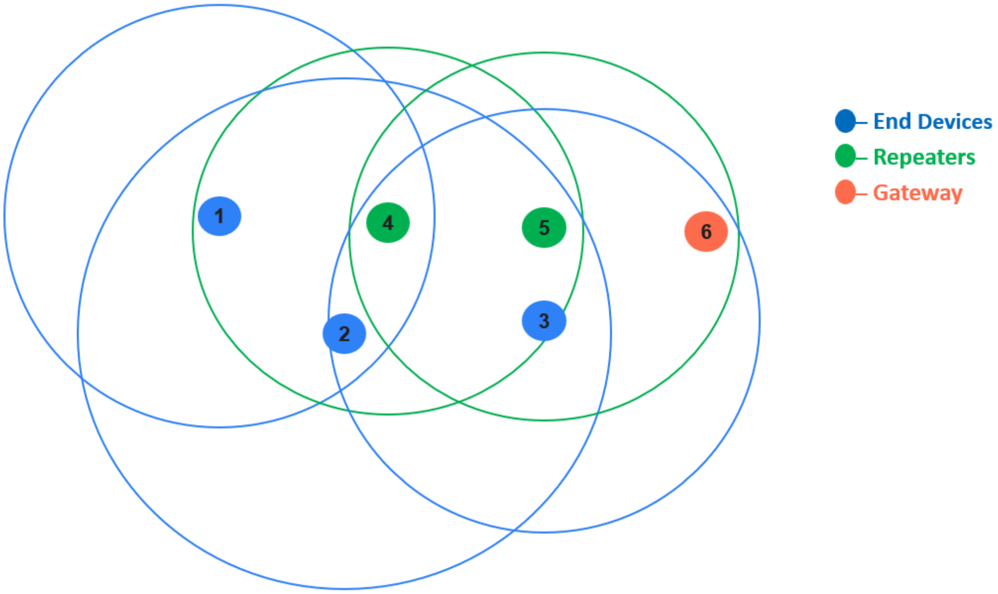
1. Introduction

LoRaWAN is a standardized technology that is commonly used in implementing star-type IoT networks where a set of end devices directly connect with a LoRaWAN gateway forming a single-hop network. LoRaWAN is the Medium Access Control (MAC) layer communication protocol that utilizes LoRa in its underlying physical layer. LoRa is a wireless modulation technology for long-range, low-power communication. Due to the popularity of its low-power, long-range capability, LoRa is now being used in multi-hop networks that utilize customized MAC layer protocols instead of LoRaWAN.

Although there are many software to simulate a LoRaWAN network, simulating a LoRa-based multi-hop network is a challenge, especially because it may not have a standardized protocol. Yet the recent research on LoRa-based multi-hop networks tends to utilize very similar wireless communication techniques such as message broadcasting and the use of LoRa repeaters. Using those common principles it is possible to create a generalized model of a LoRa mesh network, even though there is no standardized protocol. This document presents a simulation platform to simulate such a general LoRa mesh network consisting of LoRa end devices, repeaters, and gateways.

Due to its generalized style, this simulator can be used to simulate all types of mesh networks starting from very simple networks to highly complex networks such as linear chain-type LoRa networks. LoRa-based mesh-type sensor networks, actuator networks, end device-to-end device communication networks, and networks with multiple gateways are a few examples of networks that can be simulated. Even a LoRaWAN network can be simulated using this simulator. The following figure illustrates such an example network where the transmission ranges of each of the nodes overlap with many other nodes causing a lot of possibilities for contention. There are also multiple paths for traversing packets. A packet from the end device No.3 can directly reach gateway No.6 while the same packet may also hop through the No.4 and/or No.5 repeaters and then arrive at gateway No.6.



There are many user-configurable variables introduced in this simulator to customize a network. On the other hand, since the algorithms of the repeaters, end devices, and gateways are implemented in discrete Python functions, those functions can be easily overridden to implement new algorithms. Most importantly changing the repeater algorithm would result in a different network protocol. Therefore, this simulator can be easily adapted to simulate various types of network protocols that are different from the default implementations.

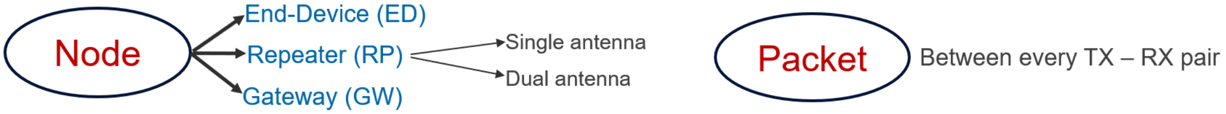
When it comes to the topic of usefulness, it is now clear that this simulator would be highly valuable in testing repeater algorithms. This simulator can also be used in simulating actual networks before deployment and simulating very large networks that are not feasible to be physically deployed, but are useful for research purposes. A scenario where a network of hundreds of nodes running for many days can be simulated in a few minutes using this simulator. The reception rates generated from such a simulation can be confidently presented as probabilities of message delivery in that network. Thus, this simulator will be immensely helpful in the development of LoRa-based mesh networks.

