

Maximum Finding in the Symmetric Radio Networks with Collision Detection

František Galčík* and Gabriel Semanišin**

Institute of Computer Science, P.J. Šafárik University, Faculty of Science
Jesenná 5, 041 54 Košice, Slovak Republic
frantisek.galcik@upjs.sk, gabriel.semanisin@upjs.sk

Abstract. We consider a problem of computing the maximal value associated to the nodes of a network in the model of unknown symmetric radio network with availability of collision detection. We assume that the nodes have no initial knowledge about the network topology, number of nodes and even they have no identifiers. The network contains one distinguished node, called initiator, that starts the process of computing. We design a series of algorithms that result into an asymptotically optimal deterministic algorithm completing the task in $\Theta(ecc + \log Max)$ rounds, where ecc is the eccentricity of the initiator and Max is the maximal value among the integer values associated to the nodes. Some other utilisations of the developed algorithm are presented as well.

1 Introduction

A *radio network* is a collection of autonomous stations that are referred as *nodes*. The nodes communicate via sending messages. Each node is able to receive and transmit messages, but it can transmit messages only to nodes, which are located within its transmission range. The network can be modeled by a directed graph called *reachability graph* $G = (V, E)$. The vertex set of G consists of the nodes of the network and two vertices $u, v \in V$ are connected by an edge $e = (u, v)$ if and only if the transmission of the node u can reach the node v . In such a case the node u is called a *neighbour* of the node v . If the transmission power of all nodes is the same, then the reachability graph is symmetric, i.e. a symmetric radio network can be modeled by an undirected graph.

Nodes of the network work in synchronised steps (time slots) called *rounds*. In every round, a node can act either as a *receiver* or as a *transmitter*. A node u acting as transmitter sends a message, which can be potentially received by every node v such that u is a neighbour of v . In the given round, a node, acting as a receiver, receives a message only if it has exactly one transmitting neighbour. The received message is the same as the message transmitted by the transmitting neighbour. If in the given round, a node u has at least two transmitting

* Research of the author is supported in part by Slovak VEGA grant number 1/3129/06 and UPJŠ VVGS grant number 38/2006.

** Research supported in part by Slovak APVT grant number 20-004104 and Slovak VEGA grant number 1/3129/06.

neighbours we say that a *collision* occurs at node u . In the case, when the nodes can distinguish collision (interference noise) from silence (background noise), we say that they have an *availability of collision detection*. It is also assumed that a node can determine its behavior in the following round within the actual round.

According to different features of the stations forming a radio network, several models of radio networks have been developed and studied. They differ in the used communications scenarios and initial knowledge assumed for nodes. The overview of the models of radio networks can be found e.g. in [7].

There has been studied many communication primitives as broadcasting, gossiping, leader election, synchronization, etc. Usually it is supposed that each node knows at least its identifier denoted by ID such that $ID \in O(n)$, where n is the number of nodes in the network. The effectiveness of the designed algorithms is mostly measured by the time required to complete the prescribed goal, i.e. by the number of required rounds. Note that throughout this paper unique identifiers of nodes are not necessary.

The goal of *broadcasting* is to distribute a message from one distinguished node, called a *source*, to all other nodes. Remote nodes of the network are informed via intermediate nodes. Similarly, the goal of *acknowledged broadcasting* is to realise broadcasting and to inform the source about finishing broadcasting.

In this paper we deal with another problem related to a communication in a radio network, namely with the problem of computing the maximal value over the values associated to nodes of the network. In our setting we have given a distinguished node called *initiator* (throughout the paper we shall denote it by s) and we assume that each node of the network possess a positive integer value. In one round the initiator starts the algorithm that computes the maximum. The remaining nodes do not know the starting round.

Our problem is motivated by the following real-world situation: Consider a multihop radio network with a distinguished central node. Every node is able to perform a measurement of a physical quantity. Sometimes, in order to perform a specific operation, the central node must find out the maximal (or minimal) value in the network. One can collect all values into the central node (e.g. by performing a gossiping algorithm), but up to now no efficient suitable algorithm is known. We provide an algorithm, that works in pipelined manner and, due to appropriate arrangement of transmissions, reduces the time necessary for completing the task.

