Network Coding Gaps for Completion Times of Multiple Unicasts

Bernhard Haeupler
Carnegie Mellon University
haeupler@cs.cmu.edu

David Wajc
Stanford University
wajc@stanford.edu

Goran Zuzic
Carnegie Mellon University
gzuzic@cs.cmu.edu

2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-7281-9621-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/FOCS46700.2020.00053

***Abstract*—We study network coding gaps for the problem of makespan minimization of multiple unicasts. In this problem distinct packets at different nodes in a network need to be delivered to a destination specific to each packet, as fast as possible. The *network coding gap* specifies how much coding packets together in a network can help compared to the more natural approach of routing.**

While makespan minimization using routing has been intensely studied for the multiple unicasts problem, no bounds on network coding gaps for this problem are known. We develop new techniques which allow us to upper bound the network coding gap for the makespan of k unicasts, proving this gap is at most polylogarithmic in k. Complementing this result, we show there exist instances of k unicasts for which this coding gap is polylogarithmic in k. Our results also hold for average completion time, and more generally any ℓ_p norm of completion times.

***Keywords*-network coding; coding gap; multiple unicast; unicast; completion time**

I. Introduction

In this paper we study the natural mathematical abstraction of what is arguably the most common network communication problem: *multiple unicasts*. In this problem, distinct packets of different size are at different nodes in a network, and each packet needs to be delivered to a specific destination as fast as possible. That is, minimizing the *makespan*, or the time until all packets are delivered.

All known multiple-unicast solutions employ (fractional) *routing* (also known as store-and-forward protocols), i.e., network nodes potentially subdivide packets and route (sub-)packets to their destination via store and forward operations, while limited by edge capacities. The problem of makespan minimization of routing has been widely studied over the years. A long line of work [1–9], starting with the seminal work of Leighton, Maggs, and Rao [8], studies makespan minimization for routing along fixed paths. The study of makespan minimization for routing (with the freedom to pick paths along which to route) resulted in approximately-optimal routing, first for asymptotically-large packet sizes [10], and then for all packet sizes [3].

It seems obvious at first that routing packets, as though they were physical commodities, is the only way to solve network communication problems, such as multiple unicasts. Surprisingly, however, results discovered in the 2000s [11] suggest that information need not flow through a network like a physical commodity. For example, nodes might not just forward information, but instead send out XORs of received packets. Multiple such XORs or linear combinations can then be recombined at destinations to reconstruct any desired packets. An instructive example is to look at the XOR $C \oplus M$ of two s-bit packets, C and M. While it is also s bits long, one can use it to reconstruct either all s bits of C or all s bits of M, as long as the other packet is given. Such network coding operations are tremendously useful for network communication problems, but they do not have a physical equivalent. Indeed, the $C \oplus M$ packet would correspond to some s ounces of a magic "café latte" liquid with the property that one can extract either s ounces of milk or s ounces of coffee from it, as long as one has enough of the other liquid already. Over the last two decades, many results demonstrating gaps between the power of network coding and routing have been published (e.g., [11–18]). Attempts to build a comprehensive theory explaining what is or is not achievable by going beyond routing have given rise to an entire research area called network information theory.

The question asked in this paper is:

> "How much faster than routing can network coding be for any multiple-unicast instance?"

In other words, what is the (multiplicative) *network coding gap* for makespan of multiple unicasts. Surprisingly, no general makespan coding gap bounds were known prior to this work. This is in spite of the vast amount of effort invested in understanding routing strategies for this problem, and ample evidence of the benefits of network coding.

This question was studied in depth for the special case of *asymptotically-large* packet sizes, otherwise known as *throughput* maximization (e.g., [12–15, 19–24]). Here, the maximum *throughput* of a multiple-unicast instance can be defined as $\sup_{w\to\infty} w/C(w)$, where $C(w)$ is the makespan of the fastest protocol for the instance after increasing all packet sizes by a factor of w (see the full version). In the throughput setting, no instances are known where coding offers *any* advantage over routing, and this is famously conjectured to be the case for all instances [13, 19]. This

[0]This paper's full version is available at https://arxiv.org/abs/1905.02805.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00053

conjecture, if true, has been proven to have surprising connections to various lower bounds [21, 25, 26]. Moreover, by the work Afshani et al. [25], a throughput coding gap of $o(\log k)$ for all multiple-unicast instances with k unicast pairs (k-*unicast instances*, for short) would imply explicit *super-linear circuit lower bounds*—a major breakthrough in complexity theory. Such a result is currently out of reach, as the best known upper bound on throughput coding gaps is $O(\log k)$, which follows easily from the same bound on multicommodity flow/sparsest cut gaps [27, 28].

In this work we prove makespan coding gaps for the general problem of *arbitrary* packet sizes. In particular, we show that this gap is at most $O(\log^2 k)$ for any k-unicast instance (for the most interesting case of similar-sized packet sizes). We note that any coding gap upper bound for this more general setting immediately implies the same bound in the throughput setting (see the full version), making our general bound only quadratically larger than the best known bound for the special case of throughput. Complementing our results, we prove that there exist k-unicast instances where the network coding gap is $\Omega(\log^c k)$ for some constant $c > 0$.

To achieve our results we develop novel techniques that might be of independent interest. The need for such new tools is due to makespan minimization for general packet sizes needing to take both source-sink distances as well as congestion issues into account. This is in contrast with the throughput setting, where bounds must only account for congestion, since asymptotically-large packet sizes make distance considerations inconsequential. For our more general problem, we must therefore develop approaches that are both congestion- and distance-aware. One such approach is given by a new combinatorial object we introduce, dubbed the *moving cut*, which allows us to provide a *universally optimal* characterization of the coding makespan. That is, it allows us to obtain tight bounds (up to polylog terms) on the makespan of *any* given multiple-unicast instance. We note that moving cuts can be seen as generalization of prior approaches that were (implicitly) used to prove unconditional lower bounds in distributed computing on specially crafted networks ([29, 30]); the fact they provide a *characterization* on all networks and instances is novel. This underlies our main result—a polylogarithmic upper bound on the makespan coding gap for any multiple-unicast instance.

A. Preliminaries

In this section we define the completion-time communication model. We defer the, slightly more general, information-theoretic formalization to the full version.

A *multiple-unicast instance* $\mathcal{M} = (G, \mathcal{S})$ is defined over a communication network, represented by a connected undirected graph $G = (V, E)$ with capacity $c_e \in \mathbb{Z}_{\geq 1}$ for each edge e. The $k \triangleq |\mathcal{S}|$ *sessions* of $\mathcal{M}$ are denoted by $\mathcal{S} = \{(s_i, t_i, d_i)\}_{i=1}^{k}$. Each session consists of source node s_i, which wants to transmit a packet to the sink t_i, consisting of $d_i \in \mathbb{Z}_{\geq 1}$ sub-packets. Without loss of generality we assume that a uniform sub-packetization is used; i.e., all sub-packets have the same size (think of sub-packets as the underlying data type, e.g., field elements or bits). For brevity, we refer to an instance with k sessions as a k-*unicast* instance.

A *protocol* for a multiple-unicast instance is conducted over finitely-many *synchronous time steps*. Initially, each source s_i knows its packet, consisting of d_i sub-packets. At any time step, the protocol instructs each node v to send a different packet along each of its edges e. The packet contents are computed with some predetermined function of packets received in prior rounds by v or originating at v. *Network coding protocols* are unrestricted protocols, allowing each node to send out any function of the packets it has received so far. On the other hand, *routing protocols* are a restricted, only allowing a node to forward sub-packets which it has received so far or that originate at this node.

We say a protocol for multiple-unicast instance has *completion times* $(T_1, T_2, \ldots, T_k)$ if for each $i \in [k]$, after T_i time steps of the protocol the sink t_i can determine the d_i-sized packet of its source s_i. The complexity of a protocol is determined by functions $\mathcal{C} : \mathbb{R}_{\geq 0}^{k} \to \mathbb{R}_{\geq 0}$ of its completion times. For example, a protocol with completion times $(T_1, T_2, \ldots, T_k)$ has *makespan* $\max_{i \in [k]} T_i$ and *average completion time* $(\sum_{i \in [k]} T_i)/k$. Minimizing these measures is a special case of minimizing weighted ℓ_p norms of completion time, namely minimizing $(\sum_{i \in [k]} w_i \cdot T_i^p)^{1/p}$ for some $\vec{w} \in \mathbb{R}^k$ and $p \in \mathbb{R}_{\geq 0}$.

