

Continuous Time Channels with Interference

Ioana Ivan* Michael Mitzenmacher† Justin Thaler‡ Henry Yuen§

Abstract

Khanna and Sudan [4] studied a natural model of continuous time channels where signals are corrupted by the effects of both noise and delay, and showed that, surprisingly, in some cases both are not enough to prevent such channels from achieving unbounded capacity. Inspired by their work, we consider channels that model continuous time communication with adversarial delay errors. The sender is allowed to subdivide time into an arbitrarily large number M of micro-units in which binary symbols may be sent, but the symbols are subject to unpredictable delays and may interfere with each other. We model interference by having symbols that land in the same micro-unit of time be summed, and we study k -interference channels, which allow receivers to distinguish sums up to the value k . We consider both a channel adversary that has a limit on the maximum number of steps it can delay each symbol, and a more powerful adversary that only has a bound on the average delay.

We give precise characterizations of the threshold between finite and infinite capacity depending on the interference behavior and on the type of channel adversary: for max-bounded delay, the threshold is at $D_{\max} = \Theta(M \log(\min\{k, M\}))$, and for average bounded delay the threshold is at $D_{\text{avg}} = \Theta(\sqrt{M \min\{k, M\}})$.

1 Introduction

We study continuous time channels with adversarial delay errors in the presence of interference. Our models are inspired by recent work of Khanna and Sudan [4], who studied continuous-time channels in the presence of both delay errors and (signal) noise errors. In this model, the communicating parties can subdivide time as finely as they wish. In each subdivided unit of time a 0 or 1 can be sent, but the sent signals are subject to unpredictable *delays*. Khanna and Sudan found (surprisingly) that the channel capacity in their model is finitely bounded only if at least one of the two sources of error (delay or signal noise) is adversarial. However, they assumed that at any instant in time, the receiver observes the *sum* of the signals delivered.

In this paper, we observe that the behavior of the channel changes dramatically if one accounts for the possibility of interference, and that this holds even in the absence of signal noise. Our model of interference is very simple; the symbols received at each time unit are summed, and the receiver sees the exact sum if it is less than k , but values greater than k cannot be distinguished from each other.

At a high level, our results are two-fold. First, we show that delay errors in the presence of interference are surprisingly powerful. Second, in the context of delay errors with interference, we find that seemingly innocuous modeling decisions can have large effects on channel behavior.

Related Work. Typically a communication channel is modeled as follows. The channel takes as input a signal f , modeled as a function from some domain \mathcal{T} to some range \mathcal{R} , and the channel outputs a received signal $\tilde{f} : \mathcal{T} \rightarrow \mathcal{R}$, which is a noisy version of f . For discrete time channels, \mathcal{T} is a finite domain $\{0, \dots, T-1\}$ where T is the time duration, and for continuous-time channels, \mathcal{T} is a continuous domain such as the interval $[0, T]$. For discrete signal

*MIT Computer Science and Artificial Intelligence Laboratory, ioanai@mit.edu.

†Harvard University, School of Engineering and Applied Sciences, michaelm@eecs.harvard.edu. This work was supported by NSF grants CCF-0915922 and IIS-0964473.

‡Harvard University, School of Engineering and Applied Sciences, jthaler@seas.harvard.edu. Supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, and in part by NSF grants CCF-0915922 and IIS-0964473.

§MIT Computer Science and Artificial Intelligence Laboratory, hyuen@csail.mit.edu. Supported by an MIT Presidential Fellowship.

channels, \mathcal{R} is a finite set such as $\{0, 1\}$, and for continuous signal channels, \mathcal{R} is an infinite set such as the interval $[0, 1]$.

Shannon showed in the discrete time setting, the capacity of the channel is finite even if the signal is continuous, as long as there is signal noise [6]. Nyquist [5] and Hartley [3] showed that even in the continuous time setting, the capacity is finite if one places certain restrictions on the Fourier spectrum of the signal.

Most relevant to us is recent work by Khanna and Sudan [4], which introduced continuous-time channels with signal noise and delay errors. They modeled their channel as the limit of a discrete process, and found that the capacity of their channel is infinite unless at least one of the error sources is adversarial.

