What Is the Use of Collision Detection (in Wireless Networks)?

Johannes Schneider[1], Roger Wattenhofer[1]

Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Abstract

We show that the asymptotic gain in the time complexity when using collision detection depends heavily on the task by investigating three prominent problems for wireless networks, i.e. the maximal independent set (MIS), broadcasting and coloring problem. We present lower and upper bounds for all three problems for the Growth-Bounded Graph such as the Unit Disk Graph. We prove that the benefit of collision detection ranges from an exponential improvement down to no asymptotic gain at all. In particular, for the broadcasting problem our deterministic algorithm is running in time $O(D \log n)$. It is an exponential improvement over prior work, if the diameter D is polylogarithmic in the number of nodes n, i.e. $D \in O(\log^c n)$ for some constant c.

1 Introduction

When studying distributed algorithms for wireless networks, the algorithm designer usually chooses between two models. The popular *radio network model* buys into worst-case thinking: Concurrent transmissions cancel each other because of interference, usually to a degree such that a potential receiver cannot even sense that there has been a message collision. On the other hand, the *local model* is used to abstract away from media access issues, allowing the nodes to concurrently communicate with all neighbors.

Clearly, the local model is too optimistic. The radio network model, however, often is too pessimistic. Most wireless devices can distinguish at least four states: (i) either the wireless node is transmitting itself and is therefore not capable of noticing any other communication, or it is silently listening, usually allowing it to differentiate between the other three states: (ii) the media is free because nobody is transmitting, (iii) at least one node is transmitting and the message can be decoded, and (iv) more than one node is transmitting but no message can be decoded. In the last case the listening node can sense that there are transmissions happening, e.g. there is energy on the channel in a wireless network. This model is called the *collision detection model*.

Furthermore, many algorithms for wireless networks are designed for general graphs. This model does not capture the nature of (somewhat) circular transmission ranges of wireless devices. Therefore, within the wireless computing community the so-called *Unit Disk Graph (UDG)* and variations of it, e.g. the *Quasi Unit Disk Graph*, have been widely adopted. In the UDG two nodes are adjacent if their distance is at most 1. We use a generalized model of these geometric graphs, i.e. *Growth-Bounded Graphs (GBG)*, which restrict the size of an independent set in the neighborhood of a node. Interestingly, we show that the lower bound for general graphs without collision detection for deterministic broadcasting can be adapted to GBG without any asymptotic change. The lower bound for randomized algorithms can be adapted as well yielding no asymptotic change already for graphs of polylogarithmic diameter (in the number of nodes n). Thus, the choice of the GBG model does not seem to render the problem more simple.

We make the same assumptions about the graph, wake-up, topology etc. in both models. In particular, we assume that an estimate of n is known. Without an estimate of n a transmission takes $\Omega(\frac{n}{\log n})$ in the radio network model, yielding a clear advantage for algorithms employing collision detection. For an overview of lower and upper bounds see Table 1. All in all, an advantage of collision detection is that it allows to design fast deterministic algorithms giving reliable bounds on the time complexity. For example, our MIS algorithm is asymptotically optimal, and also considerably faster (i.e. a factor of $\log n / \log\log n$) than the best possible MIS algorithm for the radio network model. For broadcasting, our deterministic algorithm can be exponentially faster than the best deterministic counter part in the radio network model. For coloring, the current lower and upper bound show that there cannot be an asymptotic gain for randomized algorithms for graphs of maximal degree $\Delta \in \Omega(\log^2 n)$.

Upper and Lower Bounds		
Problem	With Collision Detection	Without
MIS	$O(\log n)$ det. [This paper]	$O(\log^2 n)$ ra. [14]
	$\Omega(\log n)$ [This paper]	$\Omega(\log^2 n / \log\log n)$[9]
$\Delta + 1$ Col.	$O(\Delta + \log^2 n)$ ra. [16]	$O(\Delta + \log^2 n)$ ra. [16]
	$\Omega(\Delta + \log n)$ [This paper]	$\Omega(\Delta + \log n)$ [This paper]
Broadcast	$O(D \log n)$ det. [This paper]	$O(n \log n)$ [10] det.
	$\Omega(D + \log n)$ [This paper]	$\Omega(n \log_{n/D} n)$ det. [10][This paper]

Table 1. Comparison of deterministic (det.) and randomized (ra.) algorithms with/without collision detection for various problems in GBG

2 Related work

The MIS problem has been studied in many types of graphs using many different models, e.g. the UDG and its generalization the GBG [12] or geometric radio networks (GRN), e.g. [5]. In the weaker GRN model nodes are positioned in the plane and each node knows its coordinates by a GPS device or some other means (and sometimes also the coordinates of its neighbors or a bound on the distances). A node v is connected to all other nodes within some distance $dist(v)$. Often the distance is equal for all nodes, e.g. [5], and thus connectivity is the same (up to a scaling factor) as for UDG. In the message passing model, where all nodes can exchange messages at the same time (without collisions), an asymptotically optimal MIS algorithm was stated in [15] needing $O(\log^* n)$ communication rounds for GBG. We extend this algorithm in several ways in this paper. If collisions can occur, but may not be detected, in [14] a randomized algorithm taking time $O(\log^2 n)$ was given, which is optimal up to a factor of $O(\log \log n)$[9]. It even works for arbitrary wake-up, i.e. nodes do not share global time.