1.1 Related Work

Similar problem of finding maximum over real values associated to nodes of a multiple access broadcast network was studied in [5]. The randomized algorithm, designed in that paper, was used for solving the selection problem. The problem of finding maximum over the integer values associated to nodes of a radio network was treated in [2], too.

Our algorithm utilises the ideas and principles of the algorithm *ENCODED-BROADCAST* that was developed in [1] and its pipelined version, called *RBEM*,

that was presented in [6]. Algorithm *RBEM* is used several times as fast subroutine in order to broadcast computed information.

1.2 Model and Terminology

In this paper we consider radio networks with symmetric reachability graphs which are equipped with an availability of collision detection. No initial knowledge of nodes is assumed, i.e. the nodes have no information about the topology of the network, number of nodes in the network and even the nodes do not need identifiers. We also suppose that a reachability graph of a network is connected.

A node v , acting as a receiver in a given round, hears a μ -*signal*, if at least one from its neighbours acts as a transmitter and sends a message in this round. Otherwise we say that node v hears the λ -*signal* (i.e. no its neighbour transmits).

We shall encode information into the μ and λ -signals using transmission of *contact messages*. The contact message is a message that can be distinguished from all other messages transmitted during the work of algorithms.

The *distance* of two nodes u, v (denoted by $dist(u, v)$) is the length of a shortest $u - v$ -path in the underlying reachability graph. The *eccentricity* of a node v is defined as follows: $ecc(v) = \max\{dist(v, u) : u \in V(G) \setminus \{v\}\}$. We briefly denote the eccentricity of the initiator $ecc(s)$ by ecc . It is not difficult to see that all nodes of a reachability graph G can be partitioned into *layers* according to their distances from initiator s . Hence, we can define the sets

$$L_i = \{v \in V(G) : dist(s, v) = i\}, \quad i = 0, 1, \dots, ecc.$$

Let v be a neighbour of w , $v \in L_{i+1}$ and $w \in L_i$ for some i . Then we say that v is a (+1)-neighbour of w and w is a (-1)-neighbour of v .

2 Preprocessing Algorithms

In this section we present algorithms that compute and broadcast some parameters of the network. These parameters will be used by the main algorithm during its work. In Section 2.1 we design an algorithm *EFC*, working in $O(ecc)$ rounds, that is computing the eccentricity ecc of the initiator. After finishing this step, the value of ecc will be known only for initiator. Next, in Section 2.2, we show that in $O(ecc)$ rounds we can also broadcast the computed eccentricity of the initiator to all nodes of the network. Finally, in Section 2.3, we briefly present an algorithm called *DDC* which for each node computes its distance from the initiator. This information will be known only for the given node. This algorithm will work in $O(ecc)$ rounds too.

2.1 Computing Eccentricity of the Initiator

In this section we design an algorithm called *EFC - Eccentricity Fast Counting*. Let we start with a rough description of the algorithm.

We remind that a communication in the network is arranged into synchronised rounds. According to our algorithm, in one round each node can be either active

or inactive. Initially all nodes, except the initiator, are inactive. During the work of the algorithm, every active node knows its distance from the initiator modulo 3. The work of the algorithm is split into phases. Only initiator is informed about the number of current phase. But every active node has an information about the number of current round within the actual phase (it has its own counter of rounds that is initiated in an appropriate moment). Each phase consists of two parts. First part has 4 rounds and the second one takes 6 rounds. In the first part, the active nodes, that have been activated in the previous phase, attempt to activate their (+1)-neighbours. An inactive node becomes active whenever it receives the μ -signal in two consecutive rounds. In the second part active nodes detect whether they have some active (+1)-neighbour. If an active node does not detect any active (+1)-neighbour, it changes its state and becomes inactive. In order to avoid simultaneous transmissions in the consecutive layers, the transmissions in the second part are scheduled in a such a way, that only the nodes in the distance 3 transmit simultaneously.

Now we are going to describe the phases of the algorithm *EFC* more precisely. The algorithm is initiated by the initiator s . At that moment only the initiator is active and we consider it to be a node activated in the previous phase. The initiator starts with the tasks prescribed for the first round of first phase of *EFC*.