Our work differs from previous work in several ways. We consider channels which introduce delays adversarially, but we additionally consider a very simple model of interference. We also consider two limitations on the adversary: one where maximum delay for any symbol is bounded, and one where the average delay over all symbols is bounded. In both cases, we find that our channels display a clean threshold behavior.

We believe that the adversarial setting presented here offers a clean initial analysis of the interference model, already with surprising results. A next natural step would be to analyze the effect of *random* delays in the presence of interference, and we leave this question as an interesting direction for future work.

2 Model and Summary of Results

Modeling Time. Following [4], we model continuous time as the limit of a discrete process. More specifically, the sender and receiver may send messages that last a duration of T units of time, but also can divide every unit of time into M subintervals called *micro-intervals*, and the sender may send one bit per micro-interval. We refer to M as the *granularity of time*, and refer to a sequence of M micro-intervals as a *macro-interval*. We call T the *message duration* of the channel. A codeword c sent over the channel is therefore represented as $c \in \{0, 1\}^{MT}$.

Modeling Delays. The effect of the channel on a sent codeword $c \in \{0, 1\}^{MT}$ is to delay symbols of c by some amount, e.g. the i th symbol of c may be moved to the j th timestep of the received codeword, where $j \geq i$. The delay process is adversarial, where we assume that the adversary knows the encoding/decoding scheme of the sender and receiver, and both the symbols that get delayed and the amount they are delayed can depend on the codeword that is sent. We formalize the notions of *max-bounded delay* and *average-bounded delay* below.

Modeling Interference. If multiple symbols are delivered at the same time step, there are several natural ways the channel could behave. In [4], the receiver observes the sum of all bits delivered at that instant of time; we call this the *sum channel*. Another obvious choice is for the receiver to see the OR of all bits delivered at that instant in time; we call this the *OR channel*.

We generalize these two models to what we call the *k-interference* channel. If there are fewer than k 1s delivered at an instant in time, the receiver will see the exact number of 1s delivered; otherwise the receiver will only see that at least k 1s have arrived. Thus, the sum channel can be viewed as the ∞ -interference channel, and the OR channel as the 1-interference channel. We consider k -interference channels as k varies between the extremes of 1 and ∞ , and may depend on the granularity of time M . We call the parameter k the *collision resolution* of the channel.

Valid Codebooks. For any fixed channel and codeword c , we let $B(c)$ denote the set of possible received strings corresponding to c . For any time T , we say a codebook $C \subseteq \{0, 1\}^{MT}$ is *valid* for a channel if for any $c \neq c'$ in C , $B(c) \cap B(c') = \emptyset$. Informally, this means that the adversary cannot cause the decoder to confuse c with c' for any other codeword c' .

Rate and Capacity. For any fixed granularity of time M and time T , let $s_{M,T} := \log |C(M, T)|$, where $|C(M, T)|$ denotes the size of largest valid codebook $C(M, T) \subseteq \{0, 1\}^{MT}$ for the channel. The *capacity of the channel at granularity M* is defined as $R(M) = \limsup_{T \rightarrow \infty} \{s_{M,T}/T\}$. The *capacity of the channel* is defined as $\limsup_{M \rightarrow \infty} R(M) = \limsup_{M \rightarrow \infty} [\limsup_{T \rightarrow \infty} \{s_{M,T}/T\}]$. We stress that the order of the limits in the definition of the channel capacity is crucial, as we show in Section 5.

Encoding: For every T and M , the sender encodes $s_{T,M}$ bits as MT bits by applying an encoding function $E_T : \{0, 1\}^{s_{T,M}} \rightarrow \{0, 1\}^{MT}$. The encoded sequence is denoted X_1, \dots, X_{MT} .

Delay: The delay is modeled by a delay function $\Delta : [MT] \rightarrow \mathbb{Z}^{\geq 0}$, where $\mathbb{Z}^{\geq 0}$ denotes the non-negative integers. The delay function has to satisfy a constraint depending on the type of delay channel we have:

- *Max-bounded delay:* For all $i \in [MT]$, $\Delta(i) \leq D_{\max}$, where D_{\max} is the bound on the maximum delay.
- *Average-bounded delay:* $\sum_i \Delta(i) \leq MT \cdot D_{\text{avg}}$, where D_{avg} is the bound on the average delay.

