Gathering information in a peer-to-peer network

Laurent Beaudou* Roland Grappe† Geña Hahn‡

Abstract

In this paper, we address the problem of gathering scattered information in a given node of a peer-to-peer network. The network is modeled by a graph G. Suppose each vertex in G has a unit of information and that all the units must be collected at a vertex u in G. Assuming that a vertex can receive (from its neighbours) an unlimited number of units at each discrete moment but can only send one at a time, find the shortest collection time, $\mathrm{col}_u(G)$, needed to collect all the information at u and an optimal protocol that achieves this.

We derive lower and upper bounds for the problem, give a polynomial time algorithm in the general case, and a linear time algorithm for hypercubes.

1 Introduction

The problem has its origin in a file decomposition by Michael O. Rabin [7]. A file is split into N pieces of the same length in such a way that any K pieces are sufficient to rebuild the original file. This can be done efficiently, without significantly increasing the size of the file. Such a decomposition (obtained by an IDA: information dispersal algorithm) has key applications in file sharing since it provides at the same time security and anonymity of the data (if each server has a small number of parts, destroying one server does not destroy the data, and no server has enough of the file to be charged

*Université de Montréal, Canada. E-mail: lbeaudou@ujf-grenoble.fr
†Università di Padova, Italy. E-mail: roland.grappe@g-scop.inpg.fr
‡Université de Montréal, Canada. E-mail: hahn@iro.umontreal.ca

for its content). Recently these ideas have been applied to file sharing in peer-to-peer networks [4].

Our model is a connected graph G with $N+1$ vertices (including a receiver) that corresponds to the overlay network. Each vertex v (corresponding to a server in the network) distinct from the receiver has one of the N parts, and each of the N parts is in the graph. Each edge e of G has a weight $w(e)$ that corresponds to the propagation delays in the network.

A node can send at most one part at a time (that corresponds to limited upload speed), but can receive many parts at the same time (unlimited download speed). Two parts cannot be combined and sent as a bigger message, they have to be sent separately (sending ports have a limited buffer).

The model is synchronous: if a part P is sent by u to its neighbour v at time t, it will be received by v at time $t + w(uv)$. At any time $t \leq t' \leq t + w(uv)$, u may send nothing else. Also, v will only be able to send P to someone else after time $t + w(uv)$.

The aim is to schedule transmissions in the network so that the receiver collects K distinct parts of the original file (we refer to the protocol realizing the schedule as a *transmission* as well).

This problem looks very similar to the packet routing questions, which have been the object of a wide interest over the last decades [5, 6, 8]. The difference is that the restrictions are made on the edges (which cannot transmit more than some number of packet at each step). As close as it seems we were not able to find an easy reduction to theses problems.

We want to minimize one of the following two parameters (2 distinct problems):

1. the total bandwidth (sum of the weights of occurrences of the edges used by a transmission – each edge can be counted several times),

2. the total time of the transmission (we stop as soon as the receiver has collected K distinct parts).

In this paper, we focus on the second problem in the case $K = N$ and $w(e) = 1$ for each edge e of G. In Section 2, we define the problem formally. In Section 3, we derive lower and upper bounds for the problem. In Section 4, we give a polynomial algorithm for the problem. In Section 5, we solve the problem exactly for the hypercube of dimension d and give a linear-time algorithm for finding a transmission.

2 Definitions

Let $G = (V, E)$ be a connected graph (disconnected graphs are not relevant to the problems at hand) and let $u \in V$ be a vertex of G. We seek to solve two closely related communication problems on G. Any sending and receiving is between adjacent vertices in G.

