available within the interval. Now let us construct an algorithm A, that for interval length $T + \delta$ initially uses the exact same approach as OPT for T; choosing the same packet lengths OPT does during the initial T time of the interval. This means that it has at least the same useful payload as OPT for T, i.e., UP$_A(T + \delta, f) \geq \alpha$. Since OPT is the optimal algorithm, it must achieve at least the same useful payload as A for the interval of length $T + \delta$, i.e., UP$^*(T + \delta, f) \geq$ UP$_A(T + \delta, f)$. Hence, UP$^*(T, f) \leq$ UP$^*(T + \delta, f)$ as claimed. □

Observation 2 *The useful payload of an optimal algorithm OPT, follows a non-increasing function with respect to the number of available errors in an interval of length T, i.e., UP*$^*(T, f) \leq$ *UP*$^*(T, f - 1)$*, where* $f \geq 1$.

Proof. Let us consider an optimal algorithm OPT, with a useful payload UP$^*(T, f) = \beta$ for an interval length T with f errors available. Then, let us construct an algorithm A, that for $f - 1$ error tokens during the same interval length T, uses the exact approach as OPT for f errors; choosing the same packet lengths until $f - 1$ error tokens are used by the adversary.

Then, it schedules one packet equal to the size of the remaining interval. This means that it has at least the same useful payload as OPT does for f errors, $\text{UP}_A(T, f-1) \geq \beta$. And since OPT is the optimal algorithm, it must achieve at least the same useful payload for the same interval and $f-1$ errors, i.e., $\text{UP}^*(T, f-1) \geq \text{UP}_A(T, f-1)$. Hence, $\text{UP}^*(T, f) \leq \text{UP}^*(T, f-1)$ as claimed. □

Lemma 3. *There is an optimal algorithm OPT that is work-conserving, i.e., for each T and for each f, there is an optimal work-conserving strategy deciding the packet lengths.*

Proof. Assume by contradiction that there is some combination of interval and number of error tokens (T, f), for which no work-conserving scheduling strategy is optimal. We choose the smallest such T and consider the following:
(1) There is an optimal strategy for this pair of T and f that does not send any packet during the interval. Hence the optimal useful payload is zero, $\text{UP}^*(T, f) = 0$. In this case, sending one packet that spans the whole interval will lead to the same payload.
(2) There is a strategy that waits for Δ time at the beginning of the interval before sending a packet of length p. This packet can be jammed. Therefore,

$$\begin{aligned} \text{UP}^*(T, f) &= \min\{\text{UP}^*(T - \Delta - p, f-1), p - 1 + \text{UP}^*(T - \Delta - p, f)\} \\ &\leq \min\{\text{UP}^*(T - p, f-1), p - 1 + \text{UP}^*(T - p, f)\}. \end{aligned}$$

Where the inequality follows from Observation 1. The right side of the inequality is the useful payload obtained by the strategy that does not wait the Δ period, but instead schedules the packet of length p at the beginning of the interval (which is work-conserving). Since both cases lead to a contradiction, the claim follows. □

Lemma 4. *The optimal useful payload is a continuous function with respect to the length of the interval, T, when there are $f \geq 1$ errors available.*

Proof. Assume by contradiction that the optimal useful payload is not a continuous function. This means that there is an interval length T for which the following holds: $\lim_{\epsilon \to 0} \text{UP}^*(T - \epsilon, f) < \text{UP}^*(T, f)$. Let us fix parameter $\epsilon > 0$, and observe the behavior of a work-conserving optimal algorithm OPT for interval lengths T and $T - \epsilon$ (such an algorithm exists by Lemma 3). Let us then denote by p_O and p_ϵ the lengths of the first packet scheduled by OPT in each case respectively. These packets can be jammed or not. We observe:

$$\text{UP}^*(T - \epsilon, f) = \min\{\text{UP}^*(T - \epsilon - p_\epsilon, f-1), p_\epsilon - 1 + \text{UP}^*(T - \epsilon - p_\epsilon, f)\} \quad (1)$$

$$\text{UP}^*(T, f) = \min\{\text{UP}^*(T - p_O, f-1), p_O - 1 + \text{UP}^*(T - p_O, f)\} \quad (2)$$