Part 1 (4 rounds). In this part, only the active nodes, which were activated in the previous phase, transmit. In the first two rounds they transmit contact messages. If an inactive node receives the μ -signal in two consecutive rounds, it becomes active and sets its counter of the current round within the phase to 2. As it is shown below, an inactive node can become active if and only if it has an active (-1)-neighbour which transmits the contact messages during the first two rounds of a phase. Let $0 \leq i \leq ecc$ and $v \in L_i$ be an active node transmitting in this part of current phase. Since v is active, it knows its distance $d = i \bmod 3$ from the initiator. If $d = 2$ then the node v transmits the contact message in the round 3 of actual phase. If $d = 1$ then v transmits the contact message in the round 4. During the rounds 3 and 4 a node, that has been activated in the current phase, acts as receiver in order to learn its distance from the initiator. If the μ -signal is received in the round 3 or 4, then it knows that its distance from the initiator is 0 or 2 respectively. Otherwise, it knows that its distance is 1. (The values are considered with respect to modulo 3.).

Part 2 (6 rounds). The second part of the phase is divided into 3 couples of rounds. Since Part 1 consists of 4 rounds we number the round of this part by $5, 6, \dots, 10$.

If an active node v belongs to a layer L_i for some i then it transmits the contact message in the round with number $5 + (i \bmod 3) * 2$. It means that within Part 2 transmissions occur only in rounds with numbers 5, 7 and 9 of current phase. If an active node v belonging to L_i receives μ -signal in the round $5 + ((i + 1) \bmod 3) * 2$ then it remains active in the next phase. If a node was activated during Part 1 of current phase then it will be active in the beginning of the next phase too. All the others nodes will be inactivated.

Now we are going to show that if C stands for the number of the first phase in which the initiator is inactive then $C = 2.ecc + 2$ (the first phase of the work of the algorithm is numbered as 1 and we recall that the initiator knows the numbers of phases). In order to simplify the forthcoming consideration, we introduce two new concepts.

Definition 1. A path (v_0, v_1, \dots, v_k) is called an active path whenever v_0 is the initiator, $v_j \in L_j$ and v_j is active for all j , $0 \leq j \leq k$.

Definition 2. An active path (v_0, v_1, \dots, v_k) is called an active path in the phase i whenever it is active in the beginning of the first round of phase i .

The following lemma provides an information about the structure of active paths in a network during the work of *EFC*.

Lemma 1. Let d be a length of a longest active path in the phase i . Then for every positive integer i , $1 \leq i \leq ecc + 1$, and for every node $v \in L_{i-1}$ there is an active path in phase i of the length d ending in the node v . Moreover, in the first round of phase i , each active node belongs to an active path.

It is not very difficult to see that in the Part 1 of phase i , $1 \leq i \leq ecc$, there are activated exactly the nodes of layer L_i . The following lemma describes the active paths for the phases with number at least $ecc + 1$.

Lemma 2. Every longest active path in the phase $ecc + i$, where $1 \leq i \leq ecc + 1$, has the length $ecc - i + 1$. Moreover, no node from the layer L_j , $j > ecc - i + 1$, is active in the first round of the phase $ecc + i$.

An application of Lemma 2 for $i = ecc + 1$ yields that the initiator has no active (+1)-neighbour in the phase $2.ecc + 1$. Since the initiator is active also in all the previous phases, the initiator is inactive for the first time at the beginning of the phase $2.ecc + 2$. Using these facts we can formulate the following result.

Theorem 1. The algorithm *EFC* computes eccentricity ecc of the initiator in $O(ecc)$ rounds.

2.2 Acknowledged Broadcasting of the Eccentricity of the Initiator

We remind that after finishing the algorithm *EFC* only the initiator knows its eccentricity. Now we need to distribute this information to the remaining nodes. In order to broadcast the computed eccentricity of the initiator, we can use the algorithm *RBEM* designed by Okuwa et al. in [6]. This algorithm broadcasts a message of binary length r in $O(r + ecc)$ rounds. In our case, the eccentricity of the initiator can be binary encoded to a message with length $\log ecc$. Then the algorithm *RBEM* completes broadcasting of this message in $O(ecc)$ rounds. In general, the algorithm *RBEM* is not acknowledged, but the initiator knows the value of the parameter ecc , and therefore it has an implicit information when this task is completed. The algorithm *EFC* equipped with the previously described broadcasting ability we shall refer as *ExEFC* - extended *EFC*.