Problem 1. *Suppose each vertex has a unit of information and that all the units must be collected at u. Assuming that a vertex can receive an unlimited number of units at each discrete moment but can only send one at a time, find the shortest collection time, $col_u(G)$, needed to collect all the information at u and an optimal protocol that achieves this.*

Problem 2. *Suppose the vertex u has $|V|$ units of information that must be distributed to all the vertices of G (one unit per vertex). Assuming that each vertex can send an unlimited number of information units at each discrete moment but can only receive one, find the minimum distribution time $dist_u(G)$ needed to distribute all the information units and an optimal protocol that achieves this.*

It is also of interest to find $\text{dist}_M(G) = \max\{\text{dist}_u(G) : u \in V\}$ and $\text{dist}_m(G) = \min\{\text{dist}_u(G) : u \in V\}$ when considering the first problem and $\text{col}_M(G) = \max\{\text{col}_u(G) : u \in V\}$ and $\text{col}_m(G) = \min\{\text{col}_u(G) : u \in V\}$ when looking at the second problem.

It is easy to see that the two problems are equivalent. It suffices to reverse the protocol of one to get a protocol for the other that runs in the same time. This leads to Proposition 1.

Proposition 1. *For any graph G and any vertex u in G, $col_u(G) = dist_u(G)$.*

From now on, we only deal with Problem 1.

3 Bounds

In this section, we derive simple lower and upper bounds on $\text{col}_u(G)$ for any graph $G = (V, E)$ and a vertex u in G.

3.1 Lower bounds

For any $S \subset V \setminus \{u\}$, let the *boundary* of S, denoted by $\partial S \subseteq S$, be the set of vertices of S which have at least one neighbour in $V \setminus S$. For any set $A \subseteq V$, $d(A, u)$ denotes the minimum distance $d(u, v)$ taken over all $v \in A$.

Proposition 2. *We have:*

$$col_u(G) \geq \max_{S \subseteq V \setminus \{u\}} \left(\left\lceil \frac{|S|}{|\partial S|} \right\rceil + d(S, u) - 1 \right). \quad (1)$$

Proof. Take any set $S \subseteq V \setminus \{u\}$ (see Figure 1). At time 0, S contains $|S|$ units of information that have to be collected in u. At each step, no more than $|\partial S|$ units can leave S. This gives $(\left\lceil \frac{|S|}{|\partial S|} \right\rceil)$, the first term of (1).

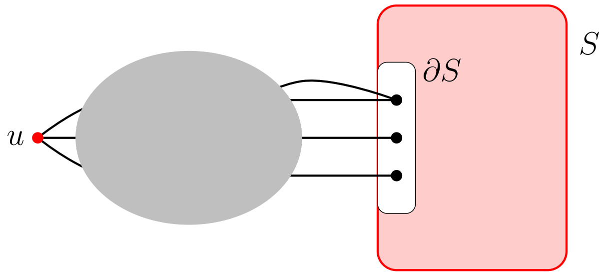

Figure 1: A set of vertices S and its boundary ∂S.

Once a unit of information has left S, it is at distance at least $d(S, u) - 1$ of u. These two terms sum together and we get $\text{col}_u(G) \geq \left\lceil \frac{|S|}{|\partial S|} \right\rceil + d(S, u) - 1$. This is true for any S, it remains to take the maximum value over these S and we obtain (1). ☐

For a vertex $v \in V$, let $\epsilon(v)$ denote the eccentricity of v in G, i.e. $\epsilon(v) = \max\{d(v, v') : v' \in V\}$. If we take $S = V \setminus \{u\}$ and $S = \{v\}$ with $d(u, v) = \epsilon(u)$, Proposition 2 implies the two following results, which are both optimal.

Corollary 1. *For any graph G and any vertex u in G, $col_u(G) \geq \epsilon(u)$.*

Corollary 2. *For any graph $G = (V, E)$ and any vertex u in G, $col_u(G) \geq \left\lceil \frac{|V|-1}{\deg(u)} \right\rceil$.*

3.2 Upper bound

We now give an upper bound.