However, if we construct an alternative algorithm A that chooses a packet of length $p'' = p_O - \epsilon$ in the case of interval of length $T - \epsilon$, and works as OPT for smaller intervals, then

$$\text{UP}_A(T - \epsilon, f) = \min\{\text{UP}^*(T - p_O, f-1), p_O - \epsilon - 1 + \text{UP}^*(T - p_O, f)\} \geq \text{UP}^*(T, f) - \epsilon.$$

Since $\text{UP}^*(T - \epsilon, f) \geq \text{UP}_A(T - \epsilon, f)$, it is then trivial to conclude that $\lim_{\epsilon \to 0} \text{UP}^*(T - \epsilon, f) = \text{UP}^*(T, f)$, which is a contradiction. Hence the optimal useful payload is a continuous function with respect to the length of the interval, as claimed. □

We will now show how Algorithm ADP(T, f) computes the packet length p of the packet π sent when $T \geq f + 1$. The computation assumes that it is possible to recursively call ADP(T', j) for any $T' < T$ and $j \leq f$, and that the useful payload of each of these recursive calls is the optimal value UP$^*(T', j)$. Then, ADP(T, f) chooses as length of packet π the smallest value $p \in [1, T]$ that satisfies the equality UP$^*(T-p, f-1) = p-1+$UP$^*(T-p, f)$. Table 1 shows the values of p chosen for some interval lengths T when $f = 2$. It also shows the useful payload achieved by the algorithm using these values of p.

T	$[1,3)$	$[3,9/2)$	$[9/2,17/3)$	$[17/3,19/3)$	$[19/3,70/9)$	$[70/9,308/36)$
p	T	$\frac{T}{3}$	$\frac{T+6}{7}$	$\frac{3T+3}{12}$	$\frac{5T+16}{26}$	$\frac{6T+42}{42}$
UP$^*(T,2)$	0	$\frac{T-3}{3}$	$\frac{3T-10}{7}$	$\frac{6T-22}{12}$	$\frac{14T-54}{26}$	$\frac{24T-98}{42}$

Table 1. Values of packet length p and optimal useful payload UP$^*(T, 2)$ achieved with Algorithm ADP$(T, 2)$.

We now prove that the described process to make the choice leads to optimality.

Theorem 3. *Given an interval of length $T \geq f + 1$, Algorithm ADP(T, f) achieves optimal useful payload by choosing the smallest value $p \in [1, T]$ that satisfies the equality*

$$UP^*(T-p, f-1) = p - 1 + UP^*(T-p, f).$$

Moreover, there are constants α_l, β_l, γ_l, α_k, β_k, and γ_k such that $UP^*(T-p, f) = \frac{\alpha_l(T-p)-\beta_l}{\gamma_l}$ and $UP^*(T-p, f-1) = \frac{\alpha_k(T-p)-\beta_k}{\gamma_k}$, and hence

$$p = \frac{(\alpha_k\gamma_l - \gamma_k\alpha_l)T + \gamma_k\gamma_l + \gamma_k\beta_l - \beta_k\gamma_l}{\gamma_k\gamma_l + \alpha_k\gamma_l - \gamma_k\alpha_l}.$$

(Observe that the parameters used in Algorithm 2 are hence $\alpha = \alpha_k\gamma_l - \gamma_k\alpha_l$, $\beta = \gamma_k\gamma_l + \gamma_k\beta_l - \beta_k\gamma_l$, and $\gamma = \gamma_k\gamma_l + \alpha_k\gamma_l - \gamma_k\alpha_l$.) The optimal useful payload obtained is then

$$UP^*(T, f) = \frac{\alpha_k\gamma_l T - (\alpha_k\gamma_l + \alpha_k\beta_l + \beta_k\gamma_l - \beta_k\alpha_l)}{\gamma_k\gamma_l + \alpha_k\gamma_l - \gamma_k\alpha_l}.$$

Proof. We prove by a double induction on the number of error tokens f and the length of the interval T, that the approach followed by Algorithm ADP(T, f) gives the optimal useful payload.

