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. However, the jammer's power is constrained by parameters ρ and σ, that represent the rate at which the adversary can jam a packet, and the length of the largest bursts of jams, respectively. This, corresponds to the translation of the Adversarial Queueing Theory (AQT) constrains, to channel jamming.
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. In this work, we focus on a simplified version of the problem, where given a transmission period T, the adversary is constrained in jamming up to f packets. We believe it to be a *building block* for solving the general problem of AQT-based jamming.
We therefore 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. Then, we consider adaptive algorithms; ones that change the packet length based on the feedback on jammed packets, and analyze them with respect to the simplified model, showing their optimality. Finally, we discuss how our solutions could be used to solve the more general problem.

1 Introduction

Motivation and Model. 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 they 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, 9–11]).

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 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.

* Thesis Director: Antonio Fernández Anta. Departamento de Ingeniería Telemática de la UC3M. Partially supported by FPU Grant from MECD. Contact at: elli.zavou@imdea.org

However, its power is constrained by two parameters, ρ and σ, whose values depend on technological aspects. Parameter σ represents the maximum number of "error tokens" available for the adversary to use at any point in time, and ρ the rate at which new error tokens become available (one at a time). Each error token represents 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, we use two efficiency measures: 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 $s > 1$. Devising an optimal solution for the overall problem seems to be challenging.

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 consider the same network setting, with a sender sending data to a receiver over an unreliable wireless channel, but this time, we focus on a fixed time interval T. Using an *online scheduling algorithm* [7, 8], the sender needs to decide the length of the packets to be sent, taking or not into consideration the history of transmissions and jams that already occurred.

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 and without loss of generality we assume that $h = 1$, i.e., $p = l + 1$ (due to this normalization, l is not necessarily an integer). In addition, we assume that the transmission time of each packet is equal to its length, that is, the channel has a constant transmission rate.

For the jams occurring in the channel, we consider an omniscient and adaptive adversary (has complete knowledge of the scheduling algorithm) with a fixed number of error tokens f that can be used in the period of length T. As in [2], we assume that the adversary jams some bit in the header of the packets in order to ensure their destruction. Finally, we assume that the power of the adversary (parameter f for a given interval length T) is known to the algorithm.

We consider two efficiency measures, the *useful payload*; the sum of payloads of the successfully transmitted packets within T, and the *goodput ratio* as defined in [2]. More formally, for a fixed algorithm A, its useful payload is denoted by $\mathrm{UP}_A(T, f) = \min_{E \in \mathcal{E}(f)} \mathrm{UP}_A(T, f, E)$, 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

$\text{UP}^*(T,f) = \max_A \text{UP}_A(T,f)$. Similary, by simply dividing the useful payload by the length of the interval, we denote by $G_A(T,f) = \text{UP}_A(T,f)/T$ the goodput of algorithm A and by $G^*(T,f) = \text{UP}^*(T,f)/T$ the optimal goodput.

Following [1, 2], we consider instantaneous feedback; 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.

Observe that if $T \leq f$, then the adversary can jam all packets sent in the interval and no useful data will be received. Hence, we focus only in time periods that are initially of length $T > f$.

Related Work. Several studies have investigated the effect of jamming in wireless channels. For example, Thuente et al. [11] 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 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 [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: 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. Another difference 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 though, are different. First they show a tight bound on the num-

ber of rounds that the malicious node can delay the communication: $2\beta + \Theta(\log |V|)$ rounds, where V is the set of possible values the two honest nodes may communicate, and then, they study the implication of this bound on more general n-node problems, such as reliable broadcast and leader election.

Our Contributions. We strongly believe that the static version of the problem as described above is a fundamental and challenging problem. We plan to use its results in order to derive optimal solutions of the continuous version, which we discuss at the end of the paper. The next sections are therefore organized as follows. In Section 2 we present one of the main results analyzed in our prior work [2] regarding the case where the scheduling algorithm is restricted in sending packets of the same length (uniform); this could be due to limitations in the communication protocol or the sender's specification. More specifically, we show bounds on the goodput and compute the quasi optimal packet size p^* that, makes goodput optimal $G^* = (1 - \sqrt{\rho})^2$. Then, in Section **??** we focus of adaptive scheduling algorithms; ones that change the packet length based on the feedback on the jammed packets so far, and show that they can be optimal for the static version of the problem (maximum f error tokens in a period of length T). We start by considering the case of $f = 1$ and show how the proposed algorithm can be generalized for any f. Finally, in Section 4 we discuss and compare the two versions of the problem considered, static (in this work) and continuous [2] and draw interesting conclusions.