Received Sequence. The final sequence seen by the receiver given delay Δ , is $Y_1, \dots, Y_{MT} \in \mathbb{Z}^{\geq 0}$, where $Y_i := \min\{k, \sum_{j \leq i \text{ s.t. } j + \Delta(j) = i} X_j\}$ and k is the collision resolution parameter of the channel. We will ignore the symbols that get delayed past timestep MT .

For brevity, we use the shorthand AVG- k channel and MAX- k channel, where the meaning is clear.

2.1 Summary of Results

We prove that in the case of max-bounded delay, the capacity is finite if $D_{\max} = \Omega(M \log(\min\{k, M\}))$, and infinite otherwise. In contrast, we prove that in the case of average-bounded delay, the capacity is finite if $D_{\text{avg}} = \Omega(\sqrt{M \cdot \min\{k, M\}})$, and infinite otherwise.

We also consider a number of variant channels and observe that seemingly innocuous modeling choices cause the behavior to change drastically. In particular, we consider settings where the granularity of time is allowed to grow with the message duration, and where adversarial signal noise can also be added. For brevity, we provide a few specific interesting results.

3 Max-Bounded Delay Channel

We give a precise characterization of the infinite/finite capacity threshold of the MAX- k channel. Here and throughout, k refers to the collision resolution parameter, and M to the granularity of time.

Theorem 3.1. *If D_{\max} is the max-delay bound for the MAX- k channel, then the capacity of the channel is infinite when $D_{\max} = o(M \log(\min\{k, M\}))$, and the capacity is finite when $D_{\max} = \Omega(M \log(\min\{k, M\}))$.*

Proof. Infinite capacity regime. Suppose $D_{\max} = cM \log(\min\{k, M\})$ for $c = o(1)$ (here, c denotes a function of M that is subconstant in M).

Assume for simplicity that $1/c$ is an integer. Also assume that $c \geq \frac{1}{\log k}$ and $k \leq M$, as smaller values of c and larger values of k only make communication easier. We give a valid codebook of size $s = 2^{T/2c}$, showing $R(M) = \omega(1)$, and thus the capacity is infinite. Given a message $x \in \{0, 1\}^{T/2c}$, the sender breaks the message x into blocks of length $\log k$. The sender then encodes each block independently, using $2cM \log k$ bits for each block as described below. The resulting codeword has length $\frac{T}{2c \log k} \cdot 2cM \log k = TM$ as desired.

A block is encoded as follows. Since each block is $\log k$ bits long, we interpret the block as an integer y , $1 \leq y \leq k$. The sender encodes the block as a string of $2cM \log k$ bits, where the first $y \leq k$ bits in the string are 1s, and all remaining bits are 0s. To decode the j 'th block of the sent message, the receiver simply looks at the j 'th set of $2cM \log k$ bits in the received string, and decodes the block to the binary representation of y , where y is the total count of 1s received in those $2cM \log k$ bits.

Since the maximum delay is bounded by $cM \log k$, and 1s only occur as the first $k \leq M \leq cM \log k$ locations of each sent block, any 1-bit must be delivered within its block. Furthermore, the count of 1 bits is preserved, because at most k 1 bits collide within a block. Correctness of the decoding algorithm follows.

Finite Capacity Regime. Suppose the delays have bounded maximum $D_{\max} = cM \log(\min\{k, M\})$, with $c = \Omega(1)$. We give an adversary who ensures that there at most $O(\log k)$ bits of information are transmitted every $c \log k$ macro-timesteps. Thus, for $c = \Omega(1)$, the rate is bounded above by $O(\frac{1}{c}) = O(1)$ for all values of M , and hence the capacity is finite.

Assume first that $k \leq M$. The adversary breaks the sent string into blocks of length D_{\max} , and delays every sent symbol to the end of its block. The adversary clearly never introduces a delay longer than D_{\max} micro-timesteps.

Each received block can only take $k + 1$ values: all bits of the received block will be 0, except for the last symbol which can take any integer value between 0 and k . Thus, only $O(\log k)$ bits of information are transmitted every $D_{\max} = cM \log k$ micro-timesteps, or $c \log k$ macro-timesteps, demonstrating finite capacity.

