Adaptive Packet Scheduling over a Wireless Channel under Constrained Jamming

Antonio Fernández Anta[a], Chryssis Georgiou[b], Dariusz R. Kowalski[c], Elli Zavou[a,d,*]

[a] *IMDEA Networks Institute, Avda. del Mar Mediterráneo 22, 28918, Madrid, Spain*
[b] *University of Cyprus, 1678 Nicosia, Cyprus*
[c] *University of Liverpool, Liverpool, United Kingdom*
[d] *Universidad Carlos III de Madrid, 28911 Madrid, Spain*

Abstract

In this work we consider the communication over a wireless link, between a sender and a receiver, being disrupted by a jammer. The objective of the sender is to transmit as much data as possible to the receiver in the most efficient way. The data is sent as the payload of packets, and becomes useless if the packet is jammed. We consider a jammer with constrained power, defined by parameters ρ and σ, which represent the rate at which the adversary may jam the channel, and the length of the largest burst of jams it can cause, respectively. This definition translates to the Adversarial Queuing Theory (AQT) constraints, typically used for packet arrivals.

We propose deterministic algorithms that decide the length of the packets sent in order to maximize the goodput rate; i.e., the amount of useful payload successfully transmitted over time. To do so, we first define and study a static version of the problem, which is used as a building block for the dynamic problem. We start by assuming packets of the same length and characterizing the corresponding quasi-optimal length. Then, we show that by adapting the length of the packets, the goodput rate can be improved. Hence, we develop optimal adaptive algorithms that choose the packet lengths depending on the jams that have occurred up to that point in time, in order to maximize the total payload transmitted successfully over a period T in the presence of up to f jams.

Keywords: Packet scheduling, Online algorithms, Wireless Channel, Unreliable communication, Adversarial jamming, Adversarial Queueing Theory

Acknowledgments. This work has been partially supported by the Regional Government of Madrid (CM) grant Cloud4BigData (S2013/ICE-2894, cofunded by FSE & FEDER), the FPU12/00505 grant from the Spanish Ministry of Education, Culture and Sports (MECD), and the Polish National Science Centre grant DEC-2012/06/M/ST6/00459.

*Corresponding author
Email address: elli.zavou@imdea.org (Elli Zavou)

1. Introduction

Motivation. Transmitting data over wireless media in a fast and reliable way, has been attracting a lot of attention from the research community for quite some time now [3, 7, 11, 12, 15, 20, 24, 25, 28, 29, 30], and continues to increase its popularity, especially due to the increment of usage of mobile devices (e.g., smart phones, tablets). One of the many challenges of wireless communication, depending on the specific model and applications, is to cope with disruptions, especially when they are caused intentionally, e.g., by malicious jamming devices. Some of the research efforts already done in addressing this challenge, have looked in different assumptions and constraints (e.g.,[4, 5, 6, 13, 16, 20, 23, 24, 25, 28]) and will be further discussed in the Related Work part of this section.

In our work we look at a wireless communication over a single channel between a sender and a receiver, being "watched" and disrupted by a malicious, adversarial jammer. The sender's goal is to fully transmit over the channel as much data possible in the most efficient way, despite the jams. More precisely, the sender has a potentially unbounded amount of data to be transmitted. Each packet sent contains a *header* of fixed size h and some *payload* whose size, l, depends on the scheduling algorithm used. Note that this payload counts towards the total size of the actual data to be transmitted. For simplicity and without loss of generality we assume that $h = 1$. We also consider *constant bit rate* for the channel (and hence constant bandwidth), which means that the transmission time of each packet is proportional to its size (in particular, a packet of size $l + 1$ takes $l + 1$ time units to be transmitted in full). What is more, when a packet is jammed, it needs to be retransmitted; hence we assume a feedback mechanism that informs the sender when a jam occurs. Our objective is to define optimal scheduling algorithms that decide the length of the packets to be sent, in particular their payload, so that they maximize the amount of data transmitted in time.

We assume that 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. Parameter ρ represents the rate at which the adversary can jam the channel and σ the largest size of a burst of jams that can be caused. More precisely, 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 [1, 2] or battery-operated military drones [14, 18]. 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). We call this model *dynamic*, due to the unpredictability and dynamic nature of the adversary and the channel jams.

To evaluate the scheduling algorithms considered, we use the *goodput rate* as our efficiency measure; successful transmission rate achieved. Under this model, we first show upper and lower bounds on the transmission time and goodput rate when the sender sends packets of the same length throughout the execution (uniform case), not taking into account the history of jams. The interesting question then is whether this bound can be surpassed by adapting the packet length depending on the channel jams. Considering first the case of $\sigma = 1$, we propose an adaptive scheduling algorithm that changes the packet length based on the feedback on jammed packets, and show 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 $\sigma > 1$. Devising an optimal solution for the overall problem seems to be a challenging task.

In order to better understand the above problem and lay the groundwork for obtaining its optimal solutions, we also consider a simpler version of the problem, for which we define a corresponding model, called *static*. 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. The sender's objective now is to correctly transmit the maximum amount of data, considering the jamming power of the adversary. The adversary is constrained only by parameter f; the maximum number of errors (packet jams) it can introduce in the corresponding interval T (all tokens are available from the very beginning of the interval). Hence, given T and f, we want to maximize the total *useful payload* transmitted within the interval of interest.[1]

[1]Since 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.

We then use the static model as a building block for the solution of the dynamic one. More details on the two models and our assumptions are detailed in Section 2.

In a previous work [4], we studied the impact of adversarial errors on packet scheduling, focusing on the long term competitive ratio of throughput, named *relative throughput*. We explored the effect of feedback delay and proposed algorithms that achieve close to optimal relative throughput under worst-case errors, and adversarial or stochastic packet arrivals. Part of the motivation to this work, was the question whether the upper bound of the relative throughput could be exceeded when the power of the adversary is constraint, one of the main differences with this work. Another difference is that in the current work the packet sizes are chosen by the sender, whereas in the previous one they were given. And last but not least, in [4], jammed packets were not retransmitted; the objective was to route packets as fast as possible and not strive to have each packet transmitted. In the current work, the choice of the packet size is precisely the most critical part from the side of the sender. Thus, we focus in devising scheduling algorithms for the decision of packet length to be used and conduct worst-case analysis for the efficiency measures.

Contributions. In this work, we first introduce our *dynamic*, AQT-based adversarial jamming model in wireless networks. AQT has been widely used for restricting packet arrivals in similar settings (see related work below). However, not much research has been done that considers the possibility of exploring its effects in the intent to "damage" a network. We compare our model with the few that have considered similar approaches in the related work below. As already mentioned, our approach of constrained adversarial jamming could be used to model battery-operated malicious devices that have bounded battery capacity and specific recharging rate. In Section 2, we formalize the constrained adversarial jamming model we consider, which we call *dynamic*, as well as the *static* version of the model (focusing on their differences), that is used as a building block to the main optimization problem in the dynamic model. To do that, we propose our approach in Section 2.3, explaining how the goodput rate of the optimal algorithm in the static model will be the goodput rate for the algorithm described for the dynamic model.

We then present the limitations these models impose on the efficiency of scheduling policies, focusing on the *goodput rate* as our main performance measure. More precisely, we start by studying the static model in Section 3, where we consider the case when an algorithm, S-UNI, is restricted in sending packets of uniform length (this could be due to limitations in the communication protocol or the sender's specification). We compute the *quasi optimal*[2] packet size p^* and show that the achievable goodput rate becomes $G_{(T,f)}(\text{S-UNI}_{p^*}) \approx (1 - \sqrt{f/T})^2$.

