On the power of the congested clique model

On the Power of the Congested Clique Model

Andrew Drucker*
Institute for Advanced Study
Princeton, NJ, USA

Fabian Kuhn
University of Freiburg
Freiburg, Germany

Rotem Oshman†
Princeton University
Princeton, NJ, USA

Abstract

We study the computation power of the congested clique, a model of distributed computation where n players communicate with each other over a complete network in order to compute some function of their inputs. The number of bits that can be sent on any edge in a round is bounded by a parameter b. We consider two versions of the model: in the first, the players communicate by unicast, allowing them to send a different message on each of their links in one round; in the second, the players communicate by broadcast, sending one message to all their neighbors.

It is known that the unicast version of the model is quite powerful; to date, no lower bounds for this model are known. In this paper we provide a partial explanation by showing that the unicast congested clique can simulate powerful classes of bounded-depth circuits, implying that even slightly super-constant lower bounds for the congested clique would give new lower bounds in circuit complexity. Moreover, under a widely-believed conjecture on matrix multiplication, the triangle detection problem, studied in [8], can be solved in $O(n^\epsilon)$ time for any $\epsilon > 0$.

The broadcast version of the congested clique is the well-known multi-party shared-blackboard model of communication complexity (with number-in-hand input). This version is more amenable to lower bounds, and in this paper we show that the subgraph detection problem studied in [8] requires polynomially many rounds for several classes of sub-

*Supported by the NSF under agreements Princeton University Prime Award No. CCF-0832797 and Sub-contract No. 00001583.

†Supported by NSF Grants No. CCF-0832797 and CCF-1149888.

graphs. We also give upper bounds for the subgraph detection problem, and relate the hardness of triangle detection in the broadcast congested clique to the communication complexity of set disjointness in the 3-party number-on-forehead model.

1 Introduction

The congested clique model, studied in [30, 32, 8, 28], features n players that communicate with each other in synchronous rounds over a complete network. In each round, each player can send b bits on each of its communication links, for a total of $\Theta(bn^2)$ bits sent per round. In the version of the model considered in [30, 32, 8, 28], players can send different messages on different links; we denote this version of the model by $\mathsf{CLIQUE\text{-}UCAST}_{n,b}$. In this paper we study the computation power of $\mathsf{CLIQUE\text{-}UCAST}_{n,b}$, and also of another variant, denoted $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$, where in each round each player can only broadcast a single b-bit message over all its links. The $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ model is essentially the classical multi-party, number-in-hand model of communication complexity, with communication over a shared blackboard [27], since writing on the blackboard can be viewed as broadcasting a message to all players.

In both models we are interested in the round complexity of computing functions $f : \{0,1\}^{n^2} \to \{0,1\}$, where initially n^2 bits of input are equally partitioned between the players (with each player receiving n bits), and the goal is for some player to eventually output the value of f. We are especially interested in the *subgraph detection problem*, studied in [8]: here, the joint input to the players is interpreted as an undirected n-node graph G, with player i receiving the list of edges adjacent to the i-th node of G, and the goal is to determine whether G contains a particular subgraph H.

As noted in [28], part of what makes the $\mathsf{CLIQUE\text{-}UCAST}$ model interesting is that it does not have any "information bottlenecks", and it is therefore not amenable to the type of arguments typically used for restricted-bandwidth networks (e.g., [23, 39, 12, 26, 20]). For this reason, any lower bounds for $\mathsf{CLIQUE\text{-}UCAST}$ would have to rely on new, "bottleneck-free" lower bound techniques. To date, no such lower bounds have been proven, and in this paper we provide a partial explanation: we show that $\mathsf{CLIQUE\text{-}UCAST}$ is sufficiently strong to simulate some well-studied parallel circuit models, such as ACC and $\mathsf{TC0}$. Proving lower bounds for explicit functions in these models is a notoriously difficult and elusive goal in the theory of Boolean circuit complexity. Our results imply that even weak lower bounds

for CLIQUE-UCAST would imply progress on these fundamental problems; specifically, even a slightly super-constant lower bound on the number of rounds required to compute some explicit function in $\mathsf{CLIQUE\text{-}UCAST}_{n,1}$ would imply a new lower bound on ACC, and an $\Omega(\log \log n)$ lower bound for $\mathsf{CLIQUE\text{-}UCAST}_{n,\log n}$ would imply new a lower bound for threshold circuits (the class TC). It seems, then, that "information bottlenecks" are essential to our ability to prove strong lower bounds, and without them we are subject to the same extremely difficult challenges facing the circuit complexity community.

We remark that, as with circuits, *non-explicit* lower bounds for the CLIQUE-UCAST model are not hard to show. In the full version we give a counting-based lower bound that shows that there exists a function that takes $(n - O(\log n))/b$ rounds to compute in $\mathsf{CLIQUE\text{-}UCAST}_{n,b}$. Since n/b rounds are sufficient for any node to learn the inputs of all other nodes (assuming n bits of input per node), this non-explicit lower bound is very close to optimal.

The other variant we consider, CLIQUE-BCAST, is used to study many areas in theoretical computer science, from streaming [1] to cryptology [14] and mechanism design [7]. Beyond being a fundamental model of communication complexity, it can also be viewed as a very abstract model of wireless communication over a single-hop network: although the theoretical distributed computing community typically assumes that in a wireless network only one node can successfully broadcast in each round, recent advances in coding (e.g., [15, 44] and others) can allow multiple packets to come through. Especially for the purpose of proving lower bounds, it seems reasonable to consider a model that allows *all* messages to be delivered.

From the perspective of lower bounds, a major difference between the $\mathsf{CLIQUE\text{-}UCAST}_{n,b}$ and $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ models is that in the broadcast model, only $\Theta(nb)$ "unique bits" (discounting duplicate messages) cross each balanced cut, compared to $\Theta(n^2 b)$ in the unicast model. This difference restores our ability to apply bottleneck arguments, and we show that in CLIQUE-BCAST, the subgraph detection problem considered in [8] is polynomially hard for several classes of subgraphs. As is often the case when working with restricted-bandwidth models, our lower bounds are obtained by reduction from 2-party communication complexity. We also provide upper bounds showing that in some (though not all) cases our lower bounds are tight. Some of our upper and lower bounds also apply to general communication networks.

One major open problem left open by our work is the complexity of triangle detection, also studied in [8], where it is shown that the problem

can be solved in time $\tilde{O}(n^{1/3}/T^{2/3})$ in $\mathsf{CLIQUE\text{-}UCAST}_{n,O(\log n)}$ when the input graph has at least T triangles. In Section 2.1 we show that if matrix multiplication can be solved by arithmetic circuits with size $O(n^\delta)$, then we can solve triangle detection in $O(n^{\delta/n^2})$ rounds in $\mathsf{CLIQUE\text{-}UCAST}_{n,1}$. It is believed plausible that matrix multiplication can be solved in size $O(n^{2+\epsilon})$ for any $\epsilon > 0$, and this would imply an $O(n^\epsilon)$-round triangle detection algorithm for $\mathsf{CLIQUE\text{-}UCAST}_{n,1}$.

In the $\mathsf{CLIQUE\text{-}BCAST}$ model, triangles are not amenable to our lower bound technique, because a triangle cannot be "partitioned" between two players — when the vertices are partitioned between the players, one of the two will always "see" all three edges of the triangle. Here we draw a connection to the 3-party number-on-forehead (NOF) model of communication complexity, in which each player can see the input of the other two players, but not its own input [27]. We show that a lower bound of $f(N)$ on the communication complexity of N-element set disjointness in the 3-party NOF model would imply a lower bound of $\Theta(f(n^2/e^{O(\sqrt{\log n})})/(nb))$ rounds for triangle detection in $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$; currently the best unconditional lower bound on randomized 3-party NOF set disjointness with N elements is $\Omega(\sqrt{N})$ [41], which is not strong enough to yield a non-trivial bound on triangle detection. For deterministic algorithms, however, it has very recently been shown that $\Omega(N)$ bits are required for 3-party NOF set disjointness, yielding a lower bound of $\Omega(n/(e^{O(\sqrt{\log n})}b)$ for deterministic triangle detection. In addition, for the randomized case, we are able to obtain a conditional and restricted lower bound, through a connection made in [35] between 3-party NOF set disjointness and the computational hardness of SAT. Even a small polynomial improvement in the lower bound for randomized 3-party NOF set disjointness would yield an unconditional polynomial lower bound for randomized triangle detection.

