# CHILDREN CHAIN

Edited by:
Calogero Lo Bue

## Overview

- Imagine a teacher that has to bring a number of children to a given destination (e.g. school trip). She/He has to look at those children continuously to avoid that someone could get lost.

- This task could be not easy, expecially if they move in a crowded area

## The Idea

- Since the teacher's task is not so easy, it could be useful to provide the children with a kind of monitoring system that warns the teacher if one or more children leave the group.

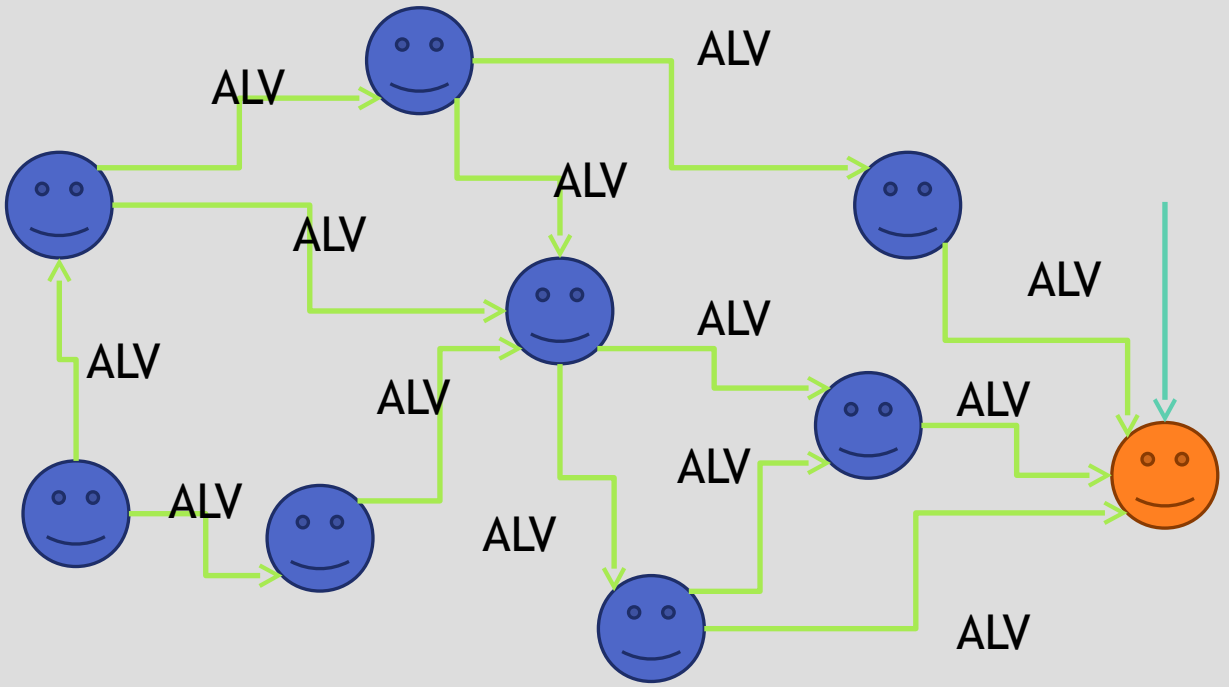
## The Solution

- Every child should be equipped with a sensor to make possible the monitoring system, such that if a child is leaving the teacher or the other children, it sends an alarm.

## Implementation

- Every child that carries a sensor, becomes a node
- Teacher becomes the sink node
- A network is established between children and teacher
- Periodically, nodes send a message to the sink to confirm their presence (ALiVe message)
- If, while checking, the sink discovers that at least one message has not been received from a node, it warns the teacher
- Teacher understands that a child has disappeared

# Example




Generating
ALVs

## The Sender

- In order to advise the sink of its presence, a sender must send an ALV message periodically to the sink.
- Two cases:
  - In a **single-hop** network, just sends its own ALV message to the sink, via unicast
  - In a **multi-hop** network, a sender:
    - Broadcasts its own ALV
    - Re-broadcasts ALVs coming from other nodes (stored in a buffer)

## The Receiver

- Receive ALV messages from the other nodes
- Associates a timer to each node
- Waits for an event to happen. Two main events:
  - MESSAGE_RECEIVED
  - TIMER_EXPIRED

## Receiver – MESSAGE_RECEIVED event

- Stands for “A message has been received”
- When this event happens, the receiver:
  - Looks for the sender id contained in the message
  - Resets the timer associated to that sender

## Receiver – TIMER_EXPIRED

- Stands for “A child has disappeared!”
- When this event happens, means that the receiver has no longer received ALV messages from the node associated with that timer
- The receiver sends an alarm in order to warn the teacher
- The associated timer will be resetted when the sink will receive an ALV message from the disappeared node (that is, the node is back again)

## Question

**Multi-hop** Vs **Single-hop**:
which is the most suitable?

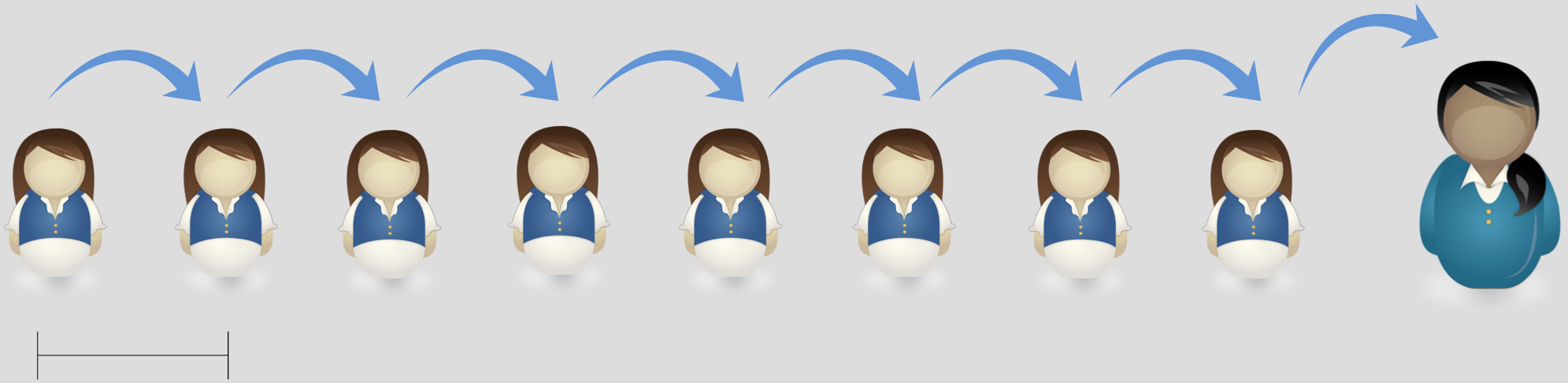
# Single-Hop

A single-hop approach could result suitable for this application:

- The main reason is that since the nodes have to send directly messages to the sink node, this means that all the nodes must be in the transmition range of the sink (less chance for a child to get lost)
- Node only transmit their data, that is, no energy consumption due to receive radio mode

# Answer (2)

The “*Single File*” problem in a **Multi-Hop** fashion



10 m

- Distance between the teacher and the farest child can be very high!

# TECHNICAL DETAILS

# Contiki OS

Contiki is the operating system used to work on sensors

- It uses by default CSMA/CA as paradigm to access the medium (no TDMA paradigm implemented)
- It uses by default ContikiMAC as protocol to save energy
- It uses RPL as routing protocol (no others routing protocol has been implemented)
- Well suited for static WSN, poor performance with mobile nodes
- Code saved in ROM memory, which is very limited
- Only unicast and broadcast communication (no multicast communication allowed)

## Others

- Sky Motes sensors used
  - CC2420 radio driver
  - Supplied by 2 AA batteries
- COOJA has been used as simulator
  - No support for mobile elements

# Radio

Chipcon SmartRF® CC2420

| Parameter | Min. | Typ. | Max. | Unit | Condition / Note |
|---|---|---|---|---|---|
| Relative accuracy | -50 | | 50 | mV | |
| **Power Supply** | | | | | |
| Current consumption in different modes (see Figure 24, page 43) | | | | | Current drawn from VREG_IN, through voltage regulator |
| Voltage regulator off (OFF) | | | 1 | µA | Voltage regulator off |
| Power Down mode (PD) | | 20 | | µA | Voltage regulator on |
| Idle mode (IDLE) | | 426 | | µA | Including crystal oscillator and voltage regulator |
| Current Consumption, receive mode | | 19.7 | | mA | |
| Current Consumption, transmit mode: | | | | | |
| P = -25 dBm | | 8.5 | | mA | The output power is delivered differentially to a 50 Ω singled ended load through a balun, see also page 53. |
| P = -15 dBm | | 9.9 | | mA | |
| P = -10 dBm | | 11 | | mA | |
| P = −5 dBm | | 14 | | mA | |
| P = 0 dBm | | 17.4 | | mA | |

# Energy

- To make some considerations about energy, radio energy consumption has been computed (only transmition, reception and idle radio mode):
  - $E_{TX} = t_{TX} * I_{TX} * V$
  - $E_{RX} = t_{RX} * I_{RX} * V$
  - $E_{idle} = t_{Idle} * I_{Idle} * V$

  Where V=3 Volt (2 AA batteries supply).
- The energy stored into two AA batteries is 18720 Joules (*)

* http://www.allaboutbatteries.com/Energy-tables.html

# Energy (2)

- A **receiver** only consumes during receiving period and idle period, thus its residual energy can be computed as
  - $RE_{RX} = IE - E_{RX} - E_{idle}$
- A **sender** consumes
  - Only during transmition and idle period if a **Single-Hop** paradigm is used, thus
    - $RE_{TX} = IE - E_{TX} - E_{idle}$
  - During transmition, reception and idle period, if a **Multi-Hop** paradigm is used, thus
    - $RE_{TX} = IE - E_{TX} - E_{RX} - E_{idle}$
- IE is the *Initial Energy* of the node, that is 18720 Joules

## Energy saving – Sender side

- In order to save energy, the ContikiMAC protocol has been used.
- An improvement consists in putting in sleep mode the radio between sending periodic messages to the sink.
- This approach is possible only in a Single-Hop paradigm.
- Practically, Contiki gives the possibility to switch radio from receiving/transmitting mode to idle mode and vice versa (no chance to put the radio in sleep mode!)
- Thus, energy consumption can be reached only giving a bound to the number of messages to send!