Next, we devise adaptive scheduling algorithms; ones that change the packet length based on the feedback on jammed packets received, in order to see whether this goodput rate can be exceeded. We start by first considering the case of $f = 1$ (Section 4). We present adaptive algorithm S-DEC, and show that it achieves a greater goodput for $T > \frac{2}{7-3\sqrt{5}} \approx 6.8541$. We continue by devising a new algorithm, S-OPT$(T, 1)$ and prove its optimality. More precisely, 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 S-OPT$(T, 1)$ 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 S-OPT$(T - p, 1)$.

Then, we generalize algorithm S-OPT$(T, 1)$ into algorithm S-OPT(T, f) and show that it obtains static-optimal useful payload for any f (Section 5). Algorithm S-OPT(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 S-OPT$(T, 1)$). 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, S-OPT(T, f) makes the recursive call S-OPT$(T-p, f-1)$. If the packet is not jammed, then it makes a recursive call to S-OPT$(T-p, f)$. Although the above algorithmic approach is quite natural, the choice of the length p of the packet to be sent as well as the algorithm's analysis of optimality, are nontrivial.

In Section 6 we analyze the uniform packet scheduling for the dynamic model, showing how it is related to the static one. We conclude in Section 7, where we summarize our results and discuss the general algorithmic approach proposed in subsection 2.3 (using the optimal algorithm of the static model to solve the scheduling problem in the dynamic model). Finally we comment on some open questions and future work. However, emphasizing on our results,

[2]We use the term "quasi optimal" because our analysis returns a packet length that is a real number, and the optimal length has to be an integer.

we show that giving guarantees in this setting, even when dealing with the simplest scenarios, is quite complex.

Related work. Several studies have investigated the effect of jamming in wireless channels and throughput maximization. Two exhaustive surveys we recommend the reader to see include the work of Pelechrinis et al. [20] where they present a detailed survey of the Denial of Service attacks. They explain 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. The second one [13], is the work of Dolev et al., where they present several existing results in adversarial interference environments in the unlicensed bands of the radio spectrum, discussing their vulnerability. Let us also present some examples of more specific works done in the are. Gummandi et al. [17] consider 802.11 networks disrupted by radio frequency interference and show that they are surprisingly vulnerable. In order to cope with these vulnerabilities they propose and analyze a channel hopping design. Tsibonis et al. [29] studied the scenario of scheduling transmissions to multiple users over a wireless channel with time-varying connectivity. Assuming saturated packet queues, they then proposed an algorithm based on the weighted sum of the throughput of the channel. Thuente et al. [28] 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).

On a different flavor, Awerbuch et al. [6] 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 [24], Richa et al. explored the design of a robust MAC protocol that takes into consideration the signal to interference plus the noise ratio (SINR) at the receiver end. In [25] they extended their work to the case of multiple co-existing networks, proposing a randomized MAC protocol that guarantees fairness between the different networks and efficient use of the bandwidth. In [23], Richa et al. considered an adaptive adversarial jammer that is also reactive; one that 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 constraint 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 and hence they have to deal with transmission collisions as well. What is more, their objective is to optimize throughput over the non-jammed time periods, whereas we include the whole execution. In a more recent work of Ogierman et al. [19], the authors introduce a new SINR model capturing various interference phenomena and propose a distributed MAC protocol that achieves constant competitive throughput. Nonetheless, the main differences with our work are similar to the ones mentioned for their previous works.

Gilbert et al. [16] worked on a theoretical analysis of the damage that can be introduced by a tiny malicious entity having limited power in the communication delay between two nodes. In particular, the nodes share a time-slotted single-hop wireless ratio channel and the malicious entity wishes to delay their communication. However, it can only broadcast a message up to β times, which is similar to the restriction imposed in our work, but our model can be viewed as a generalization of this restriction by allowing recharging. Nonetheless, the setting and objectives of their work are different. They first show a bound on the number of rounds that the malicious node can delay the communication and then study its implication on an n-node general problem, such as reliable broadcast and leader election.

Schmid and Wattenhofer [26, 27] look at TCP transmissions and assume a network where congestion varies with time while packets are lost at random due to bit rate errors in the wireless links. The authors consider two models for the congestion changes: dynamic and bursty. The second model uses an approach based on network calculus which is similar to our approach, but unlike our work they use it in order to define the congestion changes of the network and hence the available bandwidth and maximum transmission rate at any time (slotted time is assumed). Recall that we assume a constant bit rate at all times, hence the nature of the problem is different. Furthermore, unlike our approach, they perform competitive analysis.

However, none of the models studied considers an AQT modeling of the power of the adversarial entity. Adversarial queuing has been used in wireless networks as a methodology for studying their stability under worst case scenarios, removing the stochastic assumptions usually made for the generation of traffic. It concerns the arrival pro-

cess of packets in the system and it has been introduced by Borodin et al. [8] as a well defined theoretical model since 2001. A variety of works has then followed, using AQT in different network settings, such as on multiple access channels [11, 12] and routing in communication networks [9, 10]. We associate our constrained type of adversarial channel jams with the AQT model for the arrival process of packets in the following way. Classical AQT considers a *window adversary* that accounts packets being injected within a time window w in such a way that they give a total load of at most wr at each edge of the paths they need to follow, where $w \geq 1$ and $r \leq 1$. In our channel jams, for every window of duration $1/\rho$, there is exactly one new error token available for the adversary to use. In a long execution, considering for example a time interval $T > 1/\rho$, there will be up to $T\rho$ new error tokens available to the adversary.

Last but not least, as mentioned in Section 1, our adversarial jammer has limited sources of energy and can be used to model, for example, military drones or mobile jammers. Drones or *Unmanned Aerial Vehicles (UAV)* are at the peak of their development. As an upcoming technology that is rapidly improving, it has already attracted the colossi of industry, like Google or Amazon, to invest in UAV research and development, creating even commercial models. There have already been a few research works [14, 18] but the area is still being studied; the work in [14] focuses on UAV's risk analysis and the work in [18] focuses in analyzing cellular network coverage using UAV's and software defined radio. Regarding mobile jammers, in the recent years, many companies have made available battery-operated 3G/4G, WiFi or GPS mobile jammers (e.g., [1, 2]); this market can only increase, as wireless communication is becoming the dominating communication technology.

2. Model

In this section we first formalize the *dynamic* model, the main model considered in this work, and then we highlight the differences with the *static* model, which is extensively used as a *building block* for the dynamic one.

2.1. Dynamic Model

Network setting. We consider a setting of a sending station (sender) that transmits packets to a receiving station (receiver) over an unreliable wireless channel. We assume that the sender has enough data to transmit, covering any interval length T, and follows some *online scheduling* [22, 21] in order to decide the lengths of the packets to be sent in the transmission. The decisions need to be made during the course of the execution, taking into consideration (or not) the channel jams. Each packet π consists of a *header* of a fixed predefined size h and a *payload* of length l chosen by the algorithm. The payload represents the useful data to be sent across the channel and is to be chosen by the sender. The total length of the packet is then denoted by $p = h + l$. For simplicity and without loss of generality we use $h = 1$ throughout our analysis, and hence $p = l + 1$. (Note that l needs not be an integer.) Furthermore, we consider constant bit rate for the channel, which means that the transmission time of each packet is proportional to its length (i.e., a packet of size $l + 1$ takes $l + 1$ time units to be transmitted in full).