For the well-studied broadcasting problem under the assumption of unknown topology and conditional (also called non-spontaneous) wake-up, i.e. a node can perform any computation only after detecting some activity (e.g. receiving the message or detecting energy on the channel) an optimal randomized algorithm was given in [10] running in $O(D \cdot \log(n/D) + \log^2 n)$ assuming collisions (but no detection) in general (undirected) graphs. In the deterministic case in the same paper an algorithm is described requiring $O(n \cdot \log^2 D)$ steps, which is optimal up to factor of $O(\log D)$ [3]. We extend the lower bound for deterministic algorithms[10] as well as the $\Omega(D \cdot \log(n/D))$ bound for randomized algorithms [13] to GBG. The $\Omega(\log^2 n)$ lower bound[1] cannot be extended in the same manner as discussed in Section 5.

In [2] it was shown how to broadcast a message of size $O(k)$ in time $O(k \cdot D)$ by using collision detection to forward a message bit by bit in arbitrary graphs. In the same paper the currently fastest deterministic algorithm for arbitrary message size also using collision detection was given taking time $O(n \cdot D)$. Thus for the crucial class of GBG in the area of wireless networks our algorithm is an exponential improvement for graphs of polylogarithmic diameter. In [8] broadcasting is discussed with and without collision detection using "advice", i.e. each node is given some number of bits containing arbitrary information about the network. It is shown that for graphs, where constant broadcasting time is possible, $O(n)$ bits of advice suffice to achieve optimal broadcasting time without collision detection, whereas $o(n)$ bits are not enough (even with collision detection at hand). In case of GRN only a constant number of bits is sufficient.

[5] assumes a GRN, where every node can detect collisions and knows its position. This allows to assign nodes into grid cells, which is the key to achieve asymptotically optimal broadcasting time of $\Theta(D + \log n)$.

In [11] an $O(n)$ time deterministic algorithm for the problem of leader election with collision detection for arbitrary networks was given. In [18] a randomized leader election protocol is given for single-hop networks running in expected time $O(\log \log n)$. In [4] deterministic algorithms for consensus and leader election were studied for single-hop networks, i.e. the underlying graph forms a clique. With collision detection both tasks can be performed in $\Theta(\log n)$, whereas without collision detection time $\Omega(n)$ is required. As in this paper (see Algorithm

Asynchronous MIS), round coding was used to synchronize rounds. For single-hop networks time $\Omega(k(\log n)/\log k)$ [7] is needed by any deterministic algorithm until k stations out of n transmit using collision detection.

3 Model and Definition

Communication among nodes is done in synchronized rounds. In every round a node v can either listen or transmit. A listening node v can successfully receive a message in round i, if exactly one neighbor $u \in N(v)$ was transmitting in round i. We say a node v *detected transmission (dT)* in round i, if the node was listening in round i, if it has at least one transmitter in its neighborhood $N(v)$.

We assume that n is known and all nodes have unique IDs from the interval $[1, n]$ using the same number of bits, i.e. small IDs have a prefix with 0s such that all IDs have equal length.[1] All our algorithms are shown to work in case of asynchronous wake-up, i.e. each node wakes-up at an unknown point in time. Only after its wake-up it is able to follow ongoing communication. The time complexity of an algorithm denotes the number of rounds until a solution has been computed for all nodes, i.e. it denotes the time from the wake-up of the last node until all nodes have computed a solution. For broadcasting we focus on conditional (or non-spontaneous) wake-up, where nodes wake-up and can perform computations (and transmissions) only after they detected transmission for the first time.

A set S is a maximal independent set (MIS), if any two nodes $u, v \in S$ have hop distance at least 2 and every node $v \in V \setminus S$ is adjacent to a node $u \in S$. A MIS S of maximum cardinality is called a maximum independent set. We model the communication network using undirected growth-bounded (also known as bounded-independence) graphs(GBG):

Definition 1 *A graph $G = (V, E)$ is growth-bounded if there is a polynomial bounding function $f(r)$ such that for each node $v \in V$, the size of a MaxIS in the neighborhood $N^r(v)$ is at most $f(r)$, $\forall r \geq 0$.*

In particular, this means that for a constant c the value $f(c)$ is also a constant. A subclass of GBGs are (Quasi)UDGs, which have $f(r) \in \mathrm{O}(r^2)$.

We denote by $\log^{(j)} n$ the binary logarithm taken j times recursively. Thus $\log^{(1)} n = \log n$, $\log^{(2)} n = \log\log n$, etc. To improve readability we assume that $\log^{(j)} n$ is an integer for any j. The term $\log^* n$ denotes how often one has to take the logarithm to get down to 1, i.e. $\log^{(\log^* n)} n \leq 1$.

4 MIS Algorithm

We present an algorithm containing the most essential ideas assuming simultaneous wake-up of all nodes in Section 4.1. In Section 4.2 we show how to extend the algorithm to allow for arbitrary wake-up times.

[1] A polynomial bound n^c of the number of nodes n and IDs chosen from the range $[1, n^c]$, would yield the same asymptotic run time for all our algorithms.

4.1 MIS, synchronous Wake-Up

In our deterministic algorithm a node performs a sequence of competitions against neighbors. After a competition a node might immediately compete again or it might drop out and wait for a while or it might join the MIS. During a competition a node transmits a value in a bit by bit manner, i.e. one bit per round only.

