PURE TAIL CONSTRAINTS FOR ONLINE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Controlling tail risk is an important objective in online optimization, and recently it has been studied in the context of competitive analysis. Continuing this line of research, we investigate *pure tail constraints*, which capture the tradeoff between expected and worst-case competitiveness. For two fundamental search problems, online bidding and line search, we derive the Pareto-optimal frontiers of this tradeoff. We then investigate another classic problem, TCP acknowledgment, which has structure similar to the iterated ski rental problem. There, we construct an algorithm whose tradeoff coincides with the known Pareto-optimal tradeoff for ski rental. The lower bounds for this problem are substantially more involved as the problem exhibits adaptive structure: an online algorithm observes requests of the adversary (packet arrivals) and may *adaptively* adjust its actions (acknowledgments) on this basis. We emphasize that all previous work on tail risk in the context of competitive analysis was restricted to non-adaptive problems, where the feedback given to an algorithm was essentially limited to a binary indicator of whether the algorithm has succeeded or not. Nonetheless, we identify a set of constraints implied by tail bounds in this adaptive setting, and show that they imply a nontrivial lower bound on the TCP acknowledgment problem.

1 INTRODUCTION

Online optimization studies sequential decision-making with uncertainty about the future and under adversarial inputs. In many cases, randomization can substantially improve expected performance guarantees of online algorithms. However, it also introduces variability in solution quality and, consequently, the possibility of poor outcomes on individual runs. Typically, the choices in online optimization are irrevocable. An online algorithm cannot in general be rerun, and we cannot pick its best realization afterwards. Hence, an algorithm with a favorable expected performance guarantee may still exhibit a significant tail risk, i.e., a significant probability that its performance is far from the expected value. Controlling such risk is an important objective and has been studied in various fields of online optimization, including expert advice model (Even-Dar et al., 2006), multi-armed bandits (Sani et al., 2012; Galichet et al., 2013; Agrawal et al., 2021; Ayyagari & Dukkipati, 2023), Markov decision processes (Howard & Matheson, 1972; Geibel & Wysotzki, 2005; Fei et al., 2020; Ghosh & Moharrami, 2025), portfolio selection (Uziel & El-Yaniv, 2018), and submodular optimization (Soma & Yoshida, 2023).

Competitive analysis is a standard measure of performance for online algorithms (Borodin & El-Yaniv, 1998). We call a (randomized) online algorithm ρ-*competitive* if, for every input, its expected cost is at most ρ times the optimal cost with hindsight (see Definition 5 for the formal statement). Recently, risk control has been incorporated into the competitive analysis of randomized online algorithms, taking into account not only the expected performance of an algorithm but also the tail of its cost distribution.

This risk control took various forms. Christianson et al. (2024) measured the expected ratio on a fraction of worst outcomes. Their notion, called conditional value-at-risk (CVaR) competitive ratio, was applied to ski rental and one-max search, and has been later extended to the variant of the Bahncard problem with infinite discount duration by Himmelreich et al. (2026). A slightly different approach was introduced by Dinitz et al. (2024). They studied the expected competitive ratio of a randomized algorithm under the constraint that this ratio exceeds a prescribed threshold with at most a given probability. With the requirement of this probability being zero, we get so-called *pure tail constraints*, which are the focus of this paper. This framework was later extended to two-slope

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ski rental by Cui & Dinitz (2025). See Christianson et al. (2024) for a more detailed comparison of CVaR and tail-constraint-based approaches.

1.1 Our Results and Technical Overview

In this paper, we study *pure tail constraints* in the competitive analysis of three fundamental online optimization problems: *online bidding*, *line search*, and *TCP acknowledgment*. Informally, we call an algorithm γ*-dependable* if, for every input and every realization of its internal randomness, its performance is within a factor γ of the offline optimum with hindsight (see Definition 5 for the formal statement). By definition, γ cannot be smaller than the optimal deterministic competitive ratio. Moreover, as γ tends to infinity, the constraint becomes vacuous, and we expect to recover the optimal randomized competitive ratio. We investigate the tradeoff between these two extremes. For the classical ski rental problem, Dinitz et al. (2024) characterized the Pareto-optimal frontier of this tradeoff. We extend this investigation to three other canonical online problems.

Non-adaptive problems. We start with non-adaptive problems where the feedback given to an algorithm is just an indicator whether the algorithm has succeeded or not. *Online bidding* is a fundamental online problem that captures the challenge of searching for an unknown target value (Chrobak & Kenyon-Mathieu, 2006). An algorithm submits a sequence of bids until one reaches or exceeds the target, where the cost is defined as the sum of all submitted bids, including the final one. The optimal deterministic and randomized competitive ratios for this problem are 4 and e, respectively (Chrobak et al., 2008). Solutions for online bidding have been used as building blocks in algorithms and lower bounds for a variety of problems ranging from machine scheduling (Hall et al., 1997; Ebenlendr & Sgall, 2009; Epstein et al., 2010) and strip packing (Ye et al., 2011), through multi-level aggregation (Bienkowski et al., 2016) and minimum latency tours (Goemans & Kleinberg, 1998; Charikar et al., 2004), to clustering (Chrobak et al., 2008). In this paper, we characterize the Pareto-optimal frontier of the tradeoff between dependability and expected competitiveness for online bidding as follows (see Fig. 1a):

Theorem 1. *Fix $r \in [2, e]$ and let $\gamma(r) := \frac{r^2}{r-1}$. There exists a $\gamma(r)$-dependable randomized algorithm for online bidding whose expected competitive ratio is $\frac{r}{\ln r}$. Conversely, every $\gamma(r)$-dependable randomized algorithm for online bidding has expected competitive ratio at least $\frac{r}{\ln r}$.*

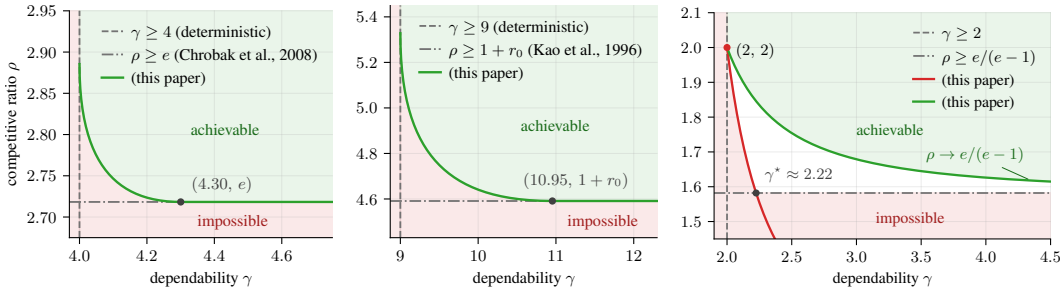
The dependability $\gamma(r)$ is always at least 4. When $r = e$, we have $\gamma(r) = e^2/(e-1) \approx 4.3$, and the expected competitive ratio reaches the unconstrained randomized optimum e.

The upper bound is an adaptation of the algorithm of Chrobak et al. (2008); our main contribution here is the lower bound. We use Yao's min-max principle and analyze a $\gamma(r)$-dependable deterministic algorithm against a random threshold. Its expected ratio is a function of the ratios between consecutive bids, and our main technical tool (Lemma 9) shows that $\gamma(r)$-dependability constrains these ratios. Roughly speaking, the algorithm cannot do substantially better than the geometric sequence in which every bid is r times larger than the previous one, up to lower-order terms. Details can be found in Section 3.

Next, we extend our results on online bidding to *line search* (Baeza-Yates et al., 1993). In this classic problem, an algorithm must locate a target at an unknown position on the line while minimizing the total distance traveled before finding it. Line search is closely related to online bidding (Chrobak & Kenyon-Mathieu, 2006), and based on this connection, we also provide the Pareto-optimal dependability-competitiveness frontier for this problem as follows (see Fig. 1b and Section 4 for details).

Theorem 2. *Fix $r \in [2, r_0]$, where $r_0 \approx 3.591$ is the solution of $r = \frac{1+r}{\ln r}$, and let $\gamma(r) := 1 + \frac{2r^2}{r-1}$. There exists a $\gamma(r)$-dependable randomized algorithm for line search whose expected competitive ratio is $1 + \frac{1+r}{\ln r}$. Conversely, every $\gamma(r)$-dependable randomized algorithm for line search has expected competitive ratio at least $1 + \frac{1+r}{\ln r}$.*

An adaptive problem. The final problem we consider in this paper is *TCP acknowledgment* (Dooly et al., 2001), in which packets sequentially arrive, and an algorithm must acknowledge each packet after its arrival. Each acknowledgment clears all pending (unacknowledged) packets. The objective is to minimize the sum of the number of acknowledgments sent and the total latency



(a) Online bidding, Theorem 1. (b) Line search, Theorem 2. (c) TCP ack., Theorems 3 and 4.

Figure 1: The dependability-competitiveness tradeoffs for online bidding, line search, and TCP acknowledgment. The curves separate the achievable pairs from the unachievable ones. For TCP acknowledgment, the two bounds do not coincide: every pair above the upper curve is achieved by the algorithm of Theorem 3, and no pair below the lower curve is achievable by Theorem 4. The two curves meet at the point $(2, 2)$, and our lower bound improves on the randomized one of $e/(e-1)$ for $\gamma < \gamma^\star \approx 2.22$.

incurred by all packets before they are acknowledged. The optimal deterministic competitive ratio is 2 (Dooly et al., 2001). By exploiting a connection to ski rental, Karlin et al. (2003) subsequently obtained a randomized algorithm with competitive ratio $e/(e-1)$, while the matching lower bound was established by Seiden (2000). The problem is also known as *lot sizing*, and has been extensively studied in the operations research literature; see, e.g., a book by Kimms (1997).

Before we describe our results, we emphasize a key difference between TCP acknowledgment and the previous two problems. In online bidding and line search, an algorithm is necessarily *non-adaptive*: the feedback available to an algorithm is essentially limited to whether the process has terminated; once the goal is reached, no further decisions are required. In contrast, in TCP acknowledgment, a deterministic algorithm can be *adaptive*: it can observe packets as they arrive and may adjust its future acknowledgments to them. This adaptivity makes the analysis of TCP acknowledgment more challenging than that of online bidding and line search. We emphasize that *all problems studied so far in CVaR and tail-constraint-based frameworks were not adaptive*: ski rental, one-max search, two-slope ski rental, and Bahncard with infinite discount duration. Our work is thus the first to study pure tail constraints for an adaptive online problem.

Inspired by the connection identified by Karlin et al. (2003), we construct an algorithm for TCP acknowledgment that attains the same dependability-competitiveness tradeoff as the ski rental algorithm of Dinitz et al. (2024) (see Fig. 1c):

Theorem 3. *Fix $r \in (0, 1]$ and let $\gamma(r) := 1 + \frac{1}{r}$. There exists a $\gamma(r)$-dependable randomized algorithm for TCP acknowledgment whose expected competitive ratio is $1 + \frac{e^r}{(1+r)\cdot e - e^r}$.*

Adaptivity makes a matching lower bound for TCP acknowledgment much harder to obtain. An algorithm plays multiple rounds against the adversary, and it can adapt its future decisions to the packets observed so far. Nevertheless, we establish a nontrivial lower bound on the achievable tradeoff (see also Fig. 1c):

Theorem 4. *Fix $r \in [\frac{\sqrt{5}-1}{2}, 1]$ and let $\gamma(r) := 1 + \frac{1}{r}$. Every $\gamma(r)$-dependable randomized algorithm for TCP acknowledgment has expected competitive ratio at least $\rho(r) = 1 + \frac{(2+r)\cdot r^{2+r}}{(1+r)^2 - r^{2+r}}$.*

To prove this bound, we extend the construction of Seiden (2000). The adversary issues bursts of packets, each much larger than all the preceding ones together, and decides on the next burst only after seeing the algorithm acknowledge the previous one. We show that $\gamma(r)$-dependability forces structural properties on every deterministic algorithm. For example, it cannot acknowledge the first burst too early. This restricts the support of the randomized algorithms we need to consider. We then construct a probability distribution over instances with at most two bursts and apply Yao's min-max principle to this restricted class.

2 PRELIMINARIES

All problems which we study are online cost minimization problems: an instance σ is revealed to an algorithm over time, and the algorithm must react to the revealed parts without any knowledge about the future. We write $\text{OPT}(\sigma)$ for the optimal offline cost, $\text{DET}(\sigma)$ for the cost of a deterministic online algorithm DET on instance σ, and

$$R(\text{DET}, \sigma) := \frac{\text{DET}(\sigma)}{\text{OPT}(\sigma)}$$

for the solution quality of DET on σ.

Definition 5. *A randomized algorithm is a probability distribution over deterministic algorithms; we denote such an algorithm by* RAND *and its support by* $\mathsf{supp}(\text{RAND})$. *It is* ρ-competitive[1] *if* $\mathbb{E}_{\text{DET}\sim\text{RAND}}[R(\text{DET}, \sigma)] \leq \rho$ *for every instance* σ, *and* γ-dependable *if* $R(\text{DET}, \sigma) \leq \gamma$ *for every* $\text{DET} \in \mathsf{supp}(\text{RAND})$ *and every instance* σ.

Note that γ-dependability corresponds to the pure tail constraint $(\gamma, 0)$ of Dinitz et al. (2024): the competitive ratio never exceeds γ, regardless of random choices of RAND.

Naming convention. Note that for a deterministic algorithm, the notions of competitiveness and dependability coincide. To avoid confusion, *we use dependability whenever we deal with deterministic algorithms*, and to emphasize the distinction we use the term *expected competitiveness* when we deal with the competitiveness of randomized algorithms. Throughout the paper, DET stands for a deterministic algorithm, RAND for a randomized one, σ for an instance, and π for a probability distribution over instances. A parameter on which an algorithm depends is written as a subscript.

Min-max principle. In our lower bounds, we show that, for fixed γ and ρ, no γ-dependable randomized algorithm has expected competitive ratio better than ρ. To this end, we use a slightly modified version of Yao's min-max principle (Yao, 1977), stated below. Its proof is given in Section B for the sake of completeness.

Lemma 6 (Min-max principle)**.** *Let* $\mathcal{D}^\star \subseteq \mathcal{D}$ *be two sets of deterministic algorithms and let* π *be a probability distribution over instances. Assume that* $\mathbb{E}_{\sigma\sim\pi}[R(\text{DET}, \sigma)] \geq \rho$ *for every deterministic algorithm* $\text{DET} \in \mathcal{D}$. *Then, every randomized algorithm* RAND *with* $\mathsf{supp}(\text{RAND}) \subseteq \mathcal{D}^\star$ *has expected competitive ratio at least* ρ.

In most applications of this lemma, $\mathcal{D}^\star$ is the set of all γ-dependable deterministic algorithms and $\mathcal{D}$ is the set of all deterministic algorithms satisfying a certain property implied by γ-dependability. The randomized algorithms in the lemma are then exactly the γ-dependable randomized algorithms.

2.1 PROBLEM DEFINITIONS

All three problems below are classic online problems. The first two problems (online bidding and line search) have a single unknown parameter that describes both the instance and the optimal solution, and are in this sense similar to the ski rental problem. The third problem (TCP acknowledgment), while having a rent-or-buy structure similar to that of the ski rental problem, has more complex instances.

Online bidding. In *online bidding* (see, e.g., Chrobak et al., 2008), an adversary fixes an unknown threshold $T \geq 1$.[2] The algorithm submits an increasing sequence of bids $x_1 < x_2 < \ldots$; a bid is *successful* if it is at least T. The process stops at the first successful bid, and the algorithm pays the sum of all submitted bids. The optimal offline cost is T, achieved by the single bid T, so an algorithm whose first successful bid is x_k has solution quality $(\sum_{i\leq k} x_i)/T$.

[1] This is so-called *strict competitiveness*; for a non-strict one, see Section B.