Base Cases. We have as base case of the induction on the number of error tokens the fact that (1) when $f = 0$ the optimal strategy is to send a single packet of length T that spans the whole interval, leading to UP$^*(T, 0) = T - 1$, and (2) that the algorithm ADP$(T, 1)$ presented in Section 4 is optimal for any T, which covers the case $f = 1$.

For a given $f > 1$, we also use induction in the length of the interval T. In this case the base case is when $T < f + 1$, which has optimal payload UP$^*(T, f) = 0$, since the adversary can jam each of the up to f packets that can be sent.

Induction Hypotheses. We first inductively assume that ADP(T, j) is optimal for any number of tokens $j < f$ available to the adversary at the beginning of the interval and any interval length $T > j$. In particular, for any $j < f$ and any $T > j$, there is a known range of lengths $R_{ij} = [a_{ij}, b_{ij})$ such that $T \in R_{ij}$, $b_{ij} = a_{(i+1)j}$ and the optimal useful payload is known to be $\text{UP}^*(T, j) = \frac{\alpha_{ij} T - \beta_{ij}}{\gamma_{ij}}$. Parameters α_{ij}, β_{ij} and γ_{ij} are known positive integers, such that $\beta_{ij} > \gamma_{ij} > \alpha_{ij}$.

We inductively also assume that for f error tokens, there are m known ranges of lengths $R_{if} = [c_{if}, d_{if})$ for $i = 1, 2, \ldots, m$, such that $\bigcup_{i=1}^{m} R_{if} = [1, d_{mf})$, $c_{1f} = 1$, and $d_{if} = c_{(i+1)f}, \forall 1 \leq i < m$. Also, for any interval length T such that $T < d_{mf}$ and $T \in R_{if} = [c_{if}, d_{if})$, the optimal useful payload is known to be $\text{UP}^*(T, f) = \frac{\alpha_{if} T - \beta_{if}}{\gamma_{if}}$. Parameters α_{if}, β_{if} and γ_{if} are known positive integers such that (1) $\beta_{if} > \gamma_{if} > \alpha_{if}$, and for any $R_{\ell f},\ R_{rf}$ where $1 \leq \ell \leq r \leq m$, it holds that (2) $\frac{\beta_{rf}}{\gamma_{rf}} \geq \frac{\beta_{\ell f}}{\gamma_{\ell f}}$ and (3) $\frac{\alpha_{rf}}{\gamma_{rf}} \geq \frac{\alpha_{\ell f}}{\gamma_{\ell f}}$.
Inductive Step. For interval length $T \in [d_{mf}, d_{mf} + 1)$, the algorithm ADP$(T, f)$ chooses the *smallest* packet length $p \in [1, T]$ that satisfies the following condition

$$\text{UP}^*(T - p, f - 1) = p - 1 + \text{UP}^*(T - p, f). \tag{3}$$

Claim. There is at least one packet length $p \in [1, T]$ that satisfies Eq. 3.

Proof of Claim. Observe that, when $p = 1$, from Observation 2 we have that $\text{UP}^*(T - p, f - 1) \geq p - 1 + \text{UP}^*(T - p, f)$. On the other hand, when $p = T$, we have that $\text{UP}^*(T - p, f - 1) = 0 \leq p - 1 + \text{UP}^*(T - p, f) = T - 1$. Hence, taking into consideration the continuity of the useful payload function of both $f - 1$ and f error tokens (Lemma 4) and the Mean Value Theorem, there always exists a packet size $p \in [1, T]$ such that $\text{UP}^*(T - p, f - 1) = p - 1 + \text{UP}^*(T - p, f)$. $\square_{Claim}$