2.3 Distributed Computing of the Distance from the Initiator

The main goal of this stage is to compute the distance from the initiator for each node of the network. After the completion of this stage, every node v knows its exact distance from the initiator that uniquely determines the layer L_i containing v . The basic idea of the suggested algorithm is that the nodes belonging to L_i transmit concurrently binary encoded number $i + 1$ (using μ and λ -signals) to their (+1)-neighbours, i.e. to nodes belonging to L_{i+1} . In order to decrease the time complexity of this task, we realise it in the pipelined fashion. We use the fact, that if we know the k lowest bits of the number i (i.e. the suffix of the binary code of i) then we also know the k lowest bits of the number $i + 1$. In order to realise the goal, we modify the algorithm *RBEM*. Particularly, we shall dynamically change the broadcasted message. In the following, we refer this modified algorithm as *DDC-Distributed Distance Counting*. Moreover *DDC* has one useful property. If the eccentricity ecc of the initiator is known for all nodes of the network, it allows us to use this algorithm for a “synchronisation of the nodes”. It means that the nodes can make an agreement about the round when they would start simultaneously some task.

Theorem 2. *The algorithm DDC computes the distance from the initiator to each node of the network in $O(ecc)$ rounds. Moreover, by an application of the algorithms *ExEFC* and *DDC* we can “synchronize” the network in $O(ecc)$ rounds.*

3 Algorithm for Computing the Maximal Value

In this section we design an algorithm *CMV* that computes the maximum of the considered values over all nodes in the network. The algorithm consists of three logical parts. In the first step the initiator estimates the maximal value by determining the minimum number of bits necessary for its binary encoding. In the second step the initiator broadcasts the estimation to the other nodes and initiates the computation of the exact value, that forms the third logical part of the algorithm. At the end of the computation, the initiator knows the desired value. The first step is described in Section 3.1, the second and third step are discussed in Section 3.2.

3.1 Estimating the Maximal Value

As we have already mentioned, we suppose that every node of the network possess a positive integer value. In what follows we show how to compute the estimation of the maximal value among them. More precisely, for the unknown value Max , the searched maximum, we want to compute the value B_{\max} such that $2^{B_{\max}-1} \leq Max < 2^{B_{\max}}$. Obviously, the value B_{\max} specifies how many bits we need to store an arbitrary value associated to a node of the network. We assume that algorithms *ExEFC* and *DDC* have been already performed and every node knows the eccentricity ecc of the initiator, it knows its distance from the initiator and the nodes are synchronized (they know the starting round of the algorithm computing the estimation).

Our algorithm, called *EMV*, works as follows. Every node performs in the loop 3 segments: receive, transmission and sleep. Every segment consists of only one round. (We use concept of segments only in order to use uniform terminology in description of algorithms.) For any node v , belonging to the layer L_i , $0 \leq i \leq ecc$, let us denote by V_v the value associated to v and let B_v be the positive integer satisfying $2^{B_v-1} \leq V_v < 2^{B_v}$.

In the first round of the algorithm, the nodes perform an activity that depends on their layer. The nodes belonging to the layers L_i , where $i = ecc - 3k$ for some integer $k \geq 0$, realise transmission segment. The nodes from the layers L_i for $i = ecc - 3k - 1$, $k \geq 0$, realise activities prescribed for receive segment and the remaining nodes realise sleep segment. Note that during the work of algorithm the nodes realising the same segment are in the layers with mutual distance at least 3.

The node $v \in L_i$ transmits the contact messages according to two rules:

1. The contact message is transmitted in all rounds r , where $r = ecc - i + 1, ecc - i + 4, \dots, ecc - i + 3(B_v - 1) + 1$. (Note that these rounds are the rounds of the transmission segment.)
2. The contact message is transmitted in every transmission segment following the receive segment during which a node received μ -signal.