Packet failures. We model the unavailability of the channel to be controlled by the omniscient and adaptive adversary (σ, ρ)-$\mathcal{A}$, which is defined by its two "restrictive" parameters, $\rho \in [0, 1]$ and $\sigma \geq 1$ as follows. The adversary has a token bucket of size σ where it stores "error tokens" and is initially full. From the beginning of the execution and up to a time t, within interval $T = [0, t]$, there will be $\lfloor \rho T \rfloor$ such error tokens created, where ρ is the rate at which they become available to the adversary. In other words, a new error token becomes available at times $1/\rho, 2/\rho, \ldots$. Note that the values of the parameters are given to the adversary (they are not chosen by it) and it can only use them in a "smart" way in order to control the packet jams in the channel. If there is at least one token in the bucket, the adversary can introduce an error in the channel and jam the packet being transmitted, consuming one token. If the token bucket is full (i.e., there are already σ error tokens in the bucket) and a new token arrives, then one token is lost. (This, models, for example, the fact that a fully charged battery cannot be further charged.) As a worst case analysis, we consider that the adversary jams some bit in the header of the packets in order to ensure their destruction. Therefore, adversary (σ, ρ)-$\mathcal{A}$ defines the *error pattern* E as a collection of jamming events on the channel, jamming the packet that is being transmitted in that instant. Finally, we assume that parameters ρ and σ are known to the scheduling algorithm.

Efficiency measures. For the efficiency of a scheduling algorithm, we look at the *goodput rate*, G; the ratio of the total amount of payload successfully transmitted over time, despite the jams in the channel.

To be more precise, let us define the amount of payload successfully transmitted as *useful payload*. We denote the useful payload of an algorithm ALG in a time interval T, under error pattern E, by $\text{UP}_T(\text{ALG}, E)$, and it is calculated as the sum of payloads of the packets successfully transmitted in the interval. Since we make a worst-case analysis, we actually calculate the *worst* useful payload of a fixed algorithm A as $\text{UP}_T(A) = \min_{E \in \mathcal{E}(\rho,\sigma)} \text{UP}_T(A, E)$, where $\mathcal{E}(\rho, \sigma)$ is the set of all possible error patterns with parameters ρ and σ. We also define the optimal useful payload as $\text{UP}^*_T = \max_{\text{ALG}} \text{UP}_T(\text{ALG})$. Now, when examining a period T in the execution of an algorithm ALG, under error pattern E, its goodput rate is defined as $G_T(\text{ALG}, E) = \text{UP}_T(\text{ALG})/T$ and the optimal goodput as $G^* = \text{UP}^*_T/T$.

For simplicity, we use the shorter notations UP and G, when the algorithm used or the time interval considered, respectively, are implied. We also overload the notation T to refer both to the interval and its length. Note finally, that in most of our analysis we avoid using *floors* and *ceilings* in order to keep the readability of our results as simple as possible for the reader. Nonetheless, this does not affect the correctness of our results since when being applied on large enough time intervals and data, the "losses" become negligible.

Feedback mechanism. As a feedback mechanism, following [4], we assume that the sender receives instantaneous feedback for a packet successfully received. We also assume that the notification packets cannot be jammed by the errors in the channel because of their relatively small size. In particular, we consider notification / acknowledgement messages sent for every packet that is received successfully. If such a message is not received by the sender, then it considers the packet to be jammed.

2.2. Static Model

We now present the static model, focusing only on the differences it has from the dynamic one.

Packet failures. We assume that the channel jams are orchestrated by an omniscient and adaptive adversary, (T, f)-$\mathcal{A}$. However, it has a constrained number of jams it can cause in a given period. Specifically, for a time interval of length T, $T \geq 1$, it can cause up to f packet jams. Thus, given a parameter 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. As in the dynamic model, 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. 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. We consider the same performance measure as in the dynamic model; *goodput rate*, and use the *useful payload* in order to calculate the exact amount of data successfully transmitted; this time for interval of length T and f error tokens.

More formally, similar to the dynamic model, we denote by $\text{UP}_{(T,f)}(\text{ALG}, E)$ the useful payload (payload successfully received) when using scheduling algorithm ALG in an interval of length T against an adversary of power f that uses error pattern E. Then, for a fixed algorithm A, its useful payload is $\text{UP}_{(T,f)}(A, E) = \min_{E \in \mathcal{E}(f)} \text{UP}_{(T,f)}(A, E)$, where $\mathcal{E}(f)$ is the set of all possible error patterns with at most f jams. From this, we also define the optimal useful payload as $\text{UP}^*_{(T,f)} = \max_A \text{UP}_{(T,f)}(A)$. For simplicity, we use the shorter notation UP. This is done when the algorithm used and the number of possible errors in the interval are implied.

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

Feedback mechanism. As in the dynamic model, we assume instantaneous feedback. Nonetheless, 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 $T > f$.

We start with proving an absolute bound on the error rate with respect to the maximum packet length.

Observation 1. *Let c be the smallest packet size used by an algorithm (i.e., $\forall p, p_{.len} \geq c$). For any error rate $\rho \geq 1/c$, no goodput larger than zero can be achieved.*

Proof: If the error rate is $\rho \geq 1/c$, a new error token arrives during the transmission of any packet (recall that packets are of size at least c). Hence, there are error tokens in the bucket at all times for the adversary to corrupt all packets being transmitted. Using an error token every c time, is sufficient to keep the goodput at zero. ■

From this observation, it can be derived that algorithms that only use packets of length $p_{.len} \geq 1/\rho$ are not interesting.

2.3. Moving from the Static Model to the Dynamic Model

Our approach is to first analyze the static model, and then explore the way its solutions can be applied in the dynamic model. In particular, we divide the executions of the continuous (dynamic) version of the problem into successive intervals of length $1/\rho$, and assume σ error tokens available at the beginning of each interval. Then these intervals become instances of the static model, where $T = 1/\rho$ and $f = \sigma$.

We therefore propose an algorithm ALG_D, that uses the optimal solution of the static model, say algorithm A, to solve the problem in the dynamic model, with parameters $1/\rho$ and σ.

Algorithm ALG_D Description:
For every time interval $T_i = \left[\frac{i}{\rho}, \frac{i+1}{\rho}\right)$, where $i = 0, 1, \ldots$, run $A(1/\rho, \sigma)$.

Observation 2. *Observe that, if the goodput of algorithm A is $G(A)$, then the goodput of ALG_D will also be $G(ALG_D) = G(A)$. This is because the goodput per-interval will be repeated throughout the whole execution.*

3. Uniform packets for the Static Model

Let us start by studying the static model, for the case when the algorithm is restricted in sending packets of equal (uniform) length. This could be due to limitations in the communication protocol or the sender's specification. We aim to define a quasi-optimal algorithm S-UNI that schedules uniform packets taking into account the parameters of the adversary. For that, we compute the quasi-optimal necessary packet length, p^*, that maximizes the minimum useful payload considering time interval T and maximum number of errors f.
Note that the approximations below are due to floors and ceilings; these approximations get closer to equality as Tf grows.