Since coding protocols subsume routing ones, for any function $\mathcal{C}$ of completion times, and for any multiple-unicast instance, the fastest routing protocol is no faster than the fastest coding protocol. Completion-time coding gaps characterize how much faster the latter is.

Definition I.1. *(Completion-time coding gaps) For any function $\mathcal{C} : \mathbb{R}_{\geq 0}^{k} \to \mathbb{R}_{\geq 0}$ of completion times, the* network coding gap for $\mathcal{C}$ *for a k-unicast instance $\mathcal{M} = (G, \mathcal{S})$ is the ratio of the smallest $\mathcal{C}$-value of any routing protocol for $\mathcal{M}$ and the smallest $\mathcal{C}$-value of any network coding protocol for $\mathcal{M}$.*

We note that the multiple-unicast instance problem can be further generalized, so that each edge has both capacity and *delay*, corresponding to the amount of time needed to traverse the edge. This more general problem can be captured by replacing each edge e with a path with unit delays of total length proportional to e's delay. As we show, despite path length being crucially important in characterizing completion times for multiple-unicast instances, this transformation does not affect the worst-case coding gaps, which are independent of the network size (including after

this transformation). We therefore consider only unit-time delays in this paper, without loss of generality.

B. Our Contributions

In this work we show that completion-time coding gaps of multiple unicasts are vastly different from their throughput counterparts, which are conjectured to be trivial (i.e., equal to one). For example, while the throughput coding gap is always one for instances with $k = 2$ sessions [31], for makespan it is easy to derive instances with $k = 2$ sessions and coding gap of $4/3$ (based on the well-known butterfly network example in network coding theory). Having observed that makespan coding gaps can in fact be nontrivial, we proceed to study the potential asymptotic growth of such coding gaps as the network parameters grow. We show that the makespan coding gap of multiple unicasts with k sessions and packet sizes $\{d_i\}_{i\in[k]}$ is polylogarithmic in the problem parameters, k and $\sum_i d_i/\min_i d_i$, but independent of the network size, n. The positive part of this result is given by the following theorem.

Theorem I.2. *The network coding gap for makespan of any k-unicast instance is at most*

$$O\left(\log(k)\cdot\log\left(\sum_i d_i/\min_i d_i\right)\right).$$

For similarly-sized packets, this bound simplifies to $O(\log^2 k)$. For different-sized packets, our proofs and ideas in [32] imply a coding gap of $O(\log k\cdot\log(nk))$. Moreover, our proofs are constructive, yielding for any k-unicast instance $\mathcal{M}$ a routing protocol which is at most $O(\log k\cdot\log(\sum_i d_i/\min_i d_i))$ and $O(\log k\cdot\log(nk))$ times slower than the fastest protocol (of any kind) for $\mathcal{M}$. We note that our upper bounds imply the same upper bounds for throughput (see the full version). Our bounds thus also nearly match the best coding gap of $O(\log k)$ known for this special case of makespan minimization.

On the other hand, we prove that a polylogarithmic gap as in Theorem I.2 is necessary, by providing an infinite family of multiple-unicast instances with unit-sized packets ($d_i = 1$ for all $i \in [k]$) exhibiting a polylogarithmic makespan coding gap (see the full version for a proof).

Theorem I.3. *There exists an absolute constant $c > 0$ and an infinite family of k-unicast instances whose makespan coding gap is at least $\Omega(\log^c k)$.*

Building on our results for makespan we obtain similar results to Theorems I.2 and I.3 for average completion time and more generally for any weighted ℓ_p norm of completion times (see the full version).

C. Techniques

Here we outline the challenges faced and key ideas needed to obtain our results, focusing on makespan.

1) Upper Bounding the Coding Gap: As we wish to bound the ratio between the best makespan of any routing protocol and any coding protocol, we need both upper and lower bounds for these best makespans. As it turns out, upper bounding the best makespan is somewhat easier. The major technical challenge, and our main contribution, is in deriving lower bounds on the optimal makespan of any given multiple-unicast instance. Most notably, we formalize a technique we refer to as the *moving cut*. Essentially the same technique was used to prove that distributed verification is hard on one *particular* graph that was designed specifically with this technique in mind [29, 30, 33]. Strikingly, we show that the moving cut technique gives an almost-tight characterization (up to polylog factors) of the coding makespan for *every* multiple-unicast instance (i.e., it gives *universally* optimal bounds).

We start by considering several prospective techniques to prove that no protocol can solve an instance in fewer than T rounds, and build our way up to the moving cut. For any multiple-unicast instance, $\max_{i\in[k]}\text{dist}(s_i,t_i)$, the maximum distance between any source-sink pair, clearly lower bounds the coding makespan. However, this lower bound can be arbitrarily bad since it does not take edge congestion into account; for example, if all source-sink paths pass through one common edge. Similarly, any approach that looks at sparsest cuts in a graph is also bound to fail since it does not take the source-sink distances into account.

Attempting to interpolate between both bounds, one can try to extend this idea by noting that a graph that is "close" (in the sense of few deleted edges) to another graph with large source-sink distances must have large makespan for routing protocols. For simplicity, we focus on instances where all capacities and demands are one, i.e., $c_e = 1$ for every edge e and $d_i = 1$ for all i, which we refer to as *simple* instances. The following simple lemma illustrates such an approach.

Lemma I.4. *Let $\mathcal{M} = (G, \mathcal{S})$ be a simple k-unicast instance. Suppose that after deleting some edges $F \subseteq E$, any sink is at distance at least T from its source; i.e., $\forall i \in [k]$ we have $\text{dist}_{G\setminus F}(s_i, t_i) \geq T$. Then any routing protocol for $\mathcal{M}$ has makespan at least $\min\{T, k/|F|\}$.*

Proof: For any sets of flow paths between all sinks and source, either (1) all source-sink flow paths contain at least one edge from F, incurring a congestion of $k/|F|$ on at least one of these $|F|$ edges, or (2) there is a path not containing any edge from F, hence having a hop-length of at least T. Either way, any routing protocol must take at least $\min\{T, k/|F|\}$ to route along these paths. ∎

Perhaps surprisingly, the above bound does not apply to general (i.e., coding) protocols. Consider the instance in Figure 1. There, removing the single edge $\{S, T\}$ increases the distance between any source-sink pair to 5, implying any routing protocol's makespan is at least 5 on this in-

stance. However, there exists a network coding protocol with makespan 3: Each source s_i sends its input to its neighbor S and all sinks t_j for $i \neq j$ along the direct 3-hop path $s_i - - - t_j$. Node S computes the XOR of all inputs, passes this XOR to T who, in turn, passes this XOR to all sinks t_j, allowing each sink t_j to recover its source s_j's packets by canceling all other terms in the XOR.

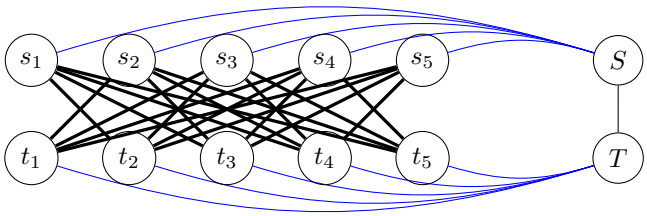

Figure 1: A family of instances with $k = 5$ pairs of terminals and makespan coding gap of $5/3$. Thick edges represent paths of 3 hops, while thin (black and blue) edges represent single edges. In other words, each of the k sources s_i has a path of 3 hops (in black and bold) connecting it to every sink t_j for all $j \neq i$. Moreover, all sources s_i neighbor a node S, which also neighbors node T, which neighbors all sinks t_j.

One can still recover a valid general (i.e., coding) lower bound by an appropriate strengthening of Lemma I.4: one has to require that *all* sources be far from *all* sinks in the edge-deleted graph. This contrast serves as a good mental model for the differences between coding and routing protocols.

Lemma I.5. *Let $\mathcal{M} = (G, \mathcal{S})$ be a simple k-unicast instance. Suppose that after deleting some edges $F \subseteq E$, any sink is at distance at least T from **any** source; i.e., $\forall \boldsymbol{i}, \boldsymbol{j} \in [k]$ we have $dist_{G \setminus F}(\boldsymbol{s_i}, \boldsymbol{t_j}) \geq T$. Then any (network coding) protocol for $\mathcal{M}$ has makespan at least* $\min\{T, k/|F|\}$.

