Maximum finding in the symmetric radio networks with collision detection

František Galčík
(joint work with Gabriel Semanišin)

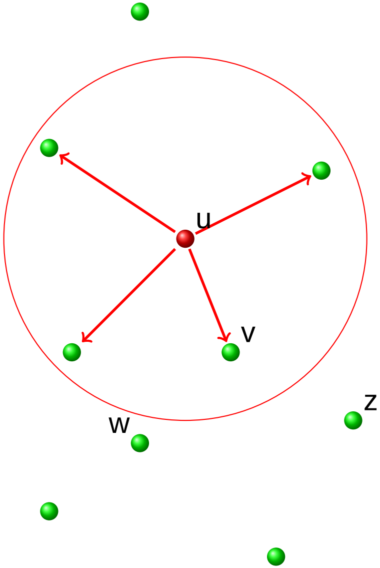
Institute of Computer Science
P.J. Šafárik University, Faculty of Science
Košice, Slovakia

SOFSEM 2007

What is radio network ?

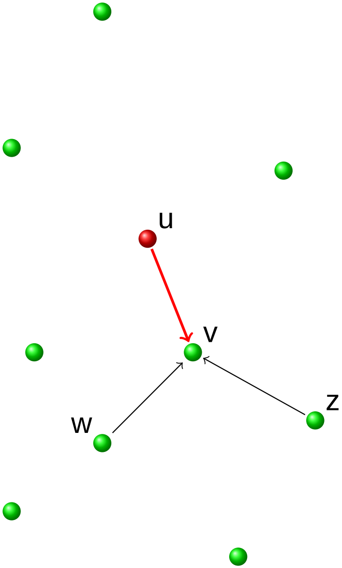
- a collection of receiver-transmitter devices - nodes
- nodes are autonomous
- communication via sending messages
- **single** shared communication frequency
- nodes work in globally synchronised time slots - rounds
- in each round, each node makes a decision:
  act either as a receiver or as a transmitter
- multihop network - modelled by a reachability graph

Communication in radio networks



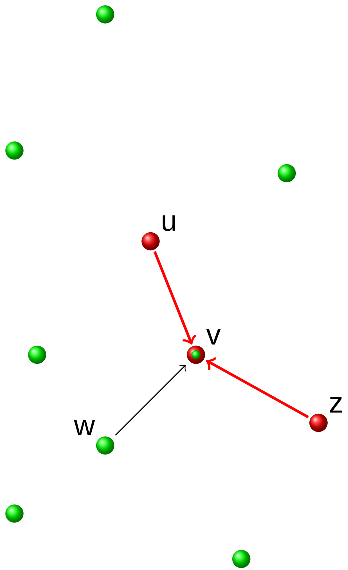
- if a node *u* transmits, then the signal from *u* goes to all nodes within its transmission range

Communication in radio networks



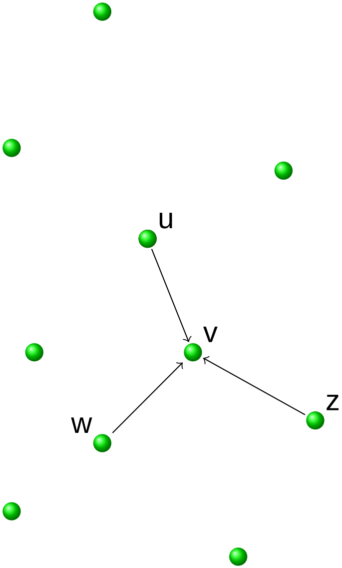
- if a node u transmits, then the signal from u goes to all nodes within its transmission range
- if the node v listen: then it receives message from u if and only if u is the only transmitting node which has v in its transmission range

Communication in radio networks



- if a node *u* transmits, then the signal from *u* goes to all nodes within its transmission range
- if the node *v* listen: and is in the range of more than one transmitting node, then collision occurs and no message is received - the node *v* hears **interference noise**

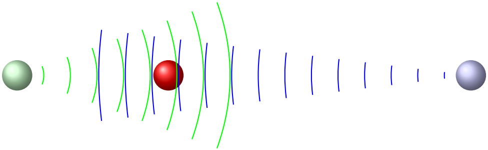
Communication in radio networks



- if the node v listen:
  and v is in the range of
  no transmitting node,
  then v hears
  background noise

Key features of radio communication

- interference of simultaneous transmissions (slowdown)
  - occurs if a node is in the range of more than one transmitting node



- omnidirectional transmission (speedup)
  - a node transmits to all nodes located within its transmission range

Reachability graph, measure of effectiveness

Reachability graph of a radio network is a directed graph $G = (V, E)$, where

- vertex set V corresponds to the nodes of the network
- an edge $e = (u, v) \in E \iff$ the node v is in the transmission range of the node u

Measure of effectiveness

- mostly the **time** required **to complete** the prescribed communication task
- other studied measures: energy consumption, combined measures, etc.
- measured according to parameters of reachability graph (diameter D, eccentricity *ecc* of the source, number of nodes n, etc.)

Communication tasks

- broadcasting - the goal is to deliver a message from a distinguished node (source) to all nodes of the network
- gossiping - all nodes have a message, the goal is to distribute each message to all nodes of the network
- maximum finding - each node possesses a (integer) value, the goal is to compute the maximal possessed value in a distinguished node (initiator)
- ...

Different communication scenarios

- directed or undirected (symmetric) network
- known network, unknown network or partially known network
- no node labels (anonymous nodes)
  nodes with labels $(1, \ldots, O(n))$
- randomized or deterministic protocol
- bounded or unbounded messages
- collision detection or no collision detection

Collision detection

- a **listening node** is able to distinguish **interference noise** (more than one transmitting neighbour) and **background noise** (no transmitting neighbour)

Effectiveness is significantly influenced by considered model.