Now, let p be the packet length chosen, and let $T - p \in R_{kj} = [a_{kj}, b_{kj})$ and $T - p \in R_{lf} = [c_{lf}, d_{lf})$. Note that R_{kj} and R_{lf} are among the known ranges from the induction hypothesis. Then, by induction hypothesis $\text{UP}^*(T - p, f) = \frac{\alpha_{lf}(T-p) - \beta_{lf}}{\gamma_{lf}}$ and $\text{UP}^*(T - p, f - 1) = \frac{\alpha_{kj}(T-p) - \beta_{kj}}{\gamma_{kj}}$. Then, solving Eq. 3 for p, the packet length is

$$p = \frac{(\alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf})T + \gamma_{kj}\gamma_{lf} + \gamma_{kj}\beta_{lf} - \beta_{kj}\gamma_{lf}}{\gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}},$$

and the useful payload obtained is

$$\begin{aligned}\text{UP}_{\text{ADP}}(T, f) = \text{UP}^*(T - p, f - 1) = p - 1 + \text{UP}^*(T - p, f) = \frac{\alpha_{kj}(T - p) - \beta_{kj}}{\gamma_{kj}} \\ = \frac{\alpha_{kj}\gamma_{lf}T - (\alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf})}{\gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}},\end{aligned}$$

as claimed. To complete the induction step, we define $\alpha = \alpha_{kj}\gamma_{lf}$, $\beta = \alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf}$ and $\gamma = \gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}$. Then, we show the following three properties (1) $\beta > \gamma > \alpha$, (2) $\frac{\beta}{\gamma} \geq \frac{\beta_{lf}}{\gamma_{lf}}$, and (3) $\frac{\alpha}{\gamma} \geq \frac{\alpha_{lf}}{\gamma_{lf}}$ as follows.

Property 1. For the new parameters $\alpha = \alpha_{kj}\gamma_{lf}$, $\beta = \alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf}$ and $\gamma = \gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}$, it holds that $\beta > \gamma > \alpha$.

Proof of Property 1. First, from the *induction hypotheses*, recall the definition of parameters α_{ij},β_{ij} and γ_{ij}, being known positive integers such that $\beta_{ij} > \gamma_{ij} > \alpha_{ij}$. Looking now at the current parameters α, β and γ individually, we have the following:

(a) $\alpha = \alpha_{kj}\gamma_{lf}$.
(b) $\beta = \alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf} = \alpha_{kj}(\gamma_{lf} + \beta_{lf}) + \beta_{kj}(\gamma_{lf} - \alpha_{lf})$.
(c) $\gamma = \gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf} = \gamma_{kj}(\gamma_{lf} - \alpha_{lf}) + \alpha_{kj}\gamma_{lf}$.

Observe that $\gamma_{kj}(\gamma_{lf} - \alpha_{lf}) + \alpha_{kj}\gamma_{lf} > \alpha_{kj}\gamma_{lf}$, since $\gamma_{kj} > 0$ and $\gamma_{lf} - \alpha_{lf} > 0$ by induction hypothesis. Hence, from (a) and (c) $\gamma > \alpha$. Also, $\alpha_{kj}(\gamma_{lf}+\beta_{lf})+\beta_{kj}(\gamma_{lf}-\alpha_{lf}) > \gamma_{kj}(\gamma_{lf} - \alpha_{lf}) + \alpha_{kj}\gamma_{lf}$, since by induction hypothesis $\beta_{kj} > \gamma_{kj}$, $\gamma_{lf} - \alpha_{lf} > 0$, and all parameters are positive. Hence, from (b) and (c) $\beta > \gamma$ holds as well. This completes the proof of the claim. $\square_{Property\ 1}$

Property 2. For the new parameters $\beta = \alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf}$ and $\gamma = \gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}$, it holds that $\frac{\beta}{\gamma} > \frac{\beta_{lf}}{\gamma_{lf}}$.