Proposition 3. *Let $G = (V, E)$ be a connected graph, let $u \in V$. Then, $col_u(G) \leq |V| - 1$.*

Proof. Let us prove this by induction on $|V|$. We actually need a stronger induction hypothesis, which is:

$\mathcal{P}(n)$: *for every connected graph $G = (V, E)$ on n vertices, and every $u \in V$, there is a protocol collecting every unit of information in at most $n - 1$ steps such that u receives at least one unit of information at each step.*

$\mathcal{P}(1)$ is trivially true. Let us suppose that $\mathcal{P}(i)$ is true for $i \leq n$, we shall prove $\mathcal{P}(n + 1)$. Let $G = (V, E)$ be a connected graph on $n + 1$ vertices and u be one of its vertices. Removing u from G leaves $k \geq 1$ connected components $C_1 \dots C_k$, each with at least one vertex $v_i \in C_i$ adjacent to u in G. Moreover, each C_i has fewer than $n + 1$ vertices. We can, therefore, use our induction hypothesis and exhibit for each $1 \leq i \leq k$ a protocol p_i collecting all the units of information of C_i in v_i in at most $|C_i| - 1$ steps and such that v_i receives at least one unit of information at each step.Thus in G, we can use the protocol p_i on C_i for each i and send one unit of information from v_i to u at each time unit. The total time required is then clearly at most $1 + \sum_{1 \leq i \leq k} |C_i| - 1 \leq |V| - 1$. □

Using the same idea, we can also find a protocol collecting every unit in at most $n-1$ steps in such way that at every step, the subgraph of G induced by u and the vertices containing some information is connected.

4 A polynomial algorithm

We prove that the following question is solvable in polynomial time:

COLLECT

INPUT: A graph $G = (V, E)$, a vertex $u \in V$ and a time $T \in \mathbb{N}^{>0}$

OUTPUT: YES if $\text{col}_u(G) \leq T$

NO if $\text{col}_u(G) > T$.

We reduce COLLECT to MAX-FLOW [2]. We use the following formulation of problem MAX-FLOW(this is standard, see, for example, [2] or [1]).

MAX-FLOW

INPUT: A directed graph $G = (V, A)$, a capacity function $c : A \to \mathbb{R}$, two vertices s and t in V, and a real number $f \in \mathbb{R}$

OUTPUT: YES if there exists an st-flow with value at least f
NO otherwise.

Consider a COLLECT instance: A graph $G = (V, E)$, a vertex u and a time $T \in \mathbb{N}^*$. Let n denote the order of G ($n = |V|$) and write $[0, T] = \{0, 1, \ldots, T-1\}$.

We build an instance of MAX-FLOW. The directed graph $\tilde{G} = (\tilde{V}, \tilde{A})$ is built as follows: $\tilde{V}$ consists of s and t, together with all the triples in $(V \setminus \{u\}) \times \{\text{in}, \text{out}\} \times [0, T-1]$.

$\tilde{A}$ consists of five types of arcs, identifying multiple arcs between the same vertices (see Figure 2).

- Type 1: $\forall v \in V \setminus \{u\}, (s, (v, \text{in}, 0)) \in \tilde{A}$
- Type 2: $\forall v \in V \setminus \{u\}, \forall \tau \in [0, T-1], ((v, \text{in}, \tau), (v, \text{out}, \tau)) \in \tilde{A}$
- Type 3: $\forall v \in V \setminus \{u\}, \forall \tau \in [0, T-2], ((v, \text{in}, \tau), (v, \text{in}, \tau+1)) \in \tilde{A}$
- Type 4: $\forall v_1, v_2 \in V \setminus \{u\}$ s.t. $v_1 v_2 \in E, \forall \tau \in [0, T-2]$, $((v_1, \text{out}, \tau), (v_2, \text{in}, \tau+1)) \in \tilde{A}$
- Type 5: $\forall v \in V$ such that $uv \in E, \forall \tau \in [0, T-1], ((v, \text{out}, \tau), t) \in \tilde{A}$

The capacity function on the arcs of $\tilde{A}$ is defined as:

$$c:\ a \longmapsto \begin{cases} 1 & \text{if } a \text{ has type 1, 2, 4 or 5.} \\ n & \text{if } a \text{ has type 3.} \end{cases}$$

Claim 1. *$\tilde{G}$ with capacity function c has an st-flow with value at least $n-1$ if and only if $col_u(G) \leq T$.*