Our setting

Model of radio network:

- undirected (symmetric) network (e.g. transmission power of all nodes is the same)
- unknown network
- no node labels (anonymous nodes)
- deterministic protocol, bounded messages
- collision detection capability

Problem: **Integer maximum finding**

- each node possesses **an integer value**
- a distinguished node (initiator) has the goal to **compute** the **maximal** possessed value
- the initiator starts in the time (round) **unknown** to other nodes

Our setting - example of motivation

- sensor devices measuring a physical quantity and communicating via low-power radio
- one central node (operator) needs to known the maximal measured value (e.g. maximal radiation)
- limited precision of real measurement - integer values

Main result

Theorem

There is an asymptotically optimal algorithm computing the value Max (in the initiator) in $\Theta(ecc + \log Max)$ rounds, where

- *Max is the maximal integer value over possessed values*
- *ecc is the eccentricity of the initiator, i.e. maximal distance from the initiator to any other node of the network.*

Used techniques

- encoding information into collisions
- pipelining (wave technique)
- utilised properties of binary encoding of integers

Subroutine algorithm RBEM

- pipelined broadcast algorithm using encoding information into collisions
- broadcasts message M of binary length m in $O(ecc + m)$, where ecc is the eccentricity of the source node
- presented by Okuwa, Chen, Wada (2003)

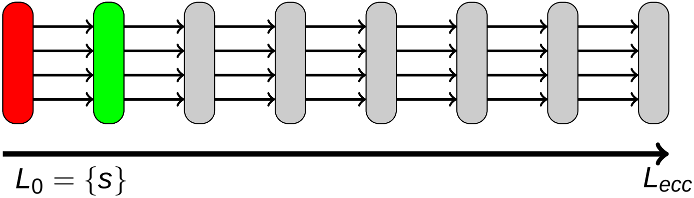
Encoding information into collisions (1)

- a distinguished message - contact message
- **encoding** of 1 bit information:
  - ”0” . . . no message is transmitted
  - ”1” . . . contact message is transmitted
- **decoding** of the received information (listening node):
  - no message received
    $\Longrightarrow$ all neighbours (if exist) said ”0”
  - contact message received or collision detected
    $\Longrightarrow$ at least one neighbour said ”1”

Encoding information into collisions (2) - example

- Layer - a set of nodes in the same distance from a distinguished node (e.g. initiator, source)
- **simple broadcast algorithm**: message transformed to
  - opening sequence (e.g. 11)
  - message content not including the closing sequence (e.g. 1 $\rightarrow$ 01, 0 $\rightarrow$ 00)
  - closing sequence (e.g. 11)

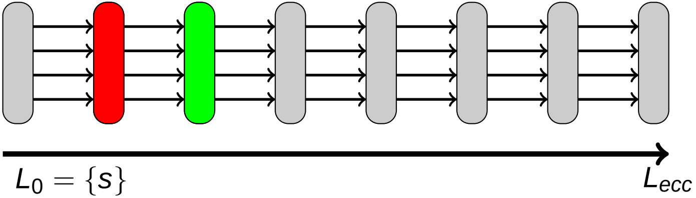
  and distributed layer by layer
- only two neighbouring layers work in the same time



Encoding information into collisions (2) - example

- Layer - a set of nodes in the same distance from a distinguished node (e.g. initiator, source)
- **simple broadcast algorithm**: message transformed to
  - opening sequence (e.g. 11)
  - message content not including the closing sequence (e.g. 1 $\rightarrow$ 01, 0 $\rightarrow$ 00)
  - closing sequence (e.g. 11)

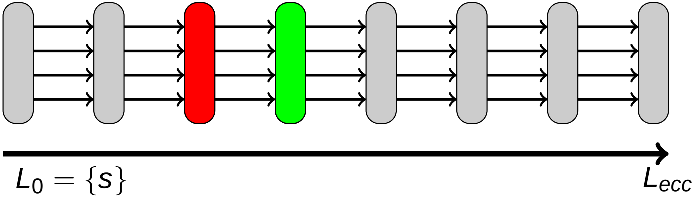
  and distributed layer by layer
- only two neighbouring layers work in the same time



Encoding information into collisions (2) - example

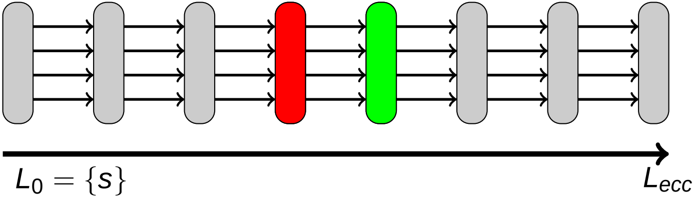
- Layer - a set of nodes in the same distance from a distinguished node (e.g. initiator, source)
- **simple broadcast algorithm**: message transformed to
  - opening sequence (e.g. 11)
  - message content not including the closing sequence (e.g. 1 $\rightarrow$ 01, 0 $\rightarrow$ 00)
  - closing sequence (e.g. 11)

  and distributed layer by layer
- only two neighbouring layers work in the same time



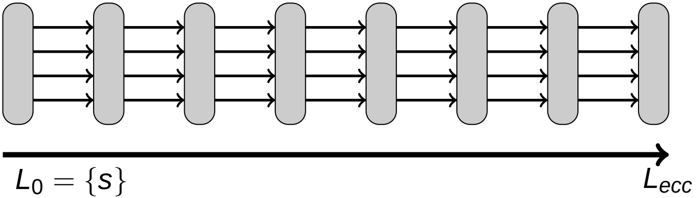
- Layer - a set of nodes in the same distance from a distinguished node (e.g. initiator, source)
- **simple broadcast algorithm**: message transformed to
  - opening sequence (e.g. 11)
  - message content not including the closing sequence (e.g. $1 \rightarrow 01$, $0 \rightarrow 00$)
  - closing sequence (e.g. 11)

  and distributed layer by layer
- only two neighbouring layers work in the same time