Related work The $\mathsf{CONGEST}$ model [33], introduced to study networks with restricted bandwidth, has received much attention recently [12, 13, 20, 23, 26, 39]. Most work on the model considers a setting where nodes communicate over some network graph G, and studies the complexity of detecting properties of G or computing various graph structures (e.g., minimum spanning trees [9, 34, 39]). Many of these results use reductions from two-player communication complexity, a technique we also use in Section 3.2.

Upper bounds for various problems in the congested clique are given in [30, 32, 8, 29, 28]. Of particular relevance to our work is [8], which gives an upper bound of $\tilde{O}(n^{(d-2)/d})$ on the round complexity of detecting

a fixed d-vertex subgraph in $\mathsf{CLIQUE\text{-}UCAST}_{n,O(\log n)}$; we give upper and lower bounds on subgraph detection in the *broadcast* version of the congested clique in Section 3.

In [28] it is shown that any "balanced" routing demand, where the number of messages that must be routed between each pair of players does not exceed $O(n)$, can be scheduled deterministically in $O(1)$ rounds in $\mathsf{CLIQUE\text{-}UCAST}_{n,O(\log n)}$. We use the routing algorithm of [28] in our simulation of circuits in Section 2.

2 From Circuits to the Congested Clique

In this section we show that the CLIQUE-UCAST model is powerful enough to simulate circuits with "simple" gates and a quasi-linear number of wires.

We model a circuit as a directed acyclic graph (DAG), where the nodes represent gates from some class $\mathcal{G}$ of Boolean functions. The complexity measures we are interested in are the *depth* of the circuit, which is the length of the longest path from any input (source node) to any output (sink node), and the *number of wires* (edges).

We now formalize what we mean by "simple gates". The following definition is a variant of *worst-partition communication complexity*, discussed in the textbook [27].

Definition 1. *A function* $f : \{0,1\}^m \to \{0,1\}$ *is* b-separable, *for* $b \in [m]$, *if for any partition* $I = (I_1, \ldots, I_k)$ *of* $[m]$ *there are functions* $\left\{g_j : \{0,1\}^{|I_j|} \to \{0,1\}^b\right\}_{j=1}^{k}$ *and* $h : \{0,1\}^{bk} \to \{0,1\}$ *s.t.* $f(x_1, \ldots, x_m) = h(g_1(x_{I_1}), \ldots, g_k(x_{I_k}))$.

We say that a gate G is *b-separable* if the function it computes is b-separable. We will analyze circuits whose gates are b-separable for small b; related restrictions on circuits have been studied previously in circuit complexity, e.g. in [24].

Theorem 2. *Suppose that* $f : \{0,1\}^{n^2} \to \{0,1\}$ *is computed by a circuit* C *of depth* D, *comprising* b*-separable gates with unbounded fan-in and fan-out. Let* $N = n^2 \cdot s$ *be the number of wires in* C. *Then for any input partition which assigns to each player* p *no more than* $n(b+s)$ *input wires, there is an* $O(D)$*-round protocol* P *for the* CLIQUE-UCAST *model with* n *players and bandwidth* $O(b+s)$, *that computes* f *under the input partition.*

Proof. For a gate $G \in C$, let $\mathsf{in}(G) \subseteq C$ and $\mathsf{out}(G) \subseteq C$ denote the set of inputs and outputs of G, respectively. We represent inputs to the circuit as

gates that have no inputs ($\mathsf{in}(G) = \emptyset$) and outputs as gates that have no output ($\mathsf{out}(G) = \emptyset$).

We define the *weight* of G, denoted $w(G)$, to be the sum of its in-degree and its out-degree: $w(G) := |\mathsf{in}(G)| + |\mathsf{out}(G)|$. We say that G is *heavy* if $w(G) \geq n \cdot s$, otherwise we say that G is *light*. Let $C_H \subseteq C$ be the set of heavy gates in C, and let $C_L := C \setminus C_H$ be the light gates.

We construct an assignment $I : C \to [n]$ of gates (including inputs and outputs) to players, such that

(1) Each player p is assigned at most one heavy gate, that is, $|I^{-1}(p) \cap C_H| \leq 1$; and

(2) For each player p, the total weight of light gates assigned to p does not exceed $2n \cdot s$.

Here $I^{-1}(p) := \{G \in C \mid I(G) = p\}$. We say that player p *owns* a gate G or a wire incoming or outgoing from G if $I(G) = p$.

Construction of I Since C has a total of $n^2 \cdot s$ wires, the number of heavy gates cannot exceed n, so we can assign each heavy gate to a unique player. As for the light gates, we go over them in arbitrary order, and assign each gate G to some player p that does not already own more than $2n \cdot s - w(G)$ wires. To see that there is always some such player, suppose for the sake of contradiction that we need to assign a light gate G, but each player already owns more than $2n \cdot s - w(G)$ wires. Since G is light we have $w(G) < n \cdot s$, so the total number of wires already assigned exceeds $n \cdot (2n \cdot s - w(G)) > n \cdot (2n \cdot s - n \cdot s) = n^2 \cdot s$, contradicting our assumption that C has a total of $n^2 \cdot s$ wires.

Evaluating the circuit We partition the gates of C into layers $L_0, \ldots, L_D$, where the first layer, defined by $L_0 := \{G \in C \mid \mathsf{in}(G) = \emptyset\}$, represents the inputs of C, and for each $r > 0$, $L_r := \left\{G \in C \mid \mathsf{in}(G) \subseteq \bigcup_{r'<r} L_{r'}\right\} \setminus \left(\bigcup_{r'<r} L_{r'}\right)$ are the gates whose inputs all belong to layers smaller than r, but which are not themselves in some smaller layer. Because the depth of the circuit is D, there are exactly D layers. We evaluate the circuit in D stages, each corresponding to one layer of the circuit and requiring $O(1)$ rounds.

The input layer, L_0, does not require any evaluation. Suppose that we have already evaluated the circuit up to layer L_{r-1}, and we now wish to evaluate L_r. For each $G \in L_r$ we proceed as follows:

(a) If G is heavy then we use the fact that G is b-separable. Let $p_1, \ldots, p_k$ be the players that own inputs to G, and let $H = \{H_1, \ldots, H_k\}$ be the partition of $\mathsf{in}(G)$ induced by I, where we have $H_i = \{G' \in \mathsf{in}(G) \mid I(G') = p_i\}$. Let $g_1, \ldots, g_k$ and h be the functions from Definition 1 with respect to H. Each player p_i computes $g_i(H_i)$ and sends its b-bit value to player $I(G)$, who then applies h to obtain the value of G.

(b) If G is light and $G' \in \mathsf{in}(G)$ is heavy, then player $I(G')$ sends the value of G' to player $I(G)$, *unless it has already done so in the past.* (It is crucial to avoid duplication of heavy-gate outputs.)

(c) The remaining wires are both inputs and outputs of light gates. These wires induce a demand pattern, where each player $I(G)$ needs to learn all the values of "light inputs" $G' \in \mathsf{in}(G) \cap C_L$. Because each player is responsible for no more than $2n \cdot s$ incoming or outgoing wires for light gates, the demand pattern is balanced, with each player requiring no more than $2n \cdot s$ bits from any other player. We route all these wires in $O(1)$ rounds using Lenzen's algorithm [28].

It is not hard to see that since each player owns at most one heavy gate, the total number of bits sent from any player p to any other player q in stage r is $O(s + b)$.