Theorem 1. *Let S-UNI use only uniform packets of length p. In an interval of length T and maximum number of errors f, the optimal packet length for these algorithms, p^*, gives a useful payload*

$$UP_{(T,f)}(\text{S-UNI}_{p^*}) = \max\left\{\frac{1}{\lfloor\sqrt{Tf}\rfloor}\left(\lfloor\sqrt{Tf}\rfloor - f\right)\left(T - \lfloor\sqrt{Tf}\rfloor\right), \frac{1}{\lceil\sqrt{Tf}\rceil}\left(\lceil\sqrt{Tf}\rceil - f\right)\left(T - \lceil\sqrt{Tf}\rceil\right)\right\}$$

and thus a corresponding goodput rate

$$G_{(T,f)}(\text{S-UNI}_{p^*}) = \max\left\{\frac{1}{T\lfloor\sqrt{Tf}\rfloor}\left(\lfloor\sqrt{Tf}\rfloor - f\right)\left(T - \lfloor\sqrt{Tf}\rfloor\right), \frac{1}{T\lceil\sqrt{Tf}\rceil}\left(\lceil\sqrt{Tf}\rceil - f\right)\left(T - \lceil\sqrt{Tf}\rceil\right)\right\}.$$

In fact, $UP(\text{S-UNI}_{p^*}) \approx T + f - 2\sqrt{Tf}$ and $G(\text{S-UNI}_{p^*}) \approx \left(1 - \sqrt{f/T}\right)^2$.

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. 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.

Let us denote by S-UNI$_n$ and S-UNI$_p$ the same algorithm, that uses n uniform packets of length p. Recall that each packet consists of the payload and a unit-size header. Its useful payload will then be $\text{UP}_{(T,f)}(\text{S-UNI}_n) = (n-f)\left(\frac{T}{n}-1\right)$. Deriving this expression with respect to n, we get

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

which implies that $\text{UP}_{(T,f)}(\text{S-UNI}_n)$ is maximized for $n = \sqrt{Tf}$. What is more, the derivative is positive for $n < \sqrt{Tf}$ and negative for $n > \sqrt{Tf}$. This means, 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^*.

Thus, the optimal useful payload is

$$\begin{aligned}\text{UP}_{(T,f)}(\text{S-UNI}_{n^*}) &= (n^*-f)\left(\frac{T}{n^*}-1\right)\\ &= \max\left\{\frac{1}{\lfloor\sqrt{Tf}\rfloor}\left(\lfloor\sqrt{Tf}\rfloor - f\right)\left(T-\lfloor\sqrt{Tf}\rfloor\right), \frac{1}{\lceil\sqrt{Tf}\rceil}\left(\lceil\sqrt{Tf}\rceil - f\right)\left(T-\lceil\sqrt{Tf}\rceil\right)\right\}\end{aligned}$$

and the corresponding goodput rate

$$\begin{aligned}G_{(T,f)}(\text{S-UNI}_{n^*}) &= \frac{\text{UP}_{(T,f)}(\text{S-UNI}_{n^*})}{T}\\ &= \max\left\{\frac{1}{T\lfloor\sqrt{Tf}\rfloor}\left(\lfloor\sqrt{Tf}\rfloor - f\right)\left(T-\lfloor\sqrt{Tf}\rfloor\right), \frac{1}{T\lceil\sqrt{Tf}\rceil}\left(\lceil\sqrt{Tf}\rceil - f\right)\left(T-\lceil\sqrt{Tf}\rceil\right)\right\},\end{aligned}$$

as claimed.

From the optimal number n^*, and the fact that $p^* = \frac{T}{n^*}$, we get that $p^* \approx \sqrt{T/f}$. Then, the optimal achievable useful payload becomes $\text{UP}_{(T,f)}(\text{S-UNI}_{p^*}) \approx T + f - 2\sqrt{Tf}$ and the corresponding optimal goodput rate, $G_{(T,f)}(\text{S-UNI}_{p^*}) \approx \left(1-\sqrt{f/T}\right)^2$, as also claimed. ■

4. Adaptive Algorithms for Static Model with $f = 1$

We now turn our attention to some adaptive algorithms; ones that change the packet sizes according to the jams they have observed so far. Starting from the case of $f = 1$ we propose algorithms that achieve a goodput rate greater than $G^*_{(T,1)}(\text{S-UNI}) \approx (1-\sqrt{1/T})^2$.

4.1. Algorithm S-DEC

The first algorithm we propose, that adapts the length of the packets sent, is called S-DEC and we show here that for time intervals T large enough, for $T > \frac{2}{7-3\sqrt{5}}$ to be exact, it achieves goodput rate greater than $G_{(T,1)}(\text{S-UNI}) \approx (1-\sqrt{1/T})^2$.

Algorithm S-DEC Description:
Each period starts by scheduling packets of decreasing length $p_i = Z - i$ for $i = 0, 1, 2, 3 \ldots$. If a packet π_j is jammed during the period, this transmission sequence is stopped, and after π_j, a single more packet is scheduled by the algorithm whose length spans the rest of the period.

Theorem 2. *Adaptive algorithm S-DEC, with* $Z = \frac{1}{2}\left(\sqrt{1+8T}-1\right)$, *achieves goodput* $G(\text{S-DEC}) = 1 - \frac{1}{2T}\left(1+\sqrt{1+8T}\right)$. *This value is larger than the upper bound for the uniform case, if* $T > \frac{2}{7-3\sqrt{5}} \approx 6.8541$.

Proof: There are two cases to be considered in a period:

(a) If the adversary jams a packet π_j, the useless data sent in the period adds to $Z+1$. This number comes from the j headers of the packets sent before π_j, plus the length $p_j = Z - j$ of the packet jammed, plus the header of the last packet sent in the period (which cannot be jammed). Hence, in this case, the useful payload of the period is $T-(Z+1)$.

Otherwise, (b) if no packet is jammed, the useless data sent in the period correspond only to the headers of the packets sent. Then, if the last packet sent in the interval is π_k, the useless data is $k+1$, and the corresponding useful payload is $T-(k+1)$. The value Z is chosen so that the total length of the packets sent in this case is equal the length of the interval. From this property, $\sum_{i=0}^{k} p_i = T$, the value of Z must satisfy $Z(k+1) - \frac{k(k+1)}{2} = T$ and hence

$$Z = \frac{k}{2} + \frac{T}{k+1}. \tag{1}$$

In a given period the choice of whether case (a) or (b) occurs is up to the adversary, since she can decide which packet to jam, if any. This means that the useful payload achieved will be the minimum of the two cases, UP $= \min\{T-(Z+1), T-(k+1)\}$. Observe from this Eq. 1 that the length Z of the initial packet increases if the number of packets k decreases. Additionally, it must hold that $Z \geq k$ and therefore UP is maximized when when $Z = k$. Hence, the optimal k is the suitable solution of the equation $k = \frac{k}{2} + \frac{T}{k+1}$, which is $k = \frac{1}{2}\left(\sqrt{1+8T}-1\right) = Z$.

The useful payload achieved is then UP(S-DEC) $= T - \left(\frac{1}{2}\sqrt{1+8T} - \frac{1}{2} + 1\right) = T - \frac{1}{2}\left(\sqrt{1+8T}+1\right)$, which is more than UP*(S-UNI) $= T \cdot G_{(T,1)}$(S-UNI) $= T\left(1-\sqrt{1/T}\right)$. The corresponding goodput is therefore G(S-DEC) $= \frac{\text{UP}}{T} = 1 - \frac{1}{2T}\left(\sqrt{1+8T}+1\right)$. ■

4.2. Algorithm S-OPT$(T,1)$: optimal for $f=1$

Since the performance of algorithm S-DEC is only better than the uniform packet scheduling approach for a limited range of intervals, i.e., $T > \frac{2}{7-3\sqrt{5}}$, we aim to improve the result given by S-DEC in the previous subsection, and see whether a goodput rate that surpasses G(S-UNI) exists for time intervals $T < \frac{2}{7-3\sqrt{5}}$. In our effort to do so, we develop the following adaptive algorithm, named S-OPT$(T,1)$, which we prove to be optimal for the static model, for $f=1$. (See the algorithm's pseudocode in Alg. 1.) By doing so, we also hope to give an intuition to the reader on *how* the optimal algorithm for any number of error tokens will work.