If $k > M$, then the adversary is the same as above, where the block size is $D_{\max} = cM \log M$. Each received block can only take one of $cM \log M + 1$ values, since all bits of the block are 0, except for the last symbol which may vary between 0 and $cM \log M$. Thus, only $\log(cM \log M) = O(c \log M)$ bits of information are transmitted every $c \log M$ macro-timesteps, completing the proof. \square

4 Average-Bounded Delay Channel

We now study the behavior of the AVG- k channel.

Theorem 4.1. *If D_{avg} is the average-delay bound for the AVG- k channel, then the capacity of the channel is infinite when $D_{\text{avg}} = o(\sqrt{M \min\{k, M\}})$, and the capacity is finite when $D_{\text{avg}} = \Omega(\sqrt{M \min\{k, M\}})$.*

Proof. Infinite capacity regime. Suppose $D_{\text{avg}} = c\sqrt{Mk}$, where $c = o(1)$ (that is, again, c is a function of M that is subconstant in M). Let T be the message duration. Assume without loss of generality that $c \geq \frac{1}{Mk}$ and $k \leq M$ (smaller values of c and larger values of k only make communication easier).

Suppose the sender wants to send a message $x \in \{0, 1\}^{s_{T,M}}$ with $s_{T,M} = T/c$. As in [4, Lemma 4.1], we use a *concatenated code*: we assume that x has already been encoded under a classical error-correcting code C that corrects a 1/5-fraction of adversarial errors (or any other constant less than 1/4), as this will only affect the rate achieved by our scheme by a constant factor. C is then concatenated with the following inner code, which is tailored for resilience against delay errors: each bit of x gets encoded into a block of length $2\ell = 2cM$: 0's map to 2ℓ 0's (called a 0-block), and 1's map to ℓ 1's followed by ℓ 0's (called a 1-block). The resulting codeword is thus MT symbols long as required.

For decoding, let $Y = Y_1, \dots, Y_{MT}$ be the received word. The receiver divides Y into blocks of length ℓ . Let $\gamma(i) = \sum_{j \in [iMT, \dots, (i+1)MT-1]} Y_j$ denote the number of 1s encountered in the i th block. The receiver decodes Y as a message $y \in \{0, 1\}^{s_{T,M}}$ where y_i is declared to be 1 if $\gamma(i) \geq \sqrt{\ell k}$, 0 otherwise. Notice $\sqrt{\ell k} \geq 1$. Finally, the receiver will decode y using the outer decoder to obtain the original message. By the error-correcting properties of the outer code C , it suffices to show that at least 4/5ths of the inner-code blocks get decoded correctly.

We use a potential argument to demonstrate that the adversary can afford to corrupt a vanishingly small fraction of the blocks. We maintain a potential function $\Phi(i)$ that measures the total amount of delay the adversary can apply after performing the i th action (where an action is delaying a single symbol some distance). Initially, $\Phi(0) = MT D_{\text{avg}}$.

Turning a 0-block into a 1-block requires the adversary to delay at least $\sqrt{\ell k}$ 1 symbols from some previous block at least a distance $\ell/2$, so this requires reducing Φ by $\Omega(\ell^{3/2}\sqrt{k})$. To turn a 1-block into a 0-block, the adversary can either 1) move 1 symbols out of the 1-block (*evicting* 1s), or 2) *collide* 1s within the 1-block, or 3) a combination of both. We show that any combination requires reducing Φ by $\Omega(\ell^{3/2}\sqrt{k})$ as well.

Suppose the adversary chooses to corrupt a 1-block by evicting δ 1 symbols, and colliding the remaining 1 symbols so that at most $\sqrt{\ell k}$ 1s remain. The adversary minimizes the amount of delays it spends to do this by evicting the last δ 1s from a block, and choosing α equally spaced ‘‘collision points’’ (CPs) within the remaining 1s, where each remaining 1 symbol is delayed to the nearest CP ahead of it. Evicting δ 1s out of the block requires the adversary to spend at least $\delta\ell$ delays. Each CP receives $(\ell - \delta)/\alpha$ 1 symbols, and the amount of delays spent per CP is $1 + 2 + \dots + (\ell - \delta)/\alpha = \Theta\left(\frac{(\ell - \delta)^2}{\alpha^2}\right)$. Thus, the total amount of delay spent by the adversary to corrupt the 1-block is $\Omega\left(\frac{(\ell - \delta)^2}{\alpha^2} + \delta\ell\right)$. This is minimized when $\delta = 0$, i.e. when no symbols are evicted. Since $\alpha k \leq \sqrt{\ell k}$ (because each CP will have value k in the received string if at least k 1s are delivered at that index), the adversary needs to use $\Omega(\ell^{3/2}\sqrt{k})$ units of potential in order to corrupt a 1-block.