[2] In the literature, the condition $T \geq 1$ is sometimes dropped. In such a case, one either allows non-strict competitiveness or defines the bids as a bi-infinite sequence that may start with an infinitesimally small value.

Line search. In *line search*, also known as the *cow-path problem* (see, e.g., Baeza-Yates et al., 1993), a searcher (a cow called Bessie) starts at the meeting point of two semi-infinite paths. It must find a target (a gate to a grazing field) placed at an unknown distance $T \geq 1$ from the start, on one of the two paths. The searcher learns the location of the target only upon reaching it, and pays the total distance traveled. The optimal offline cost is T, achieved by walking directly to the target.

TCP acknowledgment. In *TCP acknowledgment* (Dooly et al., 2001), an instance is a finite set of packets; packet j arrives at time $a_j \geq 0$ and the algorithm learns about it only then. At any moment, the algorithm may send an acknowledgment. This costs 1 and clears all pending packets. The cost of a solution is the number of acknowledgments plus the total latency $\sum_j$(acknowledgment time of packet $j - a_j$), so the algorithm has to trade acknowledgments against latency.

Related work on these and similar problems is discussed in Section A.

3 Online Bidding

In this section, we discuss Theorem 1. Its proof along with the proofs of the lemmas stated below is given in Section D; here we focus on the main ideas.

We fix $r \in [2, e]$ and let $\gamma(r) := r^2/(r-1)$. While our main contribution in this part is the lower bound, we start with a conceptually simpler upper bound. It is attained by the parametrized version of the optimal randomized algorithm by Chrobak et al. (2008), defined below.

> *Algorithm* $\textsc{Bid}_r$*: first, choose a value* $\xi \in [0, 1)$ *uniformly at random. Next, submit the increasing bids* $r^\xi, r^{\xi+1}, r^{\xi+2}, \ldots$ *until the first successful one.*

Lemma 7. *Randomized algorithm* $\textsc{Bid}_r$ *is* $\gamma(r)$*-dependable and its expected competitive ratio is at most* $r/\ln r$*.*

In the proof of the above lemma, we argue that the solution quality of $\textsc{Bid}_r$ is at most $r^{\zeta+1}/(r-1)$, where $\zeta \in [0, 1)$ is the fractional part of $\log_r$ of the last bid, which is distributed uniformly. This gives us both the expected competitive ratio and the dependability, depending on whether we take the expectation over ζ or its worst case.

For the lower bound, we use the min-max principle (Lemma 6) and construct an appropriate probability distribution over instances.

Lemma 8. *Fix* $\delta > 0$*. There exists a probability distribution* $\pi(\delta)$ *over the inputs, such that every* $\gamma(r)$*-dependable algorithm* DET *satisfies* $\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det}, \sigma)] \geq \frac{r}{\ln r} - \delta$*.*

The constructed probability distribution $\pi(\delta)$ is supported on thresholds T from the range $[1, U]$ for a sufficiently large U, and we write v_t for its density at t. Any $\gamma(r)$-dependable deterministic algorithm DET for such instances can be fully described by an increasing sequence of bids $x_1 < x_2 < \ldots$, all from range $[1, U]$; we also set $x_0 := 1$. DET places bid x_i if all its previous bids were unsuccessful, which happens with probability $\int_{x_{i-1}}^{U} v_t \, dt$, and the expected contribution of this bid to the solution quality is $x_i \cdot \int_{x_{i-1}}^{U} (v_t/t) \, dt$. To simplify the latter term, we choose $v_t = 1/(t \cdot \ln U)$, which allows us to eventually obtain

$$\mathbb{E}_{\sigma\sim\pi}[R(\textsc{Det}, \sigma)] = \frac{1}{\ln U} \cdot \sum_{i=1}^{n} \frac{x_i}{x_{i-1}} - O\left(\frac{1}{\ln U}\right). \tag{1}$$

To lower-bound (1), we observe that after the k-th bid, the total cost of DET is $\sum_{i=1}^{k} x_i$, while the adversary can terminate the instance choosing a threshold T arbitrarily close to x_{k-1}. Since the algorithm is $\gamma(r)$-dependable, this imposes the following constraints:

$$\sum_{i=1}^{k} x_i \leq \gamma(r) \cdot x_{k-1} = \frac{r^2}{r-1} \cdot x_{k-1} \qquad \text{for every } k \in \{1, \ldots, n\}.$$

The technical core of this part is the *step ratio lemma* stated below, which we use to provide a lower bound on (1). It shows that, up to lower-order terms, (1) is minimized when all consecutive ratios x_i/x_{i-1} equal r. We prove the lemma in Section C. Since we reuse it for line search, we state it in a generalized form; for online bidding, we set $c = 0$.

Lemma 9 (Step ratios). *Let $1 = x_0, x_1, \ldots, x_n$ be a sequence of positive reals. Fix $c \geq 0$ and let ρ_c denote the unique solution of $\rho \cdot \ln \rho = c + \rho$. Fix a real $r \in [2, \rho_c]$ and $\varepsilon \in (0, \frac{c+r}{\ln r})$. Let $S_k := \sum_{i=1}^{k} x_i$, and assume that*

$$S_k \leq \frac{r^2}{r-1} \cdot x_{k-1} \qquad \text{for every } k \in \{1, \ldots, n\}.$$

There is a constant F, depending on c, r and ε only, such that

$$c \cdot n + \sum_{i=1}^{n} \frac{x_i}{x_{i-1}} \geq \left(\frac{c+r}{\ln r} - \varepsilon \right) \cdot \ln x_n - F.$$

4 Line Search

In this section we focus on presenting ideas for proving Theorem 2; the complete proof is deferred to Section E. Let r_0 be defined as in the theorem, i.e., as the solution to $r = (1+r)/\ln r$. Throughout this section, we fix $r \in [2, r_0]$ and let $\gamma(r) := 1 + 2 \cdot r^2/(r-1)$.

The upper bound of Theorem 2 is attained by the algorithm of Kao et al. (1996) called SmartCow, parametrized by a number $r \in (1, +\infty)$. Kao et al. (1996) show that SmartCow$_r$ is $(1 + (1 + r)/\ln r)$-competitive. We complement this result by showing that it is also $\gamma(r)$-dependable for $r \in [2, r_0]$, and thus achieves the tradeoff of Theorem 2. The proof is deferred to Section E.

We obtain the lower bound from the min-max principle of Lemma 6 and the lemma below. Its proof is deferred to Section E, and we provide the main ideas here.

Lemma 10. *Fix $\delta > 0$. There exists a probability distribution $\pi(\delta)$ over the inputs, such that every $\gamma(r)$-dependable algorithm* Det *satisfies $\mathbb{E}_{\sigma \sim \pi(\delta)}[R(\text{Det}, \sigma)] \geq 1 + \frac{1+r}{\ln r} - \delta$.*

Similarly to the proof of Lemma 8, probability distribution $\pi(\delta)$ is defined only on values from the interval $[1, U]$ on both paths, and the probability density of the target T at any $t \in [1, U]$ is $v_t = 1/(2t \cdot \ln U)$. Any deterministic $\gamma(r)$-dependable algorithm Det can be described by

- a sequence of numbers $x_1, x_2, \ldots$ from $[1, U]$ satisfying $x_1 < x_3 < x_5 < \ldots$ and $x_2 < x_4 < x_6 < \ldots$, and
- a starting direction.

We also set $x_0 := 1$. Det first moves to points $x_1, x_2, x_3, \ldots$ on alternating paths, starting with x_1 in the chosen starting direction, until it finds the target.

If Det visits k of these points and finds the target T on its way to the $(k+1)$-st point, its total cost is $T + 2 \cdot \sum_{i=1}^{k} x_i$. The algorithm arrives at x_i if and only if T is on the same path and larger than x_i, or on the other path and larger than x_{i-1}; the probability of this event is $\int_{x_{i-1}}^{U} v_t \, dt + \int_{x_i}^{U} v_t \, dt$. This implies a bound similar to (1), albeit having extra terms because of T in the cost and the second integral.

Note that the cost of Det finding the target after visiting k points is $T + 2 \cdot \sum_{i=1}^{k} x_i$, and the adversary can choose T arbitrarily close to x_{k-1}. Since Det is $\gamma(r)$-dependable, it has to satisfy the following property on its partial cost:

$$x_{k-1} + 2 \cdot \sum_{i=1}^{k} x_i \leq \gamma(r) \cdot x_{k-1} = \left(1 + \frac{2r^2}{r-1} \right) \cdot x_{k-1} \qquad \text{for every } k \geq 1.$$

This is equivalent to the assumption required in Lemma 9, which we may use (this time with $c = 1$) to prove Lemma 10.

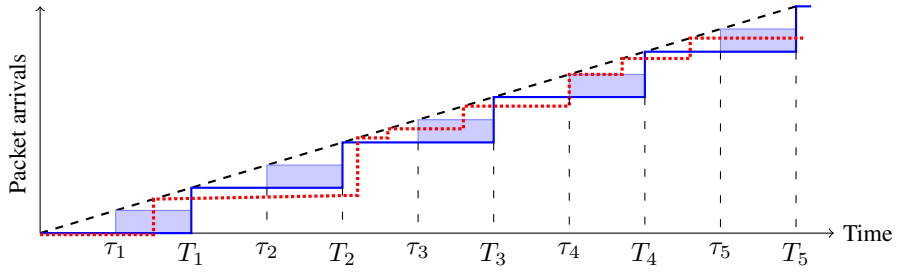


Figure 2: The latency and the acknowledgments of OPT (red dotted line) and of DET_z (blue solid line). The sequence of arrivals σ is depicted as the dashed diagonal function. The filled rectangles are the latencies that caused the acknowledgments of DET_z. Each filled rectangle has area z.

5 Upper Bound for TCP Acknowledgment

In this section, we show the main ideas behind the proof of Theorem 3; the formal arguments are deferred to Section F. To this end, we define an algorithm RAND_r, parametrized by $r \in (0, 1]$. This algorithm is inspired by that of Karlin et al. (2003), where we modify the probability distribution to guarantee the dependability of $\gamma(r) = (1 + 1/r)$.

5.1 Dependability

We start by defining a family of deterministic algorithms $\{\text{DET}_z\}_{z\in(0,1]}$ introduced by Karlin et al. (2003). Fix $z \in (0, 1]$, and let $T_0 := 0$ denote the very start of the execution. For $i \in \mathbb{Z}_{>0}$, DET_z performs the i-th acknowledgment at time T_i inductively defined as the first time for which there exists a time $\tau_i \in [T_{i-1}, T_i]$ such that

$$P(T_{i-1}, \tau_i) \cdot (T_i - \tau_i) = z, \tag{2}$$

where $P(T, T')$ denotes the number of packets that have arrived in time interval $(T, T']$ for any time points $T \leq T'$.

To provide intuitions behind DET_z, Fig. 2 shows an illustration of the behaviors of DET_z and OPT, adapted from Karlin et al. (2003). The black dashed line depicts the number of packets that have arrived so far, simplified to a straight line. The blue solid line and the red dotted line represent the numbers of packets acknowledged by DET_z and OPT, respectively, where every vertical segment of these lines corresponds to an acknowledgment of the respective algorithm.

Note that the i-th acknowledgment of DET_z at time T_i is performed only when the unacknowledged packets until time τ_i together incur latency z in time interval $[\tau_i, T_i]$, as instructed by the trigger condition (2). This latency cost is represented in Fig. 2 by a blue filled rectangle with area exactly z. Observe that these rectangles are disjoint, and the i-th rectangle touches the black dashed line at time τ_i.

We show the dependability of DET_z:

Lemma 11. *For every $z \in (0, 1]$, DET_z is $\gamma(z)$-dependable.*

The proof is twofold. First, we show that OPT must incur at least cost z between any two consecutive acknowledgments by DET_z, implying that the total number of acknowledgments by DET_z is upper-bounded within a factor $1/z$ from the optimal cost. Second, we prove that, for any set of packets acknowledged at the same time by OPT, the *additional* latency cost incurred on them by DET_z cannot exceed z by the definition of DET_z. Hence, the total latency cost of DET_z is bounded by the total cost of the optimal solution. These bounds together imply the lemma.

5.2 Competitive Algorithm

Since DET_z is $\gamma(r)$-dependable for any $z \in [r, 1]$, any randomized algorithm supported by $\{\text{DET}_z\}_{z\in[r,1]}$ is also $\gamma(r)$-dependable. Hence, we aim to attain a probability distribution over $\{\text{DET}_z\}_{z\in[r,1]}$ minimizing the expected competitive ratio, resulting in the following algorithm.

Algorithm RAND_r*: draw a value* $z \in [r, 1]$ *from a distribution that has*

- *a discrete mass of* $q_r := \frac{r \cdot e^r}{(r+1) \cdot e - e^r}$ *at* $z = r$*, and*
- *a density* $p(z) := \frac{r+1}{(r+1) \cdot e - e^r} \cdot e^z$ *for any* $z \in (r, 1]$*,*

and run DET_z*.*

Lemma 12. RAND_r *has expected competitive ratio* $1 + \frac{e^r}{(1+r) \cdot e - e^r}$.

Note that, together with Lemma 11, this lemma implies Theorem 3. To prove this lemma, we exploit the properties derived from Karlin et al. (2003).

6 Lower Bound for TCP Acknowledgment: Proof Overview

In this section, we describe the main ideas behind the proof of Theorem 4; the full proof is given in Section G. Throughout this section, we fix $r \in [(\sqrt{5} - 1)/2, 1]$ and write $\gamma(r) := 1 + 1/r$ and $c := r^2 + r - 1 \in [0, 1]$.

We recall that the deterministic algorithm for the problem can be adaptive: it observes packets as they arrive and may adjust its future acknowledgments to them. Such an algorithm is thus a function which maps the choices of the adversary made so far to the delay of the next acknowledgment. Therefore, we analyze the consequences of $\gamma(r)$-dependability by playing *multiple rounds* against the algorithm. In each round, the adversary waits for the algorithm to acknowledge, observes the latency at which this happens, and only then it decides on the next part of the input.

Structural consequences of dependability. In the first part of the proof (Section G.1), we study a deterministic algorithm DET and identify properties which every $\gamma(r)$-dependable algorithm has to satisfy. Following Seiden (2000), we run DET on instances consisting of *bursts* of packets, each much larger than all the preceding ones together. We call such instances *flat*. The latency of a burst is the total latency accrued by its packets so far. The adversary issues burst $i+1$ when burst i reaches latency y_i, and we write x_i for the latency at which DET acknowledges burst i. On flat instances, the latency accrued between two bursts is essentially caused by the last burst only, and the solution that acknowledges once, at the last burst, has cost close to $1 + \sum_i y_i$.

Right after DET acknowledges burst i, we define its *post-ack budget* Φ_i^+ as $\gamma(r)$ times the cost of the solution that acknowledges once, at this very moment, minus the cost that DET has paid so far. On flat instances, the former cost is close to $1 + \sum_{j<i} y_j + x_i$. The definition reflects a possible aggressive move of the adversary: it may issue burst $i + 1$ right after DET's acknowledgment of burst i. Then, Φ_i^+ is the amount that $\gamma(r)$-dependability still allows DET to spend on the future bursts. As the instance may also end at this moment, it holds that $\Phi_i^+ \geq 0$.

The penalizing suffix. The main ingredient of this part is a threat that the adversary may pose after any acknowledgment of DET. The adversary may append a *penalizing suffix*, i.e., a sequence of bursts, each issued immediately after DET acknowledges the previous one. In each round m of this suffix, DET pays $1 + x_m$ for its acknowledgment, while the cost of the single-acknowledgment solution grows by x_m only. Thus, roughly speaking, the post-ack budgets in consecutive rounds satisfy

$$\begin{aligned} \Phi_m^+ &\leq \Phi_{m-1}^+ + \gamma(r) \cdot x_m - (1 + x_m) \\ &= \Phi_{m-1}^+ + (\gamma(r) - 1) \cdot x_m - 1 \\ &\leq \gamma(r) \cdot (\Phi_{m-1}^+ - 1), \end{aligned}$$

where the last inequality uses $x_m \leq \Phi_{m-1}^+ - 1$, or equivalently, $1 + x_m \leq \Phi_{m-1}^+$, which is a consequence of the $\gamma(r)$-dependability of DET.