Proof of Property 2. For this proof observe first, that since $\beta > \gamma$ (as shown in Property 1), we can safely use the fact that $\frac{\beta}{\gamma} > \frac{\beta-c}{\gamma-c}$, where c is positive. Also by induction hypothesis we have that $\gamma_{lf} - \alpha_{lf} > 0$ and $\beta_{kj} - \gamma_{kj} > 0$. We therefore use some fraction inequality properties as follows:

$$\frac{\beta}{\gamma} = \frac{\alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf} + \beta_{kj}\gamma_{lf} - \beta_{kj}\alpha_{lf}}{\gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}} = \frac{\alpha_{kj}(\gamma_{lf} + \beta_{lf}) + \beta_{kj}(\gamma_{lf} - \alpha_{lf})}{\gamma_{kj}(\gamma_{lf} - \alpha_{lf}) + \alpha_{kj}\gamma_{lf}}$$
$$> \frac{\alpha_{kj}(\gamma_{lf} + \beta_{lf}) + (\beta_{kj} - \gamma_{kj})(\gamma_{lf} - \alpha_{lf})}{\alpha_{kj}\gamma_{lf}} > \frac{\alpha_{kj}\gamma_{lf} + \alpha_{kj}\beta_{lf}}{\alpha_{kj}\gamma_{lf}} = 1 + \frac{\beta_{lf}}{\gamma_{lf}} > \frac{\beta_{lf}}{\gamma_{lf}},$$

which completes the proof. $\square_{Property\ 2}$

Property 3. For the new parameters $\alpha = \alpha_{kj}\gamma_{lf}$ and $\gamma = \gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}$, it holds that $\frac{\alpha}{\gamma} > \frac{\alpha_{lf}}{\gamma_{lf}}$.

Proof of Property 3. For this proof observe first, that since $\gamma > \alpha$ (as shown in Property 1), we can safely use the fact that $\frac{\alpha}{\gamma} > \frac{\beta+c}{\gamma+c}$, where c is positive. Also by induction hypothesis we have that $\gamma_{lf} - \alpha_{lf} > 0$. We therefore use some fraction inequality properties as follows:

$$\frac{\alpha}{\gamma} = \frac{\alpha_{kj}\gamma_{lf}}{\gamma_{kj}\gamma_{lf} + \alpha_{kj}\gamma_{lf} - \gamma_{kj}\alpha_{lf}} = \frac{\alpha_{kj}\gamma_{lf} + \gamma_{kj}\alpha_{lf}}{\alpha_{kj}\gamma_{lf} + \gamma_{kj}\gamma_{lf}}$$
$$= \frac{\alpha_{kj}\alpha_{lf} + \alpha_{kj}(\gamma_{lf} - \alpha_{lf}) + \gamma_{kj}\alpha_{lf}}{\gamma_{lf}(\alpha_{kj} + \gamma_{kj})} = \frac{\alpha_{lf}(\alpha_{kj} + \gamma_{kj})}{\gamma_{lf}(\alpha_{kj} + \gamma_{kj})} + \frac{\alpha_{kj}(\gamma_{lf} - \alpha_{lf})}{\gamma_{lf}(\alpha_{kj} + \gamma_{kj})} > \frac{\alpha_{lf}}{\gamma_{lf}},$$

which completes the proof. $\square_{Property\ 3}$

Observe that the above proof holds for all Ts in the interval $[d_{mf}, d_{mf}+1)$; for each one of these, the algorithm would compute the smaller p that satisfies Eq. 3 and the computation of the parameters α, β, γ is done analogously. Therefore, the known ranges of lengths are extended in this interval.

We must now show that the useful payload is in fact optimal. Let us assume by contradiction that an algorithm A is able to achieve a larger useful payload for the pair (T, f) by sending first a different packet length $p' \neq p$. We consider the following cases.
(a) Algorithm A chooses a packet π' of length $p' > p$. Then, we assume that the adversary will jam the packet π'. Hence, the useful payload achieved by A will be upper bounded as $\text{UP}_A(T, f) \leq \text{UP}^*(T - p', f - 1)$ which by Observation 1 is smaller than