Let R be the round of the first receive segment with number at least ecc in which the μ -signal is not received by the initiator. It is possible to prove that R is well defined and moreover $(R - ecc)/3$ is the maximum over all values B_v except of the value B_s . It results into the following theorem.

Theorem 3. *Let Max be the maximal value over the values associated to the nodes. Then the algorithm *EMV* computes the value B_{max} such that $2^{B_{max}-1} \leq Max < 2^{B_{max}}$ in $O(ecc + \log Max)$ rounds.*

3.2 Computing the Maximal Value

After finishing *EMV* the value B_{max} is known only for the initiator s . Before performing the computation of Max we have to distribute its estimation B_{max} to the remaining nodes of the network. We can again utilise the algorithm *RBEM* which takes $O(ecc + \log Max)$ rounds. After finishing *RBEM*, all nodes of the network are informed how many bits are needed to store Max . Therefore the nodes can unify the representation of their values as binary sequences of the length $\lceil \log Max \rceil + 1 = B_{max}$. In *CMV* each node v computes the value E_v which is its estimation of Max . According to *CMV* the nodes, that recognize that they cannot improve the calculated value Max , eliminate themselves from the process of computation. Similarly as *EMV*, the algorithm *CMV* is working in the loop and repeatedly perform 3 segments: receive, transmission and sleep. The difference is that every segment consists of two rounds. During the algorithm the nodes can be in one of two states: active or inactive.

Now we are going to describe *CMV* in more details. In the beginning all nodes are active and the value E_v of every node $v \notin L_{ecc}$ is set to 0. We shall work with

binary representation of E_v and therefore E_v can be viewed as a finite sequence of fixed length B_{\max} consisting of 0's and 1's. During the work of the algorithm we improve the value of E_v by the modification of the particular bits. It means that in the beginning $E_v = (0, 0, \dots, 0)$. Moreover, during initialization phase, if $v \in L_{ecc}$ then E_v is set to V_v .

In the first round of the algorithm, the nodes perform an activity that depends on their layer. The nodes belonging to the layers L_i , where $i = ecc - 3k$ for some integer $k \geq 0$, realise transmission segment. The nodes from the layers L_i for $i = ecc - 3k - 1$, $k \geq 0$, realise activities prescribed for the receive segment and the remaining nodes realise sleep segment.

The inactive nodes cannot transmit. The transmission of an active node $v \in L_i$, $1 \leq i \leq ecc$, is prescribed by the following rules:

1. v transmits the contact message in all rounds r , where $r = 2(ecc - i - 2 + 3j) - 1$ for some j satisfying $j \in \{1, \dots, B_{\max}\}$, whenever j -th highest bit of binary encoding E_v is 1 (i.e. the bit corresponding to $2^{B_{\max}-j}$)
2. if the μ -signal is received by the node v in the round r , where $r = 2(ecc - i - 2 + 3j)$ for some j such that $j \in \{1, \dots, B_{\max}\}$, and the node v has not been transmitted in the round $r - 1$, then the node v becomes inactive
3. if in the round r , where $r = 2(ecc - i - 3 + 3j) - 1$ for some j satisfying $j \in \{1, \dots, B_{\max}\}$, the node v receives μ -signal or the j -th highest bit of binary encoding of V_v is 1 and $V_v \geq E_v$, then the node sends the contact message in the following round $r + 1$ and sets the j -th highest bit of the binary encoded value E_v to 1.

Note that the first and the second rule are related to the rounds that belong to a transmission segment of the node v . The third rule concerns rounds of the receive segment of the node v .

The rules can be also interpreted as follows. In the first round of the transmission segment, an active node v transmits according to j -th highest bit of the value E_v , where j is determined by the rules for the given round. Furthermore, all active nodes belonging to the same layer work with the j -th highest bit. Simultaneously a (-1)-neighbour receives μ -signal during the first round of its receive segment if and only if it has at least one active (+1)-neighbour which has j -th highest bit equal to 1. In the following round these (-1)-neighbours announce to their (+1)-neighbours how they set their j -th highest bits. After this round, every node knows whether its activity in the previous round has influenced some its (-1)-neighbours, i.e. whether its active (-1)-neighbours set their j -th highest bits according to its j -th highest bit. If a node detects that no (-1)-neighbour set its E_v according to its information, it becomes inactive. That is why no value E_v , that can be potentially computed by this node, cannot be larger than the value, which would be computed in one of its active (-1)-neighbours.