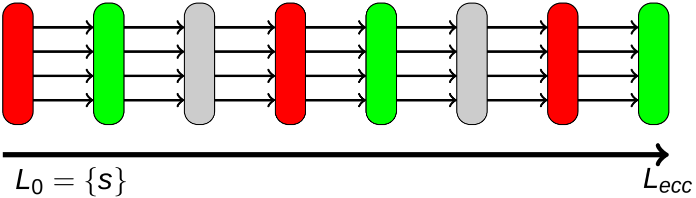
Pipelining (wave technique)

- solution for **"simultaneous" communication** in all layers **without "interference"**
- pipelined broadcasting: immediately after recognizing next bit of message, it is sent to the next layer
- layers **divided into 3 groups** (distance modulo 3), cyclic changing of action
  - transmit segment
  - receive segment
  - sleep segment - separation function



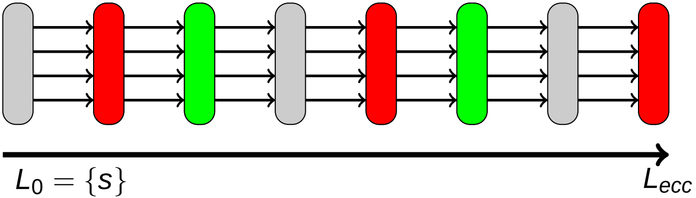
Pipelining (wave technique)

- solution for **"simultaneous" communication** in all layers **without "interference"**
- pipelined broadcasting: immediately after recognizing next bit of message, it is sent to the next layer
- layers **divided into 3 groups** (distance modulo 3), cyclic changing of action
  - transmit segment
  - receive segment
  - sleep segment - separation function



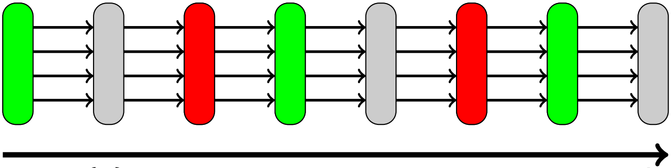
Pipelining (wave technique)

- solution for **"simultaneous" communication** in all layers **without "interference"**
- pipelined broadcasting: immediately after recognizing next bit of message, it is sent to the next layer
- layers **divided into 3 groups** (distance modulo 3), cyclic changing of action
  - transmit segment
  - receive segment
  - sleep segment - separation function



Pipelining (wave technique)

- solution for **"simultaneous" communication** in all layers **without "interference"**
- pipelined broadcasting: immediately after recognizing next bit of message, it is sent to the next layer
- layers **divided into 3 groups** (distance modulo 3), cyclic changing of action
  - transmit segment
  - receive segment
  - sleep segment - separation function



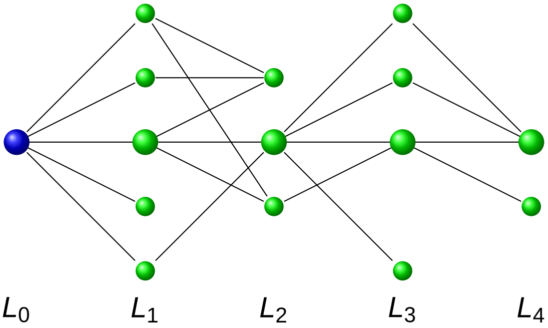
$L_0 = \{s\}$ L_{ecc}

Rough description of algorithm

- Preprocessing
  1. in the initiator, compute eccentricity of the initiator
     $O(ecc)$ rounds
  2. broadcast computed eccentricity *ecc*
     $O(ecc)$ round using *RBEM* algorithm
  3. in each node computes its distance from the initiator
     $O(ecc)$ rounds
     side effect: **synchronisation of nodes**
- Computation of the maximal value
  1. in the initiator, compute estimation B_{max} of maximal value Max such that $2^{B_{max}-1} \leq Max < 2^{B_{max}}$
     $O(ecc + \log Max)$ rounds
  2. broadcast computed estimation B_{max}
     $O(ecc + \log Max)$ rounds using *RBEM* algorithm
  3. in the initiator, compute the value *Max*
     $O(ecc + \log Max)$ rounds

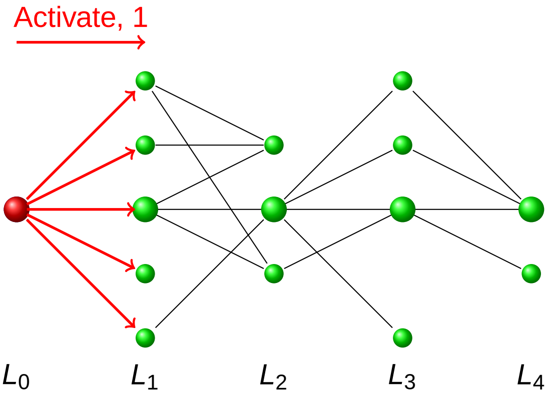
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