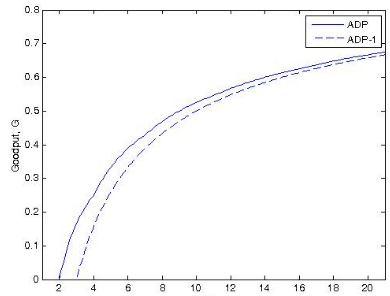

Fig. 1. The goodput rate of algorithms ADP-1 [2] and ADP$(T, 1)$ (Section 4) for $T = 1 \ldots 22$

UP$^*(T-p, f-1) = \text{UP}_{\text{ADP}}(T, f)$, since $T - p' < T - p$.
(b) Algorithm A chooses a packet π' of length $p' < p$. Observe that p' does not satisfy Eq. 3, since p is the smallest length that does. Then the adversary does not jam π'. Then, UP$_A(T, f) \leq p' - 1 + \text{UP}^*(T - p', f)$. We show now that this value is no larger than $p - 1 + \text{UP}^*(T-p, f) = \text{UP}_{\text{ADP}}(T, f)$. Let us assume that $T - p' \in R_{rf}$, where $r \geq l$. Then, UP$^*(T-p', f) = \frac{\alpha_{rf}(T-p')-\beta_{rf}}{\gamma_{rf}} \leq \frac{\alpha_{rf}}{\gamma_{rf}}(T-p') - \frac{\beta_{lf}}{\gamma_{lf}}$, since $\frac{\beta_{rf}}{\gamma_{rf}} \geq \frac{\beta_{lf}}{\gamma_{lf}}$ as shown by Property 2. Similarly, UP$^*(T-p, f) = \frac{\alpha_{lf}(T-p)-\beta_{lf}}{\gamma_{lf}} \geq \frac{\alpha_{rf}}{\gamma_{rf}}(T-p) - \frac{\beta_{lf}}{\gamma_{lf}}$, since $\frac{\alpha_{rf}}{\gamma_{rf}} \geq \frac{\alpha_{lf}}{\gamma_{lf}}$ as shown by Property 3. Finally, combining these bounds and the fact that $\frac{\alpha_{rf}}{\gamma_{rf}} < 1$ (see Property 1), we get that

$$\begin{aligned}\text{UP}_A(T, f) &\leq p' - 1 + \text{UP}^*(T-p', f) \leq p' - 1 + \frac{\alpha_{rf}}{\gamma_{rf}}(T-p') - \frac{\beta_{lf}}{\gamma_{lf}} \\ &\leq p' - 1 + \frac{\alpha_{rf}}{\gamma_{rf}}(T-p') - \frac{\beta_{lf}}{\gamma_{lf}} + (p-p') - \frac{\alpha_{rf}}{\gamma_{rf}}(p-p') \\ &= p - 1 + \frac{\alpha_{rf}}{\gamma_{rf}}(T-p) - \frac{\beta_{lf}}{\gamma_{lf}} \leq \text{UP}_{\text{ADP}}(T, f).\end{aligned}$$

In all cases the resulting useful payload is smaller than the one achieved by choosing the smallest packet size p such that UP$^*(T-p, f-1) = p - 1 + \text{UP}^*(T-p, f)$. Hence the packet size calculated by ADP(T, f) is optimal. □

6 Discussion

Recall that the problem we considered up to this point in the paper is a "static" version of the problem we considered in [2] (continuous version). In this section we discuss the use of our proposed algorithms when applied to the continuous version of the problem. (Recall from Section 1 the definitions of ρ and σ.)

We begin with the following observation: If we divide the time interval of the continuous version of the problem into successive intervals of length $1/\rho$, and σ error tokens are available at the beginning of each interval, then each of these intervals can be considered an instance of the static version of the problem, where $T = 1/\rho$ and $f = \sigma$.