That is, the post-ack budget of DET evolves according to a mapping $z \mapsto \gamma(r) \cdot (z - 1)$, and $1 + r = \gamma(r)/(\gamma(r) - 1)$ is the fixed point of this mapping. Hence, if the post-ack budget of DET drops below $1 + r$, it remains below $1 + r$, and moreover, we may show that the gap to the fixed point grows geometrically. This means that the post-ack budget of DET eventually becomes negative, which contradicts the $\gamma(r)$-dependability of DET. Lemma 27 formalizes this threat: after acknowledging any burst, a $\gamma(r)$-dependable algorithm must keep its post-ack budget at least $1 + r$.

Constraints from all bursts. By expanding the definition of the post-ack budget, we turn Lemma 27 into a family of inequalities which relate the cost of DET after burst i to its cost after burst $i-1$ (cf. Lemma 22). In particular, for a single burst, $\Phi_1^+ = (\gamma(r)-1)\cdot(1+x_1)$, and thus $1+x_1 \geq (1+r)/(\gamma(r)-1) = r\cdot(1+r) = 1+c$, i.e., the latency x_1 of the first burst is at least c. Moreover, if the adversary issues a large second burst at latency $y > x_1$, then DET pays at least $(1+r)\cdot(1+x_1+r-y)$, up to an arbitrarily small error term.

Adversarial distribution. In the second part of the proof (Section G.2), we study the behavior of a deterministic algorithm DET satisfying the properties above on a random instance. Our instances consist of one or two bursts. With probability q, the instance is a single packet; otherwise a second, large burst arrives at a random latency $y \in [c, r]$. We choose the distribution so that the expected competitive ratio of DET does not depend on x_1, and the min-max principle of Lemma 6 yields Theorem 4.

We note that Lemma 22 provides constraints for instances with any number of bursts, while our distribution uses only two of them. We believe that longer instances are a natural route towards closing the gap to Theorem 3.

7 CONCLUSIONS

In this paper, we studied pure tail constraints in competitive analysis, i.e., the tradeoff between the dependability of a randomized online algorithm and its expected competitive ratio.

For online bidding and line search, we determined the Pareto-optimal frontier of this tradeoff. In both problems, the optimal tradeoff is attained by known randomized algorithms, parametrized by the growth ratio r of consecutive steps. Our main contributions here are the lower bounds: we showed that dependability essentially forces a deterministic algorithm to follow a geometric sequence with ratio r.

For TCP acknowledgment, we gave an algorithm which achieves the tradeoff known for ski rental, and we complemented it with a lower bound that beats the randomized one of $e/(e-1)$ for dependability ratio smaller than $\gamma^\star \approx 2.22$. We do not know where the true frontier for TCP acknowledgment lies between the two curves of Fig. 1c, and closing this gap is the main problem left open by our work. Our lower bound uses instances with at most two bursts only, while dependability constrains an algorithm also on longer instances, and we believe this leaves room for a stronger bound.

AI USE STATEMENT

In this work, we used a generative AI tool (ChatGPT 5.6 Sol) for three tasks.

- First, we used it to produce the Python code that plots the dependability-competitiveness tradeoff curves in Fig. 1. We verified the produced code manually.
- Second, we used it to find the probability distribution over instances used in Section G.2, chosen so that the expected competitive ratio of an algorithm does not depend on the latency at which it acknowledges the first burst. We proved all properties of this distribution manually; the proofs are given in Section G.2.
- Conversations with it were used as an inspiration for some parts of the proof of Lemma 11. The entire proof was written and verified by hand.

We did not use generative AI tools for any other task with a required disclosure. We take full responsibility for the final content of this work.

REFERENCES

Shubhada Agrawal, Wouter M. Koolen, and Sandeep Juneja. Optimal best-arm identification methods for tail-risk measures. In *Proc. 34th Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 25578–25590, 2021.

Sara Ahmadian, Shuchi Chawla, Ravi Kumar, Manish Purohit, and Shirley Zhang. Multi-level aggregation via dual fitting: An $O(D)$-competitive algorithm. arXiv:2608.04258, 2026.

Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. A regression approach to learning-augmented online algorithms. In *Proc. 34th Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 30504–30517, 2021.

Spyros Angelopoulos and Bertrand Simon. Learning-augmented online bidding in stochastic settings. In *Proc. 38th Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 75342–75372, 2025.

Spyros Angelopoulos, Marcin Bienkowski, Christoph Dürr, and Bertrand Simon. Contract scheduling with distributional and multiple advice. In *Proc. 33rd Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 3652–3660, 2024a.

Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online computation with untrusted advice. *Journal of Computer and System Sciences*, 144:103545, 2024b.

Ranga Shaarad Ayyagari and Ambedkar Dukkipati. Risk-averse combinatorial semi-bandits. In *Proc. 2023 IEEE Int. Symp. on Information Theory (ISIT)*, pp. 1472–1477, 2023.

Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, 1993.

Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Proc. 33rd Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 20083–20094, 2020.

Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak, Neil Dobbs, Tomasz Nowicki, Maxim Sviridenko, Grzegorz Świrszcz, and Neal E. Young. Approximation algorithms for the joint replenishment problem with deadlines. *Journal of Scheduling*, 18(6):545–560, 2015.

Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukáš Folwarczný, Łukasz Jeż, Jiří Sgall, Nguyen Kim Thang, and Pavel Veselý. Online algorithms for multi-level aggregation. In *Proc. 24th European Symp. on Algorithms (ESA)*, pp. 12:1–12:17, 2016.

Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

Niv Buchbinder, Tracy Kimbrel, Retsef Levi, Konstantin Makarychev, and Maxim Sviridenko. Online make-to-order joint replenishment model: Primal-dual competitive algorithms. *Operations Research*, 61(4):1014–1029, 2013. doi: 10.1287/opre.2013.1188.

Sergio Cabello and Panos Giannopoulos. Searching in Euclidean spaces with predictions. *Theory of Computing Systems*, 70(3):49, 2026.

Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417–1440, 2004. doi: 10.1137/S0097539702418498.

Nicolas Christianson, Bo Sun, Steven Low, and Adam Wierman. Risk-sensitive online algorithms (extended abstract). In *Proc. 37th Conf. on Learning Theory (COLT)*, pp. 1140–1141, 2024.

Marek Chrobak and Claire Kenyon-Mathieu. Competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.

Marek Chrobak and Claire Kenyon-Mathieu. SIGACT news online algorithms column 10: Competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.

Marek Chrobak, Claire Kenyon, John Noga, and Neal E. Young. Incremental medians via online bidding. *Algorithmica*, 50(4):455–478, 2008. doi: 10.1007/s00453-007-9005-x.

Qiming Cui and Michael Dinitz. Controlling tail risk in two-slope ski rental. In *Proc. 23rd Workshop on Approximation and Online Algorithms (WAOA)*, pp. 81–96, 2025.

Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Controlling tail risk in online ski-rental. In *Proc. 35th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2024.

Daniel R. Dooly, Sally A. Goldman, and Stephen D. Scott. On-line analysis of the TCP acknowledgment delay problem. *Journal of the ACM*, 48(2):243–273, 2001.

Tomáš Ebenlendr and Jirí Sgall. Optimal and online preemptive scheduling on uniformly related machines. *Journal of Scheduling*, 12(5):517–527, 2009. doi: 10.1007/S10951-009-0119-7.

Leah Epstein, Asaf Levin, Alberto Marchetti-Spaccamela, Nicole Megow, Julián Mestre, Martin Skutella, and Leen Stougie. Universal sequencing on a single machine. In *Proc. 14th Int. Conf. on Integer Programming and Combinatorial Optimization (IPCO)*, pp. 230–243, 2010. doi: 10.1007/978-3-642-13036-6_18.

Eyal Even-Dar, Michael Kearns, and Jennifer Wortman. Risk-sensitive online learning. In *Proc. 17th Int. Conf. on Algorithmic Learning Theory (ALT)*, pp. 199–213, 2006.

Yingjie Fei, Zhuoran Yang, Yudong Chen, Zhaoran Wang, and Qiaomin Xie. Risk-sensitive reinforcement learning: Near-optimal risk-sample tradeoff in regret. In *Proc. 33rd Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 22384–22395, 2020.

Nicolas Galichet, Michele Sebag, and Olivier Teytaud. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Proc. 5th Asian Conf. on Machine Learning (ACML)*, pp. 245–260, 2013.

Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

Arnob Ghosh and Mehrdad Moharrami. Online learning in risk sensitive constrained MDP. In *Proc. 42nd Int. Conf. on Machine Learning (ICML)*, pp. 19406–19425, 2025.

Michel X. Goemans and Jon M. Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998. doi: 10.1007/BF01585867.

Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22(3):513–544, 1997. doi: 10.1287/moor.22.3.513.

Lukas Himmelreich, Nicolas Christianson, and Adam Wierman. Risk-sensitive peak-aware energy scheduling: Competitive and learning-augmented algorithms. In *Proc. 17th ACM Int. Conf. on Future and Sustainable Energy Systems (e-Energy)*, pp. 240–271, 2026.

Ronald A. Howard and James E. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.

Sungjin Im, Benjamin Moseley, Chenyang Xu, and Ruilong Zhang. Online state exploration: Competitive worst case and learning-augmented algorithms. In *Proc. 2023 European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pp. 333–348, 2023.

Ming-Yang Kao, John H. Reif, and Stephen R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1): 63–79, 1996. doi: 10.1006/INCO.1996.0092.

Anna R. Karlin, Claire Kenyon, and Dana Randall. Dynamic TCP acknowledgement and other stories about e/(e - 1). *Algorithmica*, 36(3):209–224, 2003.

Alf Kimms. *Multi-Level Lot Sizing and Scheduling: Methods for Capacitated, Dynamic, and Deterministic Models*. Production and Logistics. Physica-Verlag, Heidelberg, 1997. ISBN 978-3-7908-0967-1. doi: 10.1007/978-3-642-50162-3.

Dennis Komm, Rastislav Královič, Richard Královič, and Tobias Mömke. Randomized online computation with high probability guarantees. *Algorithmica*, 84(5):1357–1384, 2022. doi: 10.1007/S00453-022-00925-Z.

Stefano Leonardi, Alberto Marchetti-Spaccamela, Alessio Presciutti, and Adi Rosén. On-line randomized call control revisited. *SIAM Journal on Computing*, 31(1):86–112, 2001. doi: 10.1137/S0097539798346706.

Bruce M. Maggs, Friedhelm Meyer auf der Heide, Berthold Vöcking, and Matthias Westermann. Exploiting locality for data management in systems of limited bandwidth. In *Proc. 38th IEEE Symp. on Foundations of Computer Science (FOCS)*, pp. 284–293, 1997.

Milena Mihail and Thorben Tröbst. Online matching with high probability. In *Proc. 17th Int. Symp. on Algorithmic Game Theory (SAGT)*, pp. 21–34, 2024. doi: 10.1007/978-3-031-71033-9_2.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Proc. 31st Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 9684–9693, 2018.

Amir Sani, Alessandro Lazaric, and Rémi Munos. Risk-aversion in multi-armed bandits. In *Proc. 25th Annual Conf. on Neural Information Processing Systems (NeurIPS)*, 2012.

Steven S. Seiden. A guessing game and randomized online algorithms. In *Proc. 32nd ACM Symp. on Theory of Computing (STOC)*, pp. 592–601, 2000. doi: 10.1145/335305.335385.

Yongho Shin, Changyeol Lee, Gukryeol Lee, and Hyung-Chan An. Improved learning-augmented algorithms and (tight) lower bounds for multi-option ski rental problem. *ACM Transactions on Algorithms*, 22(1):1–30, 2025.

Tasuku Soma and Yuichi Yoshida. Online risk-averse submodular maximization. *Annals of Operations Research*, 320(1):393–414, 2023.

Guy Uziel and Ran El-Yaniv. Growth-optimal portfolio selection under CVaR constraints. In *Proc. 21st Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pp. 48–57, 2018.

Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Proc. 33rd Annual Conf. on Neural Information Processing Systems (NeurIPS)*, pp. 8042–8053, 2020.

Andrew Chi-Chih Yao. Probabilistic computation: towards a uniform measure of complexity. In *Proc. 18th IEEE Symp. on Foundations of Computer Science (FOCS)*, pp. 222–227, 1977. doi: 10.1109/SFCS.1977.24.

Deshi Ye, Xin Han, and Guochuan Zhang. Online multiple-strip packing. *Theoretical Computer Science*, 412(3):233–239, 2011. doi: 10.1016/J.TCS.2009.09.029.

A Related Work

For online bidding, folklore deterministic and randomized algorithms based on geometric scaling achieve competitive ratios of 4 and e, respectively, which were later shown to be optimal by Chrobak et al. (2008). For line search, Baeza-Yates et al. (1993) proved the optimal deterministic competitive ratio of 9, while Kao et al. (1996) established the optimal randomized competitive ratio of approximately 4.591. The standard deterministic algorithms for online bidding and line search share the same underlying doubling structure; see Chrobak & Kenyon-Mathieu (2006) for an explicit connection. This relation, however, does not extend directly to randomized algorithms.

Dooly et al. (2001) introduced the TCP acknowledgment problem and established the optimal deterministic competitive ratio of 2. The optimal randomized competitive ratio is $e/(e-1)$: a lower bound was shown by Seiden (2000) and an algorithm was given by Karlin et al. (2003), who uncovered a connection to ski rental. TCP acknowledgment is the simplest problem in a hierarchy of online aggregation problems. Its two-level extension is the online joint replenishment problem, for which Buchbinder et al. (2013) gave a 3-competitive deterministic algorithm and proved a lower bound of approximately 2.64; the lower bound was later improved to approximately 2.754 (Bienkowski et al., 2015). Both problems are special cases of the multi-level aggregation problem on trees of depth D that admits $2D$-competitive deterministic algorithms (Ahmadian et al., 2026), where TCP acknowledgment and joint replenishment correspond to $D = 1$ and $D = 2$, respectively.

Problems studied in this paper have also been studied in the context of learning-augmented online optimization (Purohit et al., 2018; Wei & Zhang, 2020; Bamas et al., 2020; Anand et al., 2021; Im et al., 2023; Angelopoulos et al., 2024a;b; Shin et al., 2025; Angelopoulos & Simon, 2025; Cabello & Giannopoulos, 2026). In this framework, an online algorithm is given a prediction of unknown quality. A central goal is to characterize the tradeoff between *consistency*, the performance under accurate predictions, and *robustness*, the worst-case performance under arbitrary predictions. The consistency-robustness tradeoff resembles the tradeoff between expected competitiveness and dependability under pure tail constraints, but is of different nature. Consistency and robustness bound the performance of an algorithm under two different qualities of the prediction, while expected competitiveness and dependability constrain it on the same input.

Finally, we note that while the tradeoffs related to tail risk are rather recent, the idea of bounding the performance of online algorithms with high probability rather than with just the expected value has been studied in literature for a long time. Examples of studied problems include, e.g., call control (Leonardi et al., 2001), data management strategies (Maggs et al., 1997), and online matching (Mihail & Tröbst, 2024). Moreover, for some problems, whose algorithms use phase-based strategies and whose costs are potentially unbounded, it is possible to transform the guarantees on the competitive ratio that hold in expectation into high-probability ones (Komm et al., 2022). This applies in particular to paging and metrical task systems.