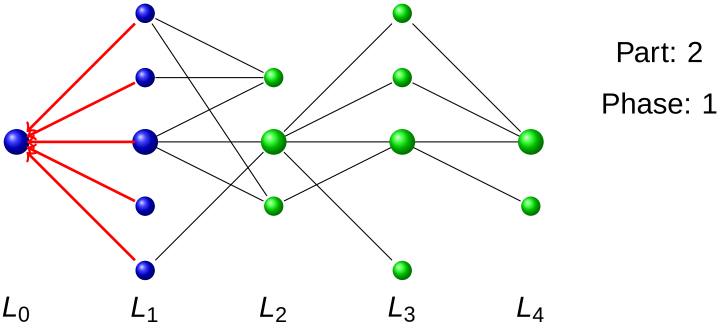


Part: 1

Phase: 1

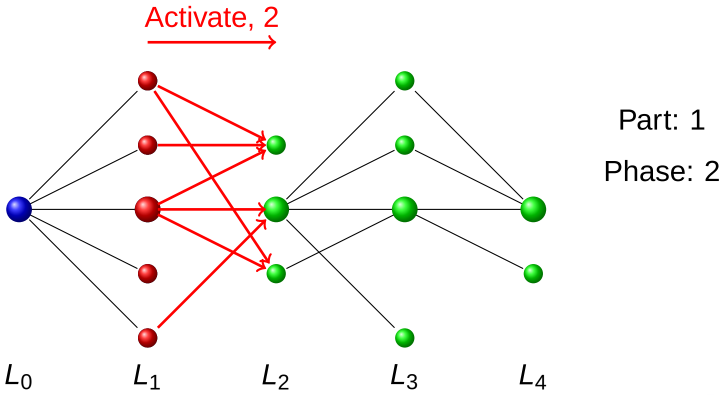
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



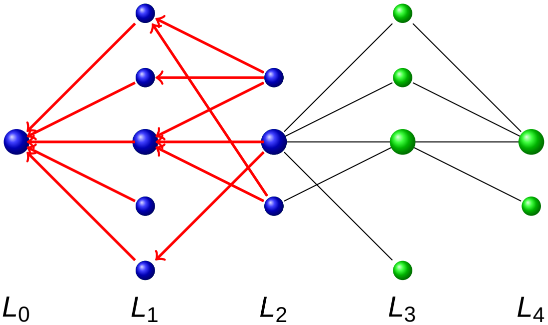
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

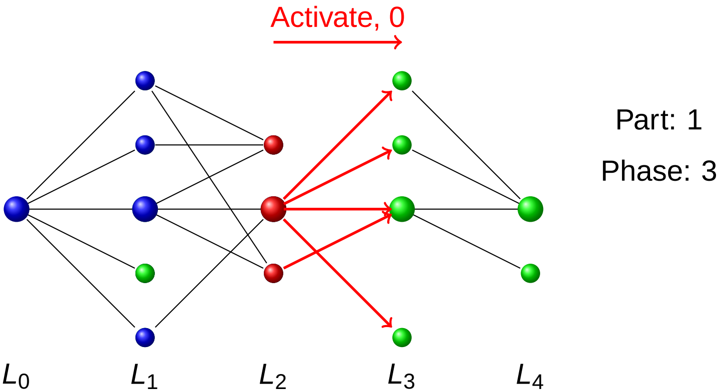


Part: 2

Phase: 2

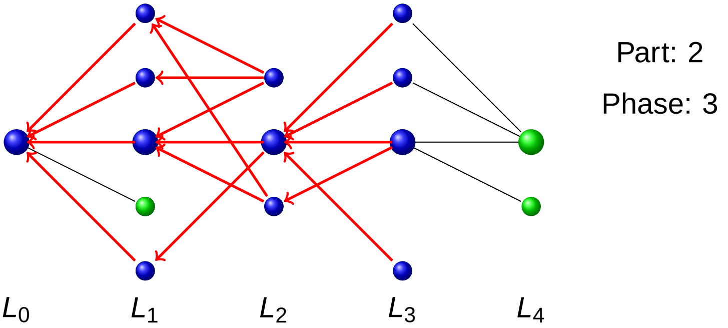
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



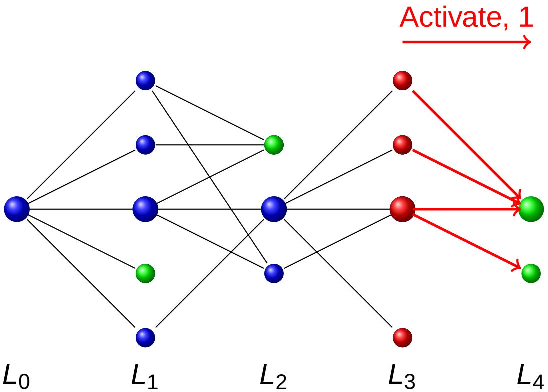
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