Since messages cannot be exchanged in parallel among interfering nodes, it looks like one communication round of a competition in the local model requires potentially $\Delta + 1$ rounds in the collision detection model. However, concurrent communication despite interference is possible, if a node v transmits its value r_v^j (with $r_v^0 := ID_v$) bit by bit (line 8 to 14), to get value r_v^{j+1} which is used to update its state and for the next competition. In case bit k of r_v^j is 1, node v transmits otherwise it listens. It starts from the highest order bit of r_v^j and proceeds bit by bit down to bit 0. As soon as it detected a transmission for the first time, say for bit l, node v sets its value r_v^{j+1} to l (line 11) and does not transmit for the remaining bits. If it has never detected a transmission while communicating r_v^j, its result is $\log^{(j)} n$. For example, consider the first competition of three nodes u, v, w, which form a triangle. Let ID_u be 1100, ID_v be 1001 and ID_w be 1101. Initially, all assume to have have highest ID, i.e. result $r_u^1 = r_v^1 = r_w^1 = 4$. In the first round all nodes transmit. In the second u and w transmit. Node v detects a transmission and sets its result to 1 and waits. In the third round no node transmits and in the fourth round w transmits and node u sets its result to 3, while w keeps its (assumed) result 4.

After each competition the states are updated in parallel (see algorithm Update State). A node starts out as *undecided* and competes against all *undecided* neighbors. For the first competition, which is based on distinct IDs, we can be sure that when node v transmitted its whole ID, i.e. has result $r_v^1 = \log n$, no other node $u \in N(v)$ has the same result $r_v^1 = r_u^1 = \log n$, since IDs differ. Thus, node v joins the MIS and informs its neighbors. All nodes in the MIS and their neighbors remain quiet and do not take part in any further competitions. For any competition $j > 1$ several nodes might be able to transmit their whole result bit by bit without detecting a transmission, e.g. $r_u^{j+1} = r_v^{j+1} = \log^{(j)} n$ for two adjacent nodes u, v. In this case, node v changes its state to marked M. A marked node is on its way into the MIS but it will not necessarily join. A neighbor of a marked node remains quiet for a while. More precisely, the algorithm can be categorized into stages (lines 3 to 17), consisting of $f(2) + 1$ phases (lines 4 to 13), being composed of a sequence of $\log^* n + 2$ competitions. A node changes its state from undecided to some other state within a phase. An M node changes back to undecided after a phase (line 16). A neighbor of a marked node, i.e. an N_M node, changes back to undecided after a stage (line 18) and competes again in the next stage.

In order to update the state of neighbors of nodes having joined the MIS or having become M, two rounds are reserved. One round is used by M nodes to signal their new presence (line 4 in Algorithm Update State) and the other by

Algorithm MIS

For each node $v \in V$
1: State s_v := *undecided*
2: **for** l:=1 to $f(f(2)+2)$ by 1 **do**
3: **for** i:=0 to $f(2)$ by 1 **do**
4: $r_v^0 := ID_v$
5: **for** j:=1 to $\log^* n + 2$ by 1 **do**
6: $r_v^j := \log^{(j)} n$
7: **for** k:=0 to $\log^{(j)} n$ by 1 **do**
8: **if** $s_v = \textit{undecided}$ **then**
9: **if** (Bit k of $r_v^{j-1} = 1) \wedge (r_v^j = \log^{(j)} n)$ **then** transmit
10: **else if** (Detected transmission) $\wedge (r_v^j = \log^{(j)} n)$ **then** $r_v^j := k$
11: **end if**
12: **end if**
13: **end for**
14: Update state s_v
15: **end for**
16: **if** $s_v = M$ **then** s_v := *undecided* **end if**
17: **end for**
18: **if** $s_v = N_M$ **then** s_v := *undecided* **end if**
19: **end for**

MIS nodes (line 7). All other nodes listen during these rounds and update their states if required (lines 10 and 11).

Theorem 1 *The total time to compute a MIS is in* $O(f(f(2)+2)\log n) = O(\log n)$ *and messages of one bit are sufficient.*

The proof for Algorithm MIS can be found in the technical report [17].

4.2 MIS, asynchronous Wake-Up

Unfortunately, asynchronism introduces some difficulties. For instance, if a node wakes up and transmits without having any information about the state of its neighbors then it might disturb and corrupt an ongoing computation of a MIS. Therefore, all nodes inform their neighbors concurrently about their state and current activity. We guarantee a synchronous execution of Algorithm MIS without disturbance of woken-up nodes by using a schedule repeating after six rounds (see Algorithm Asynchronous MIS).

The idea is that nodes involved in a computation (or in a MIS) transmit periodically and thereby, force woken-up neighbors to wait. More precisely, upon wake-up a node listens until no neighbor has transmitted for 7 rounds. If a node has detected transmission for two consecutive rounds it knows that there is a neighbor in the MIS. A node executes Algorithm MIS by iterating the six round schedule as soon as it has not detected transmission for 7 rounds. A node transmits in the first round, if it executes or is about to execute Algorithm

Algorithm Update State

For each node $v \in V$

if $(s_v = \textit{undecided}) \wedge (r_v^j = \log^{(j)} n)$ **then**
  if $j = 1$ **then**
    $s_v := MIS$
    Wait 1 round and transmit
  else
    $s_v := M$
    Transmit and wait 1 round
  end if
else
  if (Detected transmission) $\wedge (s_v = \textit{undecided})$ **then** $s_v := N_M$ **end if**
  if Detected transmission **then** $s_v := N_{MIS}$ **end if**
end if