B Competitive Ratio and Min-max Principle

Strict and non-strict competitiveness. In the literature, the competitive ratio is sometimes defined in a weaker form, in which the cost of an online algorithm may exceed the required multiple of $\textsc{Opt}(\sigma)$ by an additive constant which is independent of the instance. In this paper, both notions are *strict*, i.e., we do not allow additive constants. We do this for simplicity only: our upper bounds hold without additive constants, and our lower bounds can be extended to the weaker definition. Namely, our lower bound constructions can be made arbitrarily expensive, which makes any additive term negligible. In online bidding and line search, it suffices to place the hidden value far away, and in TCP acknowledgment, to repeat the construction many times, far apart in time.

Min-max principle. The following lemma is a slightly extended and reworded min-max principle, which we use to obtain lower bounds on the competitive ratio of randomized algorithms.

Lemma 6 (Min-max principle)**.** *Let $\mathcal{D}^\star \subseteq \mathcal{D}$ be two sets of deterministic algorithms and let π be a probability distribution over instances. Assume that $\mathbb{E}_{\sigma\sim\pi}[R(\textsc{Det},\sigma)] \geq \rho$ for every deterministic algorithm $\textsc{Det} \in \mathcal{D}$. Then, every randomized algorithm $\textsc{Rand}$ with $\mathsf{supp}(\textsc{Rand}) \subseteq \mathcal{D}^\star$ has expected competitive ratio at least ρ.*

Proof. Let $\textsc{Rand}$ be a $\rho^\star$-competitive randomized algorithm with $\mathsf{supp}(\textsc{Rand}) \subseteq \mathcal{D}^\star$; we will show that $\rho^\star \geq \rho$. Recall that $\textsc{Rand}$ is a probability distribution over deterministic algorithms, and by our assumptions, $\mathsf{supp}(\textsc{Rand}) \subseteq \mathcal{D}^\star \subseteq \mathcal{D}$. Then,

$$
\begin{aligned}
\rho^\star &\geq \sup_{\sigma} \mathbb{E}_{\textsc{Det}\sim\textsc{Rand}}[R(\textsc{Det},\sigma)] && \text{(by competitiveness of } \textsc{Rand}\text{)} \\
&\geq \mathbb{E}_{\sigma\sim\pi}\mathbb{E}_{\textsc{Det}\sim\textsc{Rand}}[R(\textsc{Det},\sigma)] \\
&= \mathbb{E}_{\textsc{Det}\sim\textsc{Rand}}\mathbb{E}_{\sigma\sim\pi}[R(\textsc{Det},\sigma)] \\
&\geq \inf_{\textsc{Det}\in\mathsf{supp}(\textsc{Rand})} \mathbb{E}_{\sigma\sim\pi}[R(\textsc{Det},\sigma)] \\
&\geq \inf_{\textsc{Det}\in\mathcal{D}} \mathbb{E}_{\sigma\sim\pi}[R(\textsc{Det},\sigma)] && \text{(by } \mathsf{supp}(\textsc{Rand}) \subseteq \mathcal{D}\text{)} \\
&\geq \rho. && \text{(by the lemma assumption)}
\end{aligned}
$$

□

C Proof of Lemma 9 (Step Ratio Lemma)

Lemma 9 (Step ratios)**.** *Let $1 = x_0, x_1, \ldots, x_n$ be a sequence of positive reals. Fix $c \geq 0$ and let ρ_c denote the unique solution of $\rho \cdot \ln \rho = c + \rho$. Fix a real $r \in [2, \rho_c]$ and $\varepsilon \in (0, \frac{c+r}{\ln r})$. Let*

$S_k := \sum_{i=1}^{k} x_i$, *and assume that*

$$S_k \leq \frac{r^2}{r-1} \cdot x_{k-1} \qquad \textit{for every } k \in \{1, \ldots, n\}.$$

There is a constant F, depending on c, r and ε only, such that

$$c \cdot n + \sum_{i=1}^{n} \frac{x_i}{x_{i-1}} \geq \left(\frac{c+r}{\ln r} - \varepsilon \right) \cdot \ln x_n - F.$$

Notation. Throughout this part, we fix sequence $\{x_i\}_{i=0}^{n}$ and parameters c, r and ε as in the assumptions of Lemma 9. For the ease of notation, we define $B := r^2/(r-1)$ and *step ratios* $q_k := x_k/x_{k-1}$ for $k \geq 1$. Note that $q_k > 0$, but, as the sequence need not be monotone, q_k may be smaller than 1. Finally, for $q > 1$, we define

$$g_c(q) := \frac{c+q}{\ln q}.$$

Then, the assumption on ε reads $\varepsilon < g_c(r)$. Proof of the lemma uses the following simple calculus claim about function g_c.

Claim 13. *It holds that $\rho_c > 2$. The function g_c is strictly decreasing on $(1, \rho_c]$ and strictly increasing on $[\rho_c, \infty)$.*

Proof of Claim 13. We have $g_c'(q) = (\ln q - (c+q)/q)/(\ln q)^2$, so the sign of g_c' is opposite to the sign of $h_c(q) := c + q - q \ln q$. By taking the derivative of h_c, we get that h_c is strictly decreasing on $(1, \infty)$. Furthermore, $h_c(1) > 0$ and $\lim_{q \to \infty} h_c(q) = -\infty$. Hence, h_c has a unique zero ρ_c in $(1, \infty)$. As h_c is positive on $(1, \rho_c)$ and negative on (ρ_c, ∞), the function g_c is monotonic on both intervals as claimed. Finally, as $h_c(2) > 0$, it holds that $\rho_c > 2$. □

Per-step bounds. Suppose we would like to prove Lemma 9 by a straightforward induction (setting for simplicity $\varepsilon = 0$ and $F = 0$). Then, the inductive step would add $c + q_k$ to the left hand side of the desired inequality and $g_c(r) \cdot \ln q_k$ to the right hand side. For $q_k = r$, the two terms are equal. For $q_k < r$, the left hand side is at least as large, as $g_c(r) > 0$ and g_c is decreasing on $(1, \rho_c]$ by Claim 13, so the inductive step would succeed also in this case. We may even allow q_k slightly larger than r, at the expense of replacing $g_c(r)$ by $g_c(r) - \varepsilon$ (cf. Lemma 14).

Hence, the hard part is to analyze the case $q_k > r$. This is where the upper bound on S_k from the lemma assumption comes into play: we show that it implies that, from some point on, the step ratios cannot exceed r by more than a small term. From this point on, we may apply Lemma 14, and we bound the initial terms by additive constant F.

Lemma 14. *Let $\{z_k\}_{k \geq 1}$ be a sequence converging to r, and assume that $q_k \leq z_k$ for every $k \geq 1$. Then, there exists K, depending on c, r, ε and the sequence $\{z_k\}$ only, such that*

$$c + q_k \geq (g_c(r) - \varepsilon) \cdot \ln q_k \qquad \textit{for every } k > K.$$

Proof. As $r \leq \rho_c$, Claim 13 implies that g_c is decreasing on $(1, r]$, and thus $g_c(q) \geq g_c(r)$ for every $q \in (1, r]$. Moreover, g_c is continuous at r. Hence, there exists small enough $\theta > 0$, such that

$$g_c(q) \geq g_c(r) - \varepsilon \qquad \text{for every } q \in (1, r + \theta]. \tag{3}$$

As the sequence $\{z_k\}$ converges to r, we may fix a K, such that $z_k \leq r + \theta$ for every $k > K$.

Now fix $k > K$; then $q_k \leq z_k \leq r + \theta$. If $q_k \leq 1$, then $\ln q_k \leq 0$, and thus $c + q_k > 0 \geq (g_c(r) - \varepsilon) \cdot \ln q_k$, where the latter inequality follows by $\varepsilon < g_c(r)$. Otherwise $q_k \in (1, r + \theta]$, and then $c + q_k = g_c(q_k) \cdot \ln q_k \geq (g_c(r) - \varepsilon) \cdot \ln q_k$ by the definition of g_c and (3). □

Recurrence. The following function $\varphi(t) := B/(B-t)$ for $t \in [0, B)$ and the recurrence

$$\tau_0 := 0, \qquad \tau_{k+1} := \varphi(\tau_k) \quad \text{for } k \geq 0 \tag{4}$$

will be later used to track the evolution of the sequence $\{S_k/x_k\}_{k \geq 0}$.

Claim 15. *The function φ is increasing and the sequence $\{\tau_k\}_{k\geq 0}$ converges to $\frac{B}{r} = \frac{r}{r-1}$.*

Proof. Since the derivative of φ is $\varphi'(t) = B/(B-t)^2 > 0$, the function φ is increasing.

The fixed points of φ satisfy $B/(B-t) = t$, or equivalently, $t^2 - B \cdot t + B = 0$. That is, the fixed points are B/r and r, where B/r is the smaller one of them, as $B/r = r/(r-1) \leq r$.

For $t \in [0, B/r)$ we have $t^2 - B \cdot t + B > 0$, or equivalently $\varphi(t) = B/(B-t) > t$. Thus, the sequence $\{\tau_k\}_{k\geq 0}$ is strictly increasing as long as it stays below B/r. It indeed stays there, as φ is increasing and $\varphi(B/r) = B/r$. Thus, it converges to some fixed point $t^* \leq B/r$, and the only fixed point with this property is B/r. □

Proof of Lemma 9. We start with applying the recurrence above to analyze how the sequence $\{S_k/x_k\}_{k\geq 0}$ evolves. Namely, by induction on k, we show that

$$S_k/x_k \geq \tau_k \qquad \text{for every } k \geq 0. \tag{5}$$

For $k = 0$, the inequality (5) holds as $S_0/x_0 = 0 = \tau_0$. For the induction step, fix $k \geq 1$ and suppose that (5) holds for $k-1$. Then,

$$\begin{aligned}
\frac{S_k}{x_k} = 1 + \frac{S_{k-1}}{x_k} &\geq 1 + \frac{S_{k-1}}{B \cdot x_{k-1} - S_{k-1}} && \text{(by } x_k + S_{k-1} = S_k \leq B \cdot x_{k-1}\text{)} \\
&= \frac{B \cdot x_{k-1}}{B \cdot x_{k-1} - S_{k-1}} = \varphi\left(\frac{S_{k-1}}{x_{k-1}}\right) \\
&\geq \varphi(\tau_{k-1}) = \tau_k, && \text{(as } \varphi \text{ is increasing by Claim 15)}
\end{aligned}$$

which concludes the induction step and proves (5).

Next, we bound the step ratios. We set $z_k := B - \tau_{k-1}$ for $k \geq 1$. By the lemma assumption, $B \geq S_k/x_{k-1} = (x_k + S_{k-1})/x_{k-1} = q_k + S_{k-1}/x_{k-1}$, so by reorganizing the terms and applying (5), we obtain

$$q_k \leq B - \frac{S_{k-1}}{x_{k-1}} \leq B - \tau_{k-1} = z_k \qquad \text{for every } k \geq 1. \tag{6}$$

By Claim 15, the sequence $\{z_k\}_{k\geq 1}$ converges to $B - B/r = r$. Thus, Lemma 14 applied to $\{z_k\}$ yields K, such that

$$c + q_k \geq (g_c(r) - \varepsilon) \cdot \ln q_k \qquad \text{for every } k > K. \tag{7}$$

Moreover, by (6), $q_i \leq B$ for every i, and thus for any k it holds that

$$\ln x_k = \sum_{i=1}^{k} \ln q_i \leq k \cdot \ln B. \tag{8}$$

Next, we set $F := (g_c(r) - \varepsilon) \cdot K \cdot \ln B$, and we consider two cases. If $n \leq K$, then $\ln x_n \leq K \cdot \ln B$. In such case, the right hand side of the inequality from the lemma is $(g_c(r) - \varepsilon) \cdot \ln x_n - F \leq (g_c(r) - \varepsilon) \cdot K \cdot \ln B - F = 0$, and the lemma follows. Thus, we may assume that $n > K$, and then

$$\begin{aligned}
c \cdot n + \sum_{k=1}^{n} q_k &\geq \sum_{k=K+1}^{n} (c + q_k) \\
&\geq (g_c(r) - \varepsilon) \cdot \sum_{k=K+1}^{n} \ln q_k && \text{(by (7))} \\
&= (g_c(r) - \varepsilon) \cdot (\ln x_n - \ln x_K) \\
&\geq (g_c(r) - \varepsilon) \cdot (\ln x_n - K \cdot \ln B) && \text{(by (8))} \\
&= (g_c(r) - \varepsilon) \cdot \ln x_n - F.
\end{aligned}$$

□

D PROOFS FOR SECTION 3 (ONLINE BIDDING)

Throughout this section, we fix $r \in [2, e]$ and let $\gamma(r) := r^2/(r-1)$.

Lemma 7. *Randomized algorithm* $\textsc{Bid}_r$ *is* $\gamma(r)$*-dependable and its expected competitive ratio is at most* $r/\ln r$.

Proof. Let $T \geq 1$ be the threshold, $\tau = \log_r(T)$, and let b be the largest bid paid by the algorithm, defined by $b/r < T \leq b$. The value $\log_r(b)$ is distributed uniformly in $[\tau, \tau+1)$. Thus, we may view the algorithm as producing bids ending at $r^{\zeta+\tau}$ for $\zeta := \log_r(b) - \tau$ uniform in $[0,1)$, each bid equal to r times the preceding one. By bounding this finite sequence by the corresponding infinite one, we obtain, for any fixed value of ζ,

$$R(\textsc{Bid}_r(\xi), T) \leq \frac{1}{T} \cdot \sum_{i=0}^{\infty} r^{\zeta+\tau-i} = r^{\zeta} \cdot \frac{r^{\tau}}{T} \cdot \sum_{i=0}^{\infty} r^{-i} = r^{\zeta} \cdot \frac{r}{r-1}. \tag{9}$$

To bound the expected competitive ratio of $\textsc{Bid}_r$, we first compute the expectation of r^{ζ} as $\mathbb{E}_{\zeta\sim[0,1)}[r^{\zeta}] = \int_0^1 r^z\, dz = (r-1)/\ln r$. Thus,

$$\mathbb{E}_{\xi\sim[0,1)}[R(\textsc{Bid}_r(\xi), T)] \leq \frac{r}{r-1} \cdot \mathbb{E}_{\zeta\sim[0,1)}\left[r^{\zeta}\right] = \frac{r}{\ln r}.$$

Next, we analyze the dependability of $\textsc{Bid}_r$. By the definition of ζ, we have $r^{\zeta} < r$. Thus, by (9), it holds that $R(\textsc{Bid}_r(\xi), T) < r \cdot r/(r-1) = \gamma(r)$ regardless of the choice of ξ. □

Lemma 8. *Fix* $\delta > 0$*. There exists a probability distribution* $\pi(\delta)$ *over the inputs, such that every* $\gamma(r)$*-dependable algorithm* DET *satisfies* $\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det}, \sigma)] \geq \frac{r}{\ln r} - \delta$.

Proof. We assume that $\delta < 4r/\ln r$, as otherwise the lemma holds trivially. Let $F(\varepsilon)$ be the parameter (depending on ε) resulting from Lemma 9, when it is applied with $c = 0$ and ε. We choose a real number $U > 1$ large enough so that

$$\frac{1}{\ln U} \leq \frac{\delta}{4} \cdot \min\left\{\frac{\ln r}{r},\ \frac{1}{F(\delta/4)},\ \frac{e\cdot(r-1)}{r^2}\right\}. \tag{10}$$

Distribution $\pi(\delta)$ is then defined as follows: for any $t \in [1, U]$, the density of the threshold at t is $v_t = 1/(t \cdot \ln U)$. Since the algorithm knows $\pi(\delta)$, we can assume without loss of generality that it never bids a number outside the range $[1, U]$.