## Energy saving – Receiver side

- In order to receive all the packets, the receiver should be always on.
- An improvement consist in putting in sleep mode the receiver for the most of time, and turning it on when senders have to send data (synchronization required).
- Practically, even if senders are synchonized, a desynchronization occurs due to the collisions (children are not so distant from each others).
- ContikiMAC performs better!

## Energy saving – Summarize

- In conclusion, reducing energy consumption is made by
  - Reducing the number of packet sent by senders
  - Applying ContikiMAC as protocol in both sender and receiver side

# SIMULATIONS

## Parameters

- SENDER_NUM = 15 // number of senders (fixed)
- Struct etimer timers[SENDER_NUM] = 40 //timers associated with each node, inizialized to 40 seconds (fixed)
- BUFFER_DIM = 1 … 5 //Dimension of the buffer where senders store ALV messages sent by other nodes (only in **Multi-Hop** paradigm) (not fixed)
- PERIODIC_SEND = random_rand()%SENDER_NUM //Time interval in which nodes can send their own ALV (fixed)
- COIN_FLIP = random_rand()%1 //Probability to accept an ALV message coming from another node (only in **Multi-Hop** paradigm) (fixed)

# Paradigms

- Single-Hop
- Multi-Hop

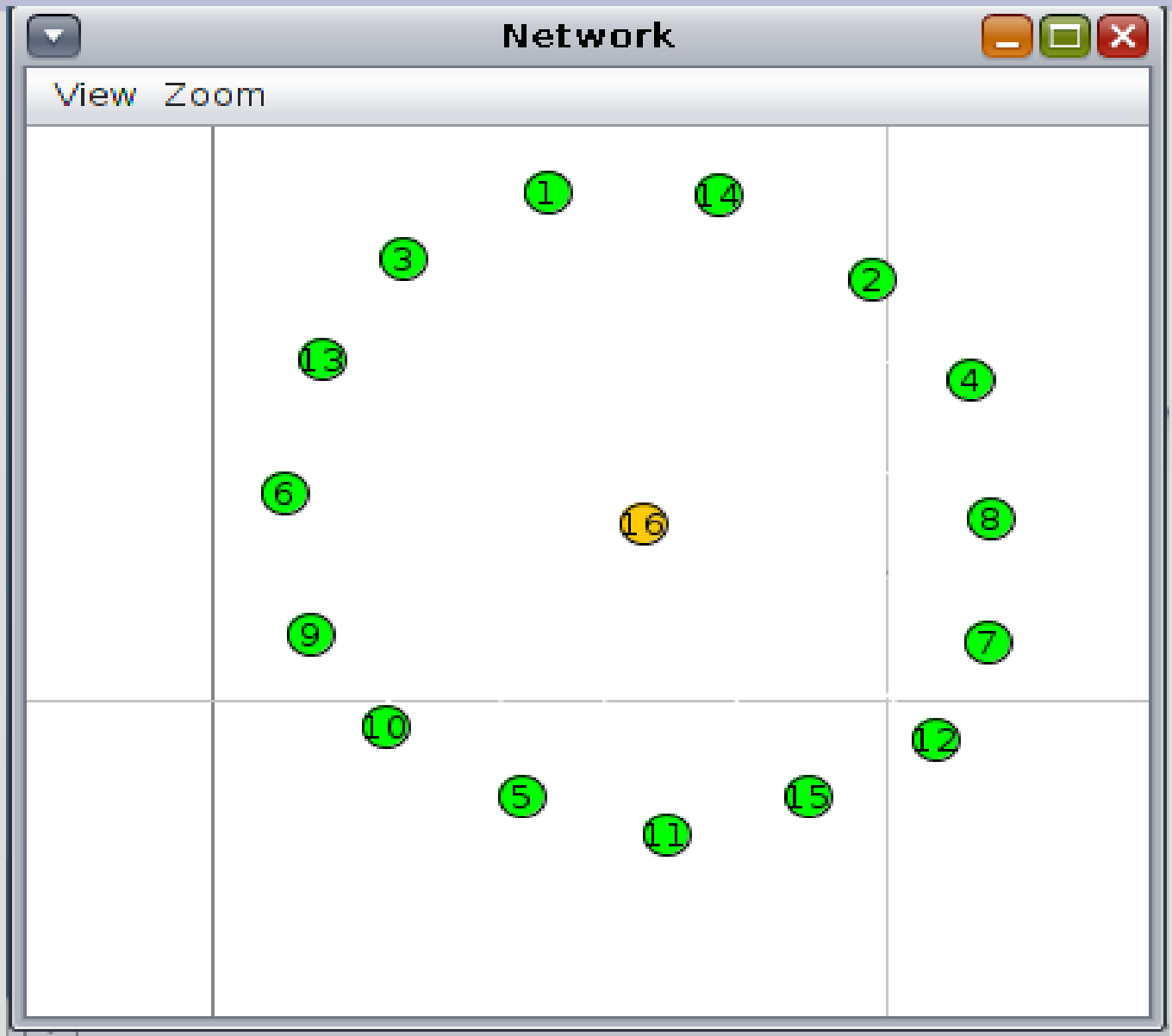
## Scenarios

- *Cirle* – **Single-Hop**
- *Two-in-a-Row* – **Single-Hop** and **Multi-Hop**
- *Two-in-a-Row (Distant Nodes)* – **Single-Hop** and **Multi-Hop**
- *Group (Distant Nodes)* – **Multi-Hop**

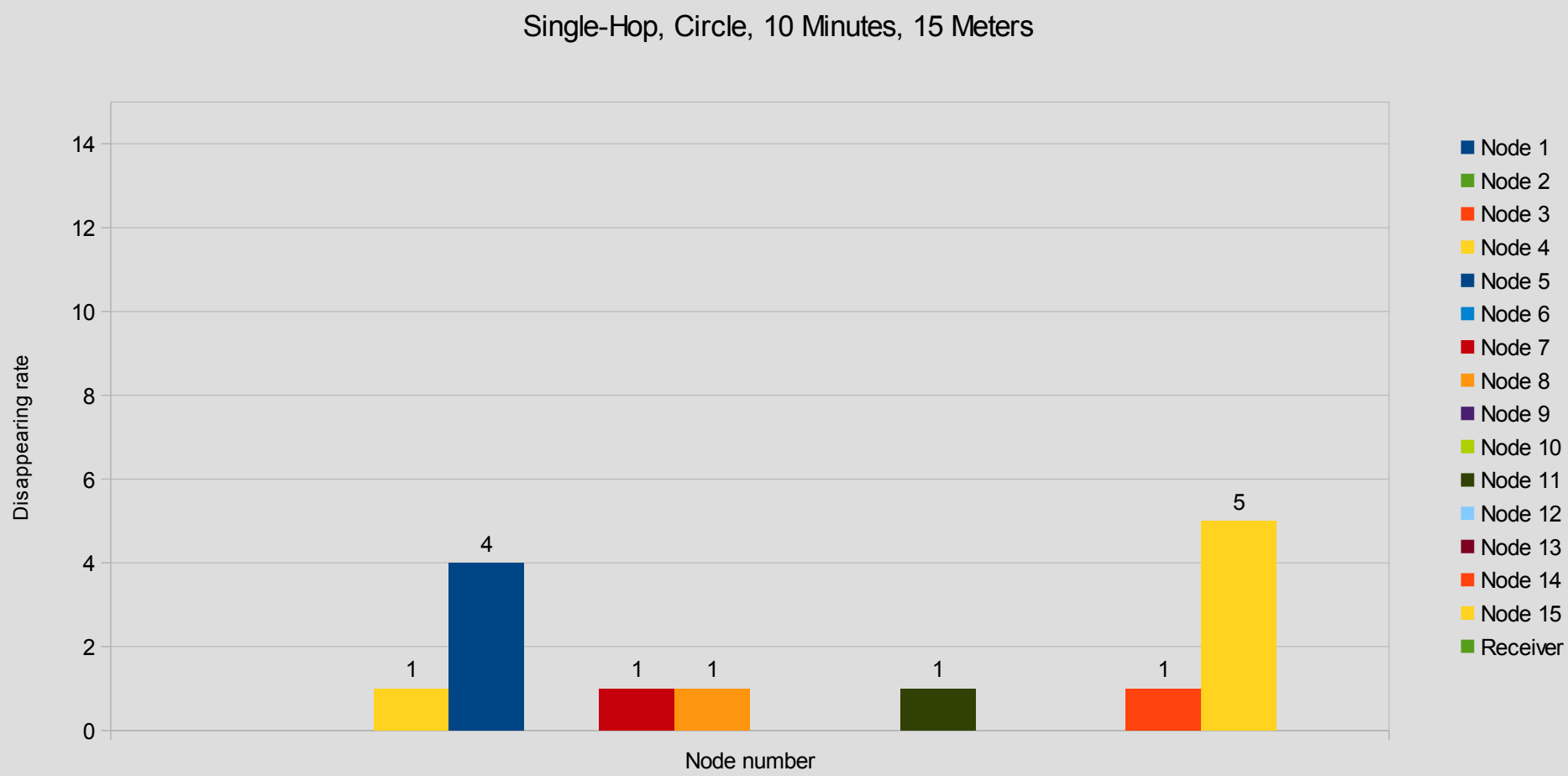
## Others

- Simulation lenght: 10 minutes
- Number of run per simulation: 10
- Area dimension: variable
- Distance loss probability applied
- Outputs:
  - Number of times that a node has disappeared
  - Number of ALVs received per node
  - Residual Energy of each node