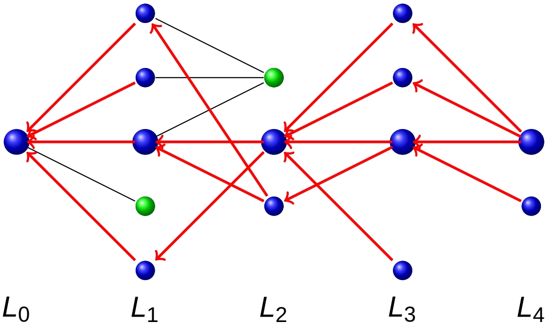


Part: 1

Phase: 4

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

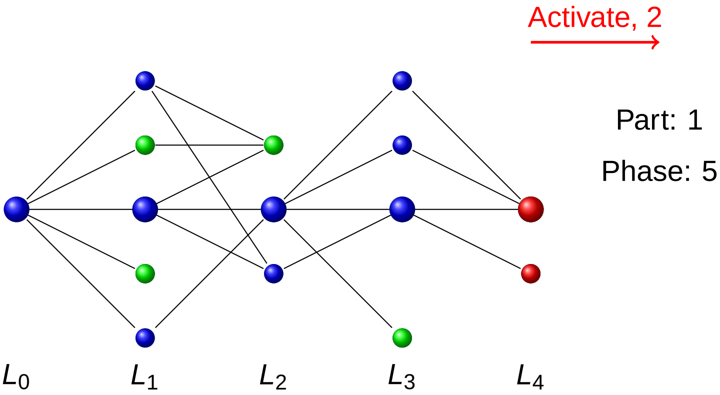


Part: 2

Phase: 4

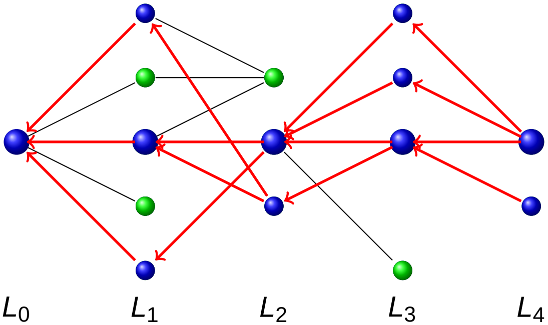
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

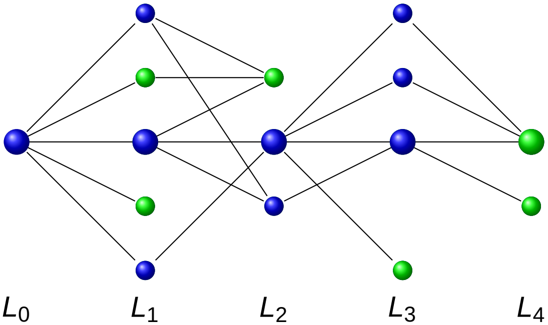


Part: 2

Phase: 5

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

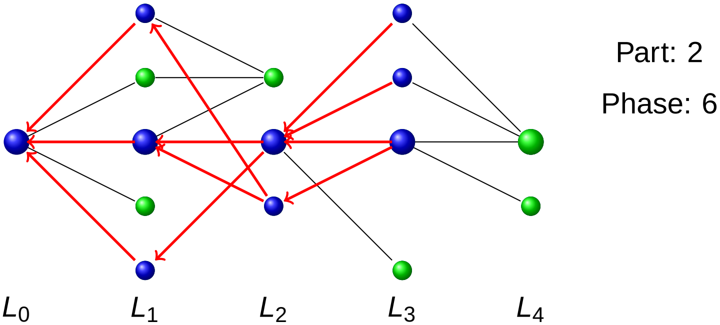


Part: 1

Phase: 6

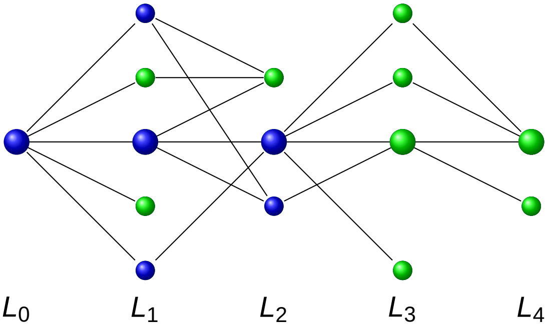
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

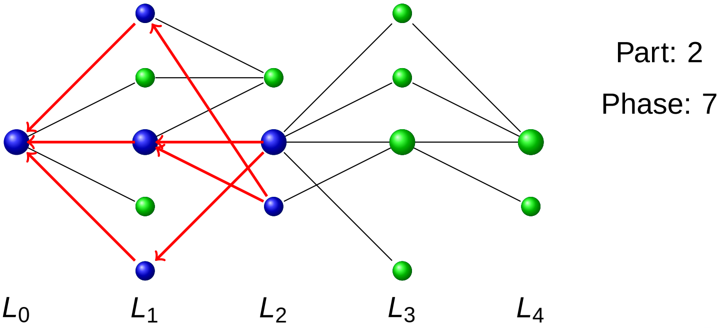


Part: 1

Phase: 7

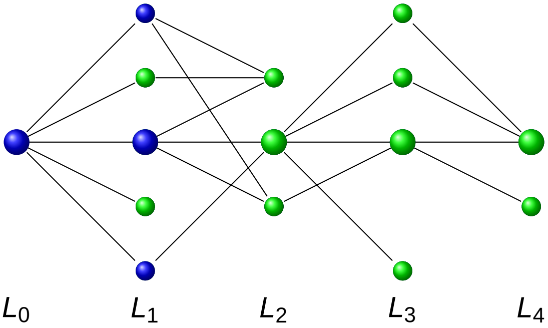
Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

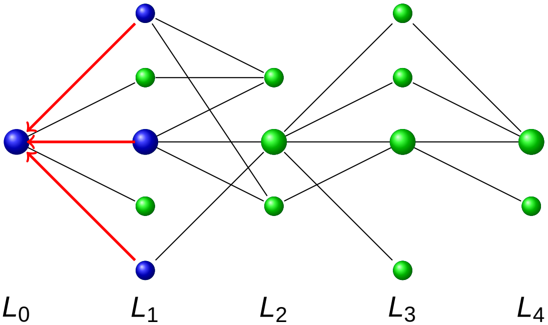


Part: 1

Phase: 8

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

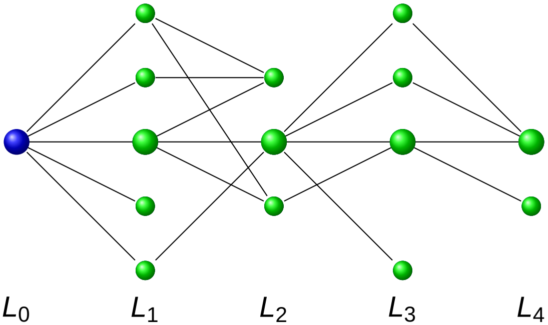


Part: 2

Phase: 8

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

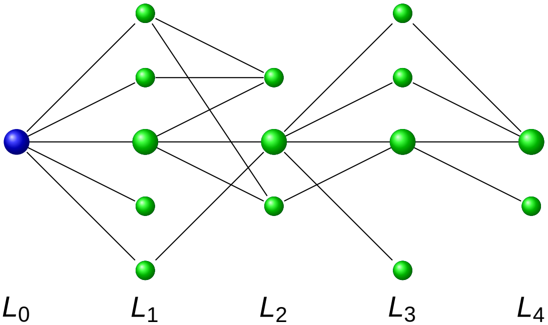


Part: 1

Phase: 9

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)