Finally, to handle any input assignment which is roughly-balanced (each player receives at most $n(b + s)$ inputs), we can use Lenzen's algorithm to route the inputs from their originally-assigned players to the player assigned them under I, using no more than $O(b + s)$ bits per message and $O(1)$ rounds. □

Remark 3. *Although the simulation is stated for functions, it is easy to see that it extends to* operators*, functions with multi-bit outputs. We partition the outputs between the players, such that no player is required to output more than $O(b + s)$ bits, and use Lenzen's algorithm to route the outputs to the correct players. This will be useful when we discuss matrix multiplication below.*

From the simulation above we conclude:

Theorem 4. *For any $s, b, R \geq 0$, if $f : \{0,1\}^{n^2} \to \{0,1\}$ cannot be computed in $\mathsf{CLIQUE\text{-}UCAST}_{n,O(b+s)}$ in R rounds, then for some constant $c > 1$, f cannot be computed by circuits of depth $c \cdot R$ using b-separable gates and $n^2 \cdot s$ wires.*

This means that any attempt to prove lower bounds for this model runs up against the following open problems in bounded-depth circuit complexity:

Threshold circuits [31] A threshold circuit is a circuit with unbounded fan-in, made up of gates that compute threshold functions — Boolean predicates of the form $a_1x_1 + a_2x_2 \ldots + a_kx_k \geq b$, where $a_1, \ldots, a_k, b \in \mathbf{Z}^+$ are called *weights*, and $x_1, \ldots, x_k$ are the (Boolean) inputs to the gate. At present, the best lower bound known for bounded-depth threshold circuits that compute Boolean functions is from [21, 42], where it is shown that a threshold circuit of depth d that computes the parity of N-bit inputs requires $N^{1+cK^{-d}}$ wires, where $c > 0$ and $K \leq 3$ are explicit constants. This holds for circuits with arbitrary weights; however, despite extensive study, no better bounds are known for the unweighted case (where all weights are set to one) beyond depth 3 [16, 18, 37, 40].

The lower bound decays quickly with d, becoming trivial (linear) at $d = \Theta(\log \log n)$. Since unweighted threshold gates are $\Theta(\log n)$-separable, our simulation implies that for some fixed constants $\alpha, \beta > 0$, obtaining a lower bound of $\alpha \log \log n$ on the round complexity of some explicit function in $\mathsf{CLIQUE\text{-}UCAST}_{n,\beta \log n}$ would imply a better lower bound than that of [21, 42] for the number of wires in unweighted threshold circuits.

ACC and CC [36] Circuits in $\mathsf{ACC}[m]$ use unbounded fan-in AND, OR, NOT and MOD_m gates, where m is constant. MOD_m gates test whether or not $(x_1 + \ldots + \ldots x_k) \bmod m = 0$; for example, a MOD_2 gate computes the parity of its inputs. A $\mathsf{CC}[m]$ circuit is one comprising only unbounded fan-in MOD_m gates.

Although there are exponential lower bounds on the size of constant-depth $\mathsf{ACC}[m]$ circuits when m is prime or a prime power [43], the general case remains very challenging; even the possibility that all of NP has depth-3, poly-size circuits consisting only of MOD_6 gates, with a linear number of gates, has not been ruled out [17]. (In a recent breakthrough, Williams showed that the larger class NEXP does not have poly-size, constant-depth $\mathsf{ACC}[m]$ circuits for any m [46]; however only lower bounds for problems in NP are generally considered "explicit".)

Currently, the best explicit lower bound on the number of wires appears in [6], where it is shown that constant-depth $\mathsf{CC}[m]$ circuits (where m may be composite) require a superlinear number of wires to compute AND or MOD_q, where q is coprime to m. Specifically, the lower bound is the following: for depth 2, $\Omega(n \log n)$ wires are required; for depth 3, $\Omega(n \log^* n)$ wires; for depth 4, $\Omega(n \log^{**} n)$, and so on. In general, the bound for depth d is $\Omega(n \cdot \lambda_d(n))$, where $\lambda_1(n) = \lceil \log_2 n \rceil$ and $\lambda_{d+1}(n) = \min\left\{ i \in \mathbb{N} \mid \lambda_d^{(i)} \leq 1 \right\}$. Here $f^{(i)}$ denotes f iterated i times.

Let $\lambda^{-1}(n) := \min\{d \in \mathbb{N} \mid \lambda_d(n) \leq 1\}$. The lower bound of [6] becomes trivial at depth $\lambda^{-1}(n)$. Since MOD_6 gates are $O(1)$-separable, we get that for some constants $\alpha, \beta > 0$, a lower bound of $\alpha \cdot \lambda^{-1}(n)$ for $\mathsf{CLIQUE\text{-}UCAST}_{n,\beta}$ would imply a bound better than [6] on $\mathsf{ACC}[m]$ for composite m.

The circuit lower bounds referred to above are the best currently known in terms of asymptotic behavior as the depth increases. For fixed depths (typically 2 or 3), or for computing operators, some better bounds are known, but the asymptotic behavior remains the same (in the case of operators, up to a multiplicative constant in the depth).

2.1 Matrix Multiplication vs. Triangle Detection

The triangle detection problem received special attention in [8], which gives an elegant randomized algorithm that runs in time $\tilde{O}(n^{1/3}/T^{2/3})$ when the input graph has at least T triangles. In the full version we show that, under a certain widely-believed conjecture on the complexity of matrix multiplication, triangle detection can be solved in time $O(n^\epsilon)$ for any $\epsilon > 0$ in $\mathsf{CLIQUE\text{-}UCAST}_{n,1}$.

We give an overview of the connection here. The connection between matrix multiplication and triangle detection is well-known [22, 45]: if one cubes the adjacency matrix of a graph over the Boolean semiring, triangles appear as non-zero entries on the diagonal. A simple randomized reduction due to Adi Shamir (described in [45, Thm. 4.1]) allows one to reduce this computation to a small number of matrix multiplications over the field $\mathbb{F}_2$. Now, it is widely conjectured (although there is no consensus) that for every $\epsilon > 0$, matrix multiplication over any field $\mathbb{F}$, and in particular over $\mathbb{F}_2$, can be computed (as a formal polynomial mapping) by arithmetic circuits of size $O(n^{2+\epsilon})$; and it is known [5] that such circuits also imply the existence of circuits for matrix multiplication with few wires *and* polylogarithmic depth. (This fact is shown by exploiting the block structure of matrix multiplication; see [5, Prop. 15.1] and the full version.)

By using the simulation from Theorem 2 (and Remark 3), we can translate small, shallow arithmetic circuits over $\mathbb{F}_2$ into a fast $\mathsf{CLIQUE\text{-}UCAST}_{n,1}$ protocol in which every entry of the output matrix is known to some player; each player can then announce whether or not it sees any non-zero diagonal entries, thereby solving triangle detection.

3 Subgraph Detection in the Congested Clique with Broadcast

We now turn our attention to the broadcast version of the congested clique, CLIQUE-BCAST, and study the problem of detecting whether a given fixed-size subgraph H is a subgraph of the input graph G. We assume that H is of fixed (constant) size, and we are interested in the complexity of the problem as the number of participants n grows. We start by describing a simple algorithm that solves the problem in the CLIQUE-BCAST model. We then describe lower bounds for subgraph detection in the CLIQUE-BCAST model; some of our lower bounds also hold for the CONGEST-UCAST model, where the communication topology is the same as the input graph G, rather than the clique.

Our bounds are based on Turán numbers for graphs, which are defined as follows.

Definition 5 (Turán Number)**.** *For a graph H and an integer $n \geq 1$, the Turán number $\mathrm{ex}(n, H)$ for graph H is the maximal possible number of edges of an n-node graph G such that G does not contain a subgraph isomorphic to H.*

Of particular relevance to us in this section are the Turán numbers for any odd-length cycle (or in general for non-bipartite graphs), which is $\Theta(n^2)$, and for the 4-cycle C_4, which is $\Theta(n^{3/2})$.

3.1 Upper Bounds on Subgraph Detection

The *degeneracy* of a graph G is the smallest integer k such that every subgraph of G has a node of degree at most k. In [2], Becker et al. give a one-round algorithm for the CLIQUE-BCAST model that allows all players to learn the entire input graph, assuming it has known degeneracy of (at most) k. In the algorithm of [2], the players simultaneously each broadcast $O(k \log n)$ bits, and based on these messages they are able to completely reconstruct the input graph.