Asynchronous MIS

Upon wake-up:

Listen until no transmission detected for 7 consecutive rounds
if ever detected transmission for 2 consecutive rounds **then** $s_v := N_{MIS}$ **else** $s_v := \textit{executing}$; SixRoundSchedule() **end if**

SixRoundSchedule():

loop forever
 if $s_v = \textit{executing}$ **then** Transmit **else** Sleep **end if**
 Sleep
 if $s_v = \textit{executing}$ **then** Execute 1 step in Algorithm MIS (Section 4.1) **else** Sleep **end if**
 Sleep
 if $s_v = MIS$ **then** Transmit twice **else** Sleep two rounds **end if**
end loop

MIS (during round 3 of the schedule). This ensures that for a node v either a neighbor starts executing Algorithm MIS concurrently with v or it waits until v has completed the algorithm. In the second and fourth round no transmissions occur. A node transmits in the fifth and sixth, if it is in the MIS. The schedule is iterated endlessly in order that nodes in the MIS continuously inform woken-up neighbors about their presence. This prevents them from attempting to join the MIS.

Let t_{MIS} denote the time Algorithm (synchronous) MIS takes for computing a MIS when all nodes start synchronously.

Theorem 2 *Algorithm Asynchronous MIS computes a MIS in time $O(t_{MIS})$.*

Proof. If a set of nodes $U \subseteq V$ start Algorithm MIS synchronously and are not disturbed by any node $w \notin U$ interfering the computation then a correct MIS is computed (see Analysis Algorithm MIS). A node v computing a MIS transmits a message at least every six rounds, since a neighbor $u \in N(v)$ must not start Algorithm MIS if it detected transmission within seven rounds, it cannot start a computation if it woke-up while v is active. Consider an arbitrary pair u, v of neighboring nodes, e.g. $u \in N(v)$ that are awake but not executing Algorithm MIS. If node v has not detected a transmission for seven rounds, it starts transmitting a message periodically every six rounds and executes Algorithm MIS. Any neighbor of v, i.e. u, that does not start at the same time, detects a transmission from v and waits.

If a node v detects two consecutive transmissions a neighbor must be in the MIS and does not take part in any new computation of a MIS. In case, it detects transmissions (but non-consecutive) ones, some neighbor $u \in N(v)$ is executing Algorithm MIS. Thus within time $O(t_{MIS})$ a node $w \in (N(u) \cup u) \subseteq N^2(v)$ within distance 2 from v joins the MIS. Since the size of a maximum independent set within distance 2 is bounded by $f(2)$ (see Model Section) within time $O(f(2)t_{MIS}) = O(t_{MIS})$ node v is in the MIS or it has a neighbor in the MIS.

4.3 Broadcast Algorithm

Our deterministic algorithm iterates the same procedure, i.e. the same schedule, using a fixed number of rounds. First, the current set of candidates (rounds 1 and 2) for forwarding the message is determined. A *candidate* is a node having the message and also having a neighbor lacking it. Second, some candidates are selected using a leader election algorithm, i.e. by computing a MIS. Finally, the chosen nodes transmit the message to all their neighbors without collision. If all nodes in the MIS transmitted the message concurrently, then no node might receive the message. This is because any node can be adjacent to more than one node in the MIS and suffer from a collision if all of them transmit concurrently. For that reason we must select subsets of the nodes in the MIS and let the nodes in each subset transmit in an assigned round. Explicitly constructing such sets is difficult in a distributed manner because a node in the MIS is unaware of the identities of the other nodes in the MIS. However, we can use the combinatorial tool of so called (n, k)-strongly selective families[6] of sets $\mathcal{F} = \{F_0, F_1, ..., F_{m-1}\}$ with $F_i \subseteq V$, which yield a direct transmission schedule of length $|\mathcal{F}| = m$ for each node in the MIS, such that every node out of the given set of k nodes can transmit to all its neighbors without collision. A node v transmits in round i if $v \in F_i$.

An essential point for making fast progress is that we distinguish between nodes that have (just) received the message and never participated in a leader election and nodes that have the message and already did so. The former ones, i.e. new candidates, are preferred for forwarding the message, since, generally, they have more neighbors lacking the message.

More precisely, in our deterministic Algorithm DetBroadcast (see Table 2) a node lacking the message (state *LackMsg*) that receives the message immediately joins the computation of leaders, i.e. of a MIS, in the next execution of the schedule by switching to state *CompMIS*. After it has participated in the leader election once, it switches to state *HaveMsg* if it has not become a leader. If it has become a leader, i.e. is in the MIS, it transmits the message and exits. A node can only reattempt to become a leader, i.e. switch back to state *CompMIS*, in case no neighbor of it has just received the message, i.e. changed from state *LackMsg* to *CompMIS*.

Schedule	State $s_v = CompMIS$	$s_v = HaveMsg$	$s_v = LackMsg$
Round 1	Transmit		Listen
2	Listen if not dT then exit		if dT then Transmit
3	Transmit	if not dT then $s_v := CompMIS$	Sleep
FOR i=1..t_{MIS}			
$3 + i$	Compute step i of Algorithm MIS	Sleep	
ENDFOR			
still round $t_{MIS} + 3$	If not joined MIS then $s_v := HaveMsg$		
FOR i=1..$\|\mathcal{F}\|$			
$3 + t_M + i$	if $v \in F_i$ then Transmit msg	Sleep	if received msg then $s_v := CompMIS$
ENDFOR			
still $3 + t_M + \|\mathcal{F}\|$	$s_v = HaveMsg$		

Table 2. Algorithm DetBroadcast, where *dT* stands for (*has*) *detected transmission* and returns true, if a node has listened and detected a transmission.