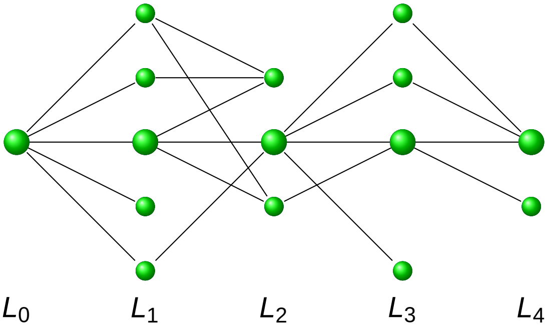


Part: 2

Phase: 9

Eccentricity computing (1)

- works in 2-part phases:
- 1^{st} part activates nodes in the next layer and provides information about modulo 3 distance (4 rounds)
- 2^{nd} part informs active nodes about existence of their active $+1$-neighbours (6 rounds)



Part: 1

Phase: 10

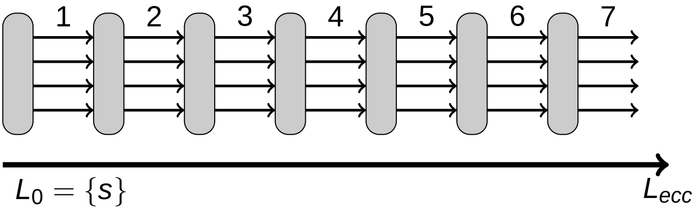
$10 = 2.4 + 2$

Eccentricity computing (2) - technical details

- first part uses special **opening sequence** and encodes **modulo 3 distance** by appropriate transmissions
- opening sequence **sets up counter** of the actual round of the current phase
- second part is scheduled using known modulo 3 distances of active nodes and **avoiding** transmission of **opening sequence** of the first part
- $C = 2.ecc + 2$, where C is the number of phases in which the initiator is active

Distance computing

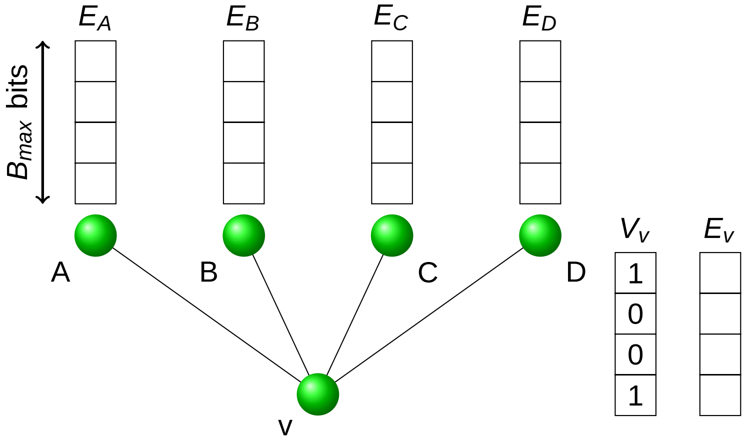
- simple modification of *RBEM* (pipelined broadcast) algorithm
- nodes of the layer L_i **dynamically change** broadcasted message to the binary encoded number $i+1$
- based on the fact, that knowing k lowest bits of a number j we know k lowest bits of the number $j+1$



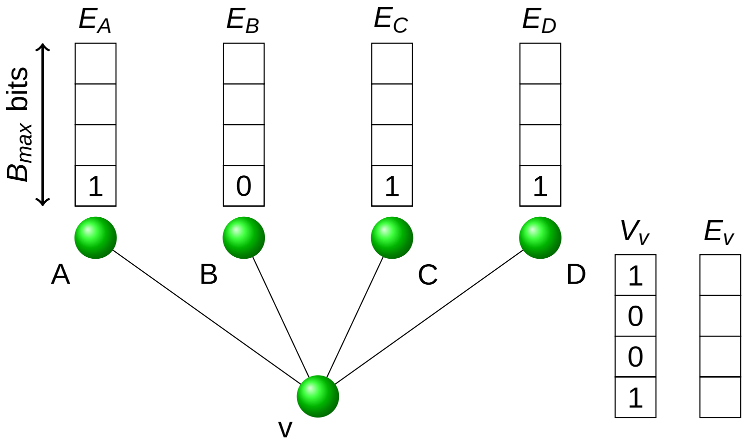
Maximum computing (1)

- suppose, that the value B_{max} is known by all nodes
  - $2^{B_{max}-1} \leq Max < 2^{B_{max}}$
  - algorithm for computing B_{max} is based on simplified idea of the maximum computing algorithm
  - B_{max} broadcasted by *RBEM*
- all values V_v (possessed integer values) can be considered as **binary sequences** of the **same length** B_{max}
- each node computes E_v - its **estimation** of the maximal value
- node is **active** only if its estimation can be **"useful" in the following computation** - otherwise it is deactivated