2. Python Program Structure

This simulator is entirely written in Python and employs libraries like SimPy and MatPlotLib. SimPy is the discrete-event simulation framework and MatPlotLib is used for simulation graphics and plotting of graphs of the simulation results.

The simulator is written in the file named *'loraMeshSimulator.py'* as a Python library that can be imported into a separate Python script (*'simulationScript.py'*) that subsequently creates, configures, and simulates a custom LoRa mesh network. For someone trying to learn using this simulator, learning how to write a simulation script is a good point to start with. After having a solid understanding of writing simulation scripts, it would be easier to refer to the source code of the simulator. Finally, the user may even alter the source code, introducing different repeater algorithms, collision models, loss models, etc.

The simulator's source code is written focusing on two Python classes: the Node class, and the Packet class.

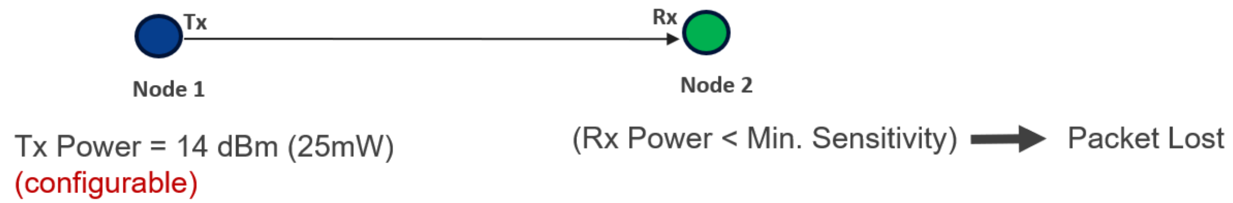


The Node class is the blueprint used to create all types of nodes (end devices, repeaters, gateways) in creating a network in the simulation script. When creating a node, it is placed on an x-y cartesian plane. The simulator facilitates pre-configuring many parameters when creating nodes such as packet sizes, transmission/waiting periods, LoRa Spreading Factor (SF), bandwidth, carrier frequency, coding rate, transmission power, etc.

Packet class is used to create virtual packets which individually represent the LoRa transmissions between each and every pair of transmitter and receiver. These virtual packets essentially model the spread of a LoRa radio signal throughout the surrounding environment of the transmitting node. Packets are used to calculate and mark the range of a transmitter according to the path loss.

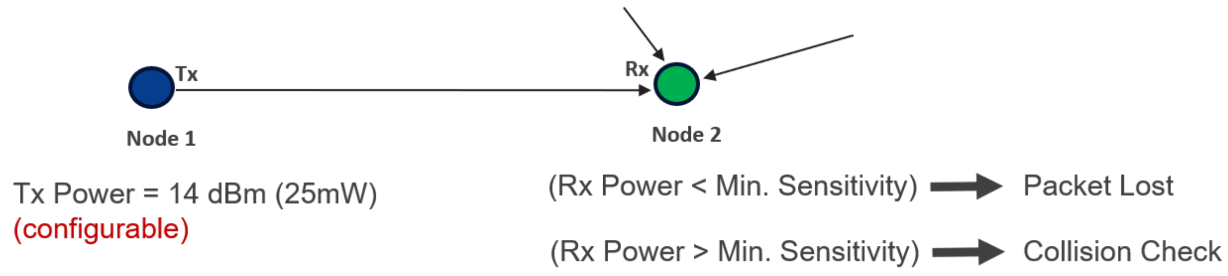
Path Loss

Currently, the simulator is using a simple log-distance path loss model (a free space path loss model) by default. It can be overridden with a more specific, advanced path loss model depending on the targeted simulation environment. The following diagram is a simple illustration of a packet transfer between two nodes. When Node 1 transmits a packet it will reach the receiver of Node 2 with some remaining signal power after the path loss. If the received signal power is less than the minimum sensitivity of the receiver, the packet is considered lost and the Node 2 is considered out of range. The minimum sensitivity is obtained from a table of sensitivity values for different LoRa configurations. This table should be made experimentally and its values may change depending on the simulation environment.