In our analysis, the minimum potential reduction $\Omega(\ell^{3/2}\sqrt{k})$ accounts for corrupting at most a block and its adjacent neighbor. Thus, the maximum number of blocks corruptable is $2\Phi(0)/\Omega(\ell^{3/2}\sqrt{k}) = O(c(M/\ell)^{3/2}T) = O(T/\sqrt{c})$. Since the original codeword had a total of T/c blocks, the maximum fraction of blocks corruptable is $O(\sqrt{c})$. However, $c = o(1)$, so a vanishingly small fraction of blocks are corrupted, and the original message can be recovered. Thus, we have constructed a valid codebook of size $2^{\Omega(T/c)}$, and this implies that the capacity is infinite.

Finite capacity regime. Suppose the delays have bounded average $D_{\text{avg}} = c\sqrt{M \min\{k, M\}}$, for some constant c . We will assume for simplicity that $c = 1$ and $k \leq M$, and explain how to handle smaller values of c and larger values of k later. We show the capacity is finite by specifying an adversary who ensures that there are a constant number of possible received strings for almost every macro-timestep.

To accomplish this, the adversary will break the sent string into blocks of length M . It scans the blocks sequentially, and adds and removes 1s so that each block will have 1s only at indices that are multiples of D_{avg} , or at the very last index of the block. The adversary ensures that it can always add 1s when it needs to by maintaining a “bank” of delayed 1s from previous blocks that will have size between D_{avg} and $2D_{\text{avg}}$ 1s whenever possible. The bank will always be small enough so that it does not contribute too many delays to the average. Once the bank reaches size D_{avg} , its size never falls below this level again. We show that the amount of information transmitted before the bank reaches this size is negligible for large T .

The adversary considers each block in turn, and its actions falls into four cases. Let ℓ denote the number of 1s in the block, and let s denote the size of the bank at the start of the block.

1. If $\ell \leq D_{\text{avg}}$ (we call the block *light*):
 - (a) If $s \geq D_{\text{avg}} + k - \ell$, the adversary will delay all 1s in the block until the final index within the block. If $\ell < k$, the adversary will also deliver $k - \ell$ 1s from the bank at the final index to ensure that the value of the final index is k . When this step completes, the size of the bank will be between D_{avg} and s .
 - (b) If $s < D_{\text{avg}} + k - \ell$, the adversary adds all 1s in the block to the bank, ensuring that the received block consists entirely of 0s. When this step completes, the bank has size least s and at most $D_{\text{avg}} + k \leq 2D_{\text{avg}}$.
2. If $\ell > D_{\text{avg}}$ (we call the block *heavy*):
 - (a) If $s \leq D_{\text{avg}}$, the adversary adds $D_{\text{avg}} - s < \ell$ of the new 1s to the bank, and it delays the rest of the 1s to the nearest integer multiple of D_{avg} .
 - (b) Otherwise, s will be at least D_{avg} . The adversary will place k 1s at every location which is an integer multiple of D_{avg} using bits from its bank (this requires at most $kM/D_{\text{avg}} = kM/\sqrt{kM} = D_{\text{avg}}$ bits), and delays the first $\ell - D_{\text{avg}}$ 1s within the current block to the nearest integer multiple of D_{avg} . The last D_{avg} 1s get added to the bank to replace the 1s lost from the bank, so the bank stays at size s .