Maximum computing (2) - example

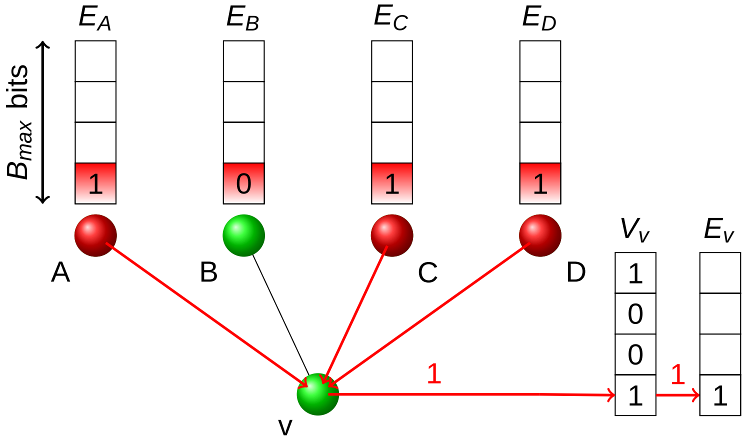


Maximum computing (2) - example



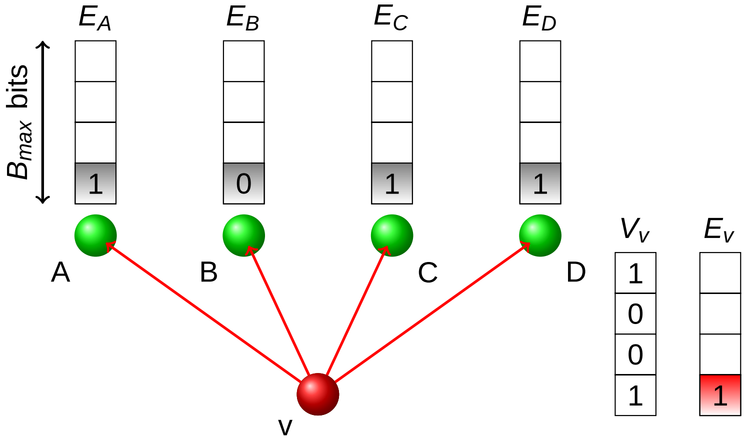
$+1$-neighbours of v have updated their estimations

Maximum computing (2) - example



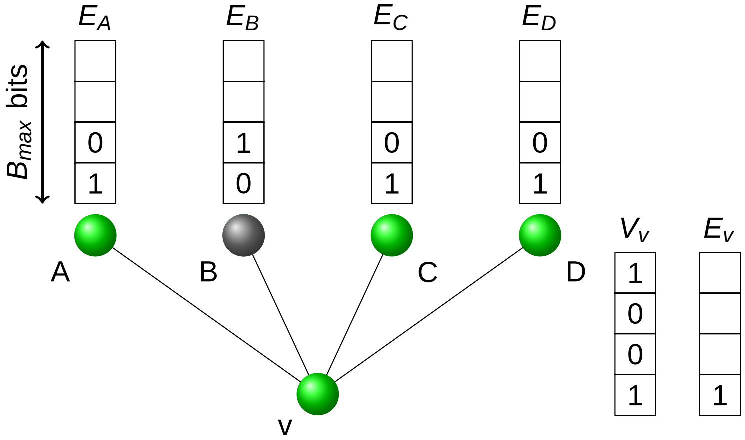
$+$1-neighbours inform v about active bits of their estimations

Maximum computing (2) - example



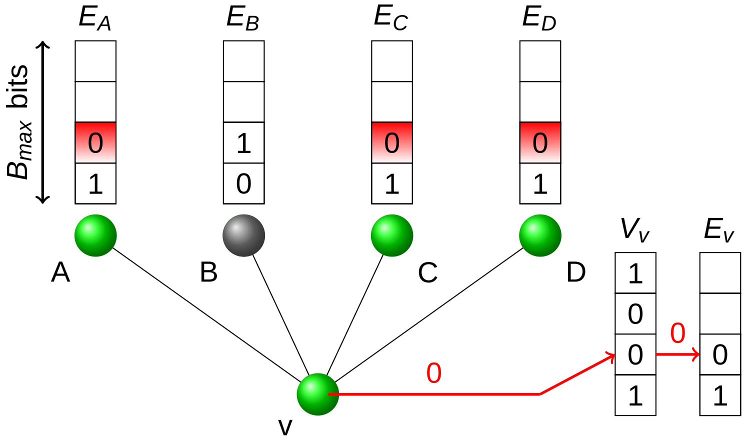
v informs $+1$-neighbours about its setting of active bit