Proof: We can assume all sources can share information among themselves for free (e.g., via a common controlling entity) since this makes the multiple-unicast instance strictly easier to solve; similarly, suppose that the sinks can also share information. Suppose that some coding protocol has makespan $T' < T$. Then all information shared between the sources and the sinks has to pass through some edge in F at some point during the protocol. However, these edges can pass a total of $|F| \cdot T'$ packets of information, which has to be sufficient for the total of k source packets. Therefore, $|F| \cdot T' \geq k$, which can be rewritten as $T' \geq k/|F|$. The makespan is therefore at least $T' \geq \min\{T, k/|F|\}$. ∎

Unfortunately, Lemma I.5 is not always tight and it is instructive to understand when this happens. One key example is the previously-mentioned instance studied in the influential distributed computing papers [29, 30, 33] (described in Figure 2), where congestion and dilation both play key roles. Informally, this network was constructed precisely to give an $\tilde{\Omega}(\sqrt{n})$ makespan lower bound (leading to the pervasive $\tilde{\Omega}(\sqrt{n} + D)$ lower bound for many global problems in distributed computing [29]). The intuitive way to explain the $\tilde{\Omega}(\sqrt{n})$ lower bound is to say that one either has to communicate along a path of length $\sqrt{n}$ or *all* information needs to shortcut significant distance over the tree, which forces all information to pass through near the top of the tree, implying congestion of $\tilde{\Omega}(\sqrt{n})$. Lemma I.5, however, can at best certify a lower bound of $\tilde{\Omega}(n^{1/4})$ for this instance. That is, this lemma's (coding) makespan lower bound can be *polynomially* far from the optimal coding protocol's makespan.

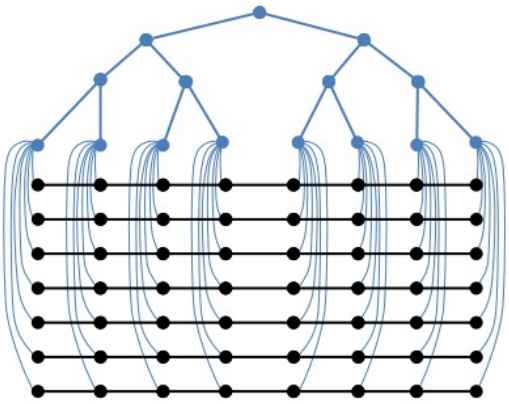

Figure 2: The hard instance for distributed graph problems [29, 30, 33], as appears in [34]. The multiple-unicast instance has $\Theta(n)$ nodes and is composed of $\sqrt{n}$ disjoint paths of length $\sqrt{n}$ and a perfectly balanced binary tree with $\sqrt{n}$ leaves. The i^{th} node on every path is connected to the i^{th} leaf in the tree. There are $\sqrt{n}$ sessions with s_i, t_i being the first and last node on the i^{th} path. All capacities and demands are one. The graph's diameter is $\Theta(\log n)$, but its coding makespan is $\tilde{\Omega}(\sqrt{n})$.

A more sophisticated argument is needed to certify the $\tilde{\Omega}(\sqrt{n})$ lower bound for this specific instance. The aforementioned papers [29, 30, 33] prove their results by implicitly using the technique we formalize as our moving cut in the following definition and lemma (proven in Section II-A).

Definition I.6. *(Moving cut) Let $G = (V, E)$ be a communication network with capacities $c : E \to \mathbb{Z}_{\geq 1}$ and let $\{(s_i, t_i) \mid i \in [k]\}$ be source-sink pairs. A **moving cut** is an assignment $\ell : E \to \mathbb{Z}_{\geq 1}$ of positive integer lengths to edges of G. We say the moving cut has **capacity** C, if $\sum_{e \in E} c_e(\ell_e - 1) = C$, and **distance** T, if all sinks and sources are at distance at least T with respect to ℓ; i.e., $\forall i, j \in [k]$ we have $d_\ell(s_i, t_j) \geq T$.*

Lemma I.7. *Let $\mathcal{M} = (G, \mathcal{S})$ be a unicast instance which admits a moving cut ℓ of capacity strictly less than $\sum_{i \in [k]} d_i$ and distance T. Then* any *(coding) protocol for $\mathcal{M}$ has makespan at least T.*

Lemma I.7 can be seen as a natural generalization of Lemma I.5, which can be equivalently restated in the following way: *"Suppose that after increasing each edge e's length from one to $\ell_e \in \{\mathbf{1}, \boldsymbol{T+1}\}$, we have that (1) $\sum_{e\in E} c_e(\ell_e - 1) < \sum_{i=1}^{k} d_i$, and (2) $dist_\ell(s_i, t_j) \geq T$. Then any (coding) protocol $\mathcal{M}$ has makespan at least T"*. Dropping the restriction on ℓ_e recovers Lemma I.7.

Strikingly, the moving cut technique allows us not only to prove tight bounds (up to polylog factors) for the instance of Figure 2—it allows us to get such tight bounds *for every multiple-unicast instance*. In order to upper bound the makespan coding gap, we therefore relate such a moving cut with the optimal routing makespan, as follows.

To characterize the optimal routing makespan, we study hop-bounded multicommodity flow, which is an LP relaxation of routing protocols of makespan T. First, we show that a fractional LP solution of high value to this LP implies a routing with makespan $O(T)$. Conversely, if the optimal value of this LP is low, then by strong LP duality this LP's dual has a low-valued solution, which we use to derive a moving cut and lower bound the coding makespan. Unfortunately, the dual LP only gives us bounds on (average) distance between source-sink pairs (s_i, t_i), and not between all sources s_i and sinks t_j (including $j \neq i$), as needed for moving cuts. For this conversion to work, we prove a generalization of the main theorem of Arora, Rao and Vazirani [35] to general metrics, of possible independent interest. (See Section II-C.) This allows us to show that a low-valued dual solution implies a moving cut certifying that no coding protocol has makespan less than $T/O(\log k \cdot \log(\sum_i d_i / \min_i d_i))$. As the above rules out low-valued optimal solutions to the LP for $T = T^* \cdot O(\log k \cdot \log(\sum_i d_i / \min_i d_i))$ with T^* the optimal coding makespan, the LP must have high optimal value, implying a routing protocol with makespan $O(T)$, and thus our claimed upper bound on the makespan coding gap.

2) Lower Bounding the Coding Gap: To complement our polylogarithmic upper bound on the makespan gap, we construct a family of multiple-unicast instances $\mathcal{M}$ that exhibit a polylogarithmic makespan coding gap. We achieve this by amplifying the gap via graph products, a powerful technique that was also used in prior work to construct extremal throughput network coding examples [36–38]. Here we outline this approach, as well as the additional challenges faced when trying to use this approach for makespan.

We use a graph product introduced by Braverman et al. [38] (with some crucial modifications). Braverman et al. [38] use their graph product to prove a conditional *throughput* coding gap similar to the one of Theorem I.3, conditioned on the (unknown) existence of a multiple-unicast instance I with non-trivial throughput coding gap. The graph product of [38] takes instances I_1, I_2 and intuitively replaces each edge of I_1 with a source-sink pair of a different copy of I_2. More precisely, multiple copies of I_1 and I_2 are created and interconnected. Edges of a copy of I_1 are replaced by the same session of different copies of I_2; similarly, sessions of a copy of I_2 replace the same edge in different copies of I_1. This product allows for coding protocols in I_1 and I_2 to compose in a straightforward way to form a fast coding protocol in the product instance. The challenge is in proving impossibility results for routing protocols, which requires more care in the definition of the product graph.

To address this challenge, copies of instances are interconnected along a high-girth bipartite graph to prevent unexpectedly short paths from forming after the interconnection. For example, to prove a throughput routing impossibility result, Braverman et al. [38] compute a dual of the multicommodity flow LP (analogous to our LP, but *without any hop restriction*) to certify a limit on the routing performance. In the throughput setting, a direct tensoring of dual LP solutions of I_1 and I_2 gives a satisfactory dual solution of the product instance. In more detail, a dual LP solution in I assigns a positive length $\ell_I(e)$ to each edge in I; each edge of the product instance corresponds to two edges $e_1 \in I_1$ and $e_2 \in I_2$, and the direct tensoring $\ell_+((e_1, e_2)) = \ell_{I_1}(e_1) \cdot \ell_{I_2}(e_2)$ provides a feasible dual solution with an adequate objective value. To avoid creating edges in the product distance of zero ℓ_+-length, they contract edges assigned length zero in the dual LP of either instance. Unfortunately for us, such contraction is out of the question when studying makespan gaps, as such contractions would shorten the hop length of paths, possibly creating short paths with no analogues in the original instance.