The following proposition provides relatively straightforward properties of binary sequences utilised for encoding integer numbers. It says that by a comparison of the highest k bits of two numbers with binary code of the same length we can obtain an important information about the size of these numbers.

Proposition 1. *Let l, A, B be positive integers and A, B have binary representations $A = (a_1 a_2 \dots a_l)_2$ and $B = (b_1 b_2 \dots b_l)_2$ respectively. If $A \leq B$ then for any k , $1 \leq k \leq l$ the following inequality $(a_1 a_2 \dots a_k)_2 \leq (b_1 b_2 \dots b_k)_2$ holds.*

The next lemma shows that the computed value E_s matches our expectations.

Lemma 3. *After the round $r = 2(\text{ecc} - 3 + 3B_{\max})$, the value E_s of the initiator s is equal to the maximal value Max over the values associated to the nodes.*

Proof. Note that we can encode the values of the nodes by binary sequences, each of them of the same length, because we have already applied the algorithm *EMV* that computes the value B_{\max} .

Using previous proposition one can easily check that the following two invariants are true during the work of the algorithm *CMV*.

1. At the end of any round $r = 2(\text{ecc} - i - 3 + 3j)$, where j , $1 \leq j \leq B_{\max}$, for each node $v \in L_i$ holds the following: $E_v \geq V_v$ and the highest j bits of the binary encoded value E_v remain the same during the rest of the algorithm and they are equal to the highest j bits of the value E_w , where w is an arbitrary active (+1)-neighbour of the node v .
2. If an inactive node $v \in L_i$ has an active (-1)-neighbour, then there is an active (-1)-neighbour $w \in L_{i-1}$ of v satisfying $E_w > E_v$.

Since the initiator has no (-1)-neighbours, it is active in every round. After the round $r = 2(\text{ecc} - 3 + 3B_{\max})$, any active path consists only of nodes with estimations of Max equal to E_s , because from the first invariant we have that all bits of E_v remain unchanged.

Consider now a node w such that $V_w = Max$. It is easy to see that in every round, every prefix of the binary encoded value E_w is equal to the prefix of the binary encoded value V_w . From the rules of the algorithm it follows, that in every round there is an active path ending in the node w . Indeed, this is true because the prefix of E_w is successively distributed and computed bit by bit in all nodes belonging to an active path ending in w . Therefore all nodes belonging to the considered active path remain active in every next round (there is no greater prefix during the work of the algorithm). Thus, in the considered round $r = 2(\text{ecc} - 3 + 3B_{\max})$, the value E_s of the initiator s is equal to the value $E_w = V_w = Max$. \square

As a consequence of the previous results we immediately have our main result.

Theorem 4. *Algorithm *CMV* computes the value Max (in the initiator) in $O(\text{ecc} + \log Max)$ rounds.*

4 Lower Bound

In this section we show that the algorithm *CMV* is asymptotically optimal. In particular, we reduce the problem of broadcasting in symmetric geometric radio

networks (shortly *GRN*) with collision detection to our problem of maximum finding. Model of *GRN* differs from our model of radio networks in two properties: nodes have an additional initial information about their positions and a reachability graph of a *GRN* should satisfy restrictions resulting from geometric locations of nodes. Note that broadcasting algorithms for *GRN* utilise unique identifiers for every node of the network (the existence of such identifiers follows for example from their geometric locations).