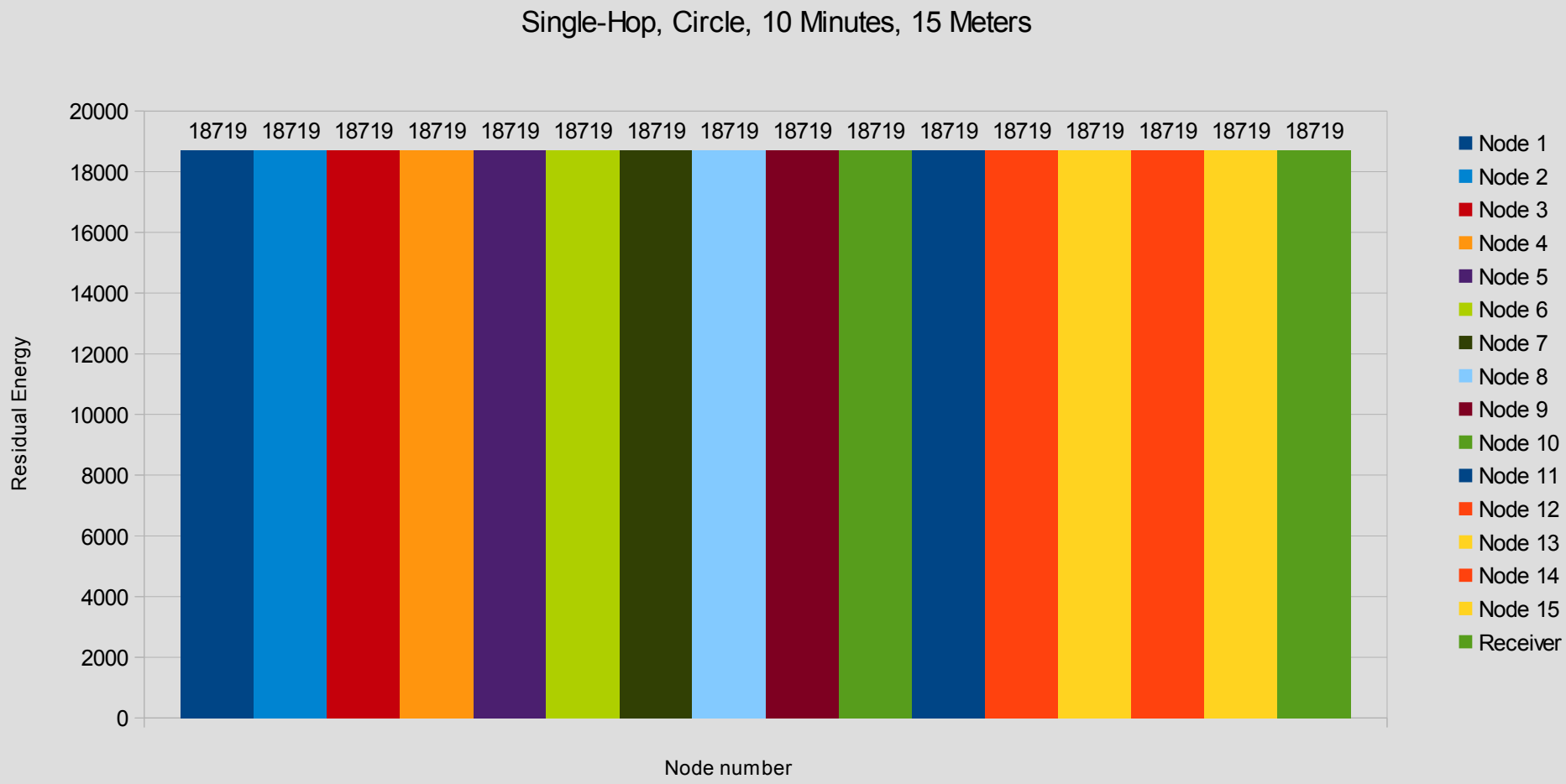
# Scenario: Circle



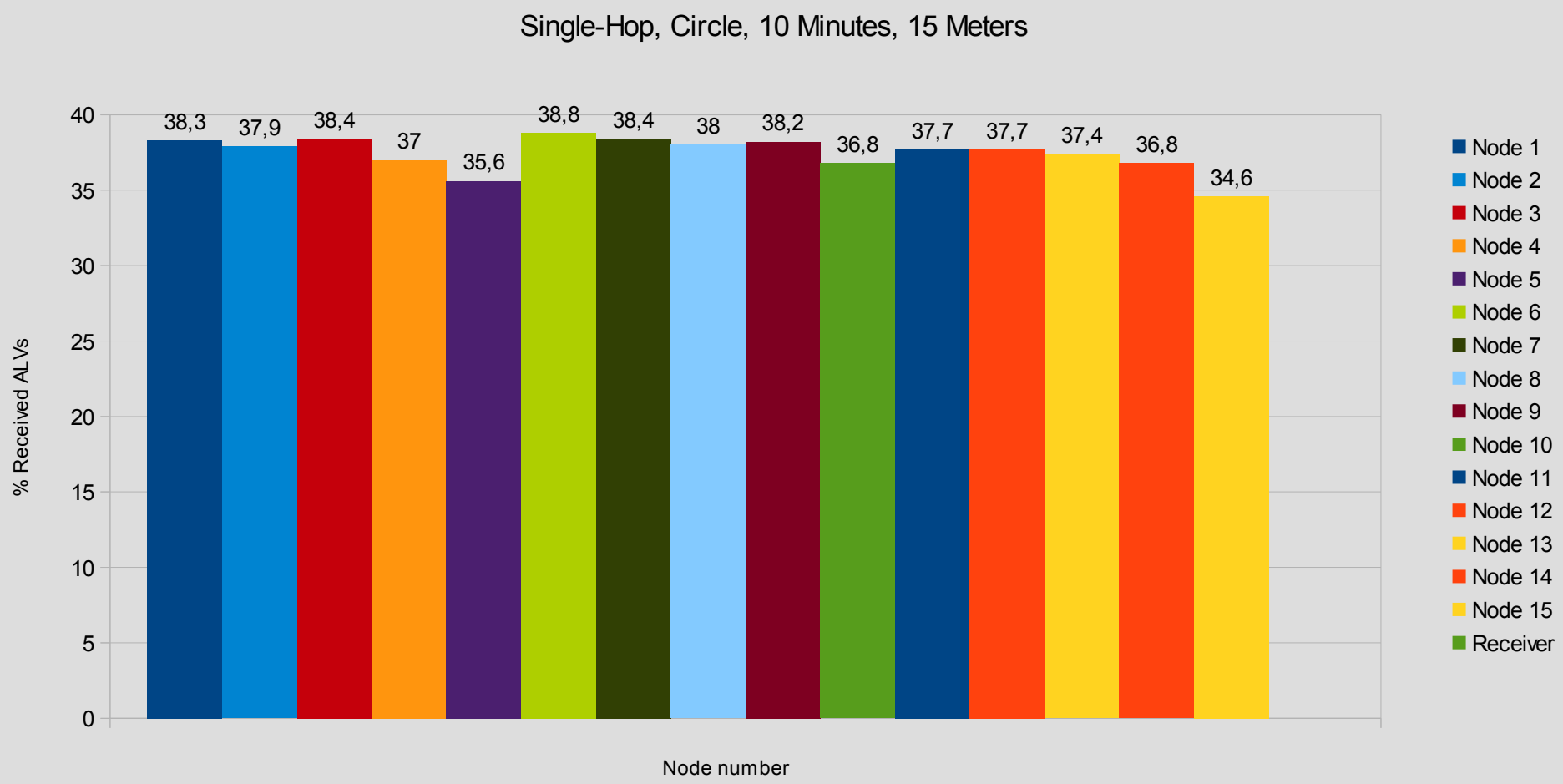
# Disappearing Rate



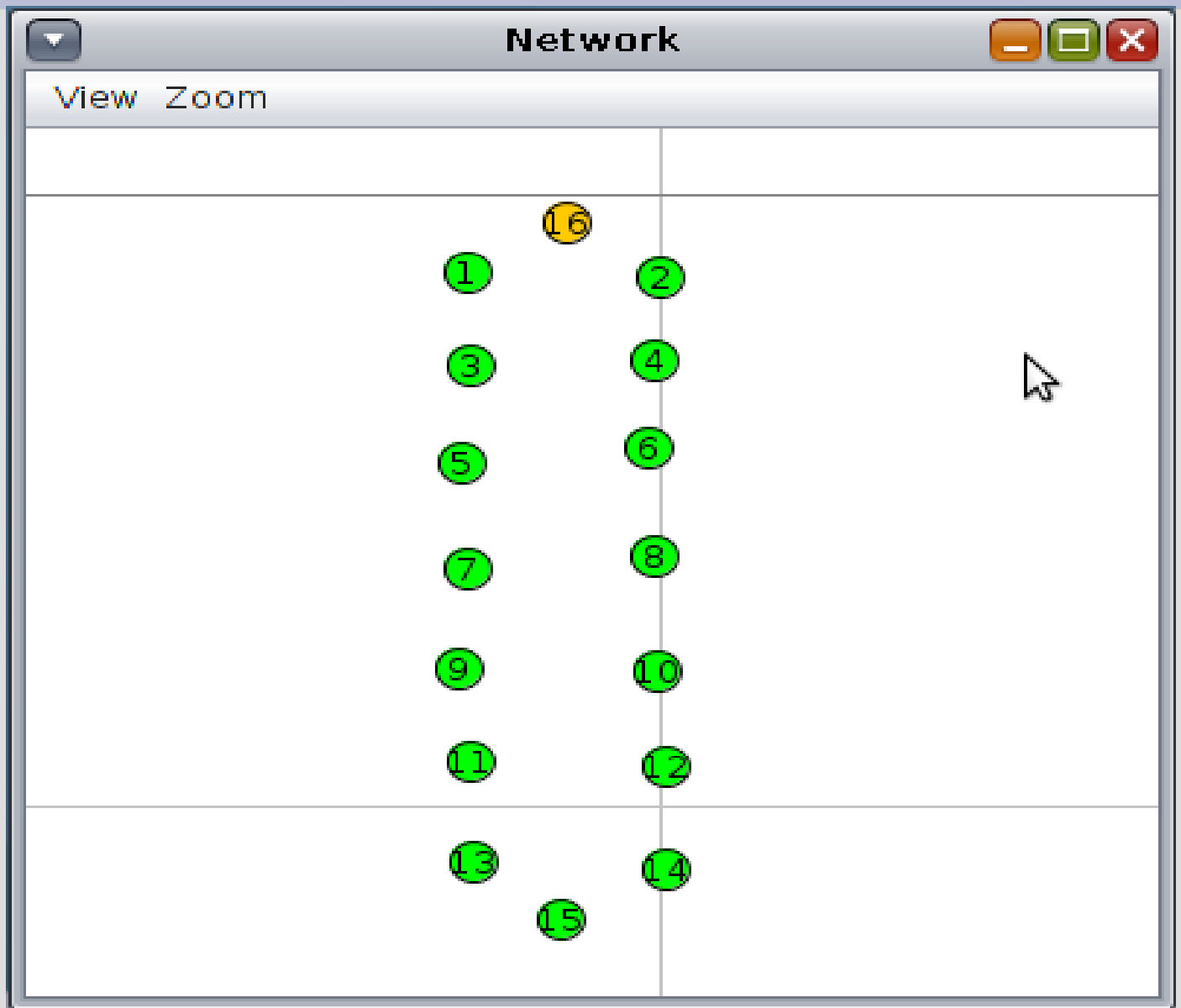
# Residual Energy



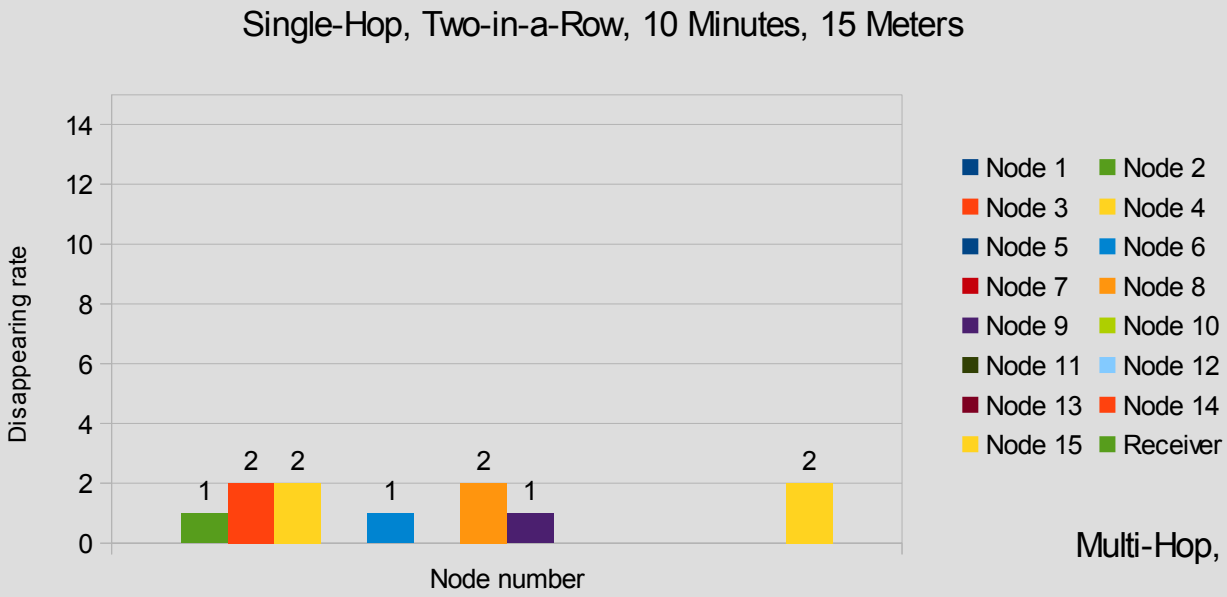