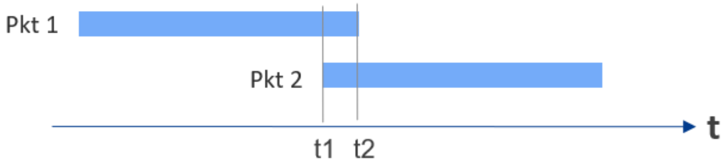
Collision Detection

In the above scenario, if the received signal power is greater than the minimum sensitivity, that packet can be successfully captured by the receiver. Still, the packet can get corrupted and unrecoverable due to interference from other signals. Therefore, a packet collision check is done as the next step.



For example, when there are 3 packets at the receiver of Node 2 at the same time, those packets may collide with each other. Depending on the LoRa configurations, timing, and signal power of each of the packets, all 3 packets may collide or only 2 packets may collide or none will collide. Even when two packets collide with each other, one may be still recoverable depending on the timing and signal power. This simulator has theoretically considered all those possible ways of collision.

For two packets to collide, firstly they should be in the same frequency channel. If the frequency channel is the same, the LoRa SFs of the two packets are checked next. If the SFs are equal those two packets collide with each other. If the SFs are different, the current implementation of the simulator concludes that there is no collision, because theoretically, all the SFs should be orthogonal, but in a practical scenario such packets with different SFs may act as noise to each other, resulting in packet corruptions. Further research with practical experiments should be carried out to accurately model such SF collisions. When both frequency and SF collisions have occurred, still the packets may be recoverable depending on their power and timing. If the received signal strength of one packet is comparatively higher than the other packet, it may overpower the other packet. In that scenario, the more powerful packet may be recoverable. In another scenario, if the latter coming packet is late enough such that only a part of its preamble collide with the other packet, that latter coming packet may be still recoverable. The following figure illustrates such a scenario.



Likewise, all possible scenarios of collision are evaluated in the simulator using 4 Python functions for frequency collisions, SF collisions, power collisions, and timing collisions.

Experiments

LoRa parameters of the transmitters in a network can be configured in many ways. A user can either write his own script for configuring LoRa parameters or use a preset given in the simulator. The *'experiment'* global variable should be assigned an integer 0-5 to select a preset. The configuration of LoRa parameters by these presets is coded in the '*createPackets'* function of the Node class.

Experiment 0 creates packets with the longest airtime for an ALOHA-style network. Experiment 1 is also similar to 0, but its packets are randomly divided into three frequency channels. Experiment 2 creates packets with the shortest airtime, still in ALOHA-style. Experiment 3 performs an auto-configuration of LoRa parameters to select the shortest possible packets for each of the transmitters depending on their distance from the surrounding receivers. Finally, experiment 4 assigns all the LoRa parameters randomly.

3. Writing a Simulation Script

A simulation script should begin with importing the simulator library named *'loraMeshSimulator'*. Then, the user may re-label the entities from the simulator that are frequently used in the simulation script. Creating nodes specifying their locations and types is the first thing to do in the script. Then their LoRa parameters can be manually pre-configured, or a preset can be selected. After placement and configurations on each individual node, the *networkConfig()* function should be called to configure the whole network. Then the transmit functions of the nodes should be called as per the requirements of the network. If the graphics are enabled the boundaries of the graphical plot should also be defined. Then the total number of packets for the simulation is given followed by starting the simulation by calling *env.run()*.

That function call returns only after the simulation is finished and there are no more events to be processed by the SimPy simulation environment. Therefore, after that function the user may log the results of the simulation accordingly.