Algorithm 1 S-OPT$(T,1)$

If $T \in [1,2)$ then
  Send packet π 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 S-OPT$(T-p,1)$

Algorithm S-OPT$(T,1)$ 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 S-OPT$(T-p,1)$.

A detailed pseudocode for 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 S-OPT$(T,1)$ in the static model.

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 $UP_{(T',1)}(S\text{-}OPT) = \frac{\alpha T' - \beta}{\alpha+1}$, where $T' = T - p$. Then, Algorithm S-OPT$(T,1)$ achieves useful payload $UP_{(T,1)}(S\text{-}OPT) = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2}$.*

Proof: Since $T \geq 2$, that Algorithm S-OPT$(T,1)$ 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 $\text{UP}_{(T,1)}(\text{S-OPT}) = T - \frac{T+\beta}{\alpha+2} - 1 = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2}$.

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

Theorem 3. *Given an interval of length $T \geq 1$, Algorithm S-OPT$(T,1)$ 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 S-OPT$(T,1)$, 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 S-OPT$(T,1)$ achieves optimal useful payload $\text{UP}^*_{(T,1)} = 0$ 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, $\text{UP}_{(T',1)}(\text{S-OPT}) = \text{UP}^*_{(T',1)} = \frac{j-1}{j}T - \frac{j+1}{2} + \frac{1}{j} = \frac{\alpha T' - \beta}{\alpha+1}$, and from Lemma 2 we have that $\text{UP}_{(T,1)}(\text{S-OPT}) = \frac{(\alpha+1)T - (\beta+\alpha+2)}{\alpha+2} = \frac{i-1}{i}T - \frac{i+1}{2} + \frac{1}{i}$.

To show that the useful payload achieved by S-OPT 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

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

(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, $\text{UP}^*_{(T'+(p-p'),1)} = \frac{j-1}{j}(T' + (p - p')) - \frac{j+1}{2} + \frac{1}{j} < \frac{j-1}{j}T' - \frac{j+1}{2} + \frac{1}{j} + (p - p') = \text{UP}^*_{(T',1)} + (p - p')$. Hence,

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

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,

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

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 S-OPT, which implies that the latter achieves optimality. ■

5. Algorithm S-OPT(T, f): optimal for any $f > 1$ in the Static Model

We now turn our focus on the case of any number of error tokens available to the adversary, for an interval of length T, i.e., $s > 1$. We present the general adaptive algorithm S-OPT(T, f) for $f > 1$ as Algorithm 2, and prove its optimality in the static model. The pseudocode of S-OPT(T, f) for $f > 1$ is similar to that of S-OPT$(T, 1)$, 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 4). 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, S-OPT(T, f) makes the call S-OPT$(T - p, f - 1)$, which could be recursive if $f > 2$, or a call to the algorithm S-OPT$(T - p, 1)$ (see Algorithm 1), if $f = 2$. It will not be surprising then that the proof of optimality of the algorithm S-OPT(T, f) will use induction on f.

Algorithm 2 S-OPT(T, f), for $f > 1$

If $T < f + 1$ then
  Send packet π 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 4
  If packet π is jammed then
    Call S-OPT$(T - p, f - 1)$
  else
    Call S-OPT$(T - p, f)$

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

Observation 3. *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 $\text{UP}^*_{(T,f)} = \alpha$, 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., $\text{UP}_{(T+\delta,f)}(A) \geq \alpha$. 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., $\text{UP}^*_{(T+\delta,f)} \geq \text{UP}_{(T+\delta,f)}(A)$. Hence, $\text{UP}^*_{(T,f)} \leq \text{UP}^*_{(T+\delta,f)}$ as claimed. ■

Observation 4. *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 $\text{UP}^*_{(T,f)} = \beta$ 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, $\text{UP}_{(T,f-1)}(A) \geq \beta$. 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., $\text{UP}^*_{(T,f-1)} \geq \text{UP}_{(T,f-1)}(A)$. 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 3. 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 that

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

$$\text{UP}^*_{(T,f)} = \min\{\text{UP}^*_{(T-p_O,f-1)}, p_O - 1 + \text{UP}^*_{(T-p_O,f)}\} \quad (3)$$

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 interval lengths, then

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

Since $\text{UP}^*_{(T-\epsilon,f)} \geq \text{UP}_{(T-\epsilon,f)}(A)$, 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 S-OPT(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 S-OPT(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 $\text{UP}^*(T', j)$. Then, S-OPT(T, f) chooses as length of packet π the smallest value $p \in [1, T]$ that satisfies the equality $\text{UP}^*_{(T-p,f-1)} = p - 1 + \text{UP}^*_{(T-p,f)}$. 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	$\frac{T}{3}$	$\frac{T+6}{7}$	$\frac{3T+3}{12}$	$\frac{5T+16}{26}$	$\frac{6T+42}{42}$
$\text{UP}^*_{(T,2)}$	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 $\text{UP}^*_{(T,2)}$ achieved with Algorithm S-OPT$(T, 2)$.

We prove that the described process to make the choice leads to optimality in the following theorem.

Theorem 4. *Given an interval of length $T \geq f + 1$, Algorithm S-OPT(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 S-OPT(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 $\text{UP}^*_{(T,0)} = T - 1$, and (2) that the algorithm S-OPT$(T, 1)$ presented before 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 $\text{UP}^*_{(T,f)} = 0$, since the adversary can jam each of the up to f packets that can be sent.

Induction Hypotheses. We first inductively assume that S-OPT(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 $R_{ij} = [a_{ij}, b_{ij})$ such that $T \in R_{ij}$, 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 $R_{if} = [c_{if}, d_{if})$ for $i = 1, 2, \ldots, m$, such that $\bigcup_{i=1}^{m} R_{if} = [1, d_{mf})$. 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 $l \leq r \leq m$ it holds that (2) $\frac{\beta_{rf}}{\gamma_{rf}} \geq \frac{\beta_{lf}}{\gamma_{lf}}$ and (3) $\frac{\alpha_{rf}}{\gamma_{rf}} \geq \frac{\alpha_{lf}}{\gamma_{lf}}$.

Inductive Step. For interval length $T \in [d_{mf}, d_{mf}+1)$, the algorithm S-OPT(T,f) chooses the smallest packet length $p \in [1,T]$ that satisfies the following condition

$$\text{UP}^*_{(T-p,f-1)} = p - 1 + \text{UP}^*_{(T-p,f)}. \tag{4}$$

Claim 1. *There is at least one packet length $p \in [1,T]$ that satisfies Eq. 4.*

Proof: Observe that, when $p = 1$, from Observation 4 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 $\text{UP}^*_{(T-p,f-1)} = p - 1 + \text{UP}^*_{(T-p,f)}$. ■

Now, let p be the packet length chosen, and assume that $T - p \in R_{kj}$ and $T - p \in R_{lf}$. 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. 4 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

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

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: 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. ■

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: 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:

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

which completes the proof. ■

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: 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:

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

which completes the proof. ■

We must now show that this useful payload is in fact optimal in the static model. 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 $\text{UP}_{(T,f)}(A) \leq \text{UP}^*_{(T-p',f-1)}$ which by Observation 3 is smaller than $\text{UP}^*_{(T-p,f-1)} = \text{UP}_{(T,f)}(\text{S-OPT})$, since $T-p' < T-p$.
(b) Algorithm A chooses a packet π' of length $p' < p$. Observe that p' does not satisfy Eq. 4, since p is the smallest length that does. Then the adversary does not jam π'. Then, $\text{UP}_{(T,f)}(A) \leq p'-1+\text{UP}^*_{(T-p',f)}$. We show now that this value is no larger than $p-1+\text{UP}^*_{(T-p,f)} = \text{UP}_{(T,f)}(\text{S-OPT})$. Let us assume that $T-p' \in R_{rf}$, where $r \geq l$. Then, $\text{UP}^*_{(T-p',f)} = \frac{\alpha_{rf}(T-p')-\beta_{rf}}{\gamma_{rf}} \leq \frac{\alpha_{rf}}{\gamma_{rf}}(T-p')-\frac{\beta_{lf}}{\gamma_{lf}}$, since $\frac{\beta_{rf}}{\gamma_{rf}} \geq \frac{\beta_{lf}}{\gamma_{lf}}$ as shown by Property 2. Similarly, $\text{UP}^*_{(T-p,f)} = \frac{\alpha_{lf}(T-p)-\beta_{lf}}{\gamma_{lf}} \geq \frac{\alpha_{rf}}{\gamma_{rf}}(T-p)-\frac{\beta_{lf}}{\gamma_{lf}}$, 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

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

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

6. Uniform packets for the Dynamic Model

The main goal for the algorithms in the dynamic model, is to maximize the data successfully transmitted to the receiver in any interval T. This, corresponds to minimizing the transmission time needed to successfully transmit a total amount of data P to the receiver, considering a value P that will eventually grow to infinity. As a consequence, this would also maximize the goodput rate, which is our main efficiency measure for the two models. Knowing both adversarial parameters, ρ and σ, let us consider algorithm D-UNI and uniform packets of size $p_{.len} = l + 1 < 1/\rho$. We can then find the quasi optimal value for the length of the payload l in each packet that minimizes the transmission time. For simplicity, we will assume that the total length of the data to be transmitted, P, is a multiple of the payload length l. (For large values of P the error introduced by this assumption is negligible.) Then, the objective is that P/l packets arrive successfully at the receiver.

Let us now derive a *lower bound* on the transmission time that can be achieved using uniform packets. We denote with $Tr(l)$ the transmission time with packets of uniform payload l. Let r be the number of packets jammed and retransmitted by the sender. Then,

$$Tr(l) = (P/l + r)(l + 1). \tag{5}$$

Observe that the last packet transmitted was correctly received, since otherwise the data would have been completely transmitted by time $Tr(l) - (l + 1)$, which contradicts the fact that $Tr(l)$ is the transmission time. Hence, the number of packets jammed and retransmitted is upper bounded as

$$r \leq \lceil (Tr(l) - (l + 1))\rho \rceil - 1 + \sigma, \tag{6}$$

where we apply the fact that the last error used by the adversary must have been available before time $Tr(l) - (l+1)$. We claim that the number of packets jammed by the adversary and retransmitted is in fact equal to the bound of Eq. 6. Otherwise, the adversary could have jammed the last packet sent (at time $Tr(l) - (l + 1)$), achieving a longer transmission time. Hence,

$$r = \lceil (Tr(l) - (l + 1))\rho \rceil - 1 + \sigma. \tag{7}$$

Moreover, since the adversary could not jam the last packet sent, it must also hold that $r + 1 \geq Tr(l)\rho + \sigma = (P/l + r)(l + 1)\rho + \sigma$, from which we can bound the value of r as

$$r \geq \frac{P\rho(l + 1) + (\sigma - 1)l}{l - l\rho(l + 1)}. \tag{8}$$

Let us define the lower bound of the transmission time when packets of uniform payload l are used, as function $LB(l)$. Then,

Lemma 5. *Using algorithm D-UNI with uniform packets of payload l, the lower bound of the transmission time is*

$$Tr(l) \geq LB(l) = \frac{P + (\sigma - 1)l}{l(1 - \rho(l + 1))}(l + 1).$$

Proof: Replacing the lower bound of r (Eq. 8) in Eq. 5 we have

$$Tr(l) \geq \left(\frac{P}{l} + \frac{P\rho(l + 1) + (\sigma - 1)l}{l - l\rho(l + 1)}\right)(l + 1) = \frac{P + (\sigma - 1)l}{l(1 - \rho(l + 1))}(l + 1),$$

which when combined with the definition of $LB(l)$, completes the proof. ■

Using calculus, we can find the payload length l^* that minimizes $LB(l)$, which yields the following theorem.

Theorem 5. *Using uniform packets the transmission time is lower bounded as*

$$Tr \geq LB(l^*) = \frac{P + (\sigma - 1)l^*}{l^*(1 - \rho(l^* + 1))}(l^* + 1)$$

and the goodput rate is upper bounded as

$$G(\textit{D-UNI}) \leq \frac{P}{LB(l^*)} = \frac{Pl^*(1-\rho(l^*+1))}{(P+(\sigma-1)l^*)(l^*+1)},$$

where

$$l^* = \frac{\sqrt{P(P\rho+(\sigma-1)(1-\rho))}-P\rho}{P\rho+\sigma-1}.$$

Obviously, when P tends to ∞, so does the transmission time Tr. However, we can derive in this case an upper bound on the goodput as follows.

Corollary 1. *Using algorithm D-UNI with uniform packets, the goodput rate is upper bounded as* $G(\textit{D-UNI}) \leq (1-\sqrt{\rho})^2$*, and in the limit as the value of* P *grows,*

$$G^* = \lim_{P\to\infty} G(\textit{D-UNI}) = (1-\sqrt{\rho})^2$$

Proof: Using calculus it can be shown that the upper bound of G(D-UNI) obtained in Theorem 5 grows with P. Observe that $\lim_{P\to\infty} G(\text{D-UNI}) = l^*(1-\rho(l^*+1))/(l^*+1)$ and $\lim_{P\to\infty} l^* = (\sqrt{\rho}-\rho)/\rho = 1/\sqrt{\rho}-1$. Replacing the latter in the former the claims follow. ■

We now show a corresponding *upper bound* on the transmission time. We start by combining Eqs. 7 and 5 as follows:

$$\begin{aligned} r &= \lceil (Tr(l)-(l+1))\rho \rceil - 1 + \sigma < (Tr(l)-(l+1))\rho + \sigma \\ &= ((P/l+r)(l+1)-(l+1))\rho + \sigma \\ &= (P/l+r)(l+1)\rho + \sigma - (l+1)\rho. \end{aligned}$$

This allows us to find an upper bound of r as

$$r < \frac{P\rho(l+1)+(\sigma-(l+1)\rho)l}{l-l\rho(l+1)}. \tag{9}$$

Let us now define the upper bound of the transmission time when packets of payload l are used, as function $UB(l)$. Then,

Lemma 6. *Using algorithm D-UNI with uniform packets of payload* l*, the upper bound of the transmission time is*

$$Tr(l) < UB(l) = \frac{P+(\sigma-(l+1)\rho)l}{l(1-\rho(l+1))}(l+1).$$

Proof: Replacing the upper bound of r (Eq. 9) in Eq. 5 we have

$$Tr(l) < \left(\frac{P}{l} + \frac{P\rho(l+1)+(\sigma-(l+1)\rho)l}{l-l\rho(l+1)}\right)(l+1) = \frac{P+(\sigma-(l+1)\rho)l}{l(1-\rho(l+1))}(l+1),$$

which when combined with the definition of $UB(l)$, completes the proof. ■

From Observation 1, $\rho < 1/(l+1)$ must hold. Then, $(l+1)\rho < 1$ and the bound obtained in the above lemma is strictly bigger than the lower bound presented in Lemma 5, as expected. In fact, the gap between bounds can be obtained as shown in the following lemma.