We can use the algorithm from [2] to solve subgraph detection in the $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ model. We first observe that the degeneracy of H-free graphs can be bounded from above in terms of the corresponding Turán number:

Claim 6. *Let H be a graph, and let G be an n-node H-free graph. Then the degeneracy of G is at most $4 \cdot \mathrm{ex}(n, H)/n$.*

Proof. Consider an n'-node subgraph G' of G (for some $n' \leq n$). Since G' is also H-free, it has at most $\mathrm{ex}(n', H)$ edges, so one of the nodes of G' must have degree at most $2 \cdot \mathrm{ex}(n', H)/n'$. It therefore suffices to show that for all $n' \leq n$ we have $\mathrm{ex}(n', H)/n' \leq 2 \cdot \mathrm{ex}(n, H)/n$. This follows because $\mathrm{ex}(n, H)$ is non-decreasing in n, and because $\mathrm{ex}(2n, H) \geq 2 \cdot \mathrm{ex}(n, H)$, since we can take two disjoint copies of an extremal H-free n-node graph to obtain an H-free $2n$-node graph with $2 \cdot \mathrm{ex}(n, H)$ edges. □

Plugging this upper bound on the degeneracy into the algorithm of [2], we obtain the following upper bound for solving the H-subgraph problem in the CLIQUE-BCAST model.

Theorem 7. *For any (fixed) graph H, the H-subgraph detection problem can be solved in the CLIQUE-BCAST$_{n,b}$ model in $O\big(\frac{\mathrm{ex}(n,H)}{n} \cdot \frac{\log n}{b}\big)$ rounds.*

The running time above is obtained by taking the single $O(\mathrm{ex}(n, H) \log(n)/n)$-bit message produced by each node in the algorithm of [2], dividing it into chunks of b bits each, and broadcasting the chunks over $O\big(\frac{\mathrm{ex}(n,H)}{n} \cdot \frac{\log n}{b}\big)$ rounds.

If H has chromatic number $\chi(H) \geq 3$, the upper bound given by Theorem 7 is $O(n \log(n)/b)$; this is trivial to achieve by simply having each node broadcast its entire neighborhood. For bipartite graphs H, however, the theorem does give non-trivial upper bounds. For example, using known bounds on Turán numbers we get that the algorithm detect cycles of length 2ℓ for any $\ell \geq 2$ in time $O(n^{1/\ell} \log(n)/b)$ [4, 10], and for $2 \leq r \leq s$, $K_{r,s}$-subgraph detection can be solved in time $O(n^{2-1/r} \log(n)/b)$ [25]. If H is a tree of or a forest of a fixed size, we have $\mathrm{ex}(n, H) = O(1)$, so Theorem 7 implies that the H-subgraph detection problem can be solved in time $O(\log(n)/b)$. In the full version we show that 4-cycle detection can also be solved in the same asymptotic time, $O(\sqrt{n} \log(n)/b)$, even when nodes can only communicate over the edges of the input graph G.

Detecting subgraphs of unknown Turán number The simple algorithm above requires nodes to know $\mathrm{ex}(n, H)$. However, for most bipartite graphs H, even the asymptotic behavior of $\mathrm{ex}(n, H)$ is not known (although there are some upper and lower bounds). We now sketch how to adapt the algorithm such that even if $\mathrm{ex}(n, H)$ is not known to the nodes, the algorithm runs in time $\tilde{O}(\mathrm{ex}(n, H)/(nb))$, and with high probability it returns an H-subgraph of G if there is one. The modified algorithm is based on the observation that if G contains an H-subgraph, an appropriate random

subgraph of G with $\Theta(\mathrm{ex}(n, H))$ edges still contains a copy of H, with high probability.

Let us recall the exact properties of the one-round algorithm $\mathcal{A}$ described in [2]. The algorithm is deterministic and has an integer parameter $k \geq 1$. When run on a graph G with parameter k, every node simultaneously writes an $O(k \log n)$-bit message on the blackboard. If the degeneracy of G is at most k, in the end, all nodes learn the complete topology of G. Otherwise, all nodes learn that the degeneracy of G is more than k. In the following, let $\mathcal{A}(G, k)$ denote algorithm $\mathcal{A}$ when applied to graph G and with parameter k.

Our algorithm will make exponentially-increasing guesses $k_i = 2^i$ for the degeneracy of G, where $i = 1, 2, \ldots, \lceil \log n \rceil$, and call $\mathcal{A}$ using the current guess. Let us first dispense with some easy cases. If G does *not* contain a copy of the subgraph H, then, as we saw in Claim 6, the degeneracy of G is at most $4\,\mathrm{ex}(n, H)/n$. Within at most $O(\log(\mathrm{ex}(n, H)/n)) = O(\log n)$ steps we will reach an appropriate guess for the degeneracy, and when we call $\mathcal{A}$ with this guess, all nodes will learn the entire topology of G, and detect that G is H-free. A similar case is when G does contain a copy of H, but its degeneracy is nevertheless bounded by $O(\mathrm{ex}(n, H)/n)$. In both cases the running time is bounded by $O(\mathrm{ex}(n, H) \log(n)/(nb))$, which we obtain by splitting the messages produced by $\mathcal{A}$ into chunks of b bits, such that each chunk can be broadcast in one round.

We therefore focus on the case where the degeneracy k of G satisfies $k \gg \mathrm{ex}(n, H)/n$. Our goal now is to reduce the degeneracy by sampling a subgraph of G, in such a way that the subgraph we sample will still contain a copy of H. By Claim 6, as long as we select a subgraph that is "not too sparse"— specifically, as long as our subgraph has degeneracy $4\,\mathrm{ex}(n, H)/n$ or greater—it will still contain a copy of H. However, we do want a subgraph with degeneracy at most $c \cdot \mathrm{ex}(n, H)/n$ for some constant $c > 4$, so that we can call $\mathcal{A}$ and obtain the desired running time.

It can be shown that if each edge of G is independently sampled with some probability $p \in [0, 1]$, then as long as $k \cdot p = \Omega(\log n)$, the random subgraph induced by all the sampled edges has degeneracy $\Theta(k \cdot p)$. Thus, a good strategy would be to sample each edge of G with probability $p = \Theta(\mathrm{ex}(n, H)/(kn))$, reducing the degeneracy to $\Theta(\mathrm{ex}(n, H)/n)$ but not less than $4\,\mathrm{ex}(n, H)/n$. Unfortunately, it is not clear how to quickly perform the sampling in a distributed way in the CLIQUE-BCAST model in such a way that each node will know which of its edges have been selected (which is necessary to run the algorithm from [2]). Indeed, when we sample each edge independently with probability $p < 1/2$, the entropy of the sample, viewed

as a Boolean assignment to the edges of G, exceeds the expected number of edges in the sampled subgraph. Learning this information seems to require too much communication. We therefore use non-uniform sampling, and show that the degeneracy of the sampled subgraph still behaves as expected.

In addition to the above, we also do not know the "right" probability $p = \Theta(\mathrm{ex}(n, H)/(kn))$, because we know neither $\mathrm{ex}(n, H)$ nor the degeneracy k of G. As we said, we will be guessing the degeneracy, but in addition we will also make exponentially-decreasing guesses for p.

More specifically, the sampling is done as follows. Let n be the number of nodes of $G = (V, E)$. We define $\ell := \lfloor \log_2 n \rfloor$ and $N := 2^\ell$ to be the largest power of 2 not exceeding n. Each node $v \in V$ independently picks an integer X_v uniformly from $\{0, \ldots, N-1\}$. For each integer $j \in \{0, \ldots, \ell\}$, we define a random subgraph $G_j = (V, E_j)$ of G, which roughly speaking corresponds to sampling each edge with probability $p = 2^{-j}$ (but the edges are not independent of each other). For each $j \in \{0, \ldots, \ell\}$, the edge set E_j is defined as follows:

$$E_j := \left\{ \{u, v\} \in E : X_u \equiv X_v \pmod{2^j} \right\}.$$

The graphs $G_0, \ldots, G_\ell$ can be constructed simultaneously in $O(\log(n)/b)$ rounds: each node $v \in V$ broadcasts its random number X_v to all its neighbors, who can then determine which of their edges are in E_j for each $j \in \{0, \ldots, \ell\}$. Since $X_v \in [n]$, this requires $O(\log(n)/b)$ rounds.

For each edge $e \in E$, the probability that $e \in E_j$ is exactly 2^{-j}. In general, the edges are not independent, but for any specific node $v \in V$, the edges of node v *are* independent of each other; therefore, for any subset $S \subseteq V$, the number of nodes in S that are adjacent to v in E_j is binomially distributed with parameters $|S|$ and 2^{-j}.