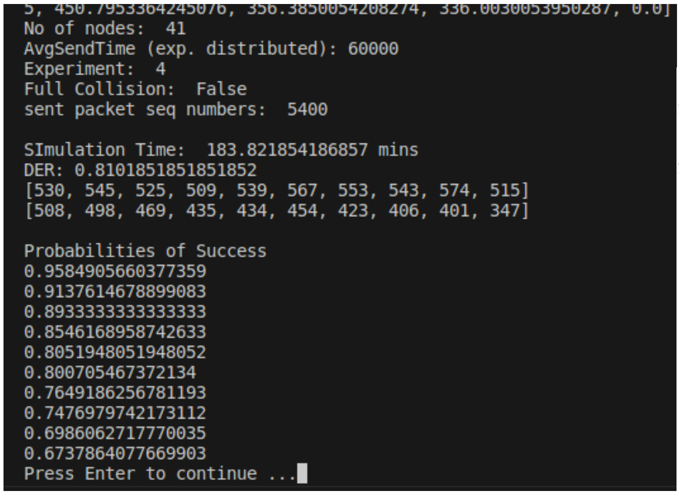
Following is an example simulation script.

```
import loraMeshSimulator as sim

#-------------------------------------------------------------------------------------
# Simulation config
#-------------------------------------------------------------------------------------
node = sim.node
nodes = sim.nodes
env =sim.env
maxDist =sim.maxDist

gw = sim.node(env, 4.6*maxDist, 1*maxDist, "gw")

e1 = node(env, 1*maxDist, 1*maxDist, "ed")
e2 = node(env, 1.5*maxDist, 1.5*maxDist, "ed")
e3 = node(env, 1.5*maxDist, 0.5*maxDist, "ed")
e3 = node(env, 2.8*maxDist, 1.5*maxDist, "ed")

r1 = node(env, 1.9*maxDist, 1*maxDist, "rp")
r2 = node(env, 2.8*maxDist, 1*maxDist, "rp")
r3 = node(env, 3.7*maxDist, 1*maxDist, "rp")

sim.networkConfig()

#Sensor Network
for i in range(len(nodes)):
    if (nodes[i].type.lower() == "ed"):
        env.process(nodes[i].transmit(env))

#Actuator Network
# for i in range(len(nodes)):
#     if (nodes[i].type.lower() == "gw"):
#         print("HELLO!!!!!!!!!!!!!")
#         env.process(nodes[i].transmit(env))

#prepare show
if (sim.graphics == 1):
    sim.plt.xlim([0, sim.xmax])
    sim.plt.ylim([0, sim.ymax])
    sim.plt.draw()
    sim.plt.show()

# start simulation
sim.totalSimPackets = 20
env.run()

#-----------------------------------------------------------------------
#Print simulation stat
print ("No of nodes: ", len(nodes)) #FIX
print ("AvgSendTime (exp. distributed):",sim.avgSendTime)
print ("Experiment: ", sim.experiment)
print ("Simulation Time: ",env.now/60000,"mins")
print ("Full Collision: ", sim.full_collision)

#print "sent packets: ", sent
print ("sent packet seq numbers: ", sim.totalSimPackets)

#print received packets at each repeater/gateway FIX
for i in range(0,len(nodes)):
    if(nodes[i].type.lower() == "rp"):
        print("packets received at Repeater",nodes[i].id, ":", nodes[i].recPackets)
    elif(nodes[i].type.lower() == "gw"):
        print("packets received at Gateway",nodes[i].id, ":", nodes[i].recPackets)
    else:
        print("packets received at End-device",nodes[i].id, ":", nodes[i].recPackets)

# print all collisions and losses
print ("collided packets: ", sim.collidedPackets)
print ("lost packets: ", sim.lostPackets)
print("gw received packets: ", sim.packetsRecBS)

# data extraction rate
der = len(sim.packetsRecBS)/float(sim.totalSimPackets)
print("DER:", der)

# this can be done to keep graphics visible
if (sim.graphics == 1):
    input('Press Enter to continue ...')

```

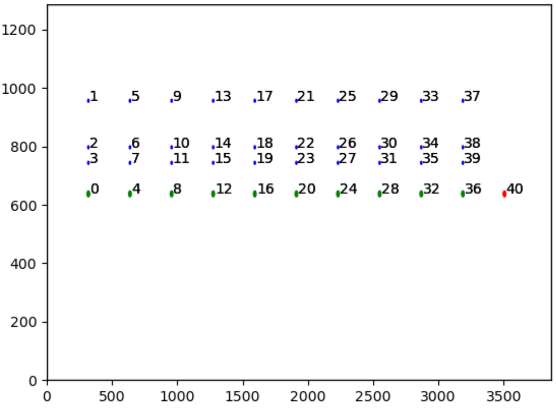
4. Simulation Results

As mentioned in the previous section, all the simulation results can be printed at the end of the simulation script as required by the user. The example simulation script includes printing the rate of successful message deliveries of the entire network. It also prints the rate of successful message deliveries for each of the message-generating end devices or gateways, which is helpful in evaluating the fairness of the network. Even different lists of collided, received, or lost packets can be printed for verification purposes.

Example Final Results Logs.