Proof. <u>From MAX-FLOW to COLLECT</u>. Suppose first that $\tilde{G}$ with capacity function c has an st-flow with value at least $n-1$. Since the arcs leaving s consists of $n-1$ arcs of capacity 1, the maximum st-flow value cannot exceed $n-1$. Therefore the maximum flow value is $n-1$. Moreover, since the capacities are integers, we know [3] that there exists a maximal flow

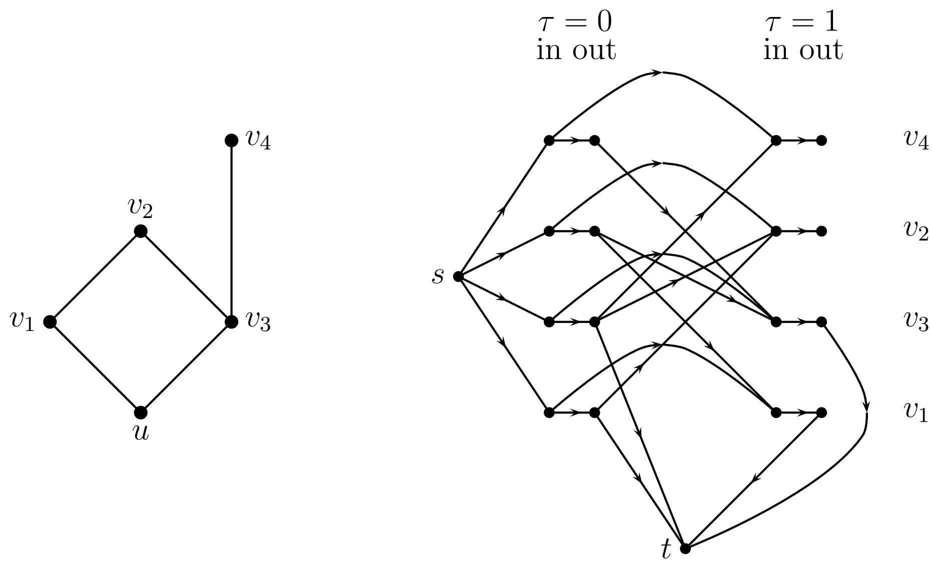

Figure 2: A graph G and most of its corresponding digraph $\tilde{G}$. Not shown: the arcs to t from (v_i, out, j), $i = 2, 4$, $j = 0, 1$.

$x : \tilde{A} \to \mathbb{N}$. From x, we will build a protocol on G so that u collects every unit of information in at most T steps.

If an arc $a = ((v_1, \text{out}, \tau), (v_2, \text{in}, \tau{+}1))$ (Type 4 arc) is such that $x(a) = 1$, then, in our protocol, vertex v_1 sends a unit of information to vertex v_2 at time $\tau + 1$. Similarly, if an arc $a = ((v_1, \text{out}, \tau), t)$ (Type 5 arc) is such that $x(a) = 1$, then, in our protocol, vertex v_1 sends a unit of information to vertex u at time $\tau + 1$. We have to prove that this protocol is valid.

Consider a vertex $v \in V \setminus \{u\}$ and a time τ. In $\tilde{G}$, (v, out, τ) has an in-capacity of 1 (the only arc arriving is $((v, \text{in}, \tau), (v, \text{out}, \tau))$ which has capacity 1). Since x is integral, there cannot be more than one arc leaving (v, out, τ) used by x. Therefore, in our protocol, no vertex sends more than one unit of information at a time. We still have to prove that when it is supposed to send a unit of information, a vertex actually has at least one unit of information. For this, we prove the following property:

For all $\tau \in [0, T-1]$ and $v \in V \setminus \{u\}$, the number of units of information at vertex v at time τ in our protocol is equal to the quantity of flow going through (v, in, τ).

Since x has a value of $n-1$ and s has only $n-1$ outgoing arcs each of them with capacity 1, all of them satisfy $x(a) = 1$. For every vertex $v \in V \setminus \{u\}$, there is only one arc ending in $(v, \text{in}, 0)$ which is $(s, (v, \text{in}, 0))$. Thus there is exactly one unit of flow going through $(v, \text{in}, 0)$. The property is then true for $\tau = 0$.