We now show that the degeneracy of the sampled subgraphs behaves as expected: it is roughly $k \cdot p$, where $p = 2^{-j}$ is the sampling probability of an edge. In the following, for each $j \in \{0, \ldots, \ell\}$, let K_j be a random variable denoting the degeneracy of G_j.

Lemma 8. *Let $c > 0$ be a sufficiently large constant. With high probability, for each $j \in \{0, \ldots, \ell\}$ such that $k \cdot 2^{-j} \geq c \log n$ we have $0.9k \cdot 2^{-j} \leq K_j \leq 1.1k \cdot 2^{-j}$.*

Sketch. Fix $j \in \{0, \ldots, \ell\}$ such that $k \cdot 2^{-j} \geq c \log n$. First let us bound the degeneracy K_j of G_j from below, by exhibiting a subgraph of G_j that has minimal degree at least $0.9k \cdot 2^{-j}$ (w.h.p.). To this end, let $F = (V_F, E_F)$ be a subgraph of G that has some node $v \in V_F$ with degree k (we know

that such a subgraph exists by definition of the degeneracy), and let $F_j := (V_F, E_F \cap E_j)$. As explained above, the distribution of the degree of v with respect to $E_F \cap E_j$ is binomial with expectation $2^{-j} \cdot k$, because each neighbor of v in F survives independently of the other neighbors with probability 2^{-j}. Since $k \cdot 2^{-j} \geq c \cdot \log n$, when we select c large enough a standard Chernoff bound implies that the degree of v in F_j is at least $0.9k \cdot 2^{-j}$ w.h.p., and therefore the degeneracy of G_j is also at least $0.9k \cdot 2^{-j}$.

Now let us show that w.h.p. the degeneracy K_j is also bounded from above by $1.1k \cdot 2^{-j}$. Because $G = (V, E)$ has degeneracy k, there is an ordering $v_1, \ldots, v_n$ of V such that for each $r \in [n]$, the degree of v_r in the subgraph $G[V_{\geq r}]$ induced by nodes $V_{\geq r} := \{v_r, \ldots, v_n\}$ is at most k. (To find such an ordering, start with a node that has degree at most k in all of G, then at each step consider the subgraph induced by the remaining nodes; since G has degeneracy k, some node has degree at most k in this subgraph, and we can select this node and continue.) By the same reasoning as above, for any $r \in [n]$, the expected degree of v_r in $G_j[V_{\geq r}]$ is at most $k \cdot 2^{-j}$. A standard Chernoff bound and a union bound over all $r \in [n]$ show that w.h.p. each node v_r has degree at most $1.1k \cdot 2^{-j}$ in $G_j[V_{\geq r}]$.

To show that G_j has degeneracy at most $1.1k \cdot 2^{-j}$, let $F = (V_F, E_F)$ be a subgraph of G_j; we must find some node with degree at most $1.1k \cdot 2^{-j}$ in F. Let r be the minimum index such that $v_r \in V_F$. Then F is a subgraph of $G_j[V_{\geq r}]$, and the degree of v_r in F is no greater than its degree in $G_j[V_{\geq r}]$, namely $1.1k \cdot 2^{-j}$. □

Combining all the ingredients above, we obtain the following algorithm for H-subgraph detection. We assume that calling $\mathcal{A}(G_j, k)$ sets a flag *success*, which indicates whether or not the parameter k was large enough for the nodes to learn the entire topology of G_j (i.e., at least as large as the true degeneracy G_j).

Theorem 9. *If G is H-free, the algorithm outputs "no H-subgraph" and terminates after $O\big(\operatorname{ex}(n, H) \log^2 n/(nb)\big)$ rounds. If G contains a subgraph isomorphic to H, the algorithm finds such a subgraph with high probability after $O\big(\operatorname{ex}(n, H) \log^2 n/(nb) + \log^3 n/b\big)$ rounds.*

3.2 Lower Bounds

Our lower bounds on subgraph detection are obtained by reduction from the 2-party set disjointness problem. The reductions all use the same basic method: we split the nodes in $[n]$ into two sets A, B, "owned" by Alice and Bob respectively. We also fix an n-vertex "template graph" G' which

```
sample G_0, ..., G_ℓ;
for i = 1, 2, ..., ⌈log_2 n⌉ do
    k_i := 2^i;
    for j ∈ {0, ..., ℓ} do
        run algorithm A(G_j, k_i);
        if success and H-subgraph H' detected then
            return H'
        else if success then
            return "no H-subgraph"
```

contains many copies of H. The players then use their inputs X, Y to construct a subgraph G of G', with each player controlling the edges internal to the set of nodes it owns, in such a way that a copy of H appears in G iff X and Y are not disjoint.

The key technical challenge is designing G' in such a way that no copies of H appear inside the subgraphs G_A, G_B induced by A and B, so that the answer to the H-subgraph-detection problem always depends on both X and Y. We will now describe a family of graphs that achieve this requirement. In the following definition we introduce a second graph F, whose edges will serve as the elements of the set disjointness instance.

Definition 10. *Let $H = (V_H, E_H)$ and $F = (V_F, E_H)$ be two graphs. We say that G' is a* (H, F)-lower bound graph *if G' satisfies the following properties:*

(1) G' contains two vertex-disjoint subgraphs $F_A = (V_A, E_A)$ and $F_B = (V_B, E_B)$ that are both isomorphic to F.

(2) There exist isomorphisms $\varphi_A : V_F \to V_A, \varphi_B : V_F \to V_B$ from F to F_A and F_B, respectively, such that

I. For every edge $\{u, v\} \in E_F$, G' contains a subgraph $H' = (V_{H'}, E_{H'})$, where

(a) H' is isomorphic to H,
(b) $\{\varphi_A(u), \varphi_A(v)\}$ and $\{\varphi_B(u), \varphi_B(v)\}$ are both edges of H',
(c) $V_{H'} \cap (V_A \cup V_B) = \{\varphi_A(u), \varphi_A(v), \varphi_B(u), \varphi_B(v)\}$.

II. The subgraphs described above are the only *subgraphs of G' isomorphic to H; that is, if H' is a subgraph of G' that is isomorphic to H, then there is an edge $\{u, v\} \in E_F$ such that (b) and (c) hold.*

We associate with each (H, F)-lower bound graph a fixed $F_A = (V_A, E_A), F_B = (V_B, E_B), \varphi_A$ and φ_B satisfying the requirements above (if there is more than one choice, we choose arbitrarily). For an edge $e = \{u, v\} \in E_F$ and a mapping φ, we let $\varphi(e) = \{\varphi(u), \varphi(v)\}$ denote the image of e under φ.

An essential property of Definition 10 is the following:

Observation 11. *Suppose that $G = (V, E)$ is a subgraph of an (H, F)-lower bound graph $G' = (V', E')$ such that $E' \setminus (E_A \cup E_B) \subseteq E$. Then G contains a copy of H iff for some edge $e \in E_F$ we have $\varphi_A(e), \varphi_B(e) \in E$.*

We will map this property into a set disjointness instance, with Alice controlling the edges of F_A and Bob controlling those of F_B. We will generally aim to have F be as dense as possible, to increase the size of the set disjointness instance.

In some cases we will be able to apply our reductions to show lower bounds for the more general CONGEST model, where the input graph G is also the communication network, and nodes communicate by unicast over the edges of G. We denote this model by CONGEST-UCAST. Here, the parameter determining the efficiency of the reduction will be the size of the cut between the nodes owned by each player.

Definition 12. *We say that an (H, F)-lower bound graph $G' = (V', E')$ is* δ-sparse*, for $\delta > 0$, if there is a partition A, B of V' such that $V_A \subseteq A, V_B \subseteq B$ and the size of the cut (A, B) is at most $\delta \cdot |V'|$.*

Before describing how we can construct lower bound graphs for specific H and F, let us show how we can use these graphs to obtain a reduction from 2-party set disjointness to H-subgraph-detection.

Lemma 13. *Suppose that for $H = (V_H, E_H)$ and $F = (V_F, H_F)$, there exists an n-node (H, F)-lower bound graph $G' = (V', E')$. Then the number of rounds required to solve H-subgraph-detection in $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ with constant error probability is at least $\Omega(|E_F|/(nb))$.*

Further, if the (H, F)-lower bound graph G' is δ-sparse for some $\delta > 0$, the number of rounds required to solve H-subgraph-detection in $\mathsf{CONGEST\text{-}UCAST}_{n,b}$ with constant error probability is at least $\Omega(|E_F|/(\delta nb))$.