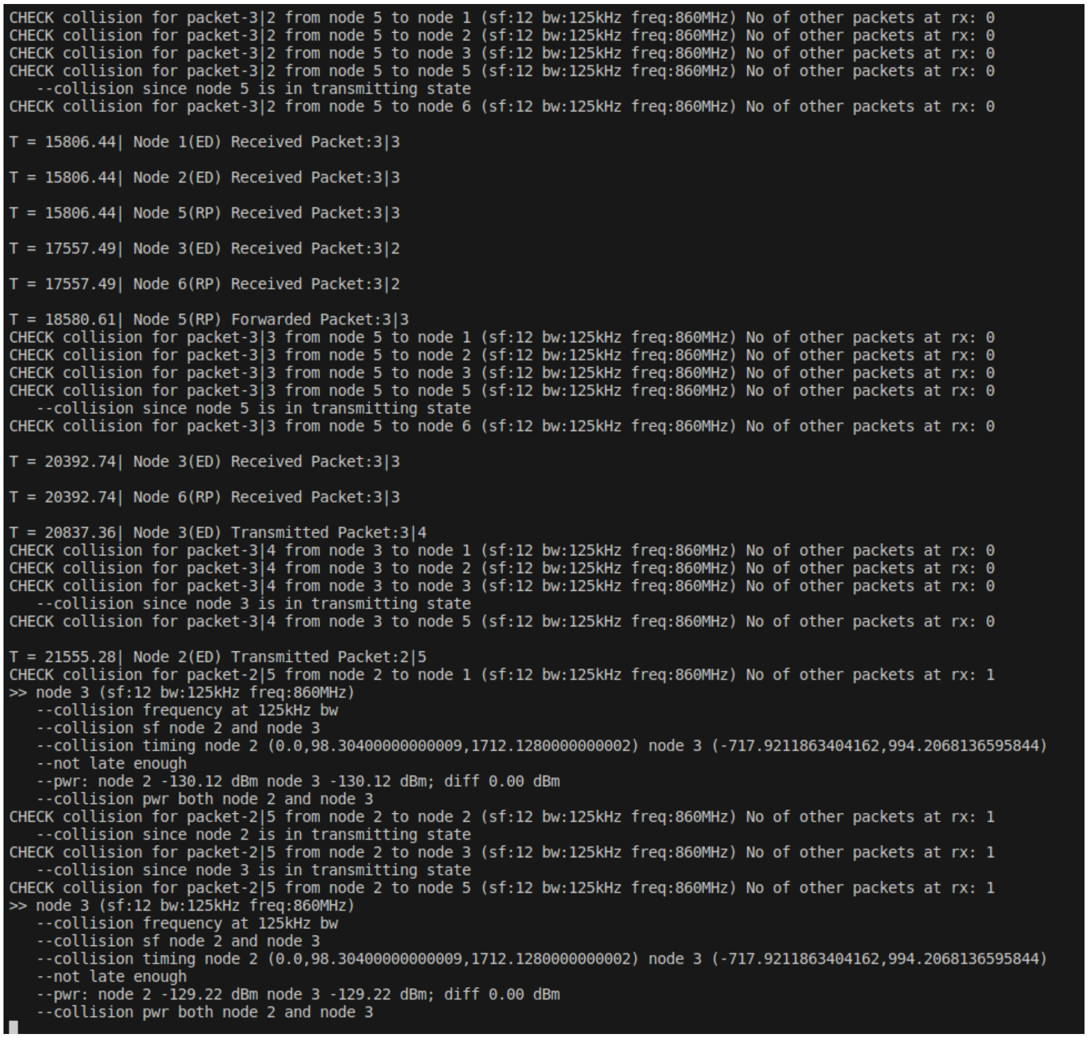


```
5, 450.7953364245076, 356.3850054208274, 336.0030053950287, 0.0]
No of nodes:  41
AvgSendTime (exp. distributed): 60000
Experiment:  4
Full Collision:  False
sent packet seq numbers:  5400

SImulation Time:  183.821854186857 mins
DER: 0.8101851851851852
[530, 545, 525, 509, 539, 567, 553, 543, 574, 515]
[508, 498, 469, 435, 434, 454, 423, 406, 401, 347]

Probabilities of Success
0.9584905660377359
0.9137614678899083
0.8933333333333333
0.8546168958742633
0.8051948051948052
0.800705467372134
0.7649186256781193
0.7476979742173112
0.6986062717770035
0.6737864077669903
Press Enter to continue ...
```