# Number of received ALVs

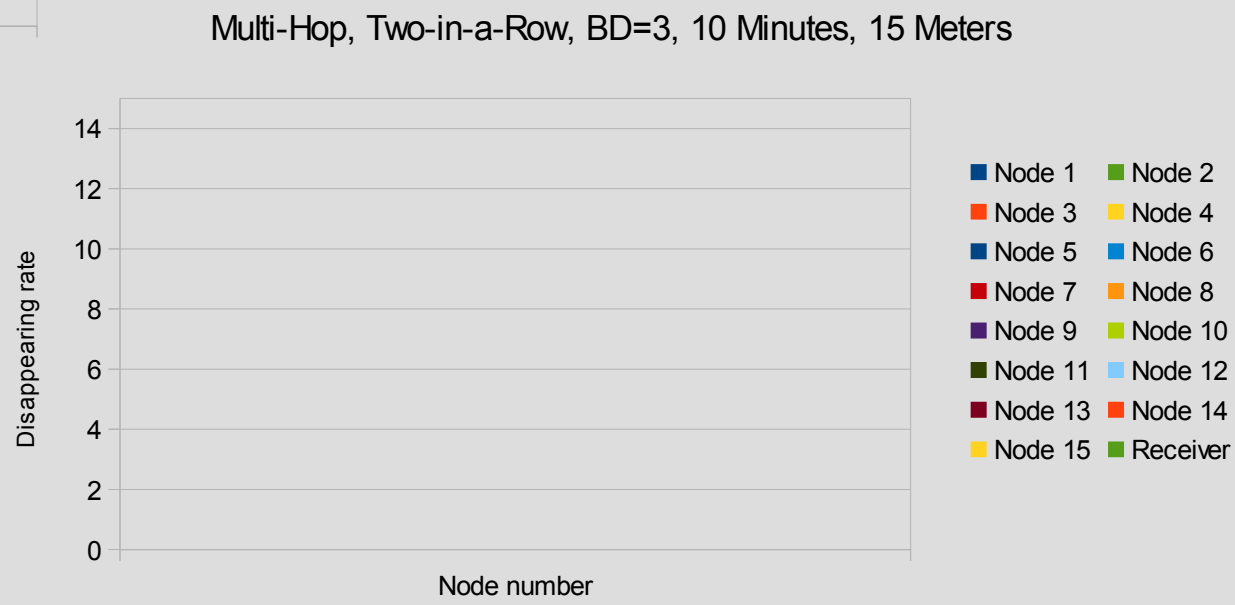


# Scenario: Two-in-a-Row

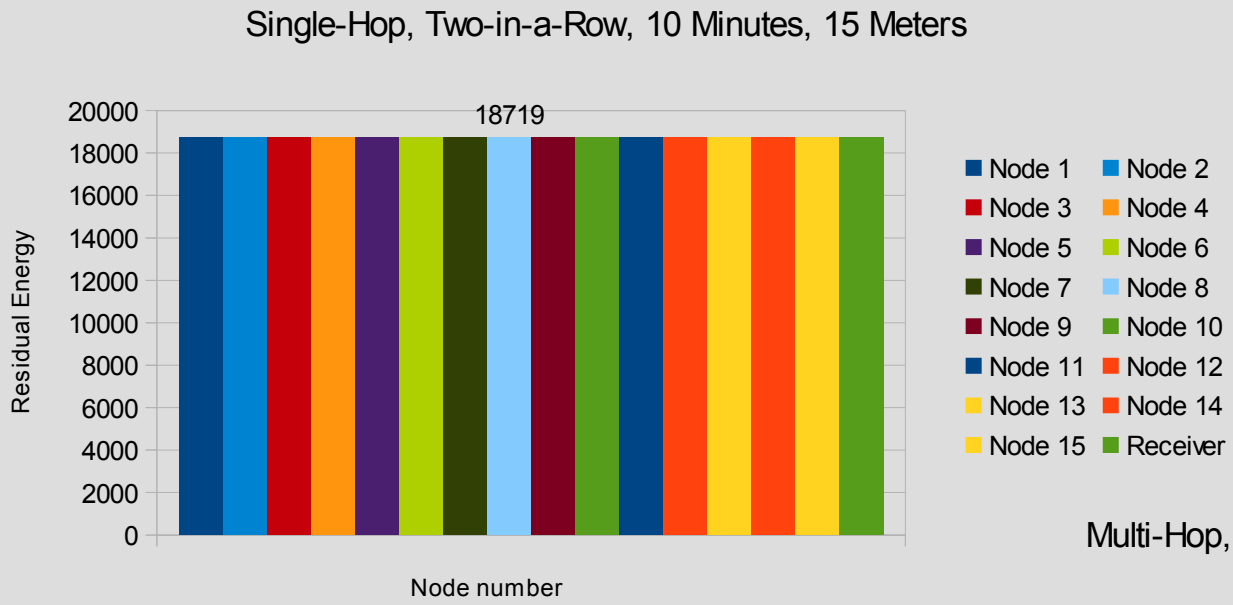


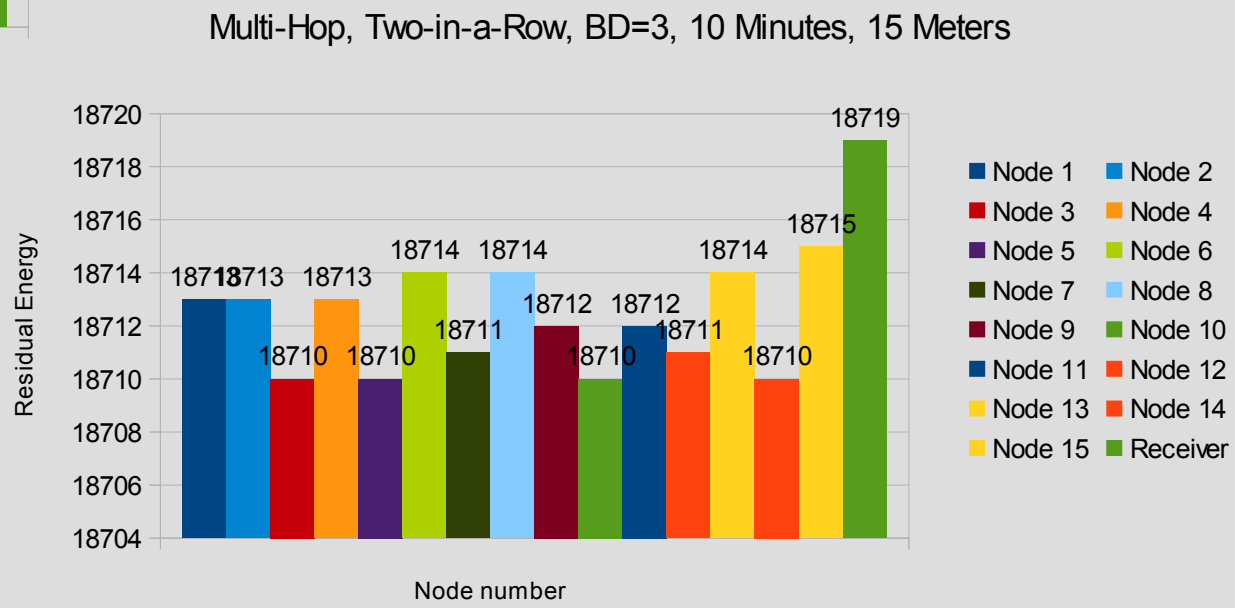
# Disappearing Rate



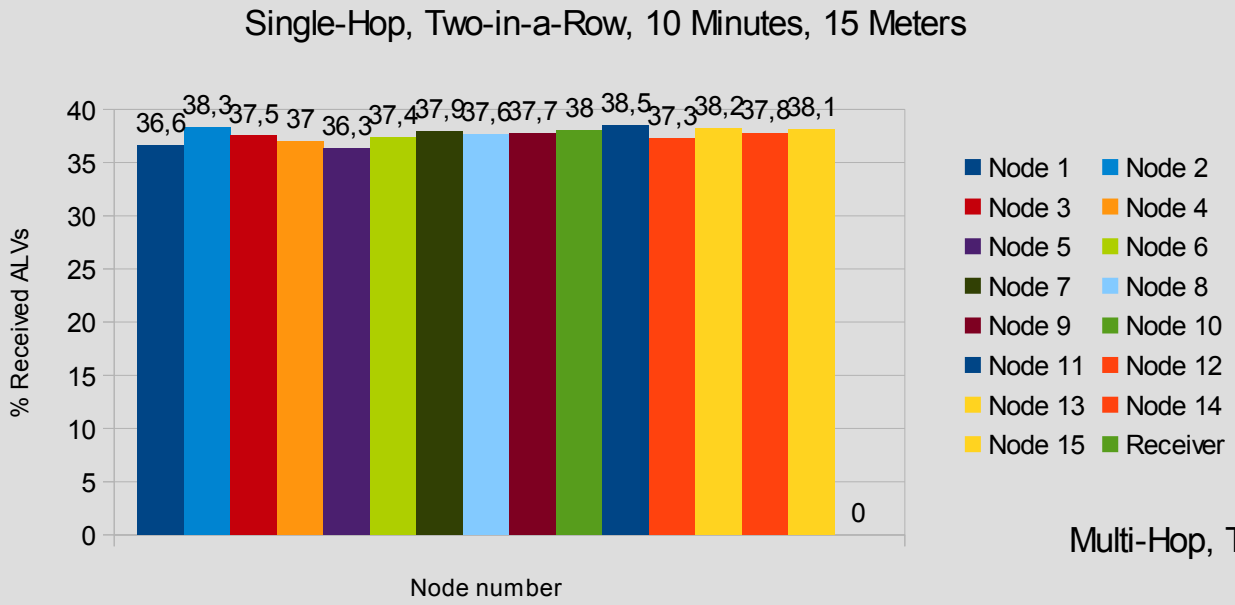