Sketch. The proof follows by reduction from 2-party set disjointness with $|E_F|$ elements, each representing an edge of F. We partition the nodes of G' between Alice and Bob, with each player simulating $\Theta(n)$ nodes, including all nodes of F_A (for Alice) or all nodes of F_B (for Bob). The players construct a subgraph of G', putting in edges of F_A or F_B only if the corresponding

edge appears in their input, and then simulate the H-subgraph detection algorithm. By Observation 11, G' contains a copy of H iff the players' inputs are non-disjoint. □

We now show how to construct (H, F)-lower bound graphs for specific H and F. In the following, we derive lower bounds for the cases where H is a clique, a cycle, or a complete bipartite graph.

3.3 Detecting Cliques

Let $\ell \geq 4$ be an integer, and consider the problem of detecting whether a given network graph G contains K_ℓ, a clique of size ℓ.

Lemma 14. *For any $\ell \geq 4, N \geq 4$, and $n \geq 1$ such that $4(N-1)+\ell \leq n$, there is a $(K_\ell, K_{N,N})$-lower bound graph $G' = (V', E')$ with n nodes. (Here $K_{N,N}$ is the complete bipartite graph on $2N$ nodes.)*

Proof. First assume that $n = 4(N-1)+\ell$ nodes. Let $S_i := \{v_{i,1} \ldots, v_{i,N}\}$ for $i = 1, 2, 3, 4$, and let the vertices of G' be $V' := \left(\bigcup_{i=1}^{4} S_i\right) \cup \{u_1, \ldots, u_{\ell-4}\}$. The edges are defined as follows:

- For each $j \in [N]$, $\{v_{1,j}, v_{2,j}\} \in E'$ and $\{v_{3,j}, v_{4,j}\} \in E'$,
- For each $(i, i') \in \{1, 2\} \times \{3, 4\}$ and $j, j' \in [N]$, $\left\{v_{i,j}, v_{i',j'}\right\} \in E$, and finally,
- Each of the nodes $u_1, \ldots, u_{\ell-1}$ is connected to all other nodes of G'.

The proof that G' satisfies Definition 10 appears in the full version; intuitively, it is because any K_4-subgraph of G' must include exactly one node from each of the sets S_1, S_2, S_3, S_4 (these being independent sets of G'), and because $S_1, S2$ and S_3, S_4 are connected by perfect matchings, one must take four nodes of the form $v_{1,i}, v_{2,i}, v_{3,j}, v_{4,j}$, which requires the edge $\{i, j\}$ to exist in both players' inputs. □

Theorem 15. *In the CLIQUE-BCAST$_{n,b}$ model, for every fixed $\ell \geq 4$, K_ℓ-subgraph detection requires at least $\Omega(n/b)$ rounds.*

Proof. Fix n, and let $N = \lfloor (n-\ell)/4 \rfloor + 1 = \Theta(n)$. By Lemma 14, there is a $(K_\ell, K_{N,N})$-lower bound graph on n vertices. Because $K_{N,N}$ has $N^2 = \Theta(n^2)$ edges, Lemma 13 implies the theorem. □

Remark 16. *Although we have assumed that the subgraph H we are trying to detect is of constant size, the proof above continues to work for cliques K_ℓ of size up to $(1-\epsilon)n$ for any constant $\epsilon \in (0, 1)$.*

3.4 Detecting Cycles

We now prove a lower bound on the H-subgraph-detection problem when H is a cycle C_ℓ of length $\ell \geq 4$. We will use the following definitions and notation from extremal graph theory:

Definition 17. *For an integer $n > 0$ and a graph H, let $\mathrm{ex}(n, H)$ denote the maximum number of edges of any graph on n vertices that does not contain H as a subgraph. We say that $G = (V, E)$ is an* extremal H-free graph *if $|E| = \mathrm{ex}(|V|, H)$.*

Lemma 18. *Fix $\ell \geq 4$, let N be an even integer, and let $n \geq \ell \cdot (N/2)$. Then for some extremal C_ℓ-free graph F on N vertices, there exists a (C_ℓ, F)-lower bound graph on n vertices.*

Proof. Assume for simplicity that $n = \ell \cdot (N/2)$ (as in Lemma 14, it is straightforward to generalize to larger n by adding isolated nodes). If ℓ is odd, then $\mathrm{ex}(N, \ell) = N^2/4$, and we let F be the complete bipartite graph $K_{N/2,N/2} = ([N], [N/2] \times ([N] \setminus [N/2]))$. If ℓ is even, then $\mathrm{ex}(N, \ell) = \Omega(N^{3/2})$, and we let F be any extremal C_ℓ-free graph.

We construct $G' = (V', E')$ as follows. V' consists of two sets, $V_A = \{v_{A,1}, \ldots, v_{A,N}\}$ and $V_B = \{v_{B,1}, \ldots, v_{B,N}\}$, as well as $(\ell - 4)N/2$ additional nodes, which we denote by $u_{i,j}$ where either $i \in [N/2]$ and $0 \leq j \leq \lfloor \ell/2 \rfloor - 1$, or $i \in [N] \setminus [N/2]$ and $0 \leq j \leq \lceil \ell/2 \rceil - 1$. As for the edges, for any $i, j \in [N]$, $\{v_{A,i}, v_{A,j}\} \in E'$ iff $\{i, j\} \in E_F$, and similarly, $\{v_{B_i}, v_{B,j}\} \in E'$ iff $\{i, j\} \in E_F$. In addition, we connect each pair $v_{A,i}, v_{B,i}$ by a path P_i comprising all the nodes of the form $u_{i,j}$ for some j. The length of the path is $\lfloor \ell/2 \rfloor - 1$ if $i \leq N/2$, or $\lceil \ell/2 \rceil - 1$ if $i > N/2$. □

By Lemma 13, we obtain:

Theorem 19. *For every fixed $\ell \geq 4$, C_ℓ-subgraph detection requires at least $\Omega(\mathrm{ex}(n, C_\ell)/(nb))$ rounds, both in the $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ and in the $\mathsf{CONGEST\text{-}UCAST}_{n,b}$ models.*

A note on triangle detection Theorems 15 and 19 both fail to address the complexity of detecting triangles; the technique we used cannot give any non-trivial lower bounds on C_3-subgraph detection. In our reductions, each node in the graph is simulated by at least one of the two players. (It is not hard to see that any *black-box reduction*, where we only access the distributed algorithm by executing it, must have this feature.) In order to simulate a node, the player must know the input edges adjacent to the

node. In the case of triangles this means that for any three nodes a, b, c, some player simulates at least two of the nodes; even if this player does not simulate the third node, since each node "knows" the edges adjacent to it, the player can determine all by itself whether there is a triangle on a, b, c or not.

Since the problem is, informally, too much overlap between the players' inputs, it seems reasonable to turn to a model of communication complexity that features such overlaps; and this is exactly what we will do in Section 3.6.

3.5 Detecting Complete Bipartite Subgraphs

Finally, let us give a lower bound for detecting $K_{\ell,m}$, the complete bipartite subgraph with ℓ vertices on one side and m on the other. For this result we need the following fact from [11]:

Observation 20. **[11]** *If* $\mathrm{ex}(n, H) = k$ *and* H *is bipartite, then there exists a* bipartite H*-free graph on* n *vertices with at least* $k/2$ *edges.*

Let F be a bipartite C_4-free graph on N vertices with at least $\mathrm{ex}(N, C_4)/2$ edges.

Lemma 21. *For every fixed* $\ell, m \geq 2$, $N \geq 2$ *and* $n \geq 2N + \ell + m - 4$, *there is a* $(K_{\ell,m}, F)$*-lower bound graph on* n *vertices.*

Proof. As usual, assume for simplicity that $n = 4N + \ell + m - 4$, otherwise add isolated nodes to make up the difference. Let $F = (L \cup R, E_F)$, where $L, R \subseteq [N]$ are the vertices of F.