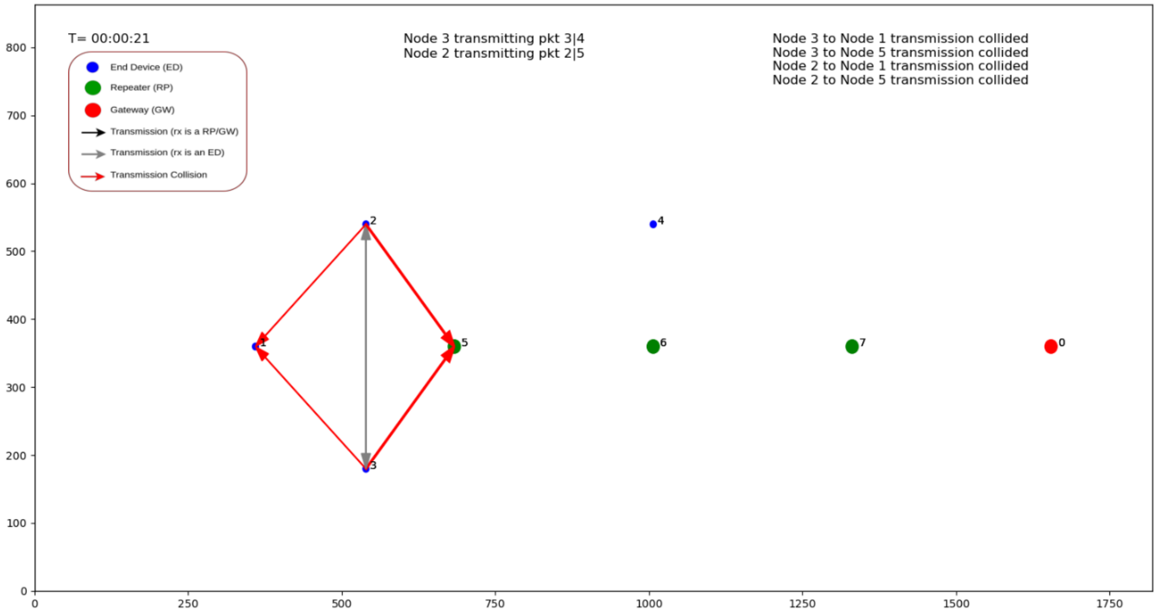


Other than the user logs in the simulation script, the simulator's source code includes a series of well-organized debug logs that can be turned ON by enabling its debug flag. This results in printing logs at each step of the simulation. Those can be used to verify how the packets traversed and in what way the collisions happened. Printing these step-by-step logs is convenient only for small-scale simulations because the amount of these logs increases immensely when the simulation becomes larger.

Example Simulator Debug Logs



```
CHECK collision for packet-3|2 from node 5 to node 1 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|2 from node 5 to node 2 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|2 from node 5 to node 3 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|2 from node 5 to node 5 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
   --collision since node 5 is in transmitting state
CHECK collision for packet-3|2 from node 5 to node 6 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0

T = 15806.44| Node 1(ED) Received Packet:3|3

T = 15806.44| Node 2(ED) Received Packet:3|3

T = 15806.44| Node 5(RP) Received Packet:3|3

T = 17557.49| Node 3(ED) Received Packet:3|2

T = 17557.49| Node 6(RP) Received Packet:3|2

T = 18580.61| Node 5(RP) Forwarded Packet:3|3
CHECK collision for packet-3|3 from node 5 to node 1 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|3 from node 5 to node 2 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|3 from node 5 to node 3 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|3 from node 5 to node 5 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
   --collision since node 5 is in transmitting state
CHECK collision for packet-3|3 from node 5 to node 6 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0

T = 20392.74| Node 3(ED) Received Packet:3|3

T = 20392.74| Node 6(RP) Received Packet:3|3

T = 20837.36| Node 3(ED) Transmitted Packet:3|4
CHECK collision for packet-3|4 from node 3 to node 1 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|4 from node 3 to node 2 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
CHECK collision for packet-3|4 from node 3 to node 3 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0
   --collision since node 3 is in transmitting state
CHECK collision for packet-3|4 from node 3 to node 5 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 0

T = 21555.28| Node 2(ED) Transmitted Packet:2|5
CHECK collision for packet-2|5 from node 2 to node 1 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 1
>> node 3 (sf:12 bw:125kHz freq:860MHz)
   --collision frequency at 125kHz bw
   --collision sf node 2 and node 3
   --collision timing node 2 (0.0,98.30400000000009,1712.1280000000002) node 3 (-717.9211863404162,994.2068136595844)
   --not late enough
   --pwr: node 2 -130.12 dBm node 3 -130.12 dBm; diff 0.00 dBm
   --collision pwr both node 2 and node 3
CHECK collision for packet-2|5 from node 2 to node 2 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 1
   --collision since node 2 is in transmitting state
CHECK collision for packet-2|5 from node 2 to node 3 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 1
   --collision since node 3 is in transmitting state
CHECK collision for packet-2|5 from node 2 to node 5 (sf:12 bw:125kHz freq:860MHz) No of other packets at rx: 1
>> node 3 (sf:12 bw:125kHz freq:860MHz)
   --collision frequency at 125kHz bw
   --collision sf node 2 and node 3
   --collision timing node 2 (0.0,98.30400000000009,1712.1280000000002) node 3 (-717.9211863404162,994.2068136595844)
   --not late enough
   --pwr: node 2 -129.22 dBm node 3 -129.22 dBm; diff 0.00 dBm
   --collision pwr both node 2 and node 3
```