Fix a deterministic $\gamma(r)$-dependable algorithm DET. Since DET is deterministic, bids of DET are a sequence of numbers $x_1 < x_2 < \ldots$. To simplify notation, we set $x_0 = 1$. Define n as an index of the bid satisfying

$$x_{n-1} < U/e \leq x_n.$$

Such an index must exist, as otherwise DET would not be competitive for thresholds greater than U/e. We charge DET only for the bids $x_1, \ldots, x_n$, and we ignore its cost of any subsequent bids. Next, let $S_k := \sum_{i=1}^{k} x_i$ denote the total cost DET pays on the first k bids. This gives the following bound

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det}, \sigma)] &\geq \sum_{i=1}^{n} \left(\int_{x_{i-1}}^{x_i} \frac{S_i}{t} \cdot v_t\, dt\right) + \int_{x_n}^{U} \frac{S_n}{t} \cdot v_t\, dt \\
&= \sum_{i=1}^{n} \left(\int_{x_{i-1}}^{x_i} \frac{\sum_{j=1}^{i} x_j}{t^2 \cdot \ln U}\, dt\right) + \int_{x_n}^{U} \frac{\sum_{j=1}^{n} x_j}{t^2 \cdot \ln U}\, dt \\
&= \frac{1}{\ln U} \cdot \sum_{i=1}^{n} x_i \int_{x_{i-1}}^{U} \frac{1}{t^2}\, dt \\
&= \frac{1}{\ln U} \cdot \sum_{i=1}^{n} x_i \left(\frac{1}{x_{i-1}} - \frac{1}{U}\right) \\
&= \frac{1}{\ln U} \cdot \sum_{i=1}^{n} \frac{x_i}{x_{i-1}} - \frac{S_n}{U \ln U}.
\end{aligned} \tag{11}$$

As DET is $\gamma(r)$-dependable, its cost on the first k bids, S_k, cannot exceed $\gamma(r)$ times the cost of OPT, which can be arbitrarily close to x_{k-1}. Hence

$$S_k \le \gamma(r) \cdot x_{k-1} = \frac{r^2}{r-1} \cdot x_{k-1} \qquad \text{for every } k \ge 1, \tag{12}$$

and thus also for $k = n$,

$$S_n \le \frac{r^2}{r-1} \cdot x_{n-1} \le \frac{r^2}{r-1} \cdot \frac{U}{e}. \tag{13}$$

By (12), we may apply Lemma 9 with $c = 0$ and $\varepsilon = \delta/4$ obtaining

$$\sum_{i=1}^{n} \frac{x_i}{x_{i-1}} \ge \left(\frac{r}{\ln r} - \frac{\delta}{4}\right) \cdot \ln x_n - F(\delta/4), \tag{14}$$

which plugged into (11) gives

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\text{DET}, \sigma)] &\ge \frac{\ln x_n}{\ln U} \cdot \left(\frac{r}{\ln r} - \frac{\delta}{4}\right) - \frac{F(\delta/4)}{\ln U} - \frac{S_n}{U \ln U} \\
&\ge \left(1 - \frac{1}{\ln U}\right) \cdot \left(\frac{r}{\ln r} - \frac{\delta}{4}\right) - \frac{F(\delta/4)}{\ln U} - \frac{r^2}{e \cdot (r-1) \cdot \ln U} \\
&\ge \frac{r}{\ln r} - \frac{\delta}{4} - \frac{r}{\ln r \cdot \ln U} - \frac{F(\delta/4)}{\ln U} - \frac{r^2}{e \cdot (r-1) \cdot \ln U} \\
&\ge \frac{r}{\ln r} - \delta. \qquad \square
\end{aligned}$$

We can now prove Theorem 1, restated below.

Theorem 1. *Fix $r \in [2, e]$ and let $\gamma(r) := \frac{r^2}{r-1}$. There exists a $\gamma(r)$-dependable randomized algorithm for online bidding whose expected competitive ratio is $\frac{r}{\ln r}$. Conversely, every $\gamma(r)$-dependable randomized algorithm for online bidding has expected competitive ratio at least $\frac{r}{\ln r}$.*

Proof. The upper bound is Lemma 7. By combining Lemma 8 with the min-max principle (Lemma 6), we obtain that no randomized $\gamma(r)$-dependable algorithm can achieve expected competitive ratio better than $r/\ln r - \delta$. As δ can be arbitrarily small, the theorem follows. $\square$

E PROOFS FOR SECTION 4 (LINE SEARCH)

Let r_0 be the solution of $r_0 = (1 + r_0)/\ln r_0$. Throughout this section, fix $r \in [2, r_0]$ and let $\gamma(r) := 1 + (2r^2)/(r-1)$.

The upper bound of Theorem 2 is attained by the following algorithm called SMARTCOW, defined in Kao et al. (1996). It is parametrized by a number $r \in (1, +\infty)$.

Algorithm SMARTCOW$_r$: first, choose a value $\xi \in [0, 1)$ uniformly at random and a random starting direction. Then move to points $r^\xi, r^{\xi+1}, r^{\xi+2}, \ldots$ in the alternating directions, until the target is found.

Lemma 16 (Kao et al., 1996, Theorem 3.1)**.** *Fix $r > 1$. Algorithm* SMARTCOW$_r$ *has expected competitive ratio $1 + \frac{1+r}{\ln r}$.*

This competitive ratio is minimized when $r = (1 + r)/\ln r$, which holds for value $r_0 \approx 3.59112$. Then the expected ratio is $1 + r_0 \approx 4.59112$.

Lemma 17. *Fix $r \in [2, r_0]$. Algorithm* SMARTCOW$_r$ *is $(1 + \frac{2r^2}{r-1})$-dependable.*

Proof. Fix the value of ξ and let k be an integer such that

$$r^{\xi+k} < T \le r^{\xi+k+1}.$$

The cost of SMARTCOW$_r$ depends on how the chosen starting direction aligns with the direction of T. If the cow travels to $r^{k+\xi}$ for the first time in the direction opposite to T, then the total cost is $2 \cdot \sum_{i=0}^{k} r^{\xi+i} + T$. Otherwise, after visiting $r^{k+\xi}$, it then visits $r^{k+1+\xi}$ in the direction opposite to T and only then comes back to reach T. Its cost is then $2 \cdot \sum_{i=0}^{k+1} r^{\xi+i} + T$.

Clearly, the second case corresponds to the higher cost. This cost can be bounded as

$$\begin{aligned} 2 \cdot \sum_{i=0}^{k+1} r^{\xi+i} + T &\le 2 \cdot \frac{r^{\xi}(r^{k+2}-1)}{r-1} + T \\ &\le \frac{2r^2 \cdot r^{\xi+k}}{r-1} + T \\ &< \frac{2r^2}{r-1} \cdot T + T. \end{aligned}$$

Since the optimal cost is T, this gives us the dependability bound. □

Combining Lemma 16 with Lemma 17 shows that SMARTCOW$_r$ gives the upper bound part of Theorem 2 for any $r \in [2, r_0]$. To prove a tight lower bound, we need the following lemma.

Lemma 10. *Fix $\delta > 0$. There exists a probability distribution $\pi(\delta)$ over the inputs, such that every $\gamma(r)$-dependable algorithm* DET *satisfies $\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\text{DET}, \sigma)] \ge 1 + \frac{1+r}{\ln r} - \delta$.*

Notation and probability distribution. We assume that $\delta < 4 \cdot (1+r)/\ln r$, as otherwise the lemma holds trivially. Let $F(\varepsilon)$ be the parameter (depending on ε) resulting from Lemma 9, when it is applied with $c = 1$ and ε. We choose a real number $U > 1$ large enough so that

$$\frac{1}{\ln U} \le \frac{\delta}{4} \cdot \min\left\{ \frac{\ln r}{1+r},\ \frac{1}{F(\delta/4)},\ \frac{e \cdot (r-1)}{2 \cdot r^2} \right\}.$$

Distribution $\pi(\delta)$ is then defined as follows: for any $t \in [1, U]$ on any of the two paths, the density of the target at t is $v_t = 1/(2t \cdot \ln U)$. Note that the distribution is symmetric on both paths. Since the algorithm knows $\pi(\delta)$, we can assume without loss of generality that it never travels further than U from the starting point.

Fix a deterministic $\gamma(r)$-dependable algorithm DET. It can be described by

- a sequence of numbers $x_1, x_2, \ldots$ from $[1, U]$ satisfying $x_1 < x_3 < x_5 < \ldots$ and $x_2 < x_4 < x_6 < \ldots$, and
- a starting direction.

DET first moves to points $x_1, x_2, x_3, \ldots$ on alternating paths, starting with x_1 in the chosen starting direction, until it finds the target. Any algorithm that works differently can be easily transformed into such an algorithm without increasing its cost. For succinctness, we set $x_{-1} = x_0 = 1$. Define n as an index satisfying

$$x_{n-1} < U/e \le x_n.$$

Such an index must exist, as otherwise DET would not be competitive for targets further than U/e. If after reaching the point x_n, the searcher has not yet reached the target, it moves back to 0 and is not required to walk any further. The cost of DET is then increased by the position of the target, which lower-bounds the cost the algorithm would pay from that point. All further costs of DET are forgiven.

Let $S_k := \sum_{i=1}^{k} x_i$. Then $2 \cdot S_k$ is the total cost of DET in the first k steps.

Lemma 18. *Fix $\delta > 0$ and let distribution $\pi(\delta)$ be defined as above. Then,*

$$\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\text{DET}, \sigma)] \ge 1 + \frac{1}{\ln U}\left(n + \sum_{i=1}^{n} \frac{x_i}{x_{i-1}} \right) - \frac{2 \cdot S_n}{U \ln U}.$$

Proof. Let t denote the position of the target. For every $i \ge 1$, if the target is positioned on the same path as x_i and in the interval $(x_{i-2}, x_i]$, then DET pays exactly $2 \cdot S_{i-1} + t$. Furthermore, if

the target is placed on the same path as x_n and $t > x_n$, due to forgiveness mentioned above, the algorithm pays $2 \cdot S_n + t$. Symmetrically, if it is placed above x_{n-1} on the other path, DET also pays $2 \cdot S_n + t$. Therefore, its expected ratio is at least

$$\begin{aligned}
&\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det},\sigma)] \\
&\quad\geq \sum_{i=1}^{n}\left(\int_{x_{i-2}}^{x_i}\frac{2S_{i-1}+t}{t}\cdot v_t\,dt\right)+\int_{x_n}^{U}\frac{2S_n+t}{t}\cdot v_t\,dt+\int_{x_{n-1}}^{U}\frac{2S_n+t}{t}\cdot v_t\,dt \\
&\quad= 1+\sum_{i=1}^{n}\left(\int_{x_{i-2}}^{x_i}\frac{2\cdot S_{i-1}}{t}\cdot v_t\,dt\right)+\int_{x_n}^{U}\frac{2\cdot S_n}{t}\cdot v_t\,dt+\int_{x_{n-1}}^{U}\frac{2\cdot S_n}{t}\cdot v_t\,dt \\
&\quad= 1+\sum_{i=1}^{n}\left(\int_{x_{i-2}}^{x_i}\frac{\sum_{j=1}^{i-1}x_j}{t^2\ln U}\,dt\right)+\int_{x_n}^{U}\frac{\sum_{j=1}^{n}x_j}{t^2\ln U}\,dt+\int_{x_{n-1}}^{U}\frac{\sum_{j=1}^{n}x_j}{t^2\ln U}\,dt
\end{aligned}$$

By changing the order of summation and, for every x_i, summing all integrals in which x_i appears, we obtain

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det},\sigma)] &\geq 1+\frac{1}{\ln U}\cdot\sum_{i=1}^{n}x_i\left(\int_{x_i}^{U}\frac{1}{t^2}\,dt+\int_{x_{i-1}}^{U}\frac{1}{t^2}\,dt\right) \\
&= 1+\frac{1}{\ln U}\cdot\sum_{i=1}^{n}x_i\left(\frac{1}{x_i}-\frac{1}{U}+\frac{1}{x_{i-1}}-\frac{1}{U}\right) \\
&= 1+\frac{1}{\ln U}\cdot\left(n+\sum_{i=1}^{n}\frac{x_i}{x_{i-1}}\right)-\frac{2\cdot S_n}{U\ln U}.
\end{aligned}$$

□

Proof of Lemma 10. As DET is $\gamma(r)$-dependable, its cost of reaching any $x \in (x_{k-1}, x_{k+1}]$, which is $x + 2 \cdot S_k$, cannot exceed $\gamma(r)$ times the cost of OPT, which is x. As x can be arbitrarily close to x_{k-1},

$$x_{k-1} + 2 \cdot S_k \leq \left(1+\frac{2r^2}{r-1}\right)\cdot x_{k-1} \qquad \text{for every } k \geq 1.$$

Equivalently,

$$S_k \leq \frac{r^2}{r-1}\cdot x_{k-1} \qquad \text{for every } k \geq 1, \tag{15}$$

and thus also for $k = n$,

$$S_n \leq \frac{r^2}{r-1}\cdot x_{n-1} \leq \frac{r^2}{r-1}\cdot\frac{U}{e}.$$

By (15), we may apply Lemma 9 with $c = 1$ and $\varepsilon = \delta/4$, obtaining

$$n+\sum_{i=1}^{n}\frac{x_i}{x_{i-1}} \geq \left(\frac{1+r}{\ln r}-\frac{\delta}{4}\right)\cdot\ln x_n - F(\delta/4). \tag{16}$$

Finally, we combine (16) with Lemma 18 to obtain

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\pi(\delta)}[R(\textsc{Det},\sigma)] &\geq 1+\frac{1}{\ln U}\cdot\left(n+\sum_{i=1}^{n}\frac{x_i}{x_{i-1}}\right)-\frac{2\cdot S_n}{U\ln U} \\
&\geq 1+\frac{\ln x_n}{\ln U}\cdot\left(\frac{1+r}{\ln r}-\frac{\delta}{4}\right)-\frac{F(\delta/4)}{\ln U}-\frac{2\cdot S_n}{U\ln U} \\
&\geq 1+\left(1-\frac{1}{\ln U}\right)\cdot\left(\frac{1+r}{\ln r}-\frac{\delta}{4}\right)-\frac{F(\delta/4)}{\ln U}-\frac{2\cdot r^2}{e\cdot(r-1)\cdot\ln U} \\
&\geq 1+\frac{1+r}{\ln r}-\frac{\delta}{4}-\frac{1+r}{\ln r\cdot\ln U}-\frac{F(\delta/4)}{\ln U}-\frac{2\cdot r^2}{e\cdot(r-1)\cdot\ln U} \\
&\geq 1+\frac{1+r}{\ln r}-\delta.
\end{aligned}$$

□

We are ready to prove Theorem 2, restated below.

Theorem 2. *Fix $r \in [2, r_0]$, where $r_0 \approx 3.591$ is the solution of $r = \frac{1+r}{\ln r}$, and let $\gamma(r) := 1 + \frac{2r^2}{r-1}$. There exists a $\gamma(r)$-dependable randomized algorithm for line search whose expected competitive ratio is $1 + \frac{1+r}{\ln r}$. Conversely, every $\gamma(r)$-dependable randomized algorithm for line search has expected competitive ratio at least $1 + \frac{1+r}{\ln r}$.*

Proof. The upper bound comes from combining Lemma 16 with Lemma 17. By combining Lemma 10 with the min-max principle (Lemma 6), we obtain that no randomized $\gamma(r)$-dependable algorithm can achieve expected competitive ratio better than $1 + (1 + r)/\ln r - \delta$. As δ can be arbitrarily small, the theorem follows. □

F Proofs for Section 5 (TCP Acknowledgment Upper Bound)

F.1 Dependability

Lemma 11. *For every $z \in (0, 1]$, $\textsc{Det}_z$ is $\gamma(z)$-dependable.*

Proof. Consider a deterministic algorithm $\textsc{Det}_z$ for some $z \in (0, 1]$ and an instance σ. Let $n_z(\sigma)$ and $W_z(\sigma)$ denote the number of acknowledgments and total latency of $\textsc{Det}_z$ on σ, respectively. We show the following two inequalities.

$$z \cdot n_z(\sigma) \le \textsc{Opt}(\sigma) \tag{17}$$

$$W_z(\sigma) \le \textsc{Opt}(\sigma) \tag{18}$$

By combining the two inequalities, we have

$$\textsc{Det}_z(\sigma) = n_z(\sigma) + W_z(\sigma) \le \left(1 + \frac{1}{z}\right) \cdot \textsc{Opt}(\sigma),$$

completing the proof.