Maximum computing (2) - example



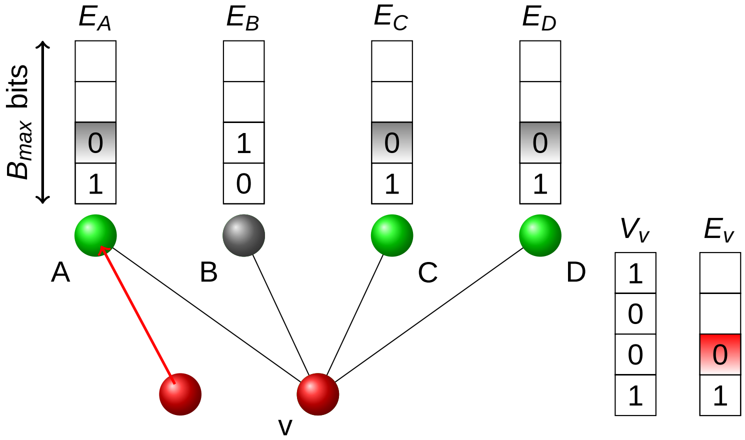
B is deactivated - it cannot help

Maximum computing (2) - example



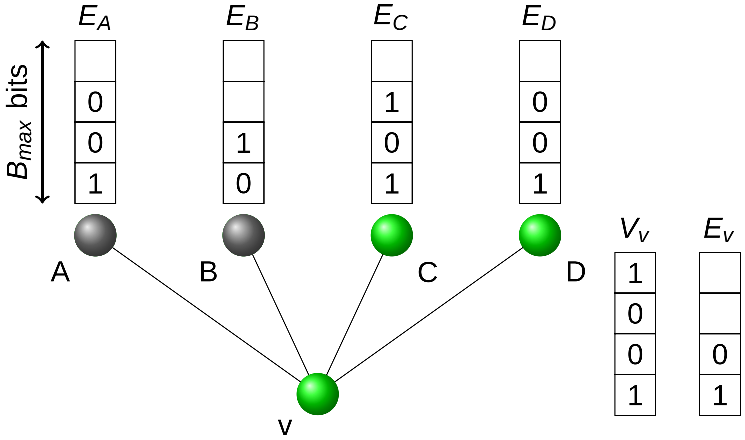
$+1$-neighbours inform v about active bits of their estimations

Maximum computing (2) - example



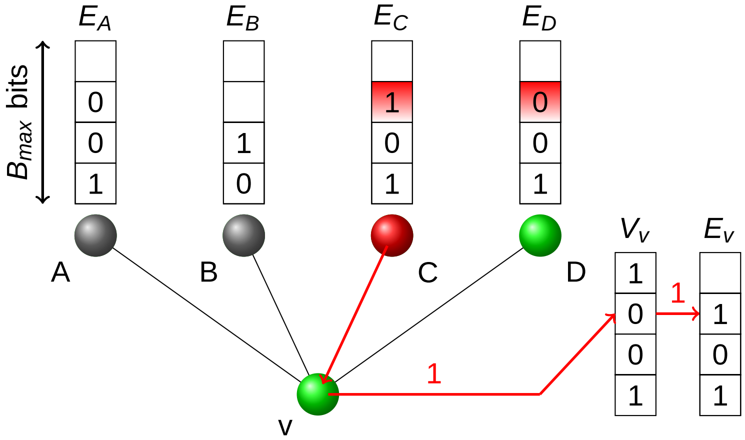
A can have other -1-neighbours ...

Maximum computing (2) - example



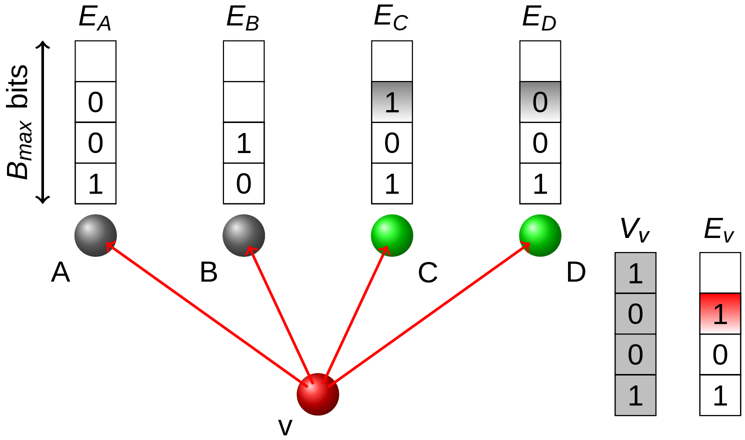
A is deactivated - stopped by a -1 neighbour

Maximum computing (2) - example



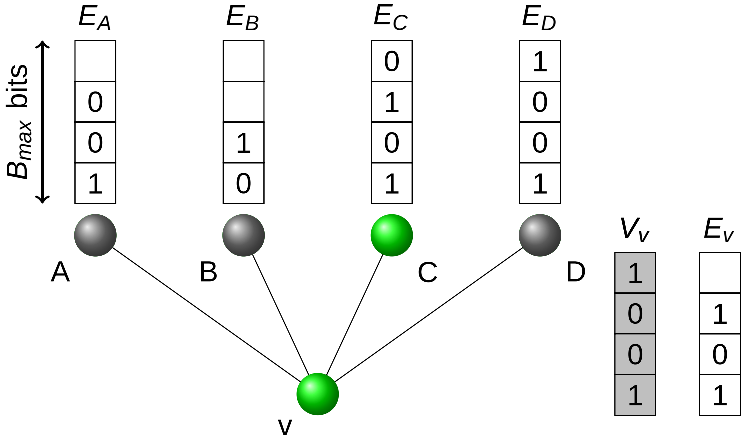
$+1$-neighbours inform v about active bits of their estimations

Maximum computing (2) - example



value V_v is too small and will be never considered

Maximum computing (2) - example



D is deactivated . . .

Maximum computing (3)

- transmissions are scheduled according to the (known) distances from the initiator
- E_v is always "greater" than V_v and "less" than all reachable estimations in higher layers
- "maximal value" always reaches initiator
- if E_s is fully computed, it is equal to the searched maximum

Total time of computation: $O(ecc + \log Max)$ rounds

Lower bound

Theorem

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 based on the result from Dessmark and Pelc: $\Omega(\log n)$ lower bound of broadcasting in symmetric geometric radio networks (GRN) with collision detection

Therefore designed algorithm is asymptotically optimal.

Conclusion

- we design $\Theta(ecc + \log Max)$ maximum finding algorithm for anonymous symmetric radio networks with collision detection
- **possible utilisations**:
  - maximum finding algorithm in the case of ”real” (binary encoded in the mantissa/exponent form) possessed values
  - computing unknown parameters of the network
  - ”synchronisation” of nodes
  - selection of particular specific node, e.g. node with maximal label
  - utilisation for broadcasting and gossiping (already utilised for broadcasting in planar graphs)
- future research:
  - networks without collision detection

Thank you for attention