Assume that the property is true for some $\tau \in [0, T-2]$, we will prove it for $\tau + 1$. Let v be a vertex of G. At time τ, it has k units of information where k is the quantity of flow going through (v, in, τ). Vertex (v, in, τ) has exactly two outgoing arcs, one to (v, out, τ) with capacity 1 denoted by a and one to $(v, \text{in}, \tau + 1)$ with capacity n denoted by b.

- If $x(a) = 1$, there must be an arc leaving (v, out, τ) used by x with capacity 1 ; this means that in our protocol, v sends a unit of information between τ and $\tau + 1$. Furthermore, $x(b) = k - 1$ which represents the number of units of information staying at v between τ and $\tau + 1$. During that step, v receives also one unit of information, for each vertex v' such that $x((v', \text{out}, \tau), (v, \text{in}, \tau + 1)) = 1$. In the end, v contains as many units of information as the quantity of flow going through $(v, \text{in}, \tau + 1)$.

- If $x(a) = 0$, all the units of information stay at v between τ and $\tau + 1$ and $x(b) = k$. Similarly, this sums with the flow coming from others vertices of $\tilde{V}$ yielding the same conclusion.

The property is then true. Consider a vertex $v \in V \setminus \{u\}$ at time $T - 1$.

- If it is not a neighbour of u, then the only arc leaving $(v, \text{in}, T-1)$ goes to $(v, \text{out}, T - 1)$ which has no outgoing arc. Thus, there cannot be any flow going through $(v, \text{in}, T-1)$. By the property, v has no unit of information at time $T - 1$.

- If it is a neighbour of u, then $(v, \text{in}, T-1)$ has exactly one outgoing arc to $(v, \text{out}, T - 1)$ which capacity is 1. Therefore, the quantity of flow going through $(v, \text{in}, T - 1)$ cannot exceed 1. We conclude that in our protocol, v has at most one unit of information at time $T - 1$. It can send it to u at time T.

This proves that our protocol is correct and collects all the information in at most T steps. So (G, u, T) is a positive instance of Collect.

From Collect to Max-Flow. Suppose there exists a protocol collecting all the information in u in less than T steps. From it, we build a flow x of value $n-1$. Set $x(a)=1$ for every arc a with type 1 (arcs leaving s). For every arc a of type 2 (between (v, in, τ) and (v, out, τ)), set $x(a)=1$ if v sends a unit of information at time $\tau+1$, and $x(a)=0$ otherwise. For every arc a with type 3 (between (v, in, τ) and $(v, \mathrm{in}, \tau+1)$), set $x(a)$ equal to the number of units of information staying at v between τ and $\tau+1$. For every arc a of type 4 or 5, set $x(a)$ to 1 if the corresponding vertex sends a unit of information at the corresponding step; otherwise, set $x(a)=0$.

The law of conservation is naturally satisfied since the number of units of information staying in v together with the number of unit of information leaving v (can be 0 or 1), is equal to the number of units of information already at v at the previous step, together with the number of vertices that send a unit of information to v at the current step.

Finally, the value of x is $n-1$ since $\sum x(s,(v, \mathrm{in}, 0)) = n-1$. □

The graph $\tilde{G}$ has $\mathcal{O}(|V|T)$ vertices and $\mathcal{O}(|V|T+|E|T)$ arcs. Since, by Proposition 3, $\mathrm{col}_u(G) \leq |V|-1$, we may consider that $T \leq |V|$. Moreover, $|E| \leq |V|^2$. So, $\tilde{G}$ has $\mathcal{O}(|V|^2)$ vertices and $\mathcal{O}(|V|^3)$ arcs. Since the maximum flow is bounded above by $|V|$, Ford-Fulkerson's algorithms runs on $\tilde{G}$ in $\mathcal{O}(|V|^4)$ [3].

5 A faster algorithm for hypercubes

For particular classes of graphs, we can do far better than $\mathcal{O}(|V|^4)$. For example, an optimal gathering can be found in linear time in trees (any shortest-path routing will be an optimal solution).