Next, we show that all neighbors of a candidate get the message within logarithmic time.

Theorem 3 *Any candidate v ends the algorithm in time $O(\log n)$.*

Proof. For any node v at most $f(2)$ nodes $u \in N^2(v)$ are in state $CompMIS$ in round $t_M + 3$, i.e. after the execution of the MIS algorithm. This follows from the correctness of the MIS algorithm and the definition of GBG. For the existence of a strongly related $(n, f(2))$ family of size $\mathrm{O}(f(2)^2 \log n)$, we refer to [6]. The time to compute a MIS is $\mathrm{O}(f(f(2)+2) \log n)$ as shown in Theorem 1. Thus one execution of the schedule takes time $O((f(f(2)+2) + f(2)^2) \log n)$.

Either a candidate v is computing a MIS itself or at least one neighbor $u \in N(v)$ does so. Assume neighbor u participates in computing a MIS S_0, joins the MIS and transmits. At least a subset of the neighbors $U \subseteq N(u)$ receives the message for the first time and any candidate $w \in U$ participates in the next computation of a MIS S_1. Assume at least one candidate $w \in U$ exists, i.e. $|U| > 0$, and a neighbor $x \in N(w)$ joins the MIS. Note that $x \notin N(u)$, since u transmitted the message to all its neighbors $w \in N(u)$. Therefore, all nodes $w \in N(u)$ have changed to state *HaveMsg* before the computation of

S_1. Therefore, some node $x \in N(w) \setminus N(u)$ receives the message for the first time and participates in the next computation of a MIS S_2. Assume a node $y \in N(x) \cap x$ joins the MIS. This node y is not adjacent to u, i.e. $y \notin N(u)$, because no nodes $N(u) \cap N(x)$ participate together with y since all neighbors $N(x)$ changed to state *HaveMsg* before the computation of S_2. Thus node y is independent of all nodes in the MIS S_0 that transmitted and also any prior nodes that transmitted. Therefore node v gets a transmitting (independent) node with three computations of a MIS within distance 4. The maximum size of any independent set is bounded by $f(4)$ within distance 4. Therefore within time $O((f(f(2)+2)+f(2)^2)f(4)\log n) = O(\log n)$ all neighbors of node v must have received the message and therefore node v cannot be a candidate any more.

Theorem 4 *Algorithm DetBroadcast finishes in time $O(D \log n)$ for a GBG.*

Proof. Due to Theorem 3 any neighbor of a node having the message also receives it within time $O(\log n)$. Therefore, within time $O(D \log n)$ any node receives the message.

5 Lower Bounds For MIS, Coloring and Broadcasting With Collision Detection

To begin with, we present two lower bounds. One showing that indeed $\Omega(\Delta)$ colors and time is needed to color a GBG and one that shows that for any Δ, i.e. also $\Delta \in O(1)$, time $\Omega(\log n)$ is needed even to make a successful transmission with high probability, i.e. $1-1/n$. The second lower bound implies a bound on the MIS and the same techniques imply an $\Omega(\log n)$ lower bound for broadcasting.

Theorem 5 *Any (possibly randomized) algorithm requires time $\Omega(\Delta)$ (in expectation) to compute a $\Delta+1$ coloring with high probability in a GBG (with or without collision detection).*

In the proof we use an argument based on information theory. Essentially, any node must figure out the identities of the nodes in its neighborhood. We show that the amount of possibly shared information about the neighborhood with $\Omega(\Delta)$ communication rounds is not sufficient to narrow down the options of distinct neighborhoods sufficiently.

Proof. Let the disconnected graph G consist of a clique C of Δ nodes and some other arbitrary subgraph such that no node $v \in C$ is adjacent to a node $u \notin C$. To color the clique C, any algorithm requires $\Delta+1$ colors. We restrict the possible choices of $\binom{n}{\Delta+1}$ cliques as follows. We pick $(\Delta+1)/2$ sets $S_0, S_1..., S_{(\Delta+1)/2-1}$, each consisting of 4 nodes, i.e. $|S_i| = 4$.(We assume that $4(\Delta+1) \leq n$.) The algorithm gets told all the sets S_i and that out of every set S_i consisting of four nodes, two nodes are in the clique C. However, it is unknown to the algorithm which two nodes out of the four are actually chosen. The algorithm must reserve two of the $\Delta+1$ colors for each set S_i, i.e. the (unknown) nodes in the set.

Assume (without loss of generality) that the algorithm assigns colors $2i$ and $2i+1$ to the chosen nodes of set S_i.