We construct $G' = (V', E')$ as follows. Define $V_A = \{u_1, \ldots, u_N\}$ and $V_B = \{v_1, \ldots, v_N\}$. We set $V' := V_A \cup V_B \cup W_L \cup W_R$, where $W_L := \{w_1^L, \ldots, w_{\ell-2}^L\}$ and $W_R := \{w_1^R, \ldots, w_{m-2}^R\}$, and in E' we include the following edges:

- For each $i, j \in [N]$, $\{u_i, u_j\} \in E'$ and $\{v_i, v_j\} \in E'$ iff $\{i, j\} \in F$. Let $\varphi_A(i) = u_i$ and $\varphi_B(i) = v_i$ be isomorphisms from F to F_A, F_B respectively.
- All nodes in W^L are connected to all nodes in $\varphi_A(R) \cup \varphi_B(L) \cup W^R$, and all nodes in W^R are connected to all nodes in $\varphi_A(L) \cup \varphi_B(R) \cup W^L$.
- We include all edges of the form $\{u_i, v_i\}$ for $i \in [N]$.

For each edge $\{i, j\} \in E_F$, the subgraph induced by the nodes $\{u_i, u_j, v_i, v_j\} \cup W$ is isomorphic to $K_{\ell,m}$: assume w.l.o.g. that $i \in L$ and $j \in R$. The sets

$W^L \cup \{u_i, v_j\}$ and $W^R \cup \{u_j, v_i\}$ are of size ℓ and m respectively. The nodes in W^L are connected to all nodes in W^R as well as to $u_j \in \varphi_A(R), v_i \in \varphi_B(L)$, and similarly for the other side. Node u_i is connected to u_j (because of the isomorphism from F) and to v_i, and similarly v_j is connected to v_i and u_j, and there are no other edges in the induced subgraph. Therefore the induced subgraph is isomorphic to $K_{\ell,m}$.

Now suppose that H is a $K_{\ell,m}$-subgraph of G', and let L^H, R^H be the two sides of H. L^H and R^H must each include at least two nodes from $V_A \cup V_B$, as $|W^L| = \ell - 2$ and $|W^R| = m - 2$ (and no mixing between W^L, W^R is possible because $W^L \times W^R \in E'$). However, it cannot be that L^H and L^R both contain two nodes from the same set V_A or V_B, because F_A and F_B are both C_4-free. Therefore the following combinations are possible:

- L^H includes at least two distinct nodes $u_i, u_j \in V_A$, and L^R does not. Then L^R contains at least one node $v_k \in V_B$. But this is impossible, because each node in L^R must be connected to each node in L^H, but the only edges between V_A and V_B are of the form $\{u_s, v_s\}$ for some $s \in [N]$. We cannot have both $\{u_i, v_k\} \in E'$ and $\{u_j, v_k\} \in E'$.
- L^R includes at least two nodes from V_A, and L^H does not: similar. Also similar are the cases where either L^H or L^R include two nodes from V_B.
- L^H includes exactly one node $u_i \in V_A$ and one node $v_j \in V_B$, and L^R includes one node $u_p \in V_A$ and one node $v_q \in V_B$. Then either $W^L \subseteq L^H$ and $W^R \subseteq R^H$, or $W^R \subseteq L^H$ and $W^L \subseteq R^H$; assume that the first holds (otherwise rename L^H and L^R). This shows that condition (c) holds.
  In addition, since H is isomorphic to $K_{\ell,m}$, we must have $\{u_i, u_p\}, \{v_j, u_p\}, \{u_i, v_q\}, \{v_j, v_q\} \in E'$, which implies, by construction, that $\{i, p\} \in E_F, j = p, i = q$ and $\{j, q\} \in E_F$; in other words, we have one edge $\{i, j\} \in E_F$ such that $\varphi_A(\{i, j\})$ and $\varphi_B(\{i, j\})$ both appear in H, as required for (b).

□

Because $\mathrm{ex}(N, C_4) = \Theta(N^{3/2})$, we obtain the following:

Theorem 22. *For any $\ell, m \geq 1$, $K_{\ell,m}$-subgraph detection in CLIQUE-BCAST$_{n,b}$ requires $\Omega(\sqrt{n}/b)$ rounds.*

3.6 Lower Bounds on Triangle Detection

We will now show that the hardness of finding triangles in CLIQUE-BCAST is related to that of solving set disjointness in the 3-party number-on-forehead

model and obtain a lower bound of $\Omega(n/(e^{O(\sqrt{n})}b))$ on deterministic triangle detection in $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$. For randomized algorithms, current lower bounds on 3-party NOF set disjointness are not strong enough to yield a non-trivial lower bound, but we can state a conditional and restricted lower bound assuming that SAT is computationally hard.

Our reduction from set disjointness uses a family of tripartite graphs with many edge-disjoint triangles. Such families can be easily obtained from dense graphs with many large disjoint induced matchings [38], which have been of interest to the theory community in various contexts including scheduling traffic in single-hop broadcast networks [3] and linearity testing [19]. The construction we will use is the following:

Claim 23 ([38])**.** *There is a family of graphs $\{G_n\}_{n>0}$ with the following properties:*

(1) G_n is a tripartite graph $G_n = (A \cup B \cup C, E)$, where $|A| = |B| = n$ and $|C| = n/3$,

(2) G_n contains $n^2/e^{O(\sqrt{\log n})}$ triangles, and each edge of G_n belongs to exactly one triangle.

Let $m(n)$ be the number of triangles in G_n (we have $m(n) \geq n^2/e^{O(\sqrt{\log n})}$), and let $R_\epsilon^{\text{3-NOF}}(\textsc{Disj}_m)$ denote the randomized communication complexity of solving set disjointness with m elements and success probability $1-\epsilon$ in the 3-party NOF model.

Theorem 24. *The number of rounds required to solve triangle detection in $\mathsf{CLIQUE\text{-}BCAST}_{n,b}$ with error probability ϵ is at least $R_\epsilon^{\text{3-NOF}}(\textsc{Disj}_m)/O(n{\cdot}b)$.*

Proof. Fix n, and let $\mathcal{T} = \{t_1, \ldots, t_m\}$, where $m = m(n) = n^2/e^{O(\sqrt{\log n})}$, be the set of edge-disjoint triangles of G_n. Note that because G_n is tripartite, each triangle $t_i \in \mathcal{T}$ can be represented as $\{a_i, b_i, c_i\} \in A \times B \times C$. Moreover, since each edge of G_n belongs to exactly one triangle, for each edge e there is exactly one index $i(e) \in [m]$ such that e is part of the triangle $t_{i(e)}$.

Let $\mathcal{A}$ be a $\mathsf{CLIQUE\text{-}BCAST}_{(7/3)n,b}$ algorithm for triangle-detection with a running time of R rounds. We use $\mathcal{A}$ to construct an 3-party number-on-forehead protocol P for $\textsc{Disj}_m$ as follows: given input $X_A, X_B, X_C \subseteq [m]$, the three parties construct a subgraph G_X of G_n, in which

- All nodes $A \cup B \cup C$ are present, with Alice simulating the nodes in A, Bob simulating the nodes in B, and Charlie simulating the nodes in C;

- An edge e of G_n is present in G_X iff $e \in A \times B$ and $i(e) \in X_C$, or $e \in B \times C$ and $i(e) \in X_A$, or $e \in C \times A$ and $i(e) \in X_B$.

Recall that since we are working with the number-on-forehead model, the set X_C is known to Alice and Bob, X_A is known to Bob and Charlie, and X_B is known to Alice and Charlie. Therefore each of the three players can tell whether an edge adjacent to any node it needs to simulate is present in G_X or not.

To simulate a run of $\mathcal{A}$ on input G_X, each player locally simulates the states of the nodes it "owns", and in each round it writes to the shared blackboard all the messages sent by these nodes. After reading the blackboard, each player feeds all messages to each of its nodes and computes their states for the next round.

The subgraph G_X contains a triangle iff there is some triangle $t_i \in \mathcal{T}$ whose edges all appear in G_X, and this occurs iff for some $i \in [m]$ we have $i \in X_A \cap X_B \cap X_C$. When $\mathcal{A}$ terminates, with probability at least $1 - \epsilon$ there is at least one player that knows whether G_X contains a triangle, and hence also whether X_A, X_B and X_C are disjoint. This player then writes the answer on the blackboard.