We now consider hypercubes. We denote by $\mathcal{Q}_d$ the hypercube of dimension d, that is, the graph on $V=\{0,1\}^d$ with two vertices adjacent if they differ in exactly one coordinate. Clearly every vertex has exactly d neighbours. The proof relies on the recursive structure of Q_d

Fix a collecting vertex x_0 (hypercubes are vertex-transitive, so any vertex will do). We shall prove that the bound implied by Corollary 2 is tight. The proof hinges on the simple obseravtion that the bound of Corollary 2 is achieved if and only if the neighbours of the gathering vertex have something to send at every time step (except possibly some of them at the very last step). We will speak of *feeding* these vertices so that this can be achieved.

Theorem 1. *In $\mathcal{Q}_d$, there exists a collecting strategy for x_0 using $\left\lceil \frac{2^d-1}{d} \right\rceil$ steps.*

Proof. Induction hypothesis $\mathcal{P}(d)$: there is a protocol that collects all information in $\mathcal{Q}_d$ at a specified vertex in $\lceil \frac{2^d-1}{d} \rceil$ steps in such a way such that the neighbours of the collecting vertex are never empty except maybe at the last step.

$\mathcal{P}(1)$, $\mathcal{P}(2)$, $\mathcal{P}(3)$, $\mathcal{P}(4)$ and $\mathcal{P}(5)$ are easily found to be true

Let us show that for any $d \geq 6$, $\mathcal{P}(d-1) \Rightarrow \mathcal{P}(d)$.

We can consider $\mathcal{Q}_d$ as the union of two instances of $\mathcal{Q}_{d-1}$, namely H_0 and H_1. We may assume that x_0 is in H_0. Let us denote by x_1 the neighbour of x_0 in H_1; it will be the gathering vertex there. Each of the two vertices has $d-1$ neighbours in its respective subgraph H_i. We denote them by $x_1^0, \ldots, x_{d-1}^0$ and $x_1^1, \ldots, x_{d-1}^1$. We want to find a strategy which collects all the units of information in $\lceil \frac{2^d-1}{d} \rceil$ steps, such that $x_1^0, \ldots, x_{d-1}^0$ and x_1 are never empty except maybe for the last step.

By $\mathcal{P}(d-1)$ we can feed $x_1^i, \ldots, x_{d-1}^i$ for $i \in \{0, 1\}$ from step 1 to step $\lceil \frac{2^{d-1}-1}{d-1} \rceil - 1$. Vertices $x_1^i, \ldots, x_k^i$ $(1 \leq k \leq d-1)$ are fed until step $\lceil \frac{2^{d-1}-1}{d-1} \rceil$. Therefore we can consider that we play on the subgraph induced by x_0, x_1 and the x_j^i's, with 1 unit of information at x_1, $\lceil \frac{2^{d-1}-1}{d-1} \rceil$ units of information at x_j^i for $1 \leq i \leq k$ and $\lceil \frac{2^{d-1}-1}{d-1} \rceil - 1$ units of information at x_j^i for $k < j \leq d-1$.

The protocol to collect all these units of information can be described as follows. Vertices x_j^0 will keep sending one unit of information to x_0 at each step. The total amount of information going through such a vertex must be between $\lceil \frac{2^d-1}{d} \rceil$ and $\lceil \frac{2^d-1}{d} \rceil - 1$. Vertex x_1 will also send one unit of information to x_0 at each step. We will ensure that exactly $\lceil \frac{2^d-1}{d} \rceil$ units of information go through this vertex.

In a first phase, vertices x_j^1 will send units of information to x_1 so that x_1 gets the needed $\lceil \frac{2^d-1}{d} \rceil$ units of information. It might be that the last step of this phase requires only some of these vertices to send a unit to x_1. In such a case, we keep the balance by considering prioritarily x_j^1 with $j \leq k$. In a second phase, each $x_{j_1}^1$ gets matched with a $x_{j_2}^0$ and will send all its information to this vertex. Once again, in order to conserve the balance, we match the exceeding vertices to the lacking ones prioritarily. This matching can be chosen as we want because of the symmetry of the hypercube. There is no need to compute the exact amount going through the x_j^0 ; since it is balanced, the amount is necessarily between $\lceil \frac{2^d-1}{d} \rceil$ and $\lceil \frac{2^d-1}{d} \rceil - 1$.