Worse yet, any approach that uses the dual of our T-hop-bounded LP is bound to fail in the makespan setting. To see why, suppose we are given two instances I_1, I_2, both of which have routing makespan at least T and expect that the product instance I_+ to have routing makespan at least T^2 by some construction of a feasible dual LP solution. Such a claim cannot be directly argued since a source-sink path in the product instance that traverses, say, $T-1$ different copies of I_2 along a path of hop-length $T+1$ in I_2 could carry an arbitrary large capacity! This is since the hop-bounded LP solution on I_2 only takes *short* paths, of hop-length at most T, into account. Since there is no direct way to compose the dual LP solutions, we are forced to use a different style of analysis from the one of [38], which in turn forces our construction to become considerably more complicated.

To bound the routing *makespan* in the product instance we rely on Lemma I.4: We keep a list of edges F along with each instance and ensure that (i) all source-sink distances in the F-deleted instance are large and that (ii) the ratio of the number of sessions k to $|F|$ is large. We achieve property (i) by interconnecting along a high-girth graph and treating the replacements of edges in F in a special way (hence deviating from the construction of [38]). Property (ii) is ensured by making the inner instance I_2 significantly

larger than the outer instance I_1, thus requiring many copies of I_1 and resulting in a large number of sessions in the product graph. To allow for this asymmetric graph product, we need an infinite number of base cases with non-trivial makespan coding gap for our recursive constructions (rather than a single base instance, as in the work of Braverman et al. [38]). This infinite family is fortunately obtained by appropriately generalizing the instance of Figure 1.

The main challenge in our approach becomes controlling the size of the product instance. To achieve this, we affix to each instance a relatively complicated set of parameters (e.g., coding makespan, number of edges, number of sessions, etc.) and study how these parameters change upon applying the graph product. Choosing the right set of parameters is key—they allow us to properly quantify the size escalation. In particular, we show that the coding gap grows doubly-exponentially and the size of the instances grow triply-exponentially, yielding the desired polylogarithmic coding gap.

D. Related Work

This work ties in to many widely-studied questions. We outline some of the most relevant here.

Routing multiple unicasts.: Minimizing the makespan of multiple unicasts using routing has been widely studied. When packets must be routed along fixed paths, two immediate lower bounds on the makespan emerge: *dilation*, the maximum length of a path, and *congestion*, the maximum number of paths crossing any single edge. A seminal result of Leighton et al. [8] proves one can route along such fixed paths in $O(\text{congestion}+\text{dilation})$ rounds, making the result optimal up to constants. Follow ups include works improving the constants in the above bound [2, 9], computing such protocols [7], simplifying the original proof [1], routing in distributed models [5, 39], and so on. When one has the freedom to choose paths, Bertsimas and Gamarnik [10] gave near-optimal routing solutions for asymptotically-large packet sizes, later extended to all packet sizes by Srinivasan and Teo [3]. The power of routing for multiple unicasts is therefore by now well understood.

Network coding gains.: The utility of network coding became apparent after Ahlswede et al. [11] proved it can increase the (single multicast) throughput of a communication network. Following their seminal work, there emerged a vast literature displaying the advantages of network coding over routing for various measures of efficiency in numerous communication models, including for example energy usage in wireless networks [40–42], delay minimization in repeated single unicast[43, 44], and makespan in gossip protocols [16–18]. The throughput of a single multicast (i.e., one single node sending to some set of nodes), arguably the simplest non-trivial communication task, was also studied in great detail (e.g., [11, 45, 46]). In particular, Agarwal and Charikar [46] showed that the throughput coding gap for a single multicast equals the integrality gap of natural min-weight Steiner tree relaxations, for which non-trivial bounds were known (see, e.g., [47, 48]). While the throughput coding gap for a single multicast is now fairly well understood, the case of multiple senders seems to be beyond the reach of current approaches.

Throughput gaps for multiple unicasts.: The routing throughput for multiple unicasts is captured by multicommodity max-flow, while the coding throughput is clearly upper bounded by the sparsest cut. Known multicommodity flow-cut gap bounds therefore imply the throughput coding gap for k unicasts is at most $O(\log k)$ [27, 28], and less for special families of instances [49–53]. In 2004, Li and Li [13] and Harvey et al. [19] independently put forward the *multiple-unicast conjecture*, which asserts that the throughput coding gap is trivial (i.e., it is one). This conjecture was proven true for numerous classes of instances [20, 21, 31]. More interestingly, a positive resolution of this conjecture has been shown to imply unconditional lower bounds in external memory algorithm complexity [21, 26], computation in the cell-probe model [21], and (recently) an $\Omega(n \log n)$ circuit size lower bound for multiplication of n-bit integers [25] (matching an even more recent breakthrough algorithmic result for this fundamental problem [54]). Given this last implication, it is perhaps not surprising that despite attempts by many prominent researchers [12, 15, 22, 23], the conjecture remains open and has established itself as a notoriously hard open problem. Indeed, even improving the $O(\log k)$ upper bound on throughput coding gaps seems challenging, and would imply unconditional *super-linear circuit size lower bounds*, by the work of Afshani et al. [25]. Improving our upper bound on makespan coding gaps to $o(\log k)$ would directly imply a similar improvement for throughput coding gaps, together with these far-reaching implications.

II. Upper Bounding the Coding Gap

In this section we prove Theorem I.2, upper bounding the makespan network coding gap. Given a multiple-unicast instance $\mathcal{M}$ we thus want to upper bound its routing makespan and lower bound its coding makespan. To characterize these quantities we start with a natural hop-bounded multicommodity flow LP, $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$, which serves as a "relaxation" of routing protocols of makespan at most T. The LP, given in Figure 3, requires sending a flow of magnitude $z \cdot d_i$ between each source-sink pair (s_i, t_i), with the additional constraints that (1) the combined congestion of any edge e is at most $T \cdot c_e$ where c_e is the capacity of the edge (as only c_e packets can use this edge during any of the T time steps of a routing protocol), and (2) the flow is composed of only *short* paths, of at most T hops. Specifically, for each $i \in [k]$, we only route flow from s_i to t_i along paths in $\mathcal{P}_i(T) \triangleq \{p : s_i \rightsquigarrow t_i \mid |p| \leq T, p \text{ is simple}\}$,

the set of simple paths of hop-length at most T connecting s_i and t_i in G.

Primal: $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$

maximize	z
subject to:	
$\forall i \in [k]$:	$\sum_{p \in \mathcal{P}_i(T)} f_i(p) \geq z \cdot d_i$
$\forall e \in E$:	$\sum_{p \ni e} f_i(p) \leq T \cdot c_e$
$\forall i \in [k], p$:	$f_i(p) \geq 0$

Dual: $\textsc{Cut}_{\mathcal{M}}(T)$

minimize	$T \cdot \sum_{e \in E} c_e \ell_e$
subject to:	
$\forall i \in [k], p \in \mathcal{P}_i(T)$:	$\sum_{e \in p} \ell_e \geq h_i$
	$\sum_{i \in [k]} d_i h_i \geq 1$
$\forall e \in E$:	$\ell_e \geq 0$
$\forall i \in [k]$:	$h_i \geq 0$

Figure 3: The concurrent flow LP relaxation and its dual.

A routing protocol solving $\mathcal{M}$ in T rounds yields a solution to $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ of value $z = 1$, almost by definition.[1] A partial converse is also true; a feasible solution to $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ of value at least $\Omega(1)$ implies a routing protocol for $\mathcal{M}$ in time $O(T)$. This can be proven using standard LP rounding [3] and $O(\text{congestion} + \text{dilation})$ path routing [8]. (See the full version of the paper).

Proposition II.1. *Let $z, \{f_i(p) \mid i \in [k], p \in \mathcal{P}_i(T)\}$ be a feasible solution for $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$. Then there exists an integral routing protocol with makespan $O(T/z)$.*

Complementing the above, we show that a low optimal LP value for $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ implies that no *coding* protocol can solve the instance in much less than T time.

Lemma II.2. *If the optimal value of $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ is at most $z^* \leq 1/10$, then the coding makespan for $\mathcal{M}$ is at least $T/(C \cdot \log(k) \cdot \log\left(\sum_i d_i / \min_i d_i\right))$ for some constant $C > 0$.*

Before outlining our approach for proving Lemma II.2, we show why this lemma together with Proposition II.1 implies our claimed upper bound for the makespan network coding gap.