Lemma 7. *Using uniform packets of payload* l*, the transmission time satisfies* $Tr(l) \in [LB(l), LB(l)+l+1)$.

Proof: Recall that the lower bound $LB(l)$ is obtained in Lemma 5. Subtracting this expression from the upper bound $UB(l)$ presented in Lemma 6, we have

$$\begin{aligned} UB(l) - LB(l) &= \frac{P + (\sigma - (l+1)\rho)l}{l(1-\rho(l+1))}(l+1) - \frac{P + (\sigma-1)l}{l(1-\rho(l+1))}(l+1) \\ &= \frac{l(1-\rho(l+1))}{l(1-\rho(l+1))}(l+1) = l+1. \end{aligned}$$

From the above and the fact that $Tr(l) < UB(l)$ the claim follows. ■

Corollary 2. *Using uniform packets of payload l, $Tr(l)$ is the only multiple of $l+1$ that falls in the interval $[LB(l), LB(l)+l+1)$.*

Finally, combining Lemma 7 with Theorem 5 we derive the following theorem.

Theorem 6. *Consider l^* as defined in Theorem 5. Then*

- *the transmission time $Tr(l^*)$ observed is less that l^*+1 (one packet) longer that the optimal. I.e., $Tr(l^*) < Tr + l^* + 1$.*
- *the goodput $G(l^*)$ converges to the optimal goodput G(D-UNI) as P grows. Additionally, when P goes to infinity the goodput matches the optimal G^*, i.e. $\lim_{P\to\infty} G(l^*) = \lim_{P\to\infty} G$(D-UNI$) = (1-\sqrt{\rho})^2$.*

Proof: The first claim follow directly from Lemma 7, since the value of l^* is the one that minimizes $LB(l)$. For the second, recall that $G(l^*) = \frac{P}{Tr(l^*)}$. Hence, observing again Lemma 7 we get that

$$G(l^*) > \frac{P}{LB(l^*) + l^* + 1} = \frac{1}{\frac{LB(l^*)}{P} + \frac{l^*+1}{P}}.$$

As P grows $\frac{l^*+1}{P}$ tends to 0, making $G(l^*)$ converge to $P/LB(l^*)$ which is an upper bound on the optimal goodput. Finally, as shown in Corollary 1, when P tends to infinity, $P/LB(l^*)$ tends to $(1-\sqrt{\rho})^2$, which completes the proof. ■

7. Discussion/Conclusions

In this paper we have applied Adversarial Queuing Theory (AQT), a well known theoretical modeling tool, to restrict adversarial packet jamming on wireless networks, creating the *dynamic model* studied. We have chosen a constrained adversarial entity, considering a bounded error-token capacity σ and an error-token availability rate ρ. This model could be applied in various battery-operated malicious devices, such as drones or mobile jammers.

We have also studied a *static* model, for which new parameters are considered; for an interval of time T the adversary is able to create at most f jams, having all f error-tokens available at the beginning of the interval. This model is used as a building block in our aim to find a solution to the problem of the dynamic model.

We have first shown an upper bound on the goodput rate of the static model, when uniform packet lengths are used, proposing algorithm S-UNI. Then, focusing on $f = 1$, we have shown that adaptive algorithms that change the packet length based on feedback received for jammed packets, can actually achieve better goodput rates, thus showing that the uniform packet scheduling is not the best approach. What might seem surprising is that even for the "simple" case of $f = 1$, the analysis of the adaptive algorithms is nontrivial, and imposes constraints also on T.

In Figure 1, you can see a graphical representation of the improvement in the goodput rate by the different algorithms developed, for the case of $\sigma = f = 1$ and $T = 1/\rho$. Unfortunately, as also shown by our analysis, algorithm S-DEC is better than S-UNI only for $T > \frac{2}{7-3\sqrt{5}}$. However, this has given a positive intuition for the fact that other adaptive algorithms may exist with better goodput rate, as well as for the smaller time intervals. Exploring this further, we proposed algorithm S-OPT$(T, 1)$, which as we have also shown analytically, exceeds the performance of S-UNI for

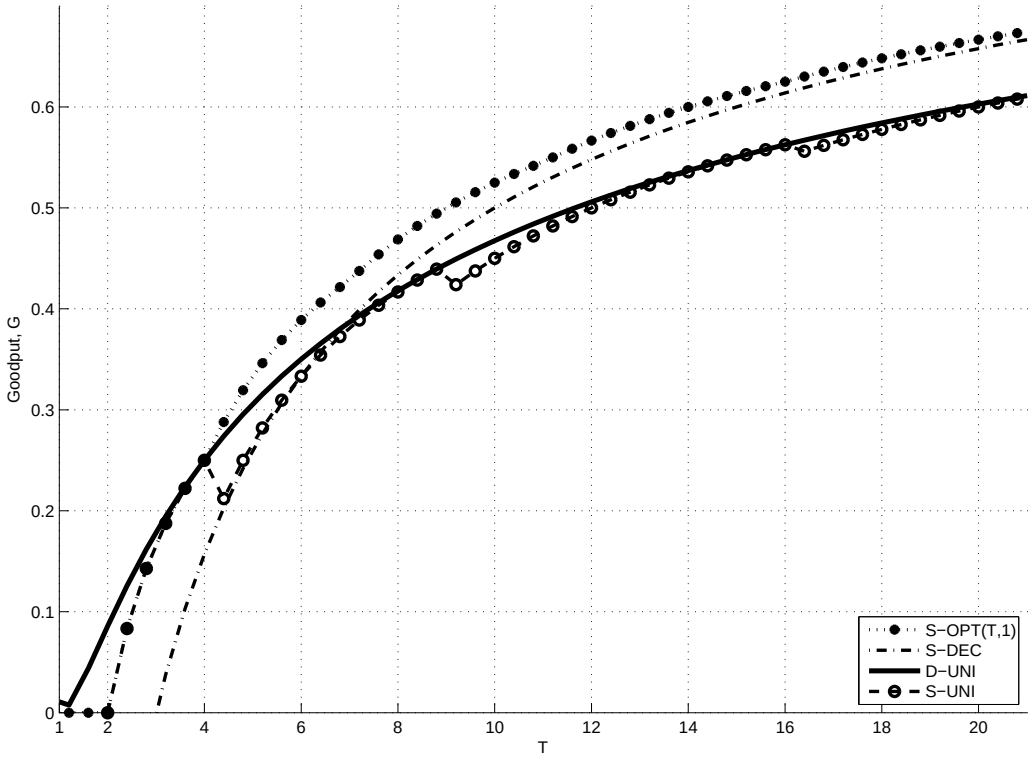

Figure 1: The goodput rate of algorithms S-OPT$(T, 1)$, S-DEC and the uniform packet scheduling for both static and dynamic models, with $\sigma = f = 1$ in a time interval $1/\rho = T = 1 \ldots 22$.

$T > 4$, and is in fact optimal. Finally, we show the goodput rate of the uniform packet scheduling algorithm D-UNI, developed for the dynamic model, which is actually better than all proposed algorithms for intervals $T < 4$. We believe that this is due to the fact that D-UNI is not restricted to fit the packet length in the intervals $1/\rho$, and further investigation is necessary to see whether there exists any other adaptive packet scheduling algorithm that exceeds that goodput rate.

In Section 2.3 we proposed a recursive algorithm ALG_D, that uses the optimal solution of the static model to solve the problem in the dynamic model. It divides the executions into consecutive intervals of length $1/\rho$, and assumes σ error tokens available at the beginning of each one. Then these intervals can be seen as instances of the static model, where $T = 1/\rho$ and $f = \sigma$. However, this algorithm may not be the best possible, as we make the pessimistic assumption that at the beginning of each interval, the adversary will have 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 further below).