Therefore, by running algorithm ADP$(1/\rho, \sigma)$ in each of these intervals we obtain a solution to the continuous version of the problem. However, this solution is possibly not the best possible, as we make the pessimistic assumption that at the beginning of each interval, the adversary has all σ error tokens available to use; this is true for the first interval, but in successive intervals this might not be the case (with the exception of the case $\sigma = 1$, which we discuss below). Based on the model defined in [2], a new error token will be arriving at the beginning of each interval. If there are already σ tokens, then a token is lost (σ represents, for example, the capacity of the battery of a jamming device – this cannot be exceeded). If in this interval, the adversary performs, say, three packet jams, then at the beginning of the next interval it will have $\sigma - 2$ available tokens. If the scheduling algorithm keeps track of this, then in this interval it should use ADP$(1/\rho, \sigma - 2)$ instead of ADP$(1/\rho, \sigma)$. So, in order to produce more efficient solutions, the scheduling algorithm needs to keep track (using the feedback mechanism) how many jams took place in the previous interval, and using its knowledge of $1/\rho$, run the appropriate version of ADP(). Although there are other subtle issues that also need to be considered, the proposed approach can be used as the basis for obtaining an optimal solution to the continuous version of the problem. We plan to pursue this direction in future research.

Regarding the case of $f = \sigma = 1$, as demonstrated in Fig. 1, algorithm ADP$(1/\rho, 1)$ obtains better results than the solution developed in [2] (called Algorithm ADP-1). In [2], for $\sigma = 1$ it was shown that the goodput rate of Algorithm ADP-1 is $1 - \frac{\rho}{2}\left(1 + \sqrt{1 + \frac{8}{\rho}}\right)$. Figure 1 depicts this goodput rate and the goodput rate of algorithm ADP$(1/\rho, 1)$ as obtained from our analysis in Section 4, for $T = 1 \ldots 22$. Since in the case of $\sigma = 1$ it is best for the adversary to use the error token (otherwise it will lose it), our improved goodput demonstrates the promise of the abovementioned approach.

References

1. A. Fernández Anta, C. Georgiou, D. R Kowalski, J. Widmer, and E. Zavou. Measuring the impact of adversarial errors on packet scheduling strategies. In *Proc. of SIROCCO*, pages 261–273, 2013.
2. A. Fernández Anta, C. Georgiou, and E. Zavou. Packet scheduling over a wireless channel: AQT-based constrained jamming. In *Proc. of NETYS*, 2015.
3. B. Awerbuch, A. Richa, and C. Scheideler. A jamming-resistant mac protocol for single-hop wireless networks. In *Proc. of PODC*, pages 45–54, 2008.
4. S. Dolev, S. Gilbert, R. Guerraoui, D. Kowalski, C. Newport, F. Kohn, and N. Lynch. Reliable distributed computing on unreliable radio channels. In *Proc. of MobiHoc S^3 workshop*, pages 1–4, 2009.
5. S. Gilbert, R. Guerraoui, and C. Newport. Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks. *Theoretical Computer Science*, 410(6):546–569, 2009.
6. K. Pelechrinis, M. Iliofotou, and S. V Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys & Tutorials, IEEE*, 13(2):245–257, 2011.
7. K. Pruhs. Competitive online scheduling for server systems. *ACM SIGMETRICS Performance Evaluation Review*, 34(4):52–58, 2007.
8. K. Pruhs, J. Sgall, and E. Torng. Online scheduling. *Handbook of scheduling: algorithms, models, and performance analysis*, pages 15–1, 2004.
9. A. Richa, C. Scheideler, S. Schmid, and Jin Zhang. Competitive and fair medium access despite reactive jamming. In *Proc. of ICDCS*, pages 507–516, 2011.
10. A. Richa, C. Scheideler, S. Schmid, and Jin Zhang. Towards jamming-resistant and competitive medium access in the sinr model. In *Proc. of the 3rd ACM workshop on Wireless of the students, by the students, for the students*, pages 33–36, 2011.
11. A. Richa, C. Scheideler, S. Schmid, and Jin Zhang. Competitive and fair throughput for co-existing networks under adversarial interference. In *Proc. of PODC*, pages 291–300, 2012.
12. D. Thuente and M. Acharya. Intelligent jamming in wireless networks with applications to 802.11 b and other networks. In *Proc. of MILCOM*, volume 6, 2006.