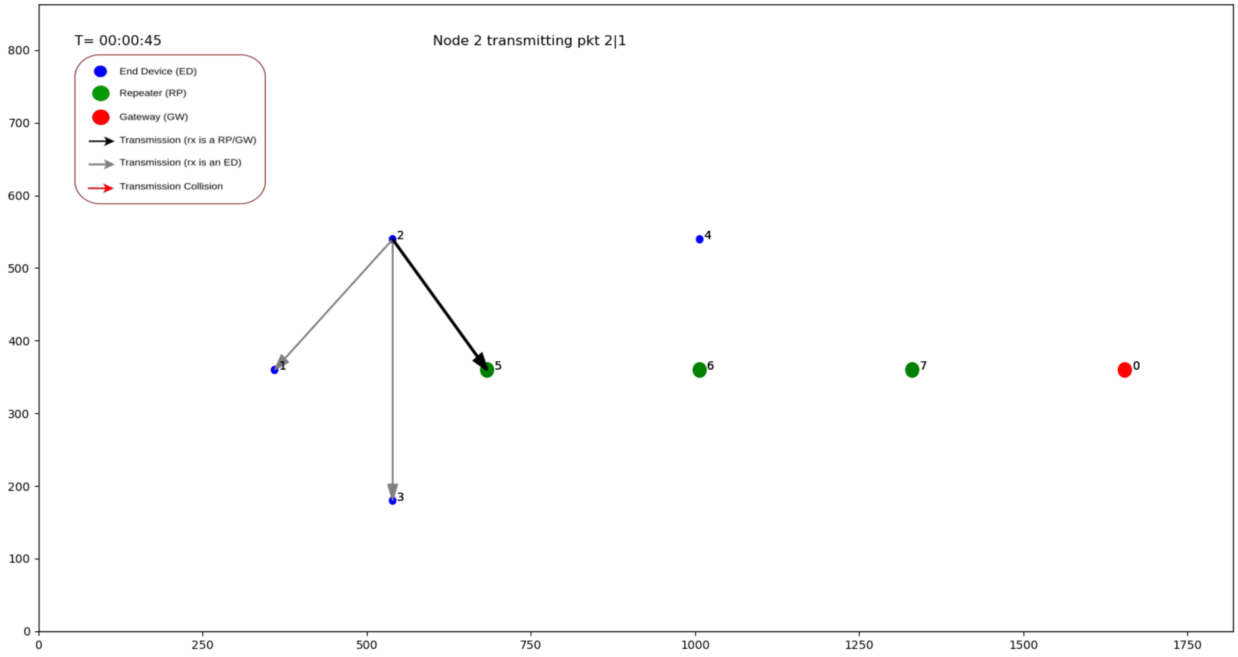

T= 00:00:21
Node 3 transmitting pkt 3|4
Node 2 transmitting pkt 2|5
Node 3 to Node 1 transmission collided
Node 3 to Node 5 transmission collided
Node 2 to Node 1 transmission collided
Node 2 to Node 5 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
800
700
600
500
400
300
200
100
0
0
250
500
750
1000
1250
1500
1750