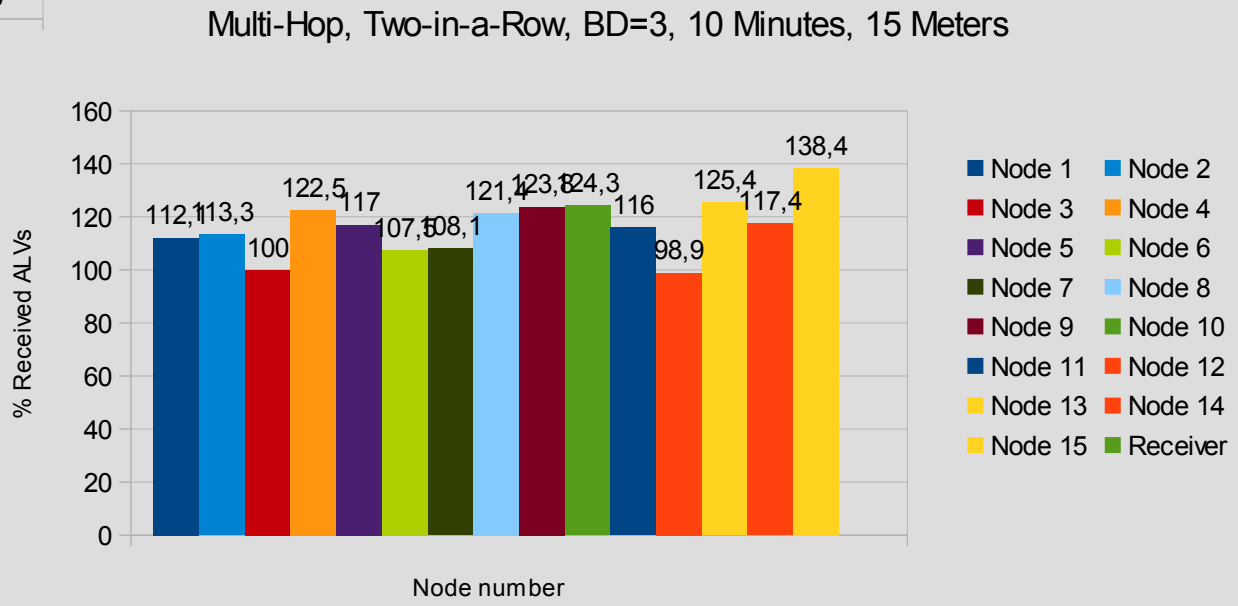
# Residual Energy



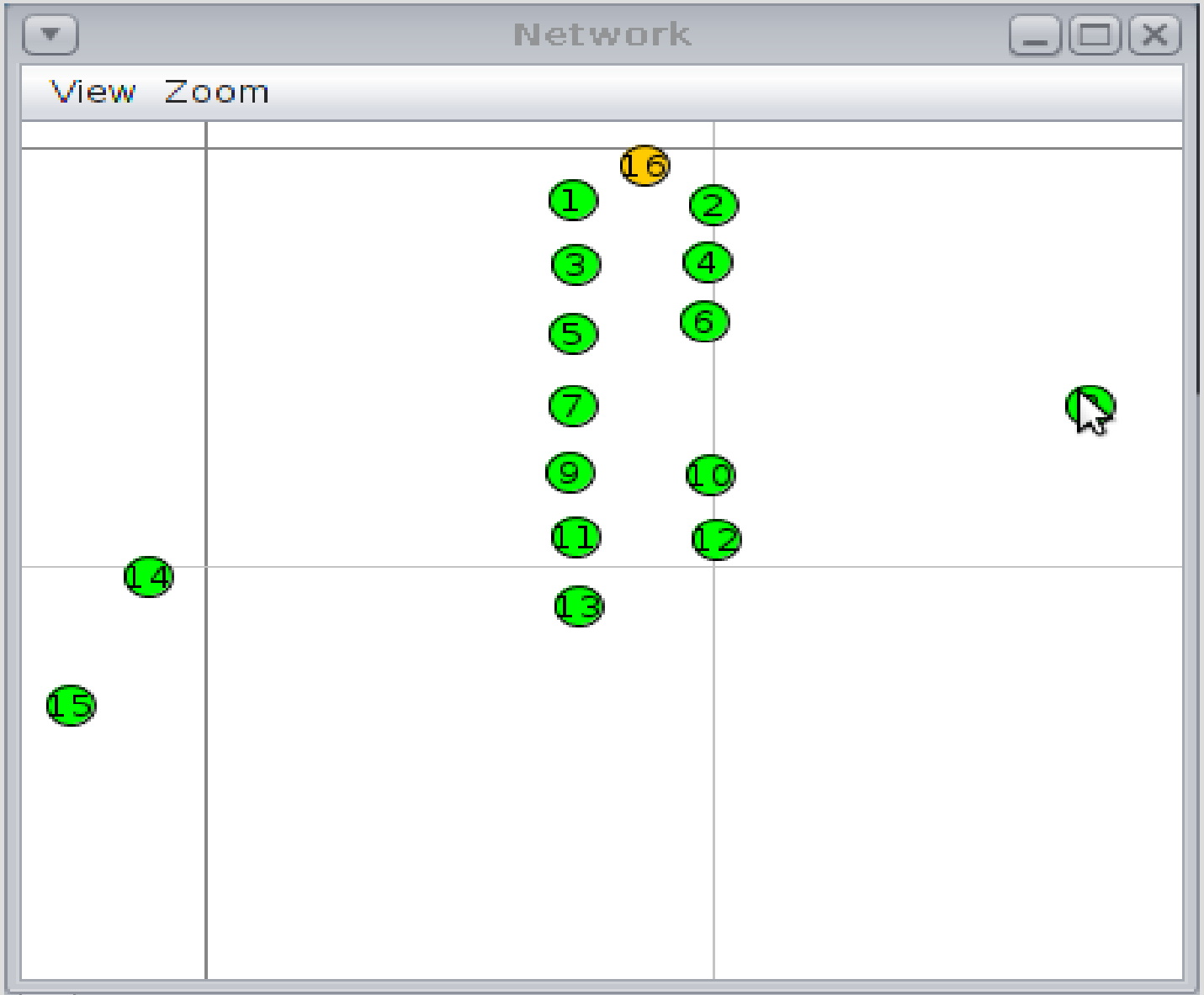



# Number of received ALVs

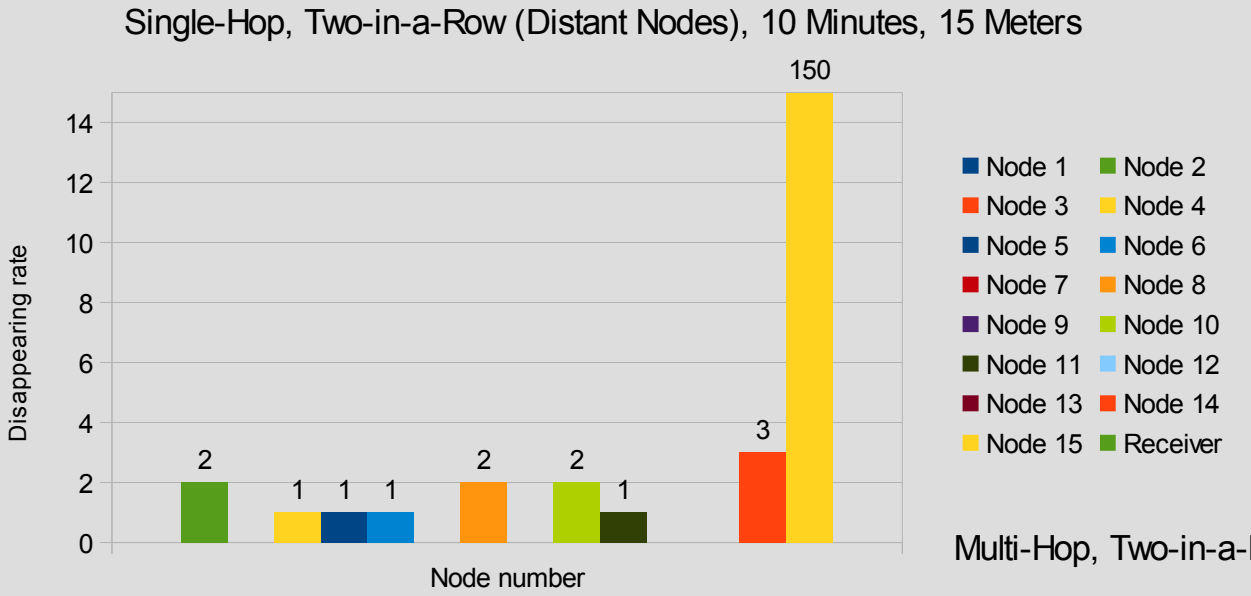


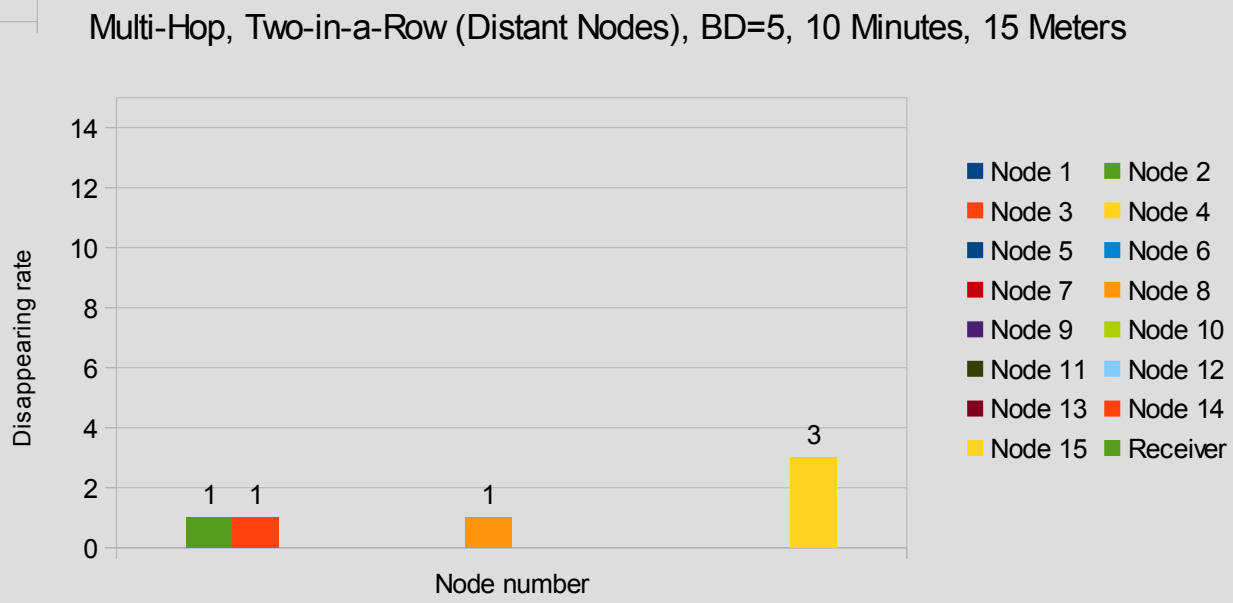