Theorem I.2. *The network coding gap for makespan of any k-unicast instance is at most*

$$O\left(\log(k) \cdot \log\left(\sum_i d_i / \min_i d_i\right)\right).$$

[1] Such a protocol must send d_i packets along paths of length at most T between each source-sink pair (s_i, t_i), and it can send at most c_e packets through each edge e during any of the T rounds, or at most $c_e \cdot T$ packets overall.

Proof: Fix a multiple-unicast instance $\mathcal{M}$. Let T^* be the minimum makespan for any coding protocol for $\mathcal{M}$. Let $T = (C+1) \cdot T^* \cdot (\log(k) \cdot \log(\sum_i d_i / \min_i d_i))$ for C as in Lemma II.2. Then, the LP $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ must have optimal value at least $z^* \geq 1/10$, else by Lemma II.2 and our choice of T, any coding protocol has makespan at least $T/O(\log(k) \cdot \log(\sum_i d_i / \min_i d_i)) > T^*$, contradicting the definition of T^*. But then, by Proposition II.1, there exists a routing protocol with makespan $O(T/z^*) = O(T^* \cdot \log(k) \cdot \log(\sum_i d_i / \min_i d_i)$. The theorem follows. ∎

The remainder of the section is dedicated to proving Lemma II.2. That is, proving that a low optimal value for the LP $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ implies a lower bound on the makespan of any coding protocol for $\mathcal{M}$. To this end, we take a low-valued LP solution to the dual LP $\textsc{Cut}_{\mathcal{M}}(T)$ (implied by strong LP duality) and use it to obtain an information-theoretic certificate of impossibility, which we refer to as a *moving cut*. Section II-A introduces a framework to prove such certificates of impossibility, which we show *completely characterizes* any instance's makespan (up to polylog terms). We then explain how to transform a low-value dual LP solution to such a moving cut in Section II-B. For this transformation, we prove a lemma reminiscent of Arora et al. [35, Theorem 1] for general metrics, in Section II-C.

A. Moving Cuts: Characterizing Makespan

In this section we prove that moving cuts characterize the makespan of a multiple unicast instance. For ease of reference, we re-state the definition of moving cuts.

Definition I.6. *(Moving cut) Let $G = (V, E)$ be a communication network with capacities $c : E \to \mathbb{Z}_{\geq 1}$ and let $\{(s_i, t_i) \mid i \in [k]\}$ be source-sink pairs. A **moving cut** is an assignment $\ell : E \to \mathbb{Z}_{\geq 1}$ of positive integer lengths to edges of G. We say the moving cut has **capacity** C, if $\sum_{e \in E} c_e(\ell_e - 1) = C$, and **distance** T, if all sinks and sources are at distance at least T with respect to ℓ; i.e., $\forall i, j \in [k]$ we have $d_\ell(s_i, t_j) \geq T$.*

We start by proving Lemma I.7, whereby moving cuts with small capacity and large distance imply makespan lower bounds.

Lemma I.7. *Let $\mathcal{M} = (G, \mathcal{S})$ be a unicast instance which admits a moving cut ℓ of capacity strictly less than $\sum_{i \in [k]} d_i$ and distance T. Then* any *(coding) protocol for $\mathcal{M}$ has makespan at least T.*

Proof: We will show via simulation that a protocol solving $\mathcal{M}$ in at most $T - 1$ rounds would be able to compress $\sum_{i=1}^{k} d_i$ random bits to a strictly smaller number of bits, thereby leading to a contradiction. Our simulation proceeds as follows. We have two players, Alice and Bob, who control different subsets of nodes. In particular, if we denote by $A_r \triangleq \{v \in V \mid \min_i \text{dist}_\ell(s_i, v) \leq r\}$ the set of

nodes at distance at most r from any source, then during any round $r \in \{0, 1, \ldots, T-1\}$ all nodes in A_r are "spectated" by Alice. By spectated we mean that Alice gets to see all of these nodes' private inputs and received transmissions during the first r rounds. Similarly, Bob, at time r, spectates $B_r \triangleq V \setminus A_r$. Consequently, if at round r a node $u \in V$ spectated by Alice sends a packet to a node $v \in V$, then Bob will see the contents of that packet if and only if Bob spectates the node v at round $r+1$. That is, this happens only if $u \in A_r$ and $v \in B_{r+1} = V \setminus A_{r+1}$. Put otherwise, Bob can receive a packet from Alice along edge e during times $r \in [\min_i \mathrm{dist}_\ell(s_i, u), \min_i \mathrm{dist}_\ell(s_i, v) - 1]$. Therefore, the number of rounds transfer can happen along edge e is at most $\min_i \mathrm{dist}_\ell(s_i, v) - \min_i \mathrm{dist}_\ell(s_i, u) - 1 \leq \ell_e - 1$. Hence, the maximum number of bits transferred from Alice to Bob via e is $c_e(\ell_e - 1)$. Summing up over all edges, we see that the maximum number of bits Bob can ever receive during the simulation is at most $\sum_{e \in E} c_e(\ell_e - 1) < \sum_{i=1}^{k} d_i$. Now, suppose Alice has some $\sum_{i=1}^{k} d_i$ random bits. By simulating this protocol with each source s_i having (a different) d_i of these bits, we find that if all sinks receive their packet in T rounds, then Bob (who spectates all t_j at time $T-1$, as $\min_i \mathrm{dist}_\ell(s_i, t_j) \geq T$ for all j) learns all $\sum_{i=1}^{k} d_i$ random bits while receiving less than $\sum_{i=1}^{k} d_i$ bits from Alice—a contradiction. ∎

Lemma I.7 suggests the following recipe for proving makespan lower bounds: Prove a lower bound on the makespan of some sub-instance $\mathcal{M}' = (G, \mathcal{S}')$ with $\mathcal{S}' \subseteq \mathcal{S}$ induced by indices $I \subseteq [k]$ using Lemma I.7. As any protocol solving $\mathcal{M}$ solves $\mathcal{M}'$, a lower bound on the makespan of $\mathcal{M}'$ implies a lower bound on the makespan of $\mathcal{M}$. So, to prove makespan lower bounds for $\mathcal{M}$, identify a moving cut for some sub-instance of $\mathcal{M}$. If this moving cut has capacity less than the sum of demands of the sub-instance and distance at least T, then the entire instance, $\mathcal{M}$, has makespan at least T.

By the above discussion, the worst distance of a (low capacity) moving cut over any sub-instance serves as a lower bound on the makespan of any instance. The following lemma, whose proof is deferred to the end of Section II-B, asserts that in fact, the highest distance of a moving cut over any sub-instance is equal (up to polylog terms) to the best routing makespan of $\mathcal{M}$. Consequently, by Theorem I.2, the strongest lower bound obtained using moving cuts is equal up to polylog terms to the optimal (coding) makespan for $\mathcal{M}$.

Lemma II.3. *If a k-unicast instance $\mathcal{M}$ has no routing protocol with makespan T, then there exists a set of sessions $I \subseteq [k]$ with a moving cut of capacity strictly less than $\sum_{i \in I} d_i$ and distance at least $T/O(\log k \cdot (\sum_i d_i / \min_i d_i))$ with respect to the unicast sub-instance induced by I.*

We now turn to leveraging moving cuts to prove makespan lower bounds. Specifically, Lemma II.2.

B. From Dual Solution to Moving Cut

Proposition II.1 shows that high objective value for the primal LP, $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ implies an upper bound on the routing time for the given instance. In this section we prove the "converse", Lemma II.2, whereby low objective value of the primal LP implies a lower bound on the *coding* time for the given instance.

Our approach will be to prove that a low objective value of the primal LP —implying a feasible dual LP solution of low value—yields a moving cut for some sub-instance. This, by Lemma I.7, implies a lower bound on protocols for this sub-instance, and thus for the entire instance, from which we obtain Lemma II.2. We turn to converting a low-valued dual LP solution to such a desired moving cut.