We argue that at most $\sqrt{Mk} \log k + O(T)$ bits of information are transmitted over T blocks by the above scheme. Once the bank reaches size D_{avg} , there are only three possible values for each received block: the all-zeros vector; the vector that is all 0s except for the final index which has value k ; and the vector that is all 0s except for indices which are integer multiples of D_{avg} , which have value exactly k . Before the bank reaches size D_{avg} , any light block is still received as either the first or second possibility just described. Finally, at most one heavy block is encountered before the bank reaches size D_{avg} , and this block can take on at most $k^{M/D_{\text{avg}}} \leq k^{\sqrt{Mk}}$ possible values. Thus, over all T blocks, at most $\sqrt{Mk} \log k + O(T)$ bits of information are transmitted, and hence the capacity is finite.

Finally, we bound the average delay incurred by the adversary. For each block, we separately bound the total delays incurred by the symbols banked at the beginning of the block and symbols within the block. The symbols within any light block are responsible for total delay at most MD_{avg} , since at most D_{avg} symbols are delayed at most M . The symbols within in any heavy block are responsible for total delay at most $2MD_{\text{avg}}$, since all but at most D_{avg} 1s are delayed only until the nearest integer multiple of D_{avg} , and the rest are delayed at most M . As the bank contains at most $2D_{\text{avg}}$ 1s, banked symbols contribute at most $2MD_{\text{avg}}$ total delays per block. The adversary therefore spends at most $4MD_{\text{avg}}$ total delays per block, for an average delay of $4D_{\text{avg}}$. To reduce this to D_{avg} , we modify the above construction to use a block length of $M/16$ micro-timesteps, decreasing the average delay appropriately while increasing the rate by only a constant factor.

It remains to explain how to handle cases $c < 1$ and $k > M$. If $c < 1$, we simply decrease the block size further, from $M/16$ to $Mc^2/16$. This decreases the average delay by a factor of c and increases the rate by only a constant factor. For $k > M$, we note the adversary described above never delivers more than M 1s at any particular micro-timestep. Thus, even if $k > M$, the the received string is the same as it would be if $k = M$. \square

5 Extensions and Alternative Models

5.1 The Order of the Limits Matters

Under the definition of capacity used in the sections above and in [4], $\limsup_{M \rightarrow \infty} \limsup_{T \rightarrow \infty} \{k_{M,T}/T\}$, the sender and receiver are not allowed to let the granularity of time grow with T . If we instead define the capacity to be $\limsup_{T \rightarrow \infty} \limsup_{M \rightarrow \infty} \{k_{M,T}/T\}$, then the channel would behave very differently. Conceptually, the reason is that if M is allowed to grow with T , the sender and receiver can choose M to be so much larger than T that a vast amount of information (relative to T) can be encoded in just the first macro-timestep, avoiding interference issues.

To demonstrate one place where this interchange of limits alters the channel capacity, we show the AVG-1 channel behaves differently under this definition.

Theorem 5.1. *If one interchanges the order of limits in the definition of channel capacity, then the capacity of the AVG-1 channel with $D_{\text{avg}} = o(M)$ is infinite.*

Proof. The idea is that the sender encodes $\omega(1)$ bits of information via the location of the *first* 1 in the entire codeword. More formally, suppose $D_{\text{avg}} = cM - 1$ with $c = o(1)$, and let $c' = \sqrt{c/2}$. Assume for simplicity that Mc' is an integer. We will construct a valid codebook $\mathcal{C} \subseteq \{0, 1\}^{MT}$ with $|\mathcal{C}| = \Omega(1/c') = \omega(1)$ such that for each message $x \in \mathcal{C}$, the last $T - 1$ macro-timesteps consist only of 0s. Thus, we only specify the first macro-timestep in each codeword x . In the first codeword, the first macro-timestep will simply be Mc' 0s followed by $M - Mc'$ 1s. In the second codeword, the first macro-timestep will be $2Mc'$ 0s followed by $M - 2Mc'$ 1s. In general, in the i th codeword, the first macro-timestep will be iMc' 0s followed by $M - iMc'$ 1s.

The decoder will look at the position L of the left-most 1 in the received string and output the largest i such that $iMc' \leq L$.

In order for the adversary to force the decoder to decode incorrectly, the decoder has to make the first 1 appear at least $c' \cdot M$ positions later than it does in the sent string. For this to happen, the adversary has to spend at least $1 + 2 + \dots + Mc' \geq M^2 c'^2 / 2$ delays in total. So the average delay has to be at least $\frac{Mc'^2}{2T} = \frac{Mc}{4T}$. For fixed T , this is $\Omega(M)$. \square

5.2 Adding noise

In this section we note that the *combination* of interference with noise yields a max-bounded adversary that is surprisingly potent.