The total cost of the simulation is $(7/3)n \cdot b \cdot R + 1$ bits, and since our protocol solves $\textsc{Disj}_m$ with error probability at most ϵ, we must have $R \geq R_\epsilon^{3\text{-NOF}}(\textsc{Disj}_m)/O(n)$. □

At present, the best lower bounds on 3-NOF $\textsc{Disj}_N$ are the following: for randomized algorithms, the best bound is the $\Omega(\sqrt{N})$ bound due to Sherstov [41], and for deterministic algorithms, Rao and Yehudayoff have recently shown a lower bound of $\Omega(N)$. The deterministic lower bound gives us the following:

Corollary 25. *Deterministic triangle detection in the* $\textsf{CLIQUE-BCAST}_{n,b}$ *model requires* $\Omega(n/(e^{O(\sqrt{\log n})}b))$ *rounds.*

The randomized lower bound is just shy of yielding a non-trivial bound for randomized triangle detection. However, we do obtain a conditional lower bound: in [35], Pătraşcu and Williams discuss a strong form of the Exponential Time Hypothesis which asserts that there is no SAT algorithm with running time $O(2^{\delta n})$ for any $\delta < 1$. They show that under this assumption, there is no protocol for 3-NOF $\textsc{Disj}_n$ with communication complexity $o(n)$ in which the players' local computation is performed in time $2^{o(n)}$.

Corollary 26. *If for all* $\delta < 1$ *there is no SAT algorithm with running time* $O(2^{\delta n})$*, then there is no triangle detection algorithm for* $\textsf{CLIQUE-BCAST}_{n,\mathrm{polylog}(n)}$

with round complexity $O(n/e^{\sqrt{\log n}})$ in which the nodes' local computation is in time $2^{o(n)}$.

4 Acknowledgments

We would like to thank Mohsen Ghaffari for interesting discussions, particularly about the triangle detection lower bound, and for pointing out the reference [38] to us.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. and Syst. Sciences*, 58(1):137–147, 1999.

[2] F. Becker, M. Matamala, N. Nisse, I. Rapaport, K. Suchan, and I. Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. In *Proc. 25th Int. Symp. on Parallel and Dist. Processing (IPDPS)*, pages 508–514, 2011.

[3] Y. Birk, N. Linial, and R. Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Trans. on Information Theory*, 39(1):186–191, 1993.

[4] J. A. Bondy and M. Simonovits. Cycles of even length in graphs. *J. Comb. Theory B*, 16:97–105, 1974.

[5] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[6] A. Chattopadhyay, N. Goyal, P. Pudlák, and D. Thérien. Lower bounds for circuits with MOD_m gates. In *Proc. 47th Symp. on Found. of Comp. Science (FOCS)*, pages 709–718, 2006.

[7] S. Dobzinski, N. Nisan, and S. Oren. Economic efficiency requires interaction. *CoRR*, abs/1311.4721, 2013.

[8] D. Dolev, C. Lenzen, and S. Peled. "Tri, tri agai": Finding triangles and small subgraphs in a distributed setting - (extended abstract). In *Proc. 25th Symp. on Distr. Comp. (DISC)*, pages 195–209, 2012.

[9] M. Elkin. Unconditional lower bounds on the time-approximation tradeoffs for the distributed minimum spanning tree problem. In *Proc. 36th Symp. on Theory of Comp. (STOC)*, pages 331–340, 2004.

[10] P. Erdős. Extremal problems in graph theory. *Theory of Graphs and its Applications*, pages 29–36, 1985.

[11] P. Erdős and M. Simonovits. Compactness results in extremal graph theory. *Combinatorica*, 2(3):275–288, 1982.

[12] S. Frischknecht, S. Holzer, and R. Wattenhofer. Networks cannot compute their diameter in sublinear time. In *Proc. 23rd Symp. on Discrete Alg. (SODA)*, pages 1150–1162, 2012.

[13] M. Ghaffari and F. Kuhn. Distributed minimum cut approximation. In *Proc. 26th Symp. on Distributed Comp. (DISC)*, pages 1–15, 2013.

[14] O. Goldreich and A. Warning. Secure multi-party computation. unpublished manuscript, 1998.

[15] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *Proc. SIGCOMM*, pages 159–170, 2008.

[16] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.

[17] K. A. Hansen and M. Koucký. A new characterization of ACC^0 and probabilistic CC^0. *Computational Complexity*, 19(2):211–234, 2010.

[18] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

[19] J. Håstad and A. Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Struct. Algorithms*, 22(2):139–160, 2003.

[20] S. Holzer and R. Wattenhofer. Optimal distributed all pairs shortest paths and applications. In *Proc. 31st Symp. on Principles of Distr. Comp. (PODC)*, pages 355–364, 2012.

[21] R. Impagliazzo, R. Paturi, and M. E. Saks. Size-depth tradeoffs for threshold circuits. *SIAM J. Comput.*, 26(3):693–707, 1997.

[22] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.

[23] A. Kipnis and B. Patt-Shamir. On the complexity of distributed stable matching with small messages. *Distributed Computing*, 23(3):151–161, 2010.

[24] M. Koucký, P. Pudlák, and D. Thérien. Bounded-depth circuits: separating wires from gates. In *Proc. 37th Symp. on Theory of Comp. (STOC)*, pages 257–265, 2005.

[25] T. Kövári, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3:50–57, 1954.

[26] F. Kuhn and R. Oshman. The complexity of data aggregation in directed networks. In *Proc. 25th Symp. on Distr. Comp. (DISC)*, pages 416–431, 2011.

[27] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[28] C. Lenzen. Optimal deterministic routing and sorting on the congested clique. In *Proc. 32nd Symp. on Principles of Distr. Comp. (PODC)*, pages 42–50, 2013.

[29] C. Lenzen and R. Wattenhofer. Tight bounds for parallel randomized load balancing: extended abstract. In *Proc. 43rd Symp. on Theory of Comp. (STOC)*, pages 11–20, 2011.

[30] Z. Lotker, E. Pavlov, B. Patt-Shamir, and D. Peleg. MST construction in $O(\log \log n)$ communication rounds. In *Proc. 15th Symp. on Parallelism in Alg. and Architectures (SPAA)*, pages 94–100, 2003.

[31] I. Parberry and G. Schnitger. Parallel computation with threshold functions. *J. Comput. Syst. Sciences*, 36(3):278–302, 1988.

[32] B. Patt-Shamir and M. Teplitsky. The round complexity of distributed sorting: extended abstract. In *Proc. 30th Symp. on Principles of Distr. Comp. (PODC)*, pages 249–256, 2011.

[33] D. Peleg. *Distributed Computing: A Locality-sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

[34] D. Peleg and V. Rubinovich. A near-tight lower bound on the time complexity of distributed MST construction. *SIAM J. Comput.*, 30(5):1427–1442, 1999.

[35] M. Pătraşcu and R. Williams. On the possibility of faster SAT algorithms. In *Proc. 21st Symp. on Discrete Algorithms (SODA)*, pages 1065–1075, 2010.

[36] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis ($\wedge$,$\oplus$). *Mathematical Notes of the Academy of Science of the USSR*, 41(4):333–338, 1987.

[37] A. A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Inf. Process. Lett.*, 45(6):303–307, 1993.

[38] I. Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorica*, 1976.

[39] A. D. Sarma, S. Holzer, L. Kor, A. Korman, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.*, 41(5):1235–1265, 2012.

[40] A. A. Sherstov. Separating AC^0 from depth-2 majority circuits. *SIAM J. Comput.*, 38(6):2113–2129, 2009.

[41] A. A. Sherstov. Communication lower bounds using directional derivatives. In *Proc. 45th Symp. on Theory of Comp. (STOC)*, pages 921–930, 2013.

[42] K.-Y. Siu, V. P. Roychowdhury, and T. Kailath. Computing with almost optimal size neural networks. In *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, pages 19–26, 1992.

[43] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proc. 19th Symp. on Theory of Comp. (STOC)*, pages 77–82, 1987.

[44] A. Tehrani, A. Dimakis, and M. Neely. Sigsag: Iterative detection through soft message-passing. *IEEE J. of Selected Topics in Signal Proc.*, 5(8):1512–1523, 2011.

[45] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.

[46] R. Williams. Nonuniform acc circuit lower bounds. *J. ACM*, 61(1):2, 2014.