By definition, a low-value feasible solution to the dual LP, $\textsc{Cut}_{\mathcal{M}}(T)$, assigns non-negative lengths $\ell : E \to \mathbb{R}_{\geq 0}$ such that (1) the c-weighted sum of ℓ-lengths is small, i.e., $\sum_{e \in E} c_e \ell_e = \tilde{O}(1/T)$, as well as (2) if h_i is the ℓ-length of the T-hop-bounded ℓ-shortest path between s_i and t_i, then $\sum_{i \in [k]} d_i \cdot h_i \geq 1$. Property (1) implies that appropriately scaling the lengths ℓ yields a moving cut given by lengths $\tilde{\ell}$ of bounded capacity, $\sum_e c_e \tilde{\ell}_e$. For this moving cut to be effective to lower bound the makespan of some sub-instance using Lemma I.7, the cut must have high distance w.r.t. this sub-instance. As a first step to this end, we use Property (2) to identify a subset of source-sink pairs $I \subseteq [k]$ with pairwise $\tilde{\ell}$-distance at least $\tilde{\Omega}(T)$. Claim II.4, proven in the full version, using a "continuous" bucketing argument, does just this. The claim introduces a loss factor α_{gap} that we use throughout this section.

Claim II.4. *Given sequences $h_1, \ldots, h_k, d_1, \ldots, d_k \in \mathbb{R}_{\geq 0}$ with $\sum_{i \in [k]} d_i \cdot h_i \geq 1$ there exists a non-empty subset $I \subseteq [k]$ with $\min_{i \in I} h_i \geq \frac{1}{\alpha_{\mathrm{gap}} \cdot \sum_{i \in I} d_i}$ for $\alpha_{\mathrm{gap}} \in \left[1, O\left(\log \frac{\sum_{i \in [k]} d_i}{\min_{i \in [k]} d_i}\right)\right]$.*

Scaling up the ℓ lengths yields a sub-instance induced by pairs $I \subseteq [k]$ and moving cut with bounded capacity and with $\tilde{\ell}$-distance between every source and its sink of at least $\tilde{\Omega}(T)$, i.e., $d_{\tilde{\ell}}(s_i, t_i) \geq \tilde{\Omega}(T)$ for all $i \in I$. However, Lemma I.7 requires $\tilde{\ell}$-distance $\tilde{\Omega}(T)$ between *any* source and sink, i.e., $d_{\tilde{\ell}}(s_i, t_j) \geq \tilde{\Omega}(T)$ for all $i, j \in I$. To find a subset of source-sink pairs with such distance guarantees, we rely on the following metric decomposition lemma, whose proof is deferred to Section II-C.

Lemma II.5. *Let (X, d) be a metric space. Given n pairs $\{(s_i, t_i)\}_{i \in [n]}$ of points in X with at most k distinct points (i.e., $|\bigcup_i \{(s_i, t_i)\}| \leq k$) and pairwise distances at least $d(s_i, t_i) \geq T$, there exists a subset of indices $I \subseteq [n]$ of size $|I| \geq \frac{n}{9}$ such that $d(s_i, t_j) \geq \frac{T}{O(\log k)}$ for all $i, j \in I$. Moreover, such a set can be computed in polynomial time.*

We are now ready to construct the moving cut.

Lemma II.6. *If the optimal value of* $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ *is at most* $z^* \leq 1/10$*, then there exists* $\tilde{I} \subseteq [k]$ *and a moving cut* $\tilde{\ell}$ *of capacity strictly less than* $\sum_{i\in\tilde{I}} d_i$ *and distance at least* $T/O(\alpha_{\text{gap}} \log k)$ *with respect to the unicast sub-instance induced by* $\tilde{I}$ *(i.e.,* $\tilde{\mathcal{M}} = (G, \tilde{\mathcal{S}})$ *where* $\tilde{\mathcal{S}} = \{(s_i, t_i, d_i)\}_{i\in\tilde{I}}$*).*

Proof: By strong duality, the dual LP $\textsc{Cut}_{\mathcal{M}}(T)$ has a feasible solution $\{h_i, \ell_e \mid i \in [k], e \in E\}$ to $\textsc{Cut}_{\mathcal{M}}(T)$ with objective value $T\sum_{e\in E} c_e\ell_e = z^*$. Fix such a solution. Let $I \subseteq [k]$ be a subset of indices as guaranteed by Claim II.4. Define $\tilde{\ell}_e \triangleq 1 + \lfloor \ell_e \cdot T \cdot \sum_{i\in I} d_i \rfloor$ for all $e \in E$ and note that $\tilde{\ell}_e \in \mathbb{Z}_{\geq 1}$. By definition of $\tilde{\ell}$ and $T\sum_e c_e\ell_e = z$, we get a bound on the capacity of $\tilde{\ell}$:

$$\sum_{e\in E} c_e(\tilde{\ell}_e - 1) \leq \sum_e c_e\ell_e \cdot T \cdot \sum_{i\in I} d_i$$
$$= z^* \cdot \sum_{i\in I} d_i \leq \frac{1}{10}\sum_{i\in I} d_i < \frac{1}{9}\sum_{i\in I} d_i.$$

We now show that $d_{\tilde{\ell}}(s_i, t_i) > T/\alpha_{\text{gap}}$ for all $i \in I$. Consider any simple path p between $s_i \rightsquigarrow t_i$. Denote by $\tilde{\ell}(p)$ and $\ell(p)$ the length with respect to $\tilde{\ell}$ and ℓ, respectively. It is sufficient to show that $\tilde{\ell}(p) > T/\alpha_{\text{gap}}$. If $p \notin \mathcal{P}_i(T)$, i.e., the hop-length of p (denoted by $|p|$) is more than T. Then $\tilde{\ell}(p) \geq |p| > T \geq T/\alpha_{\text{gap}}$, since $\tilde{\ell}_e \geq 1\ \forall e \in E$. Conversely, if $p \in \mathcal{P}_i(T)$, then by our choice of I as in Claim II.4 and the definition of h_i, we have that $\ell(p) \geq h_i \geq \frac{1}{\alpha_{\text{gap}}\sum_{i\in I} d_i}$, hence

$$\tilde{\ell}(p) \geq \ell(p) \cdot T \cdot \sum_{i\in I} d_i = \frac{T \cdot \sum_{i\in I} d_i}{\alpha_{\text{gap}} \sum_{i\in I} d_i} = T/\alpha_{\text{gap}}.$$

Finally, we choose a subset $\tilde{I} \subseteq I$ s.t. $d_{\tilde{\ell}}(s_i, t_j) > T/O(\alpha_{\text{gap}} \log k)$ for all $i, j \in \tilde{I}$. By Lemma II.5 applied to the graphic metric defined by $\tilde{\ell}$ and each pair (s_i, t_i) repeated d_i times, there exists a multiset of indices $\tilde{I} \subseteq I \subseteq [k]$ such that $d_{\tilde{\ell}}(s_i, t_j) \geq T/O(\alpha_{\text{gap}} \log k)$ for all $i, j \in \tilde{I}$ and such that $|\tilde{I}| \geq \sum_i d_i/9$. Therefore, taking each pair (s_i, t_i) indexed by $\tilde{I}$ at least once, we find a subset of sessions $\tilde{I} \subseteq [k]$ such that $\sum_{i\in\tilde{I}} d_i \geq \frac{1}{9}\sum_{i\in[k]} d_i > \sum_e c_e \cdot (\tilde{\ell}_e - 1)$ and $d_{\tilde{\ell}}(s_i, t_j) \geq T/O(\alpha_{\text{gap}} \log k)$ for all $i, j \in \tilde{I}$. In other words, $\tilde{\ell}$ is a moving cut of capacity strictly less than $\sum_{i\in\tilde{I}} d_i$ and distance $T/O(\alpha_{\text{gap}} \log k)$ with respect to the sub-instance induced by $\tilde{I}$. ∎

Combining Lemma II.6 with Lemma I.7, we obtain this section's main result, Lemma II.2.

Lemma II.2. *If the optimal value of* $\textsc{ConcurrentFlow}_{\mathcal{M}}(T)$ *is at most* $z^* \leq 1/10$*, then the coding makespan for* $\mathcal{M}$ *is at least* $T/(C \cdot \log(k) \cdot \log(\sum_i d_i / \min_i d_i))$ *for some constant* $C > 0$*.*

Remark 1.: We note that the $\log k$ term in Lemma II.2's bound is due to the $\log k$ term in the bound of Lemma II.5. For many topologies, including genus-bounded and minor-free networks, this $\log k$ term can be replaced by a constant (see Section II-C), implying smaller makespan gaps for such networks.

Remark 2.: Lemma II.3, which states that a lower bound on routing makespan implies the existence of a moving cut of high distance with respect to some sub-instance, follows by Lemma II.6 and Proposition II.1, as follows. By Proposition II.1, $\mathcal{M}$ having no routing protocol with makespan T implies that for some constant $c > 0$, the LP $\textsc{ConcurrentFlow}_{\mathcal{M}}(c \cdot T)$ has objective value at most $1/10$. Lemma II.6 then implies the existence of the moving cut claimed by Lemma II.3.