Assume an algorithm could compute a correct coloring in time Δ/c_0 for some constant $c_0 \geq 6$. Within Δ/c_0 rounds at most Δ/c_0 out of the $\Delta+1$ nodes in the clique can transmit without collision. Assume that even if there is a collision due to some transmitters, say u, v, w, in a round i, all nodes in the clique receive one message of the same node, say all nodes receive v's message.(Note, that more information about the neighborhood can only benefit a node.) Additionally, any node can detect whether there was 0,1 or more than 1 transmitter in its neighborhood. For an upper bound assume a node $v \in C$ gets to know Δ/c_0 of its neighbors, i.e. receives one message of each of these Δ/c_0 nodes, and additionally, it receives two bits of information in each round, i.e. one of the values $\{0, 1, > 1\}$ can be encoded by two bits of information, e.g. bits 11 correspond to > 1 transmitters, bits 10 correspond to 1 transmitter and bits 00 correspond to none. Thus, in total a node v gets to know at most $2\Delta/c_0$ bits. Observe that every node gets the same information, i.e. bits. The transmitted information is used to figure out, which two nodes of each set S_i are actually in the clique C in order to get a correct coloring. Since the algorithm is supposed to know already Δ/c_0 nodes, at least for $(1-1/c_0)\Delta/2$ leftover sets S_i no node of the set transmitted its identity. Therefore the algorithm can use the $2\Delta/c_0$ bits to figure out the identities of the $(1-1/c_0)\Delta$ nodes of the $(1-1/c_0)\Delta/2$ leftover sets S_i. Thus, any algorithm must decide on how many bits it spends on determining the two nodes out of the four possible in each set that are actually in its neighborhood. On average, it can use $\frac{(2\Delta/c_0)}{(1-1/c_0)\Delta/2} = 4/(c_0-1)$ bits per set. For $c_0 = 9$ for at least half all sets S_i the algorithm can use at most 1 bit. Assume set S_0 consists of nodes $\{a, b, c, d\}$. Any node in the set S_0 must make a decision, which of the two colors $\{0, 1\}$, it chooses based on its ID and a single bit. Assume node a decides in favor of color 1 given it received bit 0, i.e. $col(a|0) = 1$. Then all other nodes in the set must decide to pick color 0 if they receive bit 0, i.e. $col(b|0) = 0$, $col(c|0) = 0$ and $col(d|0) = 0$. If not, consider a node $x \in S_0$ that also decides in favor of color 1. In this case, if a and x are chosen to be in the clique C, both are adjacent and choose the same color. Thus the coloring is incorrect. Assume all nodes receive bit 1 and assume $col(a|1) = 0$ then $col(b|1) = 1$, $col(c|1) = 1$ and $col(d|1) = 1$. Thus, if out of the set S_0, nodes b, c are chosen then both decide on the same color, whatever the given bit is, i.e. they both pick color 0 if the given bit is 0 and color 1 if the bit is 1. If $col(a|1) = 1$ then $col(b|1) = 0$, $col(c|1) = 0$ and $col(d|1) = 0$ and nodes b, c decide on color 0 whatever the given bit is. Thus the coloring cannot be correct. Randomization can not increase the amount of exchanged information. Thus, in the end any algorithm must also decide on whether to choose color $\{0, 1\}$ based on a single bit. Through a case enumeration one can see that it is not possible to correctly guess the right colors with probability more than 1/4. Assume $col(a|0) = 1$ with probability $p_{\geq \frac{1}{2}} \geq 1/2$. Then all other nodes in the set must decide to pick color 0 with probability $p_{\geq \frac{1}{2}}$ to have a chance higher than 1/4 of a correct coloring. Using the same reasoning as for the deterministic case a maximum probability of 1/4 for a correct coloring of a single set using only one bit follows. Since for at least half of all $(1-1/c_0)\Delta/2$ sets S_i with unknown nodes we can use at most one bit, we expect at least $3/4(1-1/c_0)\Delta/4$ to be colored incorrectly.

Theorem 6 *There exists a graph such that for any $\Delta > 1$, any (possibly randomized) algorithm using collision detection requires time $\Omega(\log n)$ to compute a MIS (in expectation).*