In the end, we just have to prove that the first phase will not need too much time which would leave some x_j^0's empty during the protocol. In other words we have to prove that the time of the first phase $\left\lfloor \frac{\left\lceil \frac{2^d-1}{d} \right\rceil - 1}{d-1} \right\rfloor$ is less than the smallest number of unit of information on some x_j^0, $\lceil \frac{2^{d-1}-1}{d-1} \rceil - 1$.

Let $t = \left\lfloor \frac{\left\lceil \frac{2^d-1}{d} \right\rceil - 1}{d-1} \right\rfloor$.

Since $d \geq 6$,

$$
\begin{aligned}
\left(\frac{d}{2} - 1\right) \times 2^d - 2d^2 - 3d + 1 &> 0 \\
d \times [2^{d-1} - 2(d-1)] - d &> 2^d - 1 \\
2^{d-1} - 2(d-1) &> \left\lceil \frac{2^d - 1}{d} \right\rceil \\
2^{d-1} - 1 - (d-1) &> \left\lceil \frac{2^d - 1}{d} \right\rceil - 1 + (d-1) \\
\frac{2^{d-1} - 1}{d-1} - 1 &> \frac{\left\lceil \frac{2^d-1}{d} \right\rceil - 1}{d-1} + 1 \\
\left\lceil \frac{2^{d-1} - 1}{d-1} \right\rceil - 1 &> \left\lfloor \frac{\left\lceil \frac{2^d-1}{d} \right\rceil - 1}{d-1} \right\rfloor \\
\left\lceil \frac{2^{d-1} - 1}{d-1} \right\rceil - 1 &> t.
\end{aligned}
$$

Therefore, property $\mathcal{P}(d)$ holds.

In the end we have proved that $\mathcal{P}(d)$ holds for any d. □

6 Conclusion

In spite of believing for a while the problem Collect to be $NP-$complete in general, we have shown that it is polynomial-time solvable and that for the hypercube Q_d there is a simple protocol that achieves the lower bound on $col(Q_d)$. There remain some unanswered questions.

1. Can we characterize the graphs for which the lower bound on $col_u(G)$ can be achieved?

2. Is it true that for vertex-transitive graphs (where clearly $col_u(G) = col_v(G) = col_m(G) = col_M(G)$ for any vertices u and v) the lower bound can always be achieved? In particular, is this true for Cayley graphs?

3. Are there classes of graphs for which a lower-degree polynomial time algorithms is possible?

4. What can we say if the number of units of information a vertex can receive at one time is bounded? What is the number of units of information a vertex cans store is bounded?

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James Orlin: *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 846 pages, 1993.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Section 26.2: The Ford-Fulkerson method. *Introduction to Algorithms (Second ed.).* MIT Press and McGraw-Hill, pp. 651–664, 2001. ISBN 0-262-03293-7.

[3] L.R. Ford and D.R. Fulkerson: Maximal flow through a network. *Canadian Journal of Mathematics* **8**:399–404, 1956.

[4] `http://www.freehaven.net/`

[5] B. Hoppe and É. Tardos: The quickest transshipment problem. *Math. Oper. Res.* **25**(1):36–62, 2000.

[6] F. T. Leighton, B. M. Maggs, and S. B. Rao: Packet routing and job-shop scheduling in O(congestion+dilation) steps. *Combinatorica* **14**:167–186, 1994.

[7] Michael O. Rabin: Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, **36**(2):335–348, 1989.

[8] A. Srinivasan and C.-P. Teo: A Constant-Factor Approximation Algorithm for Packet Routing and Balancing Local vs. Global Criteria. *SIAM J. Comput.* **30**(6):2051–2068, 2000.