C. From Pairwise to All-Pairs Distances

This section is dedicated to a discussion and proof of the following Lemma that seems potentially useful beyond the scope of this paper.

Lemma II.5. *Let* (X, d) *be a metric space. Given* n *pairs* $\{(s_i, t_i)\}_{i\in[n]}$ *of points in* X *with at most* k *distinct points (i.e.,* $|\bigcup_i \{(s_i, t_i)\}| \leq k$*) and pairwise distances at least* $d(s_i, t_i) \geq T$*, there exists a subset of indices* $I \subseteq [n]$ *of size* $|I| \geq \frac{n}{9}$ *such that* $d(s_i, t_j) \geq \frac{T}{O(\log k)}$ *for all* $i, j \in I$*. Moreover, such a set can be computed in polynomial time.*

We note that the above lemma is similar to the main Theorem of Arora et al. [35]. Our result holds for general metrics with a factor of $O(\log k)$ in the distance loss, while their holds for ℓ_2^2 metrics with a factor of $O(\sqrt{\log k})$. The results are incomparable and both are tight. (The tightness of Lemma II.5 can be shown to be tight for graph metrics, for example in graph metrics of constant-degree expanders.)

To prove Lemma II.5 we rely on *padded decompositions* [55]. To define these, we introduce some section-specific notation. Let (X, dist) be a metric space. Let the (weak) diameter of a set of points $U \subseteq X$ be denoted by $\text{diam}(U) \triangleq \max_{x,y\in U} \text{dist}(x, y)$. We say a partition $P = \{X_1, X_2, \ldots, X_t\}$ of X is Δ-*bounded* if $\text{diam}(X_i) \leq \Delta$ for all i. Next, for $U \subseteq X$ and a partition P as above, we denote by $U \subseteq P$ the event that there exists a part $X_i \in P$ containing U in its entirety; i.e., $U \subseteq X_i$. Let $B(x, \rho) \triangleq \{y \in X \mid \text{dist}(x, y) \leq \rho\}$ denote the ball of radius $\rho \geq 0$ around $x \in X$.

Definition II.7. *Let* $(X, dist)$ *be a metric space. We say that a distribution* $\mathcal{P}$ *over* Δ*-bounded partitions of* X *is* (β, Δ)-padded *if, for some universal constant* δ*, it holds that for every* $x \in X$ *and* $0 \leq \gamma \leq \delta$*,*

$$\Pr_{P\sim\mathcal{P}}[B(x, \gamma\Delta) \not\subseteq P] \leq \beta\gamma.$$

In words, each part of the partition has diameter at most Δ and the probability of any point x in the metric being

at distance less than $\gamma\Delta$ from a different part than its own part is at most $\beta\gamma$. Such decompositions were presented, for example, by Gupta et al. [55].

Lemma II.8 ([55]). *Any metric $(X, dist)$ on k points admits a (β, Δ)-padded decomposition, for any $\Delta > 0$ and some $\beta = O(\log k)$. Such a decomposition can be computed in polynomial time.*

We are now ready to prove Lemma II.5.

Proof of Lemma II.5: First note that we can focus on the metric space induced by the k distinct points. Let $\mathcal{P}$ be a Δ-bounded β-padded decomposition with $\Delta = T - 1$ and $\beta = O(\log k)$. We first note that for all $i \in [k]$, s_i and t_i are contained in different parts since the diameter of each part X_i is at most $\Delta = T - 1$ and $\text{dist}(s_i, t_i) \geq T$. Furthermore, letting $\gamma = \frac{1}{2\beta}$, we have that $\Pr[B(s_i, \gamma\Delta) \subseteq P] \geq \frac{1}{2}$. Let $I' \subseteq [n]$ be the subset of indices i with $B(s_i, \gamma\Delta) \subseteq P$. Then we have $\Pr[i \in I'] \geq \frac{1}{2}$ for all $i \in [n]$.

Flip a fair and independent coin for each part in P. Let $U \subseteq X$ be the set of points in parts whose coin came out heads, and $V \subseteq X$ be the analogous set for tails. Then for each $i \in I'$ we have that $\Pr[s_i \in U \text{ and } t_i \in V] = \frac{1}{4}$. Let $I \subseteq I'$ be the subset of indices i with $s_i \in U$ and $t_i \in V$, giving $\Pr[i \in I] = \Pr[i \in I'] \cdot \Pr[i \in I \mid i \in I'] = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$ $\forall i \in [n]$. We also have that $\text{dist}(s_i, t_j) > \rho = \frac{T-1}{2\beta}$ for all $i, j \in I$, since $B(s_i, \rho) \subseteq U$ for all $i \in I \subseteq I'$ and $\{t_j\}_{j \in I} \cap U = \emptyset$. Therefore, this random process yields a subset of indices $I \subseteq [n]$ such that $\text{dist}(s_i, t_j) > \frac{T-1}{2\beta}$ for all $i, j \in I$, of expected size at least $\mathbb{E}[|I|] \geq \sum_{i \in [n]} \Pr[i \in I] \geq \frac{n}{8}$. As $n - \mathbb{E}[|I|]$ is a non-negative random variable, Markov's inequality implies that with constant probability $n - |I| \leq \frac{64}{63} \cdot (n - \mathbb{E}[|I|]) \leq \frac{8n}{9}$. The lemma follows. ∎

Remark: The $O(\log k)$ term in Lemma II.5's bound is precisely the smallest possible β for which (β, Δ)-padded decompositions of the metric exist. For many graphic metrics, such as those of minor-excluded, bounded-genus, and bounded-doubling-dimension networks, padded decompositions with smaller β exist [50, 55, 56]. This improves the bounds of Lemma II.5 and thus Lemma II.2 by $(\log k)/\beta$, implying the same improvement for our makespan coding gaps for such networks.

III. Conclusions and Open Questions

In this paper we study completion-time coding gaps; i.e., the ratio for a given multiple-unicast instance, of the fastest routing protocol's completion time to the fastest coding protocol's completion time. We provide a strong characterization of these gaps in the worst case, showing they can be polylogarithmic in the problem parameters, but no greater. The paper raises a few exciting questions and research directions.

Probably the most natural question is to close our upper and lower bounds. We show that the network coding gap is polylogarithmic, but what polylog? Another question, motivated by the super-constant speedups we prove coding can achieve over routing for this basic communication problem, is whether there exist efficient algorithms to compute the fastest coding protocol, mirroring results known for the fastest routing protocols for multiple unicasts [7, 8], and for the highest-throughput coding protocols for multicast [57]. Another natural question is extending this study of network coding gaps for completion times to other widely-studied communication problems, such as multiple multicasts.

Implications to other fields.: As discussed in Section I and Section I-D, the conjectured non-existence of throughput coding gaps for multiple unicast has been used to prove (conditional) lower bounds in many seemingly-unrelated problems. It would be interesting to see whether our upper and lower bounds on the coding gap for multiple unicasts' completion times can be used to prove *unconditional* lower bounds for other models of computation. We already have reason to believe as much; using techniques developed in this paper, combined with many other ideas, the authors have obtained the first non-trivial universal lower bounds in distributed computation (see details in the full version). It would be interesting to see what other implications this work might have to other areas of Theory. Perhaps most exciting would be to investigate whether completion-time coding gaps imply new results in circuit complexity for depth-bounded circuits.

Acknowledgments

The authors would like to thank Mohsen Ghaffari for suggesting an improvement to Theorem I.2 which resulted in a coding gap independent of n, Anupam Gupta for pointing out a simplification of Lemma II.5 and the Lemma's similarity to [35, Theorem 1], and Paritosh Garg for bringing [38] to our attention. This work was supported in part by NSF grants CCF-1618280, CCF-1814603, CCF-1527110, NSF CAREER award CCF-1750808, a Sloan Research Fellowship, and a DFINITY scholarship.

References

[1] T. Rothvoß, "A simpler proof for $O(Congestion + Dilation)$ packet routing," in *Proceedings of the 16th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2013, pp. 336–348.

[2] B. Peis and A. Wiese, "Universal packet routing with arbitrary bandwidths and transit times," in *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2011, pp. 362–375.

[3] A. Srinivasan and C.-P. Teo, "A constant-factor approximation algorithm for packet routing and balancing local vs. global criteria," *SIAM Journal on Computing (SICOMP)*, vol. 30, no. 6, pp. 2051–2068, 2001.