Proof. Consider a (disconnected) graph where every node has degree 1, i.e. a single neighbor. Assume every node $v \in V$ knows that its degree in the network is one but it is unaware of the identity of its neighbor u. Consider a sequence of $\frac{\log n}{8}$ rounds. For every node $v \in V$ we can calculate the probability that node v transmits in round $0 \leq i < \frac{\log n}{8}$ given that it has not yet received a message but transmitted itself or listened without detecting any transmission by its neighbor for rounds $0 \leq j < i$. There must exist a set U of $n^{7/8}$ nodes such that for every round i with $i \in [0, \frac{\log n}{8} - 1]$ all nodes in U transmit either with probability $p_{\geq \frac{1}{2}}$ at least $\frac{1}{2}$ or with probability $p_{<\frac{1}{2}}$ less than $\frac{1}{2}$, since a node has only two choices in each round (transmit or not). Thus, out of n nodes on average at least $n/2^{\frac{\log n}{8}} \geq n^{7/8}$ must decide to transmit (and listen) in the same rounds for all $\frac{\log n}{8}$ rounds. Consider an arbitrary pair $u, v \in U$ and assume they are adjacent. The chance of a transmission from u to v or the other way around is at most $(1-p_{<\frac{1}{2}}) \cdot p_{<\frac{1}{2}} + p_{\geq \frac{1}{2}} \cdot (1-p_{\geq \frac{1}{2}}) \leq \frac{1}{2}$ for one round, since any term $p \cdot (1-p)$ can be at most 1/4. After all $\frac{\log n}{8}$ rounds the probability is at most $1 - \frac{1}{4^{\frac{\log n}{8}}} \leq 1 - \frac{1}{n^{1/4}}$. Assume, we randomly create $n^{7/8}/2$ pairs of nodes from the set U, such that the two nodes from each pair are adjacent. We expect for $n^{7/8}/2/n^{1/4} = n^{5/8}/2$ pairs that no message is exchanged. Thus the nodes from these pairs must make the decision, whether to join or not to join the MIS based on the same information. Let $U_1 \subseteq U$ be all nodes that decide to join the MIS with probability at least 1/2 if they have never received a message, i.e. transmitted in the same rounds as their neighbor(s), and let all other nodes be in set $U_2 = U \setminus U_1$. Either U_1 or U_2 is of size at least $|U|/2$. Assume it holds for U_1. If we pick pair after pair then the probability that both nodes are taken from U_1 is at least 1/16 for a pair independent of all previously picked pairs as long as at most $|U_1|/4 \geq |U|/8$ pairs have been chosen, i.e. for the remaining nodes in U_1 holds $|U_1| \geq |U|/4$. Thus, we expect $|U|/8/16 = |U|/128$ pairs to have both nodes in the same set U_1 or U_2. The chance that both join for a pair in U_1 is $p_{\geq \frac{1}{2}} p_{\geq \frac{1}{2}} \geq 1/4$ or none does for a pair in U_2 is $(1-p_{<\frac{1}{2}})(1-p_{<\frac{1}{2}}) \geq 1/4$. Thus, 1/4 of all the pairs having transmitted in the same round and being from the same set U_1 or U_2 either both join the MIS or not in expectation. The probability that at least half of the expected $n^{5/8}/2/128/4 = n^{5/8}/1024$ pairs transmit in the same rounds, i.e. do not exchange a message, and both nodes from the pair join the MIS or both do not join is larger than $1 - 1/n^c$ for some arbitrary large constant c using a Chernoff bound. Thus the probability that the algorithm finishes in time less than $\log n/8$ is at most $1/n^c$.

The argument for the deterministic case is analogous, i.e. an equally large set (as in the randomized case) of nodes must transmit in the same rounds and it is not possible that all pairs correctly decide to join the MIS or not for all possible choices of neighborhoods. More precisely, consider three nodes u, v, w that all transmit in the same round (given their only neighbor is also one of u, v, w). If in a graph G_1 u, v are adjacent then either u or v must join the MIS without having received a message from its neighbor, i.e. v is unaware whether its neighbor is u, v or w. Assume u joins the MIS then v cannot join. If in a graph G_2 v, w are adjacent then both transmit the same sequence and since v does not join w has to. Now if in a third graph u, w are adjacent then both join the MIS violating the independence condition of a MIS.

Observe that the above theorem even holds for synchronous wake-up. Since one can compute a MIS from a coloring in constant time, the lower bound also holds for the MIS. More precisely, for the graph in the above proof with $\Delta = 1$ one can compute a MIS from a $\Delta + 1$ coloring by putting all nodes with color 0 in the MIS.

Corollary 7 *Any (possibly randomized) algorithm using collision detection requires time $\Omega(\log n)$ to compute a coloring in GBG (in expectation).*

For broadcasting with conditional wake-up a lower bound of D is trivial. A lower bound of $\Omega(\log n)$ for networks of diameter two can be proven using the same idea as for the proof of Theorem 6.

Theorem 8 *There exists a graph such that for any $\Delta > 1$, any (possibly randomized) algorithm using collision detection requires time $\Omega(\log n)$ to make a transmission among all nodes with probability $1 - \frac{1}{n}$.*

Proof. Assume the following network of diameter two. The source is adjacent to two nodes and these two nodes in turn are adjacent to all other nodes. Consider a sequence of $\frac{\log n}{2}$ rounds. For every node $v \in V$ and every round i we can calculate the probability that node v transmits in round $0 \le i < \frac{\log n}{2}$ given that it has received the message (and possibly other information). There must exist two nodes u, v such that for all $\frac{\log n}{2}$ rounds either they both send with probability at least $\frac{1}{2}$ or less than $\frac{1}{2}$ given they received the same information, since any node has only these two options and we have that $n > 2^{\frac{\log n}{2}}$. Thus the chance of a successful transmission of either u or v to its neighbor is at most $\frac{1}{2}$ for one round and at most $1 - \frac{1}{2^{\frac{\log n}{2}}} = 1 - \frac{1}{\sqrt{n}}$ after $\frac{\log n}{2} \in \Omega(\log n)$ rounds.

6 Lower bound for broadcasting without collision detection in GBG

The lower bounds for randomized [13] as well as for deterministic [10] algorithms for general graphs can be adapted to GBG. Both rely on constructing a graph with layers $L_0, L_1, ..., L_{\Omega(D)}$, where nodes L_i in layer i are independent and they are at distance i from the source, i.e. broadcast initiator. In [10] the graph consists of two alternating layers consisting of a single node in layer i that is connected to all nodes L_{i+1} in layer $i+1$. Only a subset $W_{i+1} \subseteq L_{i+1}$ of the nodes in layer $i+1$ is connected to the single node in layer $i+2$. In the lower bound graph in [13] the nodes in layer i are connected to some nodes $W_{i+1} \subseteq L_{i+1}$ in layer L_{i-1} and L_{i+1}. There are no fixed layers of single nodes. The difficulty for the algorithm is figuring out the number of nodes in L_i. If it knows the number $|L_i|$ it can transmit with probability $1/|L_i|$, yielding an O(1) algorithm to get to the next layer. However, in [13] one could also use the same topology as in [10], i.e. every second layer consists only of a single node. Though this allowed the algorithm to pass every second layer in one round by transmitting with probability 1 for the other half of the layers the algorithm is unaware of the number of nodes in a layer.