Theorem 5.2. *Suppose the adversary is allowed to flip t bits per macro-timestep, and delay each bit a maximum of D_{max} micro-timesteps. Then the capacity of the 1-interference channel is finite if $D_{\text{max}} \cdot t = \Omega(M)$, and is infinite if $D_{\text{max}} \cdot t = o(M)$. In particular, the capacity is finite if $t = D_{\text{max}} = \Omega(\sqrt{M})$.*

Proof. Finite capacity regime. Suppose for simplicity that $D_{\text{max}} \cdot t = M$. The key observation is that the adversary can turn *any* macro-timestep into the unique string consisting of all zeros, except for 1s at indices which are integer multiples of D_{max} , while staying within its budget. The adversary breaks each macro-timestep into t blocks of length D_{max} , and delays each bit to the end of its block. If no 1s are sent in a block, the adversary flips a single bit in the block to create a 1. This totals at most t bit-flips per macro-timestep, giving the result.

Infinite capacity regime. Suppose $D_{\text{max}} \cdot t = o(M)$. We again use a concatenated code to construct a valid codebook of size $2^{\Omega(\frac{MT}{D_{\text{max}}})}$. The sender starts with a string $x \in \{0, 1\}^{\frac{MT}{D_{\text{max}}+1}}$ encoded under a classical error-correcting code with constant rate which can tolerate up to a $1/5$ fraction of adversarial errors.

The sender then replaces each bit of x with a block of length $D_{\text{max}} + 1$: if $x_i = 0$, the i 'th block is set to the all-zeros string, and if $x_i = 1$, the i 'th block is set to the string consisting of a 1 followed by D_{max} zeros. The decoder decodes a block to 1 if the block contains at least one 1, and decodes a block to 0 otherwise. It suffices to show that at least a $4/5$ fraction of the blocks get decoded correctly.

Call a block *dirty* if even a single bit within it is flipped, and call the block *clean* otherwise. Since the adversary can afford to flip only $T \cdot t = o(\frac{MT}{D_{\text{max}}})$ bits of the course of the entire message, only an $o(1)$ fraction of blocks are dirty. For any clean block, the adversary cannot afford to delay the first bit in the block beyond the final bit in the

block. Thus, any clean block will get decoded correctly. It follows that decoding will always be successful, yielding the result. \square

6 Discussion

We studied a variety of natural models for continuous-time channels with delay errors in the presence of interference. Our results show that these channels exhibit a clean threshold behavior. We note our finite capacity results hold even for *computationally simple* adversaries in the sense of Guruswami and Smith [2]; that is, our adversaries process the sent string sequentially in linear time, using just $O(M)$ space. Our results can be viewed as a counterweight to those of Khanna and Sudan [4], by showing that other natural additional restrictions can lead to finite capacity in their model.

Many questions remain for future work. Our results only address adversarial delays; random delays under our interference model remains open. One might also consider different models of interference, or different limitations on the delays introduced by the adversary.

References

- [1] J. Aspnes, E. Blais, M. Demirbas, R. O’Donnell, A. Rudra, and S. Uurtamo. k^+ decision trees - (extended abstract). In C. Scheideler, editor, *ALGOSENSORS*, volume 6451 of *Lecture Notes in Computer Science*, pages 74–88. Springer, 2010.
- [2] V. Guruswami and A. Smith. Codes for computationally simple channels: Explicit constructions with optimal rate. In *FOCS*, pages 723–732. IEEE Computer Society, 2010.
- [3] R. V. L. Hartley. Transmission of information. *Bell Syst. Tech. Journal*, 7:535–563, 1928.
- [4] S. Khanna and M. Sudan. Delays and the capacity of continuous-time channels. In R. Ostrovsky, editor, *FOCS*, pages 758–767. IEEE, 2011.
- [5] H. Nyquist. Certain Factors Affecting Telegraph Speed. *Bell System Technical Journal*, page 324, 1924.
- [6] C. E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, Jan. 1949.