[4] R. Koch, B. Peis, M. Skutella, and A. Wiese, "Real-time message routing and scheduling," in *Proceedings of the 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2009, pp. 217–230.

[5] R. Ostrovsky and Y. Rabani, "Universal $O(congestion + dilation + \log^{1+\epsilon} n)$ local control packet switching algorithms," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, vol. 29, 1997, pp. 644–653.

[6] F. M. auf der Heide and B. Vöcking, "Shortest-path routing in arbitrary networks," *Journal of Algorithms*, vol. 31, no. 1, pp. 105–131, 1999.

[7] T. Leighton, B. Maggs, and A. W. Richa, "Fast algorithms for finding o (congestion+ dilation) packet routing schedules," *Combinatorica*, vol. 19, no. 3, pp. 375–401, 1999.

[8] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in O(congestion+ dilation) steps," *Combinatorica*, vol. 14, no. 2, pp. 167–186, 1994.

[9] C. Scheideler, *Universal routing strategies for interconnection networks*. Springer, 2006, vol. 1390.

[10] D. Bertsimas and D. Gamarnik, "Asymptotically optimal algorithms for job shop scheduling and packet routing," *Journal of Algorithms*, vol. 33, no. 2, pp. 296–318, 1999.

[11] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[12] N. J. Harvey, R. Kleinberg, and A. R. Lehman, "On the capacity of information networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2345–2364, 2006.

[13] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Allerton Conference on Communications*, vol. 16, 2004, p. 8.

[14] M. Langberg and M. Médard, "On the multiple unicast network coding, conjecture," in *47th Annual Allerton Conference on Communication, Control, and Computing*, 2009, pp. 222–227.

[15] A. Al-Bashabsheh and A. Yongaçoglu, "On the k-pairs problem," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 1828–1832.

[16] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2486–2507, 2006.

[17] B. Haeupler, "Simple, fast and deterministic gossip and rumor spreading," *Journal of the ACM (JACM)*, vol. 62, no. 6, p. 47, 2015.

[18] ——, "Analyzing network coding (gossip) made easy," *Journal of the ACM (JACM)*, vol. 63, no. 3, p. 26, 2016.

[19] N. J. Harvey, R. D. Kleinberg, and A. R. Lehman, "Comparing network coding with multicommodity flow for the k-pairs communication problem," Tech. Rep., 2004.

[20] G. Kramer and S. A. Savari, "Edge-cut bounds on network coding rates," *Journal of Network and Systems Management*, vol. 14, no. 1, p. 49, 2006.

[21] M. Adler, N. J. Harvey, K. Jain, R. Kleinberg, and A. R. Lehman, "On the capacity of information networks," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 241–250.

[22] K. Jain, V. V. Vazirani, R. Yeung, and G. Yuval, "On the capacity of multiple unicast sessions in undirected graphs," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2805–2809, 2006.

[23] X. Yin, Z. Li, Y. Liu, and X. Wang, "A reduction approach to the multiple-unicast conjecture in network coding," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4530–4539, 2018.

[24] T. Xiahou, Z. Li, C. Wu, and J. Huang, "A geometric perspective to multiple-unicast network coding," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2884–2895, 2014.

[25] P. Afshani, C. B. Freksen, L. Kamma, and K. G. Larsen, "Lower bounds for multiplication via network coding," in *Proceedings of the 46th International Colloquium on Automata, Languages and Programming (ICALP)*, 2019, pp. 10:1–10:12.

[26] A. Farhadi, M. Hajiaghayi, K. G. Larsen, and E. Shi, "Lower bounds for external memory integer sorting via network coding," in *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, 2019, p. To Appear.

[27] Y. Aumann and Y. Rabani, "An o (log k) approximate min-cut max-flow theorem and approximation algorithm," *SIAM Journal on Computing (SICOMP)*, vol. 27, no. 1, pp. 291–301, 1998.

[28] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, no. 2, pp. 215–245, 1995.

[29] A. Das Sarma, S. Holzer, L. Kor, A. Korman, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer, "Distributed verification and hardness of distributed approximation," *SIAM Journal on Computing (SICOMP)*, vol. 41, no. 5, pp. 1235–1265, 2012.

[30] D. Peleg and V. Rubinovich, "A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction," *SIAM Journal on Computing (SICOMP)*, vol. 30, no. 5, pp. 1427–1442, May 2000.

[31] T. C. Hu, "Multi-commodity network flows," *Operations research*, vol. 11, no. 3, pp. 344–360, 1963.

[32] S. Plotkin and É. Tardos, "Improved bounds on the max-flow min-cut ratio for multicommodity flows," *Combinatorica*, vol. 15, no. 3, pp. 425–434, 1995.

[33] M. Elkin, "An unconditional lower bound on the time-approximation trade-off for the distributed minimum spanning tree problem," *SIAM Journal on Computing (SICOMP)*, vol. 36, no. 2, pp. 433–456, 2006.

[34] M. Ghaffari and B. Haeupler, "Distributed algorithms for planar networks II: Low-congestion shortcuts, mst, and min-cut," in *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2016, pp. 202–219.

[35] S. Arora, S. Rao, and U. Vazirani, "Expander flows, geometric embeddings and graph partitioning," *Journal of the ACM (JACM)*, vol. 56, no. 2, p. 5, 2009.

[36] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Lexicographic products and the power of non-linear network coding," in *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*, 2011, pp. 609–618.

[37] S. Lovett, "Linear codes cannot approximate the network capacity within any constant factor." in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 21, 2014, p. 141.

[38] M. Braverman, S. Garg, and A. Schvartzman, "Coding in undirected graphs is either very helpful or not helpful at all," in *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, 2017, pp. 18:1–18:18.

[39] Y. Rabani and É. Tardos, "Distributed packet switching in arbitrary networks," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, vol. 96, 1996, pp. 366–375.

[40] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on communications*, vol. 53, no. 11, pp. 1906–1918, 2005.

[41] A. Goel and S. Khanna, "On the network coding advantage for wireless multicast in euclidean space," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 64–69.

[42] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 2, pp. 450–463, 2008.

[43] C. Chekuri, S. Kamath, S. Kannan, and P. Viswanath, "Delay-constrained unicast and the triangle-cast problem," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 804–808.

[44] C.-C. Wang and M. Chen, "Sending perishable information: Coding improves delay-constrained throughput even for single unicast," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 252–279, 2016.

[45] Z. Li and B. Li, "Network coding in undirected networks," in *Conference on Information Systems and Sciences (CISS)*, 2004.

[46] A. Agarwal and M. Charikar, "On the advantage of network coding for improving network throughput," in *Information Theory Workshop, 2004. IEEE*, 2004, pp. 247–249.

[47] L. Zosin and S. Khuller, "On directed steiner trees," in *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2002, pp. 59–63.

[48] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang, "Integrality ratio for group steiner trees and directed steiner trees," *SIAM Journal on Computing (SICOMP)*, vol. 36, no. 5, pp. 1494–1511, 2007.

[49] S. Rao, "Small distortion and volume preserving embeddings for planar and euclidean metrics," in *Proceedings of the 15th Symposium on Computational geometry (SoCG)*, 1999, pp. 300–306.

[50] J. R. Lee and A. Sidiropoulos, "Genus and the geometry of the cut graph:[extended abstract]," in *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010, pp. 193–201.

[51] C. Chekuri, F. B. Shepherd, and C. Weibel, "Flow-cut gaps for integer and fractional multiflows," *Journal of Combinatorial Theory, Series B*, vol. 103, no. 2, pp. 248–273, 2013.

[52] R. Krauthgamer, J. R. Lee, and H. Rika, "Flow-cut gaps and face covers in planar graphs," in *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019, pp. 525–534.

[53] C. Chekuri, S. Khanna, and F. B. Shepherd, "Multicommodity flow, well-linked terminals, and routing problems," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005, pp. 183–192.

[54] D. Harvey and J. Van Der Hoeven, "Polynomial multiplication over finite fields in time $O(n \log n)$," 2019.

[55] A. Gupta, R. Krauthgamer, and J. R. Lee, "Bounded geometries, fractals, and low-distortion embeddings," in *Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS)*, 2003, p. 534.

[56] I. Abraham, C. Gavoille, A. Gupta, O. Neiman, and K. Talwar, "Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs," *SIAM Journal on Computing (SICOMP)*, vol. 48, no. 3, pp. 1120–1145, 2019.

[57] S. Jaggi, P. A. Chou, and K. Jain, "Low complexity algebraic multicast network codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2003, pp. 368–368.