2 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 goodput in the continuous model, thoroughly studied in [2].

Theorem 1. *Having a fixed amount of data to be transmitted, P, error token availability rate ρ and error token capacity σ, using packets with payload $l^* = \frac{\sqrt{P(P\rho+(\sigma-1)(1-\rho))}-P\rho}{P\rho+\sigma-1}$, the goodput is upper bounded us $G \leq \frac{Pl^*(1-\rho(l^*+1))}{(P+(\sigma-1)l^*)(l^*+1)}$ and as P grows to infinity, it converges to the optimal $G^* = \lim_{P\to\infty} G = \left(1 - \sqrt{\rho}\right)^2$, which does not depend on σ.*

3 Adaptive Optimal Algorithms

In this Section, we turn our focus on the static model of maximum number of errors f in an interval of length T. We look at two adaptive algorithms, ADP$(T, 1)$ and ADP(T, f), that tackle the cases of one error token available to the adversary during the interval, and any number of error tokens f, respectively. We show that in both cases, they give optimal useful payload and hence goodput ratio for the static version of the problem. Note that, although the algorithmic approach they follow is natural, the choice of the length p of the packet to be sent as well as the algorithms' analysis of optimality, are nontrivial.

Algorithm ADP$(T, 1)$ description. This algorithm is used in a time recursive fashion, with respect to the length of the interval of interest, T. In particular, 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 now guaranteed not to be jammed. If the first packet goes through, then the algorithm is invoked recursively as ADP$(T - p, 1)$.

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$(T - p, 1)$

We fix the interval length $T \geq 1$ and 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 pseudocode. We 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 $T' = T - p$. Then, we prove the following Theorem for interval length T.

Theorem 2. *Algorithm ADP*$(T, 1)$ *achieves optimal useful payload* $UP^*(T, 1) = \frac{i-1}{i}T - \frac{i+1}{2} + \frac{1}{i}$.

Algorithm ADP(T, f) *description.* This algorithm is also used in a recursive fashion, with respect not only to the length of the interval of interest, T, but also to the number of error tokens available at the beginning of T, f. In particular, it chooses the length of p of the first packet to be transmitted depending on T. If the packet is jammed, there are still some errors available to the adversary; hence, instead of sending a packet that spans the rest of the interval, it makes a call to ADP$(T - p, f - 1)$. Otherwise, it calls ADP$(T - p, f)$. Both cases are recursive.

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 $p = \frac{\alpha T+\beta}{\gamma}$ // 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$(T - p, f)$

Then, for the proof of optimality, we use induction on both T and f and obtain the following result.

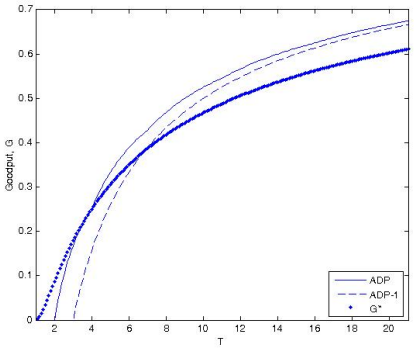

Fig. 1. The goodput rate of algorithms ADP-1 [2] and ADP$(T, 1)$ (Section 3), as well as the optimal goodput rate for the uniform packet approach, $G^* = (1 - \sqrt{\rho})^2$ (Section 2) for $T = 1 \ldots 22$

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 $\gamma = \gamma_k\gamma_l + \alpha_k\gamma_l - \gamma_k\alpha_l$.) Then the optimal useful payload obtained is

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

4 Discussion

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.)

Observe first, that 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 Figure 1 demonstrates, by using algorithm ADP$(1/\rho, 1)$ we obtain 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 3, for $T = 1 \ldots 22$. It also shows that both algorithms exceed the optimal goodput for the uniform packet case, as presented in Section 2 and [2], $G^* = (1 - \sqrt{\rho})^2$, each for different values of ρ. More precisely, the goodput achieved by Algorithm ADP(T, f) exceeds G^* as soon as $1/\rho \geq 4$. 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. Springer, 2013.
2. A.Fernández Anta, C. Georgiou, and E. Zavou. Packet scheduling over a wireless channel: AQT-based constrained jamming. In *Proc. of NETYS, accepted*, 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, ACM, 2008.
4. S. Dolev, S. Gilbert, R. Guerraoui, D.R Kowalski, C. Newport, F. Kohn, and N. Lynch. Reliable distributed computing on unreliable radio channels. In *Proc. of the 2009 MobiHoc S 3 workshop on MobiHoc S 3*, pages 1–4. ACM, 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 J. 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 J. 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. ACM, 2011.
11. 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.