# Scenario: Two-in-a-Row (Distant Nodes)



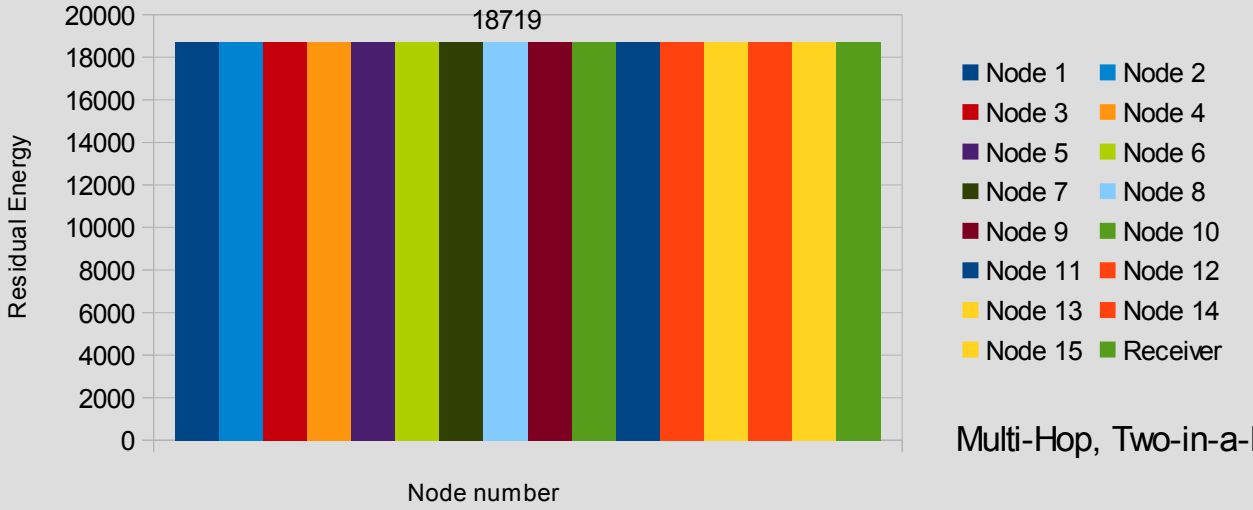
# Disappearing Rate



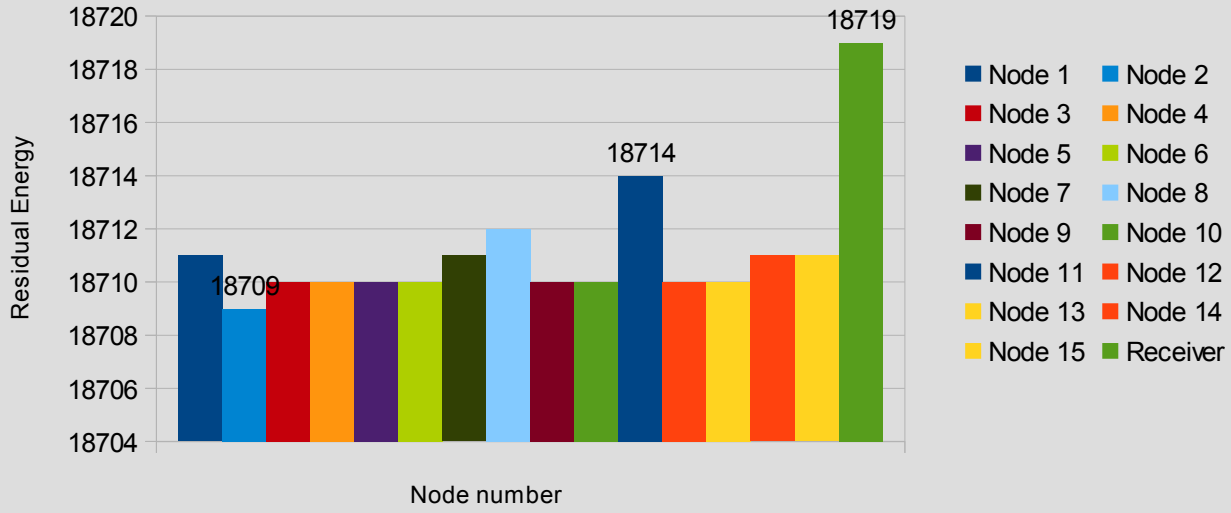



# Residual Energy

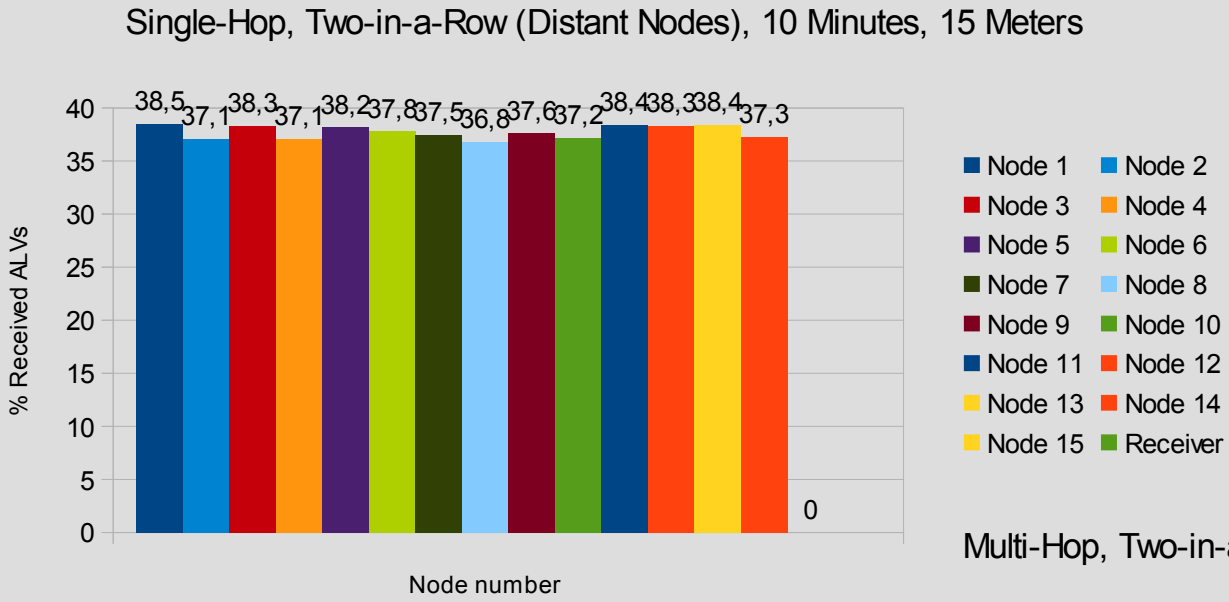
Single-Hop, Two-in-a-Row (Distant Nodes), 10 Minutes, 15 Meters