Based on the dynamic model, 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 S-OPT$(1/\rho, \sigma - 2)$ instead of S-OPT$(1/\rho, \sigma)$. 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 S-OPT$()$. 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 above, algorithm S-OPT$(1/\rho, 1)$ obtains better results than Algorithm S-DEC. 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. Nonetheless, we have also noticed that the uniform packet scheduling algorithm D-UNI still achieves better goodput rate for some small values of T. Apart from whether *that* can be exceeded, an intriguing open question is whether it is still possible to obtain better efficiency than the uniform packet lengths "policy", with adaptive algorithms for $\sigma > 1$. Considering for example $\sigma = 2$ seems to already be a challenging task.

We believe that our results are only the beginning of further interesting research lines. In general, they motivate studies on the aspect of data partitioning in case of threads. Our work could also be seen in the context of distributed or parallel job executions. An interesting future direction is to investigate the case where one or both parameters ρ and σ are not known; here one will need to monitor the history of the observed jams in an attempt to estimate these parameters. On the other hand, the adversary will try to "hide" the true value of these parameters, yielding an interesting gameplay between the adversary and an algorithm. Another direction to follow would be to consider in addition the channel errors due to congestion and transmission rate. A related future research line could also be to consider multi-channel settings and study the benefits that similar approaches could have depending on the strength of the adversaries assumed. Furthermore, conducting a study on randomized algorithms and giving lower bounds for that case would be valuable in order to identify their limits as well.

References

[1] http://alljammer.com/ (last accessed: April 8, 2015).

[2] http://www.jammer-store.com/ (last accessed: April 8, 2015).

[3] Matthew Andrews, Baruch Awerbuch, Antonio Fernández, Tom Leighton, Zhiyong Liu, and Jon Kleinberg. Universal-stability results and performance bounds for greedy contention-resolution protocols. *Journal of the ACM (JACM)*, 48(1):39–69, 2001.

[4] Antonio Fernández Anta, Chryssis Georgiou, Dariusz R Kowalski, Joerg Widmer, and Elli Zavou. Measuring the impact of adversarial errors on packet scheduling strategies. In *Structural Information and Communication Complexity (SIROCCO)*, pages 261–273. Springer, 2013.

[5] Antonio Fernández Anta, Chryssis Georgiou, and Elli Zavou. Packet scheduling over a wireless channel: AQT-based constrained jamming. In *Proceedings of the 2015 International Conference on Networked Systems (NETYS), accepted*, 2015.

[6] Baruch Awerbuch, Andrea Richa, and Christian Scheideler. A jamming-resistant mac protocol for single-hop wireless networks. In *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*, PODC '08, pages 45–54, New York, NY, USA, 2008. ACM.

[7] Pravin Bhagwat, Partha Bhattacharya, Arvind Krishna, and Satish K Tripathi. Enhancing throughput over wireless lans using channel state dependent packet scheduling. In *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, volume 3, pages 1133–1140. IEEE, 1996.

[8] Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P Williamson. Adversarial queuing theory. *Journal of the ACM (JACM)*, 48(1):13–38, 2001.

[9] Bogdan S Chlebus, Vicent Cholvi, and Dariusz R Kowalski. Stability of adversarial routing with feedback. In *Networked Systems*, pages 206–220. Springer, 2013.

[10] Bogdan S Chlebus, Vicent Cholvi, and Dariusz R Kowalski. Universal routing in multi hop radio networks. In *Proceedings of the 10th ACM international workshop on Foundations of mobile computing*, pages 19–28. ACM, 2014.

[11] Bogdan S Chlebus, Dariusz R Kowalski, and Mariusz A Rokicki. Adversarial queuing on the multiple-access channel. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 92–101. ACM, 2006.

[12] Bogdan S Chlebus, Dariusz R Kowalski, and Mariusz A Rokicki. Stability of the multiple-access channel under maximum broadcast loads. In *Stabilization, Safety, and Security of Distributed Systems*, pages 124–138. Springer, 2007.

[13] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, Dariusz R Kowalski, Calvin Newport, Fabian Kohn, and Nancy Lynch. Reliable distributed computing on unreliable radio channels. In *Proceedings of the 2009 MobiHoc S 3 workshop on MobiHoc S 3*, pages 1–4. ACM, 2009.

[14] Michelle S Faughnan, Brian J Hourican, G Collins MacDonald, Megha Srivastava, JA Wright, YY Haimes, E Andrijcic, Zhenyu Guo, and JC White. Risk analysis of unmanned aerial vehicle hijacking and methods of its detection. In *Systems and Information Engineering Design Symposium (SIEDS), 2013 IEEE*, pages 145–150. IEEE, 2013.

[15] Zhenghua Fu, Petros Zerfos, Haiyun Luo, Songwu Lu, Lixia Zhang, and Mario Gerla. The impact of multihop wireless channel on tcp throughput and loss. In *INFOCOM 2003. Twenty-second annual joint conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1744–1753. IEEE, 2003.

[16] Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks. *Theoretical Computer Science*, 410(6):546–569, 2009.

[17] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. *ACM SIGCOMM Computer Communication Review*, 37(4):385–396, 2007.

[18] Michal Jakubiak. Cellular network coverage analysis using uav and sdr. Master's thesis, Tampere University of Technology, 2014.

[19] Adrian Ogierman, Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive mac under adversarial sinr. In *INFOCOM, 2014 Proceedings IEEE*, pages 2751–2759. IEEE, 2014.

[20] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys & Tutorials, IEEE*, 13(2):245–257, 2011.

[21] Kirk Pruhs. Competitive online scheduling for server systems. *ACM SIGMETRICS Performance Evaluation Review*, 34(4):52–58, 2007.

[22] Kirk Pruhs, Jiri Sgall, and Eric Torng. Online scheduling. *Handbook of scheduling: algorithms, models, and performance analysis*, pages 15–1, 2004.

[23] A. Richa, C. Scheideler, S. Schmid, and Jin Zhang. Competitive and fair medium access despite reactive jamming. In *31st International Conference on Distributed Computing Systems (ICDCS)*, pages 507–516, 2011.

[24] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Towards jamming-resistant and competitive medium access in the sinr model. In *Proceedings of the 3rd ACM workshop on Wireless of the students, by the students, for the students*, pages 33–36. ACM, 2011.

[25] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive and fair throughput for co-existing networks under adversarial interference. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, pages 291–300. ACM, 2012.

[26] Stefan Schmid and Roger Wattenhofer. Dynamic internet congestion with bursts. In *International Conference on High-Performance Computing*, pages 159–170. Springer, 2006.

[27] Stefan Schmid and Rogert Wattenhofer. A tcp with guaranteed performance in networks with dynamic congestion and random wireless losses. In *Proceedings of the 2nd annual international workshop on Wireless internet*, page 9. ACM, 2006.

[28] David Thuente and Mithun Acharya. Intelligent jamming in wireless networks with applications to 802.11 b and other networks. In *Proc. of MILCOM*, volume 6, 2006.

[29] Vagelis Tsibonis, Leonidas Georgiadis, and Leandros Tassiulas. Exploiting wireless channel state information for throughput maximization. *Information Theory, IEEE Transactions on*, 50(11):2566–2582, 2004.

[30] Elif Uysal-Biyikoglu, Balaji Prabhakar, and Abbas El Gamal. Energy-efficient packet transmission over a wireless link. *IEEE/ACM Trans. Netw.*, 10(4):487–499, August 2002.