We prove (17) by showing that, for every acknowledgment of $\textsc{Det}_z$, we can disjointly charge at least z to the optimal solution. Consider the i-th acknowledgment of $\textsc{Det}_z$ at time T_i for $i \in \mathbb{Z}_{>0}$. If the optimal solution performs an acknowledgment in time interval $(T_{i-1}, T_i]$, we charge $z \le 1$ to this acknowledgment of the optimal solution for the i-th acknowledgment of $\textsc{Det}_z$. Otherwise, if the optimal solution does not perform any acknowledgment in interval $(T_{i-1}, T_i]$, observe that every packet that has arrived in interval $(T_{i-1}, \tau_i]$ is unacknowledged until time T_i in the optimal solution, implying that the optimal solution incurs latency at least

$$P(T_{i-1}, \tau_i) \cdot (T_i - \tau_i) = z$$

in interval $(T_{i-1}, T_i]$, where the equality is due to (2) from the definition of $\textsc{Det}_z$. Hence, we can charge z to the latency incurred by the optimal solution during $(T_{i-1}, T_i]$ for the i-th acknowledgment of $\textsc{Det}_z$. Notice that every fraction of the optimal cost $\textsc{Opt}(\sigma)$ is charged at most once, completing the proof of (17).

We now turn to proving (18). Let $n_{\textsc{Opt}}(\sigma)$ and $W_{\textsc{Opt}}(\sigma)$ respectively denote the number of acknowledgments and latency of the optimal solution, i.e., $\textsc{Opt}(\sigma) = n_{\textsc{Opt}}(\sigma) + W_{\textsc{Opt}}(\sigma)$. It thus suffices to show

$$W_z(\sigma) - W_{\textsc{Opt}}(\sigma) \le n_{\textsc{Opt}}(\sigma). \tag{19}$$

Let $K_{\textsc{Opt}}$ denote the set of all acknowledgment times of the optimal solution, and for any $t \in K_{\textsc{Opt}}$, let R_t denote the set of all packets acknowledged at time t by the optimal solution. We remark that $\{R_t\}_{t \in K_{\textsc{Opt}}}$ partitions the entire set of input packets. For each packet $j \in R_t$, let d_j denote the time at which packet j is acknowledged by $\textsc{Det}_z$. Let $R'_t \subseteq R_t$ denote the set of packets in R_t that are acknowledged by $\textsc{Det}_z$ later than by the optimal solution, i.e., $R'_t := \{j \in R_t : d_j > t\}$.

We claim that, for any acknowledgment time $t \in K_{\textsc{Opt}}$ of the optimal solution,

$$\sum_{j \in R'_t} (d_j - t) \le z. \tag{20}$$

Observe that, if this claim is true, we immediately derive (19) since

$$W_z(\sigma) - W_{\text{OPT}}(\sigma) = \sum_{t \in K_{\text{OPT}}} \sum_{j \in R_t} (d_j - t) \leq \sum_{t \in K_{\text{OPT}}} \sum_{j \in R'_t} (d_j - t)$$
$$\leq \sum_{t \in K_{\text{OPT}}} z \leq z \cdot n_{\text{OPT}}(\sigma) \leq n_{\text{OPT}}(\sigma),$$

where the equality comes from the fact that $\{R_t\}_{t \in K_{\text{OPT}}}$ partitions the entire set of input packets.

To prove the claim, we assume that R'_t is non-empty, as otherwise the claim is trivial. Observe first that all packets in R'_t are acknowledged at the same time, say T_i, by DET_z because they are all unacknowledged at time t in the execution of DET_z. Moreover, R'_t is a subset of the packets unacknowledged by DET_z at time t, yielding that $T_{i-1} \leq t$ and $|R'_t| \leq P(T_{i-1}, t)$. We can thus bound the left-hand side of (20) as

$$\sum_{j \in R'_t} (d_j - t) = |R'_t| \cdot (T_i - t) \leq P(T_{i-1}, t) \cdot (T_i - t).$$

Suppose toward contradiction that $\sum_{j \in R'_t} (d_j - t) > z$. We then have $P(T_{i-1}, t) \cdot (T_i - t) > z$, implying that DET_z would have performed the i-th acknowledgment strictly earlier than T_i by definition. This yields a contradiction to the choice of T_i, completing the proof of the claim, and hence the entire proof. □

Remark 19. *The proof of Lemma 11 actually shows $\gamma(z)$-dependability not only for DET_z, but also for any (deterministic or randomized) algorithm such that, for any $i \in \mathbb{Z}_{>0}$, the i-th acknowledgment time T_i is determined as the first time for which there exists $\tau_i \in [T_{i-1}, T_i]$ satisfying*

$$P(T_{i-1}, \tau_i) \cdot (T_i - \tau_i) = z_i,$$

where $T_0 := 0$, and $z_i \in [z, 1]$ is chosen at time T_{i-1} using any information revealed so far.

F.2 COMPETITIVENESS

In our analysis of RAND_r, we use the following properties shown by Karlin et al. (2003).

Lemma 20. *Let $n_z(\sigma)$ denote the number of acknowledgments made by DET_z on input σ. Then,*

$$\text{DET}_z(\sigma) \leq \text{OPT}(\sigma) + n_z(\sigma) - \int_z^1 n_w(\sigma)\, dw; \tag{21}$$

$$\text{OPT}(\sigma) \geq \int_0^1 n_w(\sigma)\, dw. \tag{22}$$

Indeed, (21) can be derived by combining (1) and Lemma 4 of Karlin et al. (2003), while (22) is Corollary 5 of the same paper.

We are now ready to prove Lemma 12, restated below for convenience. Recall that RAND_r initially draws a value $z \in [r, 1]$ from the mixed probability distribution that has

- a discrete mass of $q_r := \frac{r \cdot e^r}{(r+1) \cdot e - e^r}$ at $z = r$, and
- a density $p(z) := \frac{r+1}{(r+1) \cdot e - e^r} \cdot e^z$ for any $z \in (r, 1]$,

and then executes DET_z with the sampled z.

Lemma 12. RAND_r *has expected competitive ratio* $1 + \frac{e^r}{(1+r) \cdot e - e^r}$.

Proof. Consider an input σ. The expected cost incurred by RAND_r is written as

$$\mathbb{E}[\text{RAND}_r(\sigma)] = q_r \cdot \text{DET}_r(\sigma) + \int_r^1 p(z) \cdot \text{DET}_z(\sigma)\, dz.$$

For any $z \in [r,1]$, let $\textsc{Det}'_z(\sigma) = n_z(\sigma) - \int_z^1 n_w(\sigma)\,dw$. Using (21), we can bound the expected cost by

$$\mathbb{E}\left[\textsc{Rand}_r(\sigma)\right] \le \textsc{Opt}(\sigma) + q_r \cdot \textsc{Det}'_r(\sigma) + \int_r^1 p(z) \cdot \textsc{Det}'_z(\sigma)\,dz.$$

Note that the last term can be rephrased as follows:

$$\begin{aligned}
\int_r^1 p(z) \cdot \textsc{Det}'_z(\sigma)\,dz &= \int_r^1 p(z) \cdot \left(n_z(\sigma) - \int_z^1 n_w(\sigma)\,dw\right) dz \\
&\overset{(a)}{=} \int_r^1 \left(p(z) \cdot n_z(\sigma) - n_z(\sigma) \cdot \int_r^z p(w)\,dw\right) dz \\
&\overset{(b)}{=} \int_r^1 \Big(p(z) \cdot n_z(\sigma) - n_z(\sigma) \cdot \big(p(z) - p(r)\big)\Big)\,dz \\
&= p(r) \cdot \int_r^1 n_z(\sigma)\,dz,
\end{aligned}$$

where (a) follows from changing the order of integration, and (b) from the definition of $p(\cdot)$.

We moreover remark that the number of acknowledgments $n_z(\sigma)$ by $\textsc{Det}_z$ is non-increasing as z increases. To see this, fix z and z' with $z \le z'$, and consider two consecutive acknowledgments of $\textsc{Det}_{z'}$, where the start of the execution counts as an acknowledgment; let T'_{i-1} and T'_i denote the times of these acknowledgments of $\textsc{Det}_{z'}$. Note that $\textsc{Det}_z$ must acknowledge at least once in $(T'_{i-1}, T'_i]$ since otherwise all packets unacknowledged by $\textsc{Det}_{z'}$ at time $t \in (T'_{i-1}, T'_i]$ are also unacknowledged by $\textsc{Det}_z$ at the same time t, from which we deduce that (2) for $\textsc{Det}_{z'}$ contradicts the absence of acknowledgments of $\textsc{Det}_z$ in $(T'_{i-1}, T'_i]$. We therefore have $n_z(\sigma) \ge n_{z'}(\sigma)$, yielding that

$$r \cdot n_r(\sigma) \le \int_0^r n_z(\sigma)\,dz. \tag{23}$$

We thus derive

$$\begin{aligned}
\mathbb{E}\left[\textsc{Rand}_r(\sigma)\right] &\le \textsc{Opt}(\sigma) + q_r \cdot \textsc{Det}'_r(\sigma) + p(r) \cdot \int_r^1 n_z(\sigma)\,dz \\
&= \textsc{Opt}(\sigma) + q_r \cdot \left(n_r(\sigma) - \int_r^1 n_z(\sigma)\,dz\right) + p(r) \cdot \int_r^1 n_z(\sigma)\,dz \\
&= \textsc{Opt}(\sigma) + q_r \cdot n_r(\sigma) + (p(r) - q_r) \cdot \int_r^1 n_z(\sigma)\,dz \\
&\overset{(a)}{=} \textsc{Opt}(\sigma) + q_r \cdot n_r(\sigma) + \frac{q_r}{r} \cdot \int_r^1 n_z(\sigma)\,dz \\
&\overset{(b)}{\le} \textsc{Opt}(\sigma) + \frac{q_r}{r} \cdot \int_0^r n_z(\sigma)\,dz + \frac{q_r}{r} \cdot \int_r^1 n_z(\sigma)\,dz \\
&= \textsc{Opt}(\sigma) + \frac{q_r}{r} \cdot \int_0^1 n_z(\sigma)\,dz \\
&\overset{(c)}{\le} \left(1 + \frac{q_r}{r}\right) \cdot \textsc{Opt}(\sigma),
\end{aligned}$$

where (a) follows by $p(r) = (1 + 1/r) \cdot q_r$, (b) by (23), and (c) by (22). Notice that $1 + q_r/r = 1 + \frac{e^r}{(1+r)\cdot e - e^r}$. □

We can now prove Theorem 3, restated below.

Theorem 3. *Fix $r \in (0,1]$ and let $\gamma(r) := 1 + \frac{1}{r}$. There exists a $\gamma(r)$-dependable randomized algorithm for TCP acknowledgment whose expected competitive ratio is $1 + \frac{e^r}{(1+r)\cdot e - e^r}$.*

Proof. For any $z \in [r,1]$, $\textsc{Det}_z$ is $\gamma(z)$-dependable by Lemma 11 and therefore $\gamma(r)$-dependable. For any $r \in (0,1]$, $\textsc{Rand}_r$ is then supported only by $\gamma(r)$-dependable algorithms, and thus it is $\gamma(r)$-dependable. Moreover, by Lemma 12, the algorithm is $(1 + \frac{e^r}{(1+r)\cdot e - e^r})$-competitive. □

G LOWER BOUND FOR TCP ACKNOWLEDGMENT

In this section, we show a nontrivial lower bound for the TCP acknowledgment problem. Recall that no algorithm is better than 2-dependable (Dooly et al., 2001) and no algorithm is better than $e/(e-1)$-competitive (Seiden, 2000). Here, we show a lower bound on the dependability-competitiveness tradeoff, which is stronger than these two bounds combined.

Similarly to Theorem 3, we parametrize both the dependability and the resulting bound on the expected competitive ratio by a single real $r \in [(\sqrt{5}-1)/2, 1]$, setting

$$\begin{aligned} c(r) &:= r^2 + r - 1 && \in [0,1], \\ \gamma(r) &:= 1 + 1/r && \in [2, (3+\sqrt{5})/2], \\ \rho(r) &:= 1 + \frac{(2+r)\cdot r^{2+r}}{(1+r)^2 - r^{2+r}}. \end{aligned}$$

The range of r is chosen so that $c(r) \in [0,1]$. We show that every $\gamma(r)$-dependable randomized algorithm for TCP acknowledgment has expected competitive ratio at least $\rho(r)$, which proves Theorem 4. See Fig. 1c for a plot of $\rho(r)$ versus $\gamma(r)$.

Throughout this section, we fix $r \in [(\sqrt{5}-1)/2, 1]$ and write $c := c(r)$, $\gamma := \gamma(r)$ and $\rho := \rho(r)$. We extensively use the following relationships:

$$\gamma \geq 2, \qquad \frac{\gamma}{\gamma - 1} = 1 + r \leq 2, \qquad 1 + c = r \cdot (1 + r).$$

The proof consists of two parts. First, in Section G.1, we study instances similar to those by Seiden (2000), consisting of bursts of packets whose sizes grow quickly. We show that γ-dependability forces any deterministic algorithm to satisfy certain inequalities. In particular, such an algorithm has to acknowledge the first burst no earlier than at latency c; more complex inequalities hold also for the subsequent bursts. Second, in Section G.2, we apply the inequalities obtained this way for the first two bursts. We construct a probability distribution over instances with one or two bursts, on which every deterministic algorithm satisfying the bounds of Section G.1 has expected competitive ratio at least ρ. Theorem 4 then follows by the application of the min-max principle (Lemma 6).

Burst sequences. For a vector $\mathbf{x}$, we denote its length by $|\mathbf{x}|$ and its i-th component by x_i. We write $\perp$ for the vector of length 0 and $\mathbb{R}^*_{\geq 0} := \bigcup_{k \geq 0} \mathbb{R}^k_{\geq 0}$ for the set of all finite vectors with non-negative entries. For $\mathbf{x} \in \mathbb{R}^k$ and $j \leq k$, we write $\mathbf{x}_{\leq j} := \langle x_1, \ldots, x_j \rangle$ for the prefix of $\mathbf{x}$ of length j.

An instance $E(\mathbf{s}, \mathbf{y})$, defined for a vector $\mathbf{s} \in \mathbb{Z}^n_{\geq 1}$ of *burst sizes* and a vector $\mathbf{y} \in \mathbb{R}^{n-1}_{\geq 0}$ of *gaps*, consists of n *bursts*; burst i arrives at time t_i and consists of s_i packets. We define $t_1 = 0$ and $t_{i+1} = t_i + y_i/s_i$ for $i \geq 1$.

We say that an unacknowledged burst i has *latency* z at time $t_i + z/s_i$, i.e., the latency of a burst is the total latency which its packets accrued so far. Note that burst $i+1$ arrives at latency y_i of burst i. We also define the *overhead* of burst i as $\theta_i := (\sum_{j \leq i} s_j)/s_i \geq 1$. While burst i accrues latency z, the bursts $1, \ldots, i$ together accrue latency $\theta_i \cdot z$. Note that $\theta_1 = 1$ and that θ_i is close to 1 whenever burst i is much larger than all the preceding ones together.