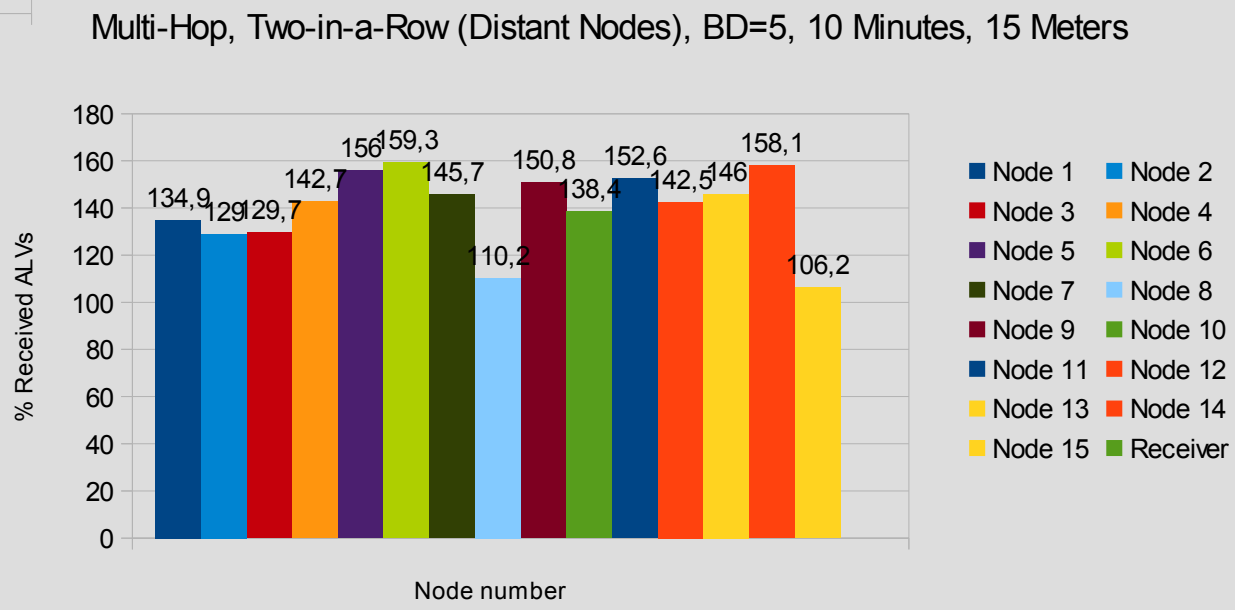



Multi-Hop, Two-in-a-Row (Distant Nodes), BD=5, 10 Minutes, 15 Meters

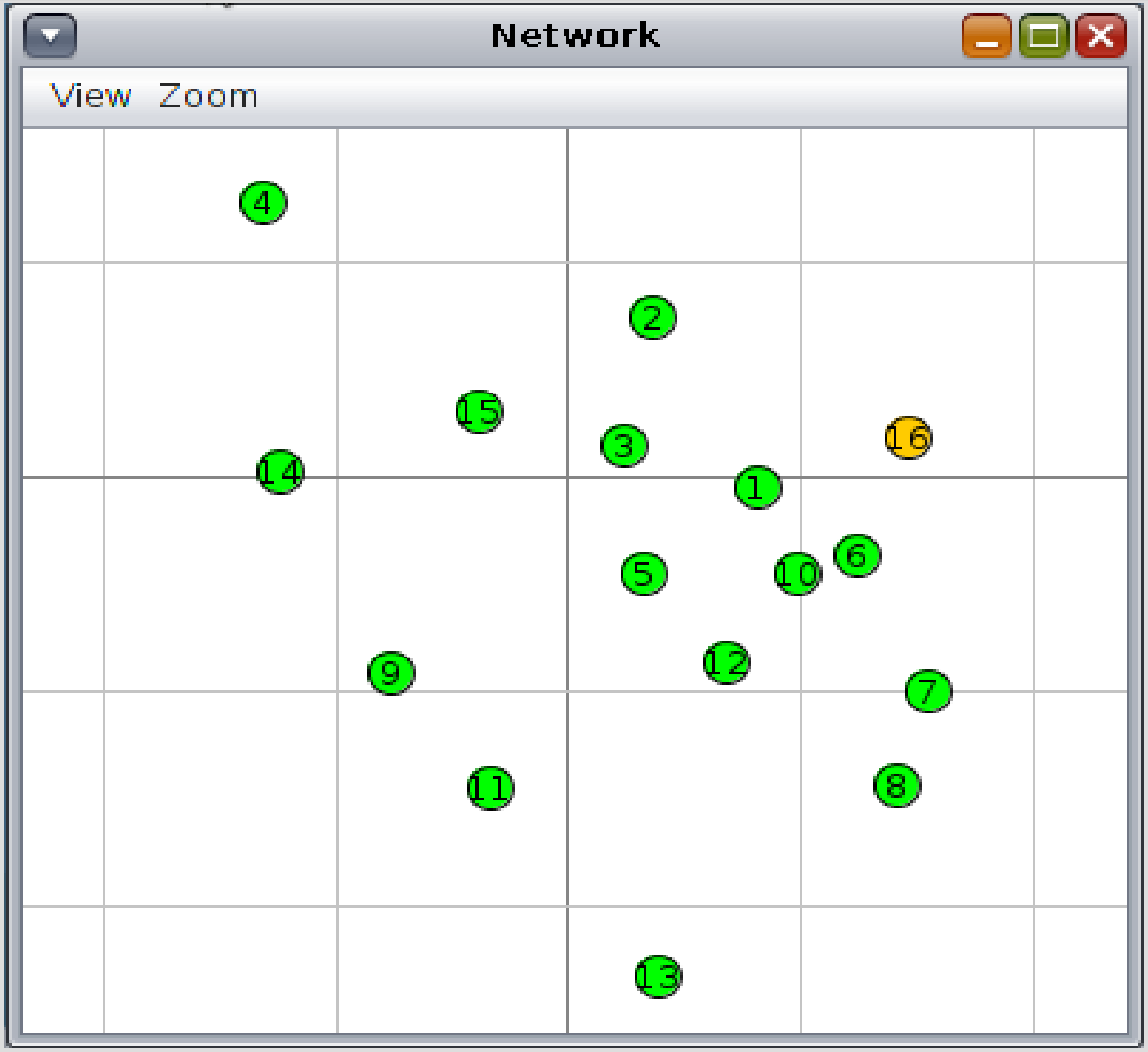


# Number of received ALVs

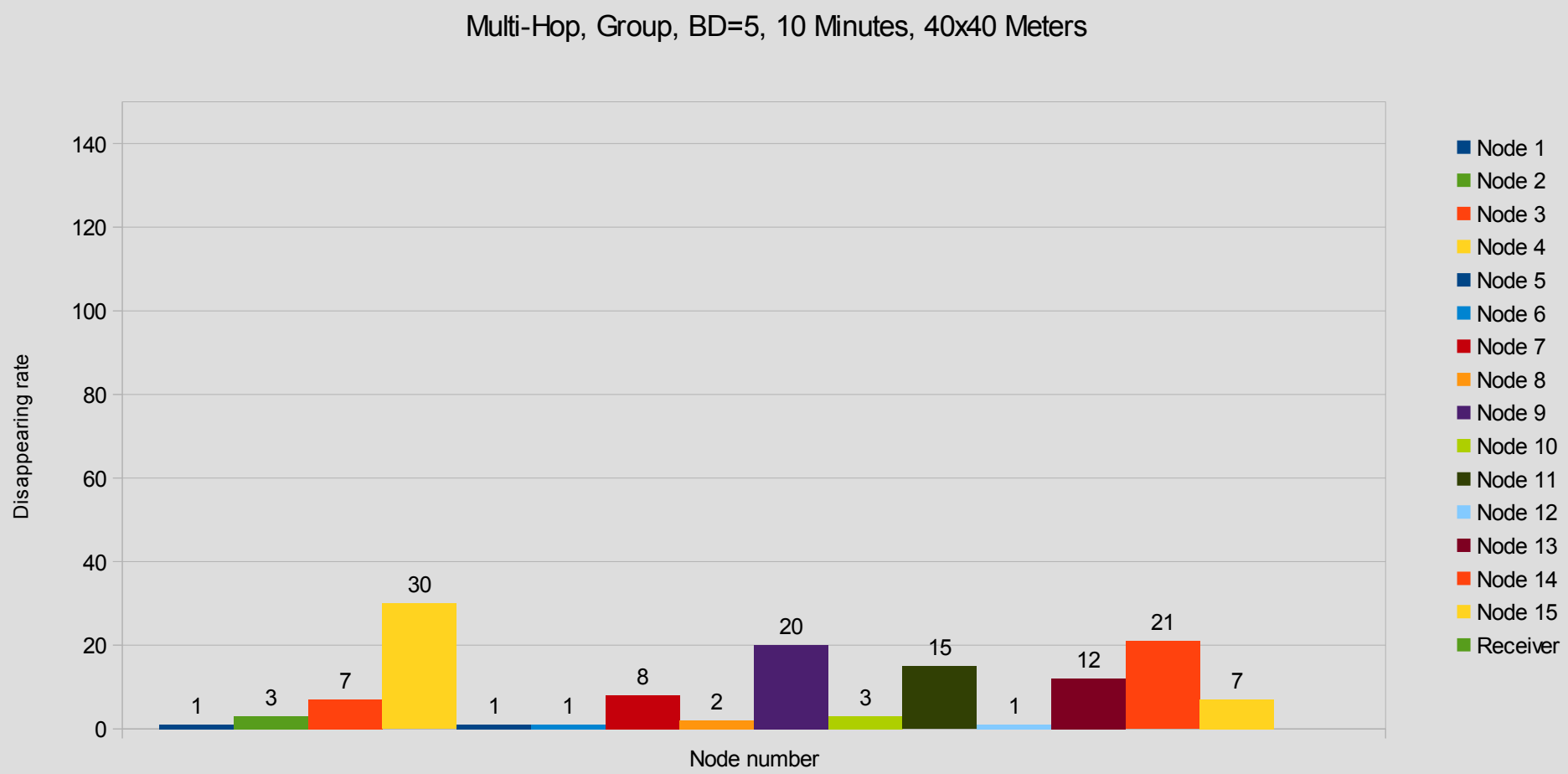


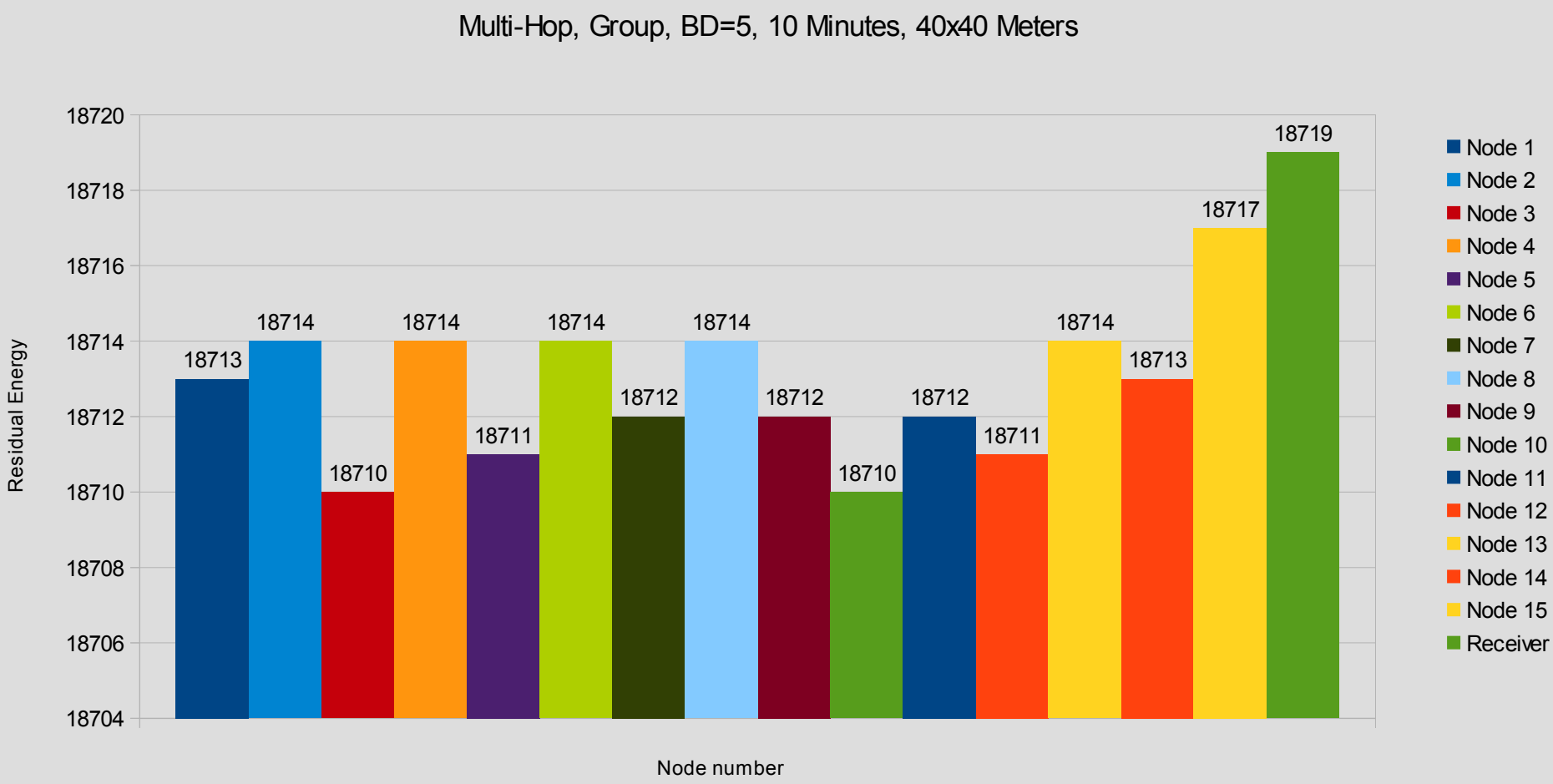
# Scenario: Group (Distant Nodes)



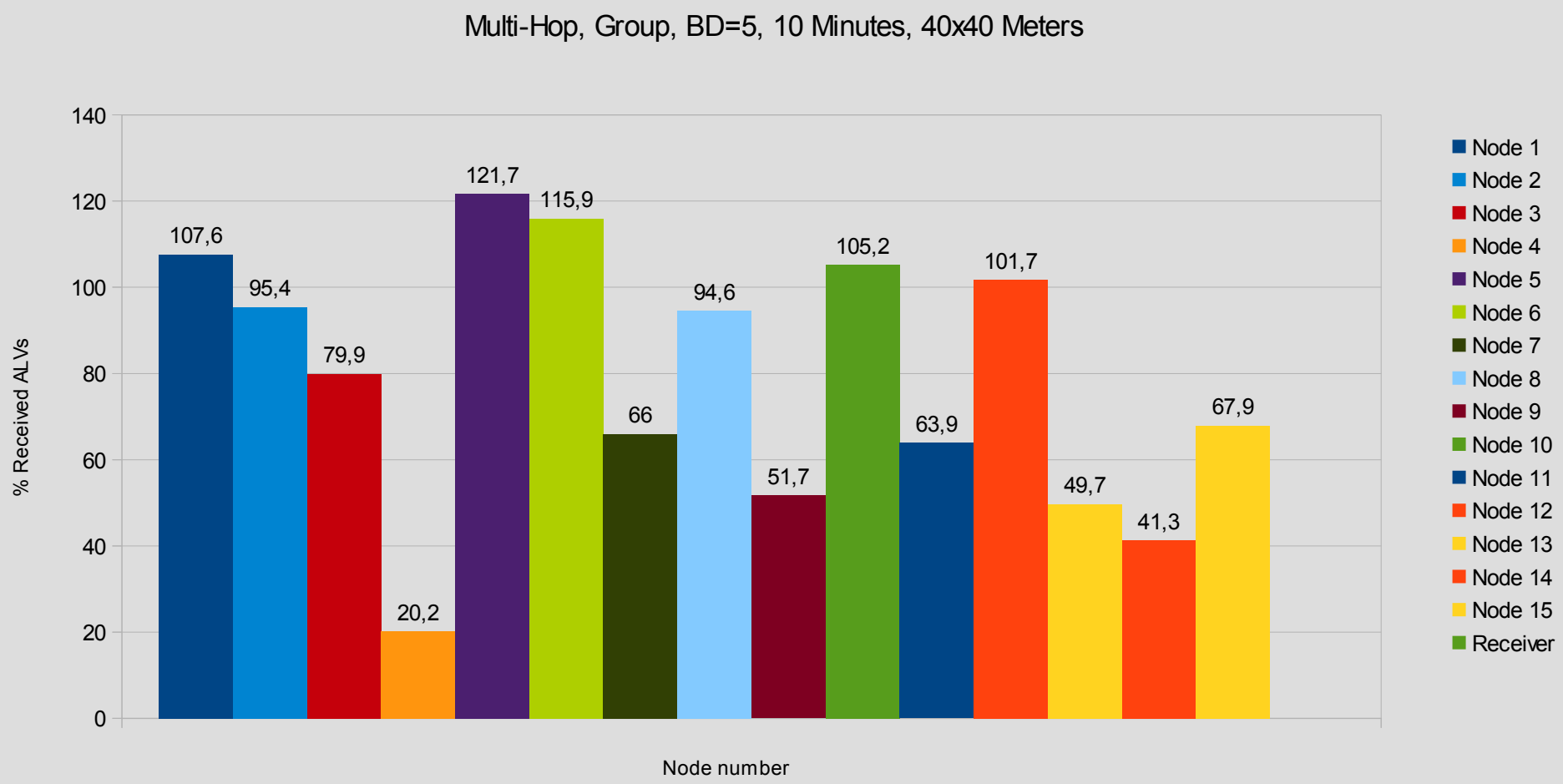
# Disappearing Rate



# Residual Energy



# Number of received ALVs



## Conclusions

- Multi-Hop performes well then Single-Hop paradigm in terms of disappearing rate
- Single-Hop performes well then Multi-Hop paradigm in terms of energy saving


THANKS FOR THE ATTENTION