Theorem 5. *For any maximum finding algorithm with collision detection there exists a symmetric radio network of diameter 2 and such an assignment of values associated to nodes that the algorithm requires $\Omega(\log Max)$ rounds.*

Proof. Dessmark and Pelc showed in [3] that for every broadcasting algorithm with collision detection there exists a class of symmetric geometric radio network with diameter 2 for which this algorithm requires $\Omega(\log n)$ rounds, where n is the number of nodes. More precisely, for given n this network has the following form: nodes of the network are labeled $1, \dots, n$, node 1 is the *source* and node n is the *sink*. The set $\{2, \dots, n-1\}$ is partitioned into sets X and Y , where $|Y| = 2$. Nodes $1, \dots, n-1$ form a complete graph. Nodes from Y are connected to the sink n . We shall refer such a network as a *network of class \mathcal{H}* .

In what follows we show how to utilise a maximum finding algorithm in a broadcasting algorithm for networks of class \mathcal{H} . Now, let \mathcal{A} be an algorithm for the maximum finding in symmetric radio networks with collision detection and G be a n -node network of the class \mathcal{H} . An associated broadcasting algorithm (to the algorithm \mathcal{A}) for the network G works as follows: In the first round the source (node 1) transmits a source message. Nodes in the distance 1 from the source become informed. Next, we perform algorithms *ExECC* and *DDC* with the node 1 as the initiator. After the $O(1)$ -rounds, we can distinguish the sink n (a node in the distance 2 from the initiator 1). In order to distinguish the nodes of the set Y , the sink n transmits the contact message in the following round. All nodes, except the nodes of Y , set their values to 1. Two nodes of the set Y set their associated values to their identifiers. Performing algorithm \mathcal{A} for maximum finding problem, we compute a label of one node from the set Y in the initiator 1 (a label of the node in Y with larger label). After that the initiator transmits a message containing the computed label of one node from Y . In the following round, only the node with this label transmits the source message and the sink n become informed.

Obviously, the asymptotical time complexity of the associated broadcasting algorithm is the same as the complexity of algorithm \mathcal{A} . Since $Max \in O(n)$, the previously mentioned result from [3] implies that time complexity of \mathcal{A} is $\Omega(\log Max)$. \square

Combining the previous result and the trivial lower bound $\Omega(ecc)$ we obtain that algorithm *CMV* is asymptotically optimal from the view of parameters *ecc* and *Max*.

5 Conclusion

We have designed the algorithm *CMV* that computes the maximal value over the values associated to nodes of the network. The designed algorithm is asymptotically optimal and works in time $\Theta(ecc + \log Max)$ where *ecc* is the eccentricity of the initiator and *Max* is the searched maximum. Besides of the studied problem of finding maximum over the values of a measured physical quantity the algorithm can be successfully utilised in some other situations. For example, we can use it to compute some parameters of the network, e.g. maximal identifier among the identifiers of nodes (potentially with a specified property - see Theorem 5), to compute the logical sum and logical product of the one-bit values associated to the nodes, etc. Moreover, the algorithm can serve for designing of broadcasting algorithm of time complexity $O(D \cdot \log n / D \cdot \log^3 n)$ for unknown symmetric radio networks with collision detection whose underlying reachability graph is planar (see [4]).

References

1. Chlebus, B.S., Gasieniec, L., Gibbons, A., Pelc, A., and Rytter, W.: Deterministic Broadcasting in Unknown Radio Networks. In Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'00) (2000) 861–870
2. Chrobak, M., Gasieniec, L., and Rytter, W.: Fast Broadcasting and Gossiping in Radio Networks. In Proceedings of 41st Annual Symposium on Foundations of Computer Science (FOCS'2000) 575–581
3. Dessmark, A. and Pelc, A.: Broadcasting in Geometric Radio Networks. Journal of Discrete Algorithms (to appear)
4. Galčík, F.: Broadcasting in Radio Networks with Specific Topology. (2006) (manuscript)
5. Martel, C.U.: Maximum Finding on a Multiple Access Broadcast Network. Information Processing Letters **52** (1994) 7–13
6. Okuwa, T., Chen, W., and Wada, K.: An Optimal Algorithm of Acknowledged Broadcasting in Ad Hoc Radio Networks. Second International Symposium on Parallel and Distributed Computing (2003) 178–184
7. Pelc, A.: Broadcasting in Radio Networks. Handbook of Wireless Networks and Mobile Computing, I. Stojmenovic(ed.) John Wiley and Sons, Inc., New York (2002) 509–528