In all our constructions, $s_1 = 1$, i.e., the first burst is a single packet, and $E(\langle 1 \rangle, \perp)$ is the instance which consists of this packet only.

For $i \in \{1, \ldots, n\}$, we write $\sigma_{\leq i} := E(\mathbf{s}_{\leq i}, \mathbf{y}_{\leq i-1})$ for the instance consisting of the first i bursts.

Prefixes and costs. Consider a deterministic algorithm DET with finite dependability, and assume that an adversary creates an instance $E(\mathbf{s}, \mathbf{y})$ *adaptively against* DET: it chooses the size s_{i+1} of burst $i+1$ and places this burst at latency y_i of burst i, and it fixes both values only after it observes the acknowledgment of burst i by DET. (Such acknowledgment must eventually occur, as the dependability of DET is finite.) The process results in an instance $E(\mathbf{s}, \mathbf{y})$ with $n := |\mathbf{s}|$ bursts. We denote the latency at which DET acknowledges burst $i \leq n$ by x_i, which yields the vector

$$\mathbf{x}(\text{DET}, \mathbf{s}, \mathbf{y}) = \langle x_1, \ldots, x_n \rangle,$$

for which $x_i < y_i$ for every $i < n$.

For brevity, we write $x_1(\text{DET})$ for the latency at which DET acknowledges the single burst of $E(\langle 1\rangle, \bot)$, i.e., $\mathbf{x}(\text{DET}, \langle 1\rangle, \bot) = \langle x_1(\text{DET})\rangle$.

Fix $j < n$. As DET is deterministic and online, when it is executed on instance $E(\mathbf{s}_{\le j+1}, \mathbf{y}_{\le j})$ (a prefix of $E(\mathbf{s}, \mathbf{y})$), it acknowledges all bursts *at the same latencies* $x_1, \dots, x_{j+1}$ as on $E(\mathbf{s}, \mathbf{y})$.

For $i \in \{0, \dots, n\}$, we write $C_i := \sum_{j \le i}(1 + x_j)$ and when $i < n$, we let $Y_i := \sum_{j \le i} y_j$. In particular, $C_0 = 0$ and $Y_0 = 0$. Note that on the instance $E(\mathbf{s}_{\le i}, \mathbf{y}_{\le i-1})$ DET pays 1 for each of its first i acknowledgments, and the packets of burst $j \le i$ accrue latency x_j. Hence C_i is the cost of DET on such instance.

Flat instances. The bounds which we present are the strongest when all overheads θ_i are close to 1. The adversary can easily enforce this by making every burst much larger than all the previous ones together.

For a real $\xi > 0$, we call an instance $E(\mathbf{s}, \mathbf{y})$ *ξ-flat* if $s_1 = 1$ and $\theta_i \le 1 + \xi$ for every burst i. As $\theta_1 = 1$, the condition constrains only bursts $i \ge 2$, and the adversary may satisfy it simply by choosing $s_i \ge (\sum_{j<i} s_j)/\xi$. The gaps y_i on flat instances may be arbitrary.

For a deterministic algorithm DET, an integer $n \ge 1$ and a real $\xi > 0$, we denote by $\mathcal{I}_n(\text{DET}, \xi)$ the set of all ξ-flat instances with n bursts and gaps $y_1, \dots, y_{n-1} \le 1$ that can be generated adaptively against DET. We have $\mathcal{I}_1(\text{DET}, \xi) = \{E(\langle 1\rangle, \bot)\}$ for every ξ.

Note that an adaptively generated instance has to satisfy the condition $y_i > x_i$. Thus, if DET decides to acknowledge burst $i < n$ at latency $x_i \ge 1$, then the resulting instance does not belong to $\mathcal{I}_n(\text{DET}, \xi)$. In particular, if $x_1(\text{DET}) \ge 1$, then $\mathcal{I}_n(\text{DET}, \xi)$ is empty for every $n \ge 2$.

Definition 21. *For an integer $n \ge 1$ and a real $\varepsilon > 0$, we let*

$$\xi_n(\varepsilon) := \frac{r \cdot \varepsilon}{(1+r)^2 \cdot n},$$

and we let $\mathcal{D}_\varepsilon^{(n)}$ be the set of all deterministic algorithms DET *which satisfy the following two properties.*

- (first burst) $x_1(\text{DET}) \ge c$*, i.e.,* DET *acknowledges the single burst of $E(\langle 1\rangle, \bot)$ no earlier than at latency c.*
- (recursion) *For every instance in $\mathcal{I}_n(\text{DET}, \xi_n(\varepsilon))$, it holds that*

$$C_i \ge (1+r) \cdot (C_{i-1} + r - Y_{i-1}) - \varepsilon \qquad \text{for every } i \in \{2, \dots, n\}.$$

The recursion property lower-bounds the cost of DET after each burst in terms of its cost after the previous one. Note that the first-burst property corresponds to the recursion property for $i = 1$, but without the error term: as $C_0 = Y_0 = 0$, we have $C_1 = 1 + x_1(\text{DET})$ and $(1 + r) \cdot r = 1 + c$. The recursion property is vacuous whenever $\mathcal{I}_n(\text{DET}, \xi_n(\varepsilon))$ is empty, e.g., if $x_1(\text{DET}) \ge 1$. We will handle such deterministic algorithms in Section G.2 by a separate straightforward argument.

The lemma below states that classes $\mathcal{D}_\varepsilon^{(n)}$ contain all γ-dependable deterministic algorithms; its proof is deferred to Section G.1.

Lemma 22. *Consider a deterministic γ-dependable algorithm* DET*, an integer $n \ge 1$, and a real $\varepsilon > 0$. Then* $\text{DET} \in \mathcal{D}_\varepsilon^{(n)}$.

In Section G.2, we only need the properties of algorithms belonging to the class $\mathcal{D}_\varepsilon^{(2)}$ on sequences whose first gap is at most 1. In this case, Definition 21 reduces to the following statement.

Lemma 23. *Fix a real $\varepsilon > 0$ and a deterministic algorithm* $\text{DET} \in \mathcal{D}_\varepsilon^{(2)}$*, and let $x_1 := x_1(\text{DET})$. Then $x_1 \ge c$ and, for every integer $s \ge 1/\xi_2(\varepsilon)$ and every real $y \in (x_1, 1]$, it holds that*

$$\text{DET}(E(\langle 1, s\rangle, \langle y\rangle)) \ge (1+r) \cdot (1 + x_1 + r - y) - \varepsilon.$$

Proof. The bound $x_1 \ge c$ is the first-burst property of Definition 21. Next, fix s and y as in the statement. The instance $E(\langle 1, s\rangle, \langle y\rangle)$ is generated adaptively against DET, as its second burst

arrives at latency $y > x_1$ of the first one. It is $\xi_2(\varepsilon)$-flat, as $\theta_1 = 1$ and $\theta_2 = 1 + 1/s \leq 1 + \xi_2(\varepsilon)$, and its only gap satisfies $y \leq 1$ by the lemma assumption. Thus, it belongs to $\mathcal{I}_2(\textsc{Det}, \xi_2(\varepsilon))$, and therefore the recursion property of Definition 21 together with $C_1 = 1 + x_1$, $Y_1 = y$ and $C_2 = \textsc{Det}(E(\langle 1, s\rangle, \langle y\rangle))$ yields the lemma. □

G.1 Structural Consequences of Dependability

In this section, we prove Lemma 22. The idea behind our construction is the following: when DET acknowledges too early, the adversary can append burst after burst, each of them arriving just after DET acknowledges the previous one. We call such a chain of bursts a *penalizing suffix*. This forces DET to pay 1 for an acknowledgment in every round, while OPT may acknowledge the whole sequence at the last burst only. To remain γ-dependable, DET then has to make its acknowledgment latencies smaller and smaller, until this becomes impossible.

We make this argument applicable at an arbitrary moment of an arbitrary adaptively generated instance. More formally, we define the *post-ack budget*, a quantity which is computed on the basis of the history. Whenever it drops below $1 + r$, the adversary may append the penalizing suffix of bursts, which contradicts the γ-dependability of DET. As this construction can be applied after each burst of a flat instance, we obtain one bound per burst, and these bounds are exactly the inequalities stated in Definition 21.

Throughout this subsection, we fix deterministic algorithm DET with finite dependability. All instances considered below are generated adaptively against DET, and we let $\mathbf{x}(\textsc{Det}, \mathbf{s}, \mathbf{y}) = \langle x_1, \ldots, x_n\rangle$. Recall that $\textsc{Det}(\sigma_{\leq i}) = C_i$.

Lemma 24. *Let σ be an instance with n bursts generated adaptively against* DET. *For every $i \in \{1, \ldots, n\}$, it holds that* $\textsc{Opt}(\sigma_{\leq i}) \leq 1 + \sum_{j<i} \theta_j \cdot y_j$ *and* $\textsc{Opt}(\sigma_{\leq i}) \leq i$.

Proof. The term $1 + \sum_{j<i} \theta_j \cdot y_j$ is the cost of the schedule that serves $\sigma_{\leq i}$ by a single acknowledgment at the time of burst i. Such schedule pays 1 for the acknowledgment, and gathers latency $\theta_j \cdot y_j$ within the gap between bursts j and $j + 1$. Next, the term i is the cost of the schedule that serves $\sigma_{\leq i}$ by acknowledging at each of its i bursts. □

Post-ack budgets. For an instance σ with n bursts and $i \in \{1, \ldots, n\}$, we define the *budget at burst i* as

$$\Phi_i := \gamma \cdot (1 + \sum_{j<i} \theta_j \cdot y_j) - C_{i-1}. \tag{24}$$

By Lemma 24, the first term is at least $\gamma \cdot \textsc{Opt}(\sigma_{\leq i})$, i.e., at least the total cost which γ-dependability allows DET to spend on $\sigma_{\leq i}$. C_{i-1} is the amount which DET has already paid before burst i arrived, and thus Φ_i is the part which is not spent yet. Note that $\Phi_1 = \gamma$.

Lemma 25. *Let σ be an instance with n bursts generated adaptively against* DET. *If* DET *is γ-dependable, then for every $i \in \{1, \ldots, n\}$, $x_i \leq \Phi_i - 1$ and $x_i \leq i/r$.*

Proof. Fix $i \in \{1, \ldots, n\}$. As DET is γ-dependable, $C_i = \textsc{Det}(\sigma_{\leq i}) \leq \gamma \cdot \textsc{Opt}(\sigma_{\leq i})$. Thus, by Lemma 24 and (24), we have $C_i \leq \gamma \cdot (1 + \sum_{j<i} \theta_j \cdot y_j) = \Phi_i + C_{i-1}$. As $C_i = C_{i-1} + 1 + x_i$, the first inequality of the lemma follows.

By Lemma 24 again, $C_i \leq \gamma \cdot i$. As $C_{i-1} \geq i - 1$, we obtain $1 + x_i = C_i - C_{i-1} \leq \gamma \cdot i - (i - 1)$, and thus $x_i \leq (\gamma - 1) \cdot i = i/r$. □

By the budget definition (24), for every instance with n bursts and every $i < n$,

$$\Phi_{i+1} = \Phi_i + \gamma \cdot \theta_i \cdot y_i - (1 + x_i). \tag{25}$$

Assume that DET has just acknowledged burst i at latency x_i and that the adversary is about to issue burst $i + 1$. The cheapest continuation for the adversary is to place this burst at latency y_i which is arbitrarily close to x_i. By (25), the resulting budget Φ_{i+1} is then arbitrarily close to the amount

denoted Φ_i^+, which we call *post-ack budget at burst i*:

$$\begin{aligned}\Phi_i^+ &:= \Phi_i + \gamma \cdot \theta_i \cdot x_i - (1 + x_i) \\ &= \gamma \cdot (1 + \sum_{j<i} \theta_j \cdot y_j + \theta_i \cdot x_i) - C_i. \end{aligned} \tag{26}$$

The second equality above follows by (24) and $C_i = C_{i-1} + 1 + x_i$. It is worth noting that Φ_i^+ depends only on the history up to the acknowledgment of burst i by DET. In particular, it does not depend on the future choices of the adversary. By Lemma 25, $\Phi_i^+ \geq \gamma \cdot \theta_i \cdot x_i \geq 0$.

The penalizing suffix. Recall the definition (26) of the post-ack budget. The lemma below states that the post-ack budget of a γ-dependable algorithm can never drop below $1 + r$.

The idea behind the lemma is that $1+r$ is the fixed point of the mapping $z \mapsto \gamma \cdot (z-1)$, which, up to lower-order terms, describes how the post-ack budget evolves from burst to burst once the adversary appends bursts greedily. We show that if the post-ack budget falls below $1 + r$, then its distance to $1 + r$ grows geometrically with each subsequent burst. Eventually, the budget drops below zero, which contradicts Lemma 25.

We will use the following technical claim, whose proof is deferred to Section G.3.

Claim 26. *Fix reals $a \geq 2$ and $\beta \in (0, 2]$, and let $\ell := \lceil \log_a \frac{2}{\beta} \rceil$. If the sequence $z_0, z_1, \ldots, z_\ell$ satisfies*

$$\begin{aligned} z_0 &\leq \frac{a}{a-1} - 2 \cdot \beta, \\ z_{i+1} &\leq a \cdot (z_i - 1) + \beta \qquad\qquad \text{for every } i \in \{0, \ldots, \ell - 1\}, \end{aligned}$$

then $z_\ell < 0$.

Lemma 27. *Let σ be an instance with n bursts generated adaptively against* DET. *If* DET *is γ-dependable, then $\Phi_k^+ \geq 1 + r$ for every $k \in \{1, \ldots, n\}$.*

Proof. Assume towards contradiction that $\Phi_k^+ < 1 + r$ for some $k \leq n$. We show that a suitable adaptive extension of $\sigma_{\leq k}$ contradicts the γ-dependability of DET. To this end, we use the following constants.

- $\Delta := (1 + r) - \Phi_k^+$,
- $L := \lceil \log_\gamma(8/\Delta) \rceil$ and $M := k + L + 1$,
- $\delta := \Delta/(8 \cdot \gamma \cdot \theta_k)$,
- $\eta := \Delta/(8 \cdot \gamma \cdot (M/r + \delta))$.

As $0 < \Delta \leq 1 + r \leq 2$, all constants are positive and $L \geq 1$. Note that they depend on $\sigma_{\leq k}$ and on the behavior of DET on it only.

The extension consists of bursts $k + 1, \ldots, M$. For $m \in \{k + 1, \ldots, M\}$, once DET acknowledges burst $m - 1$ at latency x_{m-1}, the adversary issues burst m at latency $y_{m-1} := x_{m-1} + \delta$ of burst $m - 1$, and it chooses its size $s_m := \lceil (\sum_{j<m} s_j)/\eta \rceil$. Thus, $\theta_m \leq 1 + \eta$ for every $m \in \{k + 1, \ldots, M\}$, while the overheads $\theta_1, \ldots, \theta_k$ inherited from $\sigma_{\leq k}$ can be arbitrary.