For GBGs it is not possible for a node v in layer i to have an arbitrary number of independent nodes in layer $i+1$. Thus, we assume that all nodes in layer i form a clique. Therefore, a node in layer i knows all the successful transmissions that occurred in layer i. However, by elongating any protocol P by a factor of 4, the nodes in layer i in a general graph also know all successful transmissions in layer i. The idea is to let all layers with single nodes repeat their received message to the other layers. Since a single node is adjacent to two layers, it might face a collision if a node from each of its adjacent layers transmits concurrently and thus is not able to tell any layer if the transmission was successful. Thus, we repeat 8 rounds in a round robin fashion. Any node in layer i executes one step of the algorithm in round t if $2i \mod 8 = t \mod 8$.[2] Any node in layer i transmits in round t if $2i-1 \mod 8 = t \mod 8$ and if it received a message in the previous round. Thus a node in layer i having transmitted the message in round t knows that some node in layer i transmitted without collision, if and only if it detects transmission in round $t+1$.

Thus, we have arrived at the following proposition:

Proposition 1. *Any deterministic broadcasting algorithm takes time $\Omega(n \log_{n/D} n)$ and any randomized broadcasting algorithm takes time $\Omega(D \log \frac{n}{D})$ for GBG.*

The lower bound of $\Omega(\log^2 n)$ [1] cannot be extended in the same manner. The lower bound graph consists of two layers L_1 and L_2, where nodes within a layer are independent. In particular, layer L_2 consists of $\Omega(\log n)$ nodes. Each node l in L_2 is connected to some set H_l of nodes in L_1. There are $\Omega(\log n)$ sets H and in some round exactly one node from each set H_l must transmit in order that all nodes in L_2 receive the message. In GBG this is not the case, since in L_2 some nodes are adjacent, i.e. by definition of a GBG at most $f(2)$ nodes in L_2 can be independent. Thus, a node in L_2 having received the message might forward it to other nodes in L_2 and it might be sufficient that for only $f(2)$ out of all $\Omega(\log n)$ sets H from L_1 a node transmit to its neighbor in layer L_2.

[2] Note, that all nodes having the message are synchronized, i.e have global clocks, if the message includes the current time t.

References

1. N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *J. Comput. Syst. Sci.*, 43(2), 1991.
2. B. Chlebus, L. Gcasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. In *Symp. on Discrete Algorithms(SODA)*, 2000.
3. A. E. F. Clementi, A. Monti, and R. Silvestri. Distributed broadcast in radio networks of unknown topology. *Theor. Comput. Sci.*, 302(1-3), 2003.
4. J. Czyzowicz, L. Gasieniec, D. R. Kowalski, and A. Pelc. Consensus and mutual exclusion in a multiple access channel. In *Int. Symposium on Distributed Computing (DISC)*, 2009.
5. A. Dessmark and A. Pelc. Broadcasting in geometric radio networks . In *Journal of Discrete Algorithms*, volume 5, 2007.
6. P. Erdoes, P. Frankl, and Z. Fiiredi. Families of finite sets in which no set is covered by the union of r others. In *Israel J. of Math*, volume 51, 1985.
7. A. G. Greenberg and S. Winograd. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *J. ACM*, 32(3), 1985.
8. D. Ilcinkas, D. R. Kowalski, and A. Pelc. Fast radio broadcasting with advice. *Theor. Comput. Sci.*, 411(14-15), 2010.
9. T. Jurdziński and G. Stachowiak. Probabilistic Algorithms for the Wakeup Problem in Single-Hop Radio Networks. In *Int. Symp. on Algorithms and Computation (ISAAC)*, 2002.
10. D. R. Kowalski and A. Pelc. Broadcasting algorithms in radio networks with unknown topology. In *Distributed Computing*, volume 18, 2005.
11. D. R. Kowalski and A. Pelc. Leader election in ad hoc radio networks: A keen ear helps. In *ICALP (2)*, 2009.
12. F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In *Int. Symp. on Distributed Computing (DISC)*, 2005.
13. E. Kushilevitz and Y. Mansour. An omega(d log(n/d)) lower bound for broadcast in radio networks. In *Symp. on Principles of Distributed Computing (PODC)*, 1993.
14. T. Moscibroda and R. Wattenhofer. Maximal Independent Sets in Radio Networks. In *Symp. on Principles of Distributed Computing (PODC)*, 2005.
15. J. Schneider and R. Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In *Symp. on Principles of Distributed Computing(PODC)*, 2008.
16. J. Schneider and R. Wattenhofer. Coloring Unstructured Wireless Multi-Hop Networks. In *Symp. on Principles of Distributed Computing (PODC)*, 2009.
17. J. Schneider and R. Wattenhofer. What Is the Use of Collision Detection (in Wireless Networks)? In *TIK Technical Report 322, ftp://ftp.tik.ee.ethz.ch/pub/publications/TIK-Report-322.pdf*, 2010.
18. D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. In *SIAM Journal on Computing*, volume 15, 1986.