We denote the resulting instance by σ'. As σ' and σ share the prefix $\sigma_{\leq k}$, DET acknowledges the first k bursts of σ' at the same latencies $x_1, \ldots, x_k$, and Φ_k and Φ_k^+ are the same for both instances. We first bound Φ_{k+1} as

$$\begin{aligned} \Phi_{k+1} &= \Phi_k + \gamma \cdot \theta_k \cdot (x_k + \delta) - (1 + x_k) && \text{(by (25) and } y_k = x_k + \delta) \\ &= \Phi_k^+ + \gamma \cdot \theta_k \cdot \delta && \text{(by the definition of } \Phi_k^+) \\ &= (1 + r) - \Delta + \Delta/8 && \text{(by the definitions of } \Delta \text{ and } \delta) \\ &\leq (1 + r) - \Delta/2. \end{aligned}$$

Next, for $m \in \{k+1, \dots, M-1\}$, we bound Φ_{m+1} in terms of Φ_m. By Lemma 25, $x_m \le m/r \le M/r$, and thus

$$\eta \cdot \gamma \cdot (x_m + \delta) \le \eta \cdot \gamma \cdot (M/r + \delta) = \Delta/8. \tag{27}$$

We obtain

$$\begin{aligned}
\Phi_{m+1} &= \Phi_m + \gamma \cdot \theta_m \cdot (x_m + \delta) - (1 + x_m) && \text{(by (25) and } y_m = x_m + \delta) \\
&\le \Phi_m + \gamma \cdot (1+\eta) \cdot (x_m + \delta) - (1 + x_m) && \text{(as } \theta_m \le 1 + \eta) \\
&= \Phi_m + (\gamma - 1) \cdot x_m - 1 + \gamma \cdot \delta + \gamma \cdot \eta \cdot (x_m + \delta) \\
&\le \Phi_m + (\gamma - 1) \cdot x_m - 1 + \Delta/8 + \Delta/8 && \text{(as } \theta_k \ge 1 \text{ and by (27))} \\
&\le \gamma \cdot (\Phi_m - 1) + \Delta/4,
\end{aligned}$$

where the last inequality follows by Lemma 25, i.e., $x_m \le \Phi_m - 1$.

Now we apply Claim 26 to the sequence $z_i := \Phi_{k+1+i}$ for $i \in \{0, \dots, L\}$, with $a = \gamma$, $\beta = \Delta/4$ and $\ell = \lceil \log_a(2/\beta) \rceil = L$; the two bounds above are its assumptions. The claim gives $\Phi_M - 1 = z_L - 1 < -1 < x_M$, which contradicts Lemma 25 for burst M. □

Constraints from all bursts. By applying Lemma 27 after every burst of a flat instance, we obtain one inequality per burst, which proves Lemma 22, restated below.

Lemma 22. *Consider a deterministic γ-dependable algorithm* DET*, an integer $n \ge 1$, and a real $\varepsilon > 0$. Then* DET $\in \mathcal{D}_\varepsilon^{(n)}$.

Proof. Consider an arbitrary instance σ generated adaptively against DET. By Lemma 27, for every burst i of σ, it holds that $\Phi_i^+ \ge 1 + r$, and thus (26) implies

$$C_i \le \gamma \cdot (1 + \sum_{j<i} \theta_j \cdot y_j + \theta_i \cdot x_i) - (1 + r). \tag{28}$$

We start with showing the first-burst property. On the instance $E(\langle 1 \rangle, \perp)$, we have $C_1 = 1 + x_1(\text{DET})$ and $\theta_1 = 1$, so (28) gives $C_1 \le \gamma \cdot C_1 - (1 + r)$. This gives us $C_1 \ge (1 + r)/(\gamma - 1) = (1 + r) \cdot r = 1 + c$, i.e., $x_1(\text{DET}) \ge c$.

It remains to show the recursion property. Let $\xi := \xi_n(\varepsilon)$, cf. Definition 21. Consider an instance $E(\mathbf{s}, \mathbf{y}) \in \mathcal{I}_n(\text{DET}, \xi)$ with $\mathbf{x}(\text{DET}, \mathbf{s}, \mathbf{y}) = \langle x_1, \dots, x_n \rangle$, and fix $i \in \{2, \dots, n\}$. As all gaps are at most 1, we have $Y_{i-1} \le i - 1$, and by Lemma 25, $x_i \le i/r$. Thus, $Y_{i-1} + x_i \le n \cdot (1 + 1/r)$, and hence

$$\gamma \cdot \xi \cdot (Y_{i-1} + x_i) \le \gamma \cdot \xi \cdot n \cdot (1 + 1/r) = \varepsilon/r. \tag{29}$$

Now, using (28) we have

$$\begin{aligned}
C_i &\le \gamma \cdot (1 + Y_{i-1} + x_i) + \gamma \cdot \xi \cdot (Y_{i-1} + x_i) - (1 + r) && \text{(as } \theta_j \le 1 + \xi) \\
&\le \gamma \cdot (1 + Y_{i-1} + x_i) - (1 + r) + \varepsilon/r && \text{(by (29))} \\
&= \gamma \cdot (Y_{i-1} + C_i - C_{i-1}) - (1 + r) + \varepsilon/r. && \text{(as } 1 + x_i = C_i - C_{i-1})
\end{aligned}$$

By reorganizing the terms, we get $(\gamma - 1) \cdot C_i \ge \gamma \cdot (C_{i-1} - Y_{i-1}) + (1 + r) - \varepsilon/r$. The lemma follows by multiplying both sides by $r = 1/(\gamma - 1)$ and using $r \cdot \gamma = 1 + r$. □

G.2 Randomized Lower Bound

The argument below uses properties of deterministic algorithm DET $\in \mathcal{D}_\varepsilon^{(2)}$ given by Lemma 23: the lower bound on $x_1(\text{DET})$ and the lower bound on its cost on a two-burst instance.

We note that the first bound alone already yields a nontrivial lower bound. Indeed, consider a γ-dependable randomized algorithm RAND. Every DET $\in$ supp(RAND) is γ-dependable, so $x_1(\text{DET}) \ge c$ by Lemma 22. On the instance $E(\langle 1 \rangle, \perp)$, OPT pays 1 and DET pays $1 + x_1(\text{DET}) \ge 1 + c$. Thus, the expected competitive ratio of RAND is at least $1 + c = r \cdot (1 + r)$. In what follows, we construct a distribution over instances on which the two bounds together imply Theorem 4, restated below for convenience.

Theorem 4. *Fix $r \in [\frac{\sqrt{5}-1}{2}, 1]$ and let $\gamma(r) := 1 + \frac{1}{r}$. Every $\gamma(r)$-dependable randomized algorithm for TCP acknowledgment has expected competitive ratio at least $\rho(r) = 1 + \frac{(2+r) \cdot r^{2+r}}{(1+r)^2 - r^{2+r}}$.*

Probability distribution. Probability distribution π_ε defined below consists of two parts: an atom on the single-packet instance and a density over two-burst instances, in which the second burst is large enough to make the instance flat, and the gap y between the two bursts ranges over $[c, r]$.

For $y \in [c, r]$, we write $w(y) := (1+r) - (1+y)/(1+r)$. The function w is decreasing, $w(c) = 1$ as $1 + c = r \cdot (1 + r)$, and $w(r) = r$, i.e., w maps $[c, r]$ onto $[r, 1]$. Let

$$q := \frac{(2+r) \cdot r^{1+r}}{(1+r)^2 - r^{2+r}} \text{ and}$$
$$p(y) := q \cdot \frac{(1+r-w(y)) \cdot w(y)^r}{r^{1+r}} = q \cdot \frac{(1+y) \cdot w(y)^r}{(1+r) \cdot r^{1+r}} \quad \text{for } y \in [c, r].$$

Note that $\rho = 1 + q \cdot r$. Fix a real $\varepsilon > 0$ and let $s := \lceil 1/\xi_2(\varepsilon) \rceil$, cf. Definition 21. We define π_ε as the following probability distribution:

- with probability q, the instance is $E_1 = E(\langle 1 \rangle, \perp)$;
- with density $p(y)$ on $y \in [c, r]$, the instance is $E_2 = E(\langle 1, s \rangle, \langle y \rangle)$.

Note that π_ε is indeed a probability distribution.

Claim 28. *It holds that* $q + \int_c^r p(y)\, dy = 1$.

Proof. We substitute $w = w(y)$, so that $dy = -(1+r)\, dw$, and w runs from 1 down to r as y runs from c to r. Thus,

$$\begin{aligned}
\int_c^r p(y)\, dy &= \frac{q \cdot (1+r)}{r^{1+r}} \cdot \int_r^1 (1+r-w) \cdot w^r\, dw \\
&= \frac{q \cdot (1+r)}{r^{1+r}} \cdot \left[w^{1+r} - \frac{w^{2+r}}{2+r} \right]_r^1 \\
&= \frac{q \cdot (1+r)}{r^{1+r}} \cdot \frac{(1+r) - 2 \cdot r^{1+r}}{2+r} \\
&= q \cdot \left(\frac{(1+r)^2}{(2+r) \cdot r^{1+r}} - \frac{2 \cdot (1+r)}{2+r} \right) \\
&= q \cdot \left(\frac{1}{q} + \frac{r}{2+r} - \frac{2 \cdot (1+r)}{2+r} \right) \qquad \text{(by the definition of } q\text{)} \\
&= q \cdot (1/q - 1) = 1 - q.
\end{aligned}$$

□

Below, we show that, up to an additive ε, the expected ratio on π_ε of any algorithm from $\mathcal{D}_\varepsilon^{(2)}$ which acknowledges the first burst at latency $t \in [c, r]$ is at least $F(t)$, where

$$F(t) := q \cdot (1+t) + \int_c^t p(y)\, dy + (1+r) \cdot \int_t^r \frac{1+t+r-y}{1+y} \cdot p(y)\, dy.$$

Claim 29. *It holds that* $F(t) = \rho$ *for every* $t \in [c, r]$.

Proof. By substituting $u = w(y)$, we obtain, for every $t \in [c, r]$,

$$\int_t^r w(y)^r\, dy = (1+r) \cdot \int_r^{w(t)} u^r\, du = w(t)^{1+r} - r^{1+r}.$$

Next, by the definition of p, we have

$$F(t) = q \cdot (1+t) + \int_c^t p(y)\, dy + \frac{q}{r^{1+r}} \cdot \int_t^r (1+t+r-y) \cdot w(y)^r\, dy.$$

We show that $F' = 0$ on $[c, r]$. Fix $t \in [c, r]$. At $y = t$, the integrand of the last integral equals $(1 + r) \cdot w(t)^r$, and its derivative with respect to t is $w(y)^r$. Thus,

$$\begin{aligned} F'(t) &= q + p(t) - \frac{q \cdot (1 + r) \cdot w(t)^r}{r^{1+r}} + \frac{q}{r^{1+r}} \cdot \int_t^r w(y)^r \, dy \\ &= q + p(t) - \frac{q \cdot (1 + r) \cdot w(t)^r}{r^{1+r}} + \frac{q \cdot w(t)^{1+r}}{r^{1+r}} - q \\ &= p(t) - q \cdot \frac{(1 + r - w(t)) \cdot w(t)^r}{r^{1+r}} \\ &= 0. && \text{(by the definition of } p\text{)} \end{aligned}$$

Thus, F is constant on $[c, r]$, and

$$\begin{aligned} F(t) = F(r) &= q \cdot (1 + r) + \int_c^r p(y) \, dy \\ &= q \cdot (1 + r) + 1 - q && \text{(by Claim 28)} \\ &= \rho && \text{(by the definition of } \rho\text{).} \end{aligned}$$

□

Lemma 30. *For every deterministic algorithm* $\textsc{Det} \in \mathcal{D}_\varepsilon^{(2)}$*, it holds that* $\mathbb{E}_{\sigma \sim \pi_\varepsilon}[R(\textsc{Det}, \sigma)] \geq \rho - \varepsilon$.

Proof. Consider a deterministic algorithm $\textsc{Det} \in \mathcal{D}_\varepsilon^{(2)}$ and let $x_1 := x_1(\textsc{Det})$; by Lemma 23, $x_1 \geq c$. We may assume that x_1 is finite, as otherwise $R(\textsc{Det}, E_1) = \infty$ and the lemma holds trivially. We bound the ratio of DET in the following three cases.

- On E_1: DET pays $1 + x_1$ and OPT pays 1, so $R(\textsc{Det}, E_1) = 1 + x_1$.
- On E_2 with $y \leq x_1$: the second burst arrives before DET acknowledges the first one, so Lemma 23 does not apply, and we use the trivial bound $R(\textsc{Det}, E_2) \geq 1$. This case covers also all algorithms with $x_1 \geq 1$.
- On E_2 with $y > x_1$: as $y \leq r \leq 1$, Lemma 23 yields $\textsc{Det}(E_2) \geq (1+r) \cdot (1+x_1+r-y) - \varepsilon$, and OPT may acknowledge once at the second burst, paying $1 + y$. Thus, as $1 + y \geq 1$,
$$R(\textsc{Det}, E_2) \geq \frac{(1 + r) \cdot (1 + x_1 + r - y) - \varepsilon}{1 + y} \geq \frac{(1 + r) \cdot (1 + x_1 + r - y)}{1 + y} - \varepsilon.$$

First, consider the case $x_1 > r$. As y is chosen from $[c, r]$, we have $y \leq r < x_1$, so the second case above applies to every two-burst instance, and $\mathbb{E}_{\sigma \sim \pi_\varepsilon}[R(\textsc{Det}, \sigma)] \geq q \cdot (1 + x_1) + (1 - q) = 1 + q \cdot x_1 > 1 + q \cdot r = \rho$.

Second, consider the case $x_1 \in [c, r]$. The three cases above yield

$$\begin{aligned} \mathbb{E}_{\sigma \sim \pi_\varepsilon}[R(\textsc{Det}, \sigma)] &\geq F(x_1) - \varepsilon \cdot \int_{x_1}^r p(y) \, dy \\ &\geq F(x_1) - \varepsilon \\ &= \rho - \varepsilon. && \text{(by Claim 29)} \end{aligned}$$

□

Proof of Theorem 4. Fix a real $\varepsilon > 0$. By Lemma 22, every γ-dependable deterministic algorithm belongs to $\mathcal{D}_\varepsilon^{(2)}$, and by Lemma 30, each of them has expected ratio at least $\rho - \varepsilon$ on π_ε. The min-max principle (Lemma 6) with $\mathcal{D} = \mathcal{D}_\varepsilon^{(2)}$ gives the same bound for every γ-dependable randomized algorithm. As $\varepsilon > 0$ can be arbitrarily small, the theorem follows. □

G.3 Proofs of the Technical Claims

Claim 26. *Fix reals* $a \geq 2$ *and* $\beta \in (0, 2]$*, and let* $\ell := \lceil \log_a \frac{2}{\beta} \rceil$*. If the sequence* $z_0, z_1, \ldots, z_\ell$ *satisfies*

$$\begin{aligned} z_0 &\leq \frac{a}{a - 1} - 2 \cdot \beta, \\ z_{i+1} &\leq a \cdot (z_i - 1) + \beta && \text{for every } i \in \{0, \ldots, \ell - 1\}, \end{aligned}$$

then $z_\ell < 0$.

Proof. For every $i \in \{0, \ldots, \ell\}$, we define the *gap* $g_i := \frac{a-\beta}{a-1} - z_i$. Using $a \geq 2$, we have

$$g_0 = \frac{a-\beta}{a-1} - z_0 \geq \frac{a-\beta}{a-1} - \frac{a}{a-1} + 2 \cdot \beta = -\frac{\beta}{a-1} + 2 \cdot \beta \geq -\beta + 2 \cdot \beta = \beta,$$

and for every $i \in \{0, \ldots, \ell - 1\}$,

$$\begin{aligned} g_{i+1} &= \frac{a-\beta}{a-1} - z_{i+1} \geq \frac{a-\beta}{a-1} - a \cdot (z_i - 1) - \beta \\ &= \frac{a-\beta}{a-1} + a - \beta - a \cdot z_i = a \cdot \left(\frac{a-\beta}{a-1} - z_i \right) = a \cdot g_i. \end{aligned}$$

Thus, $g_i \geq a^i \cdot \beta$ for every $i \in \{0, \ldots, \ell\}$. In particular, $g_\ell \geq a^\ell \cdot \beta \geq (2/\beta) \cdot \beta = 2$. Furthermore, $a/(a-1) \leq 2$ as $a \geq 2$, and thus

$$z_\ell = \frac{a}{a-1} - \frac{\beta}{a-1} - g_\ell \leq 2 - \frac{\beta}{a-1} - 2 < 0. \qquad \square$$