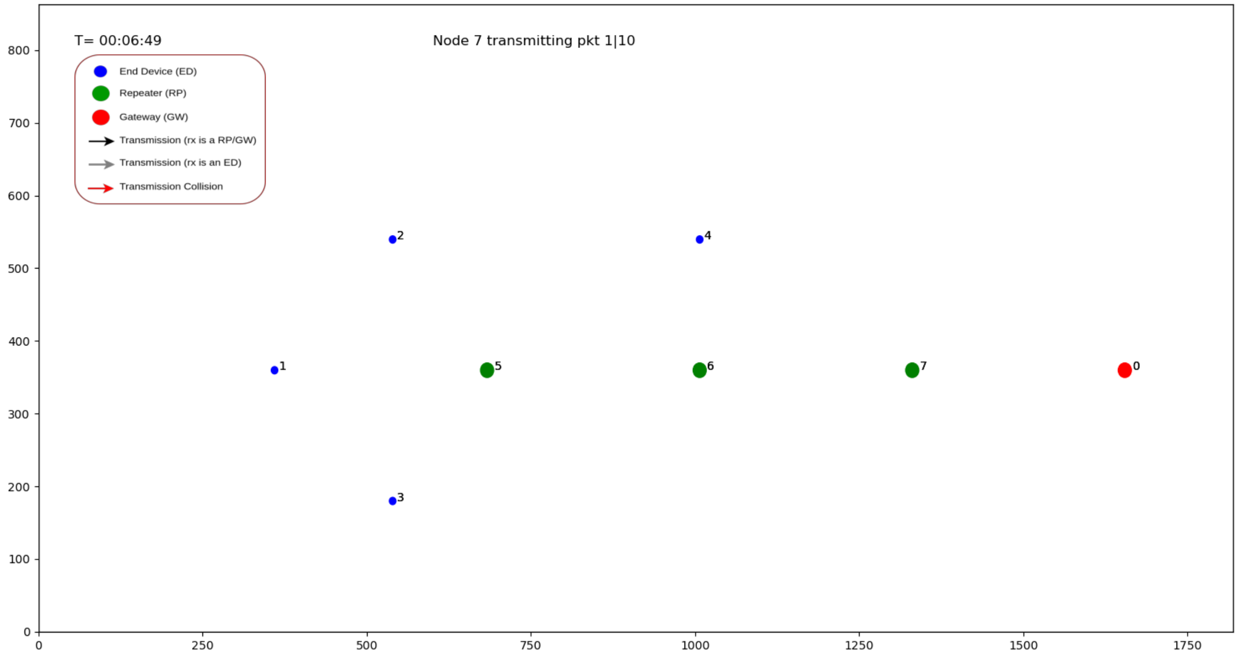
Matplotlib is the library used to graphically display the simulations. There are 2 flags in the simulator named *'graphics'* and *'realtime_graphics'*. Enabling only the *'graphics'* flag will show a single cartesian plot of the nodes, which sufficiently illustrates how the nodes are placed. It is advisable to enable only *'graphics'* flag when simulating large networks because there can be millions of steps in such a network.

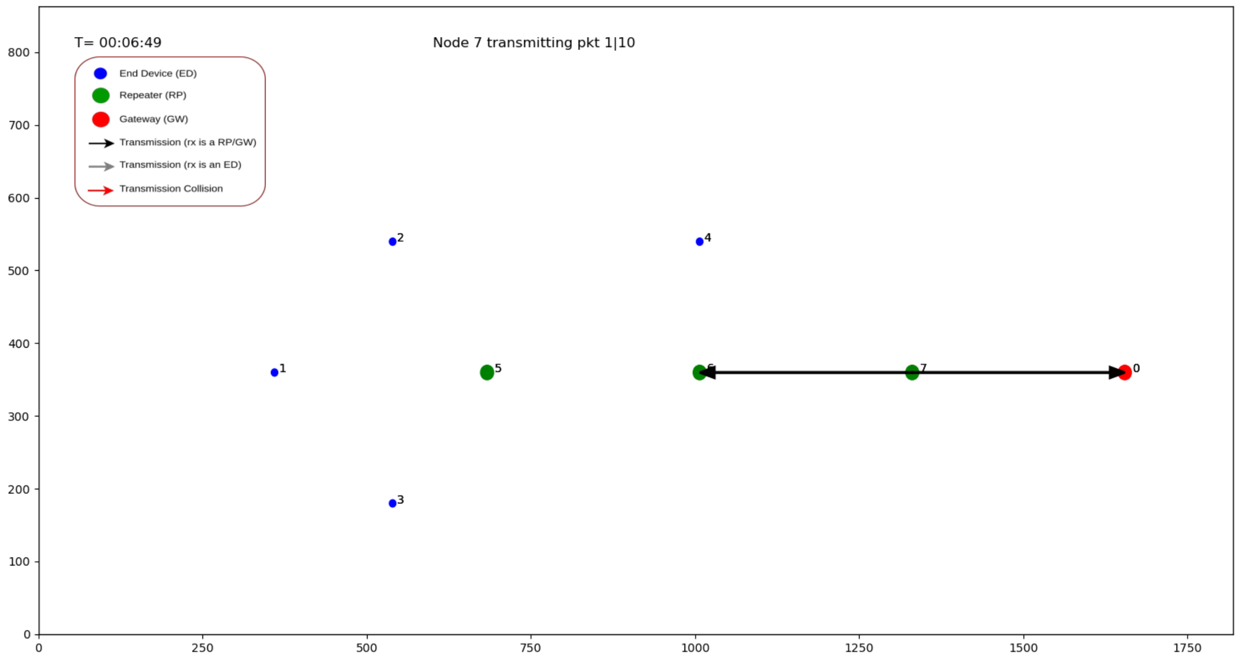
The flag *'realtime_graphics'* can be enabled to plot each step of LoRa transmissions in the simulation. These plots can be shown as a slide show by assigning the variable *'slideShowPause'* with the number of seconds for the slide transition period. Assigning 0 to *'slideShowPause'* will enable the plot transitions using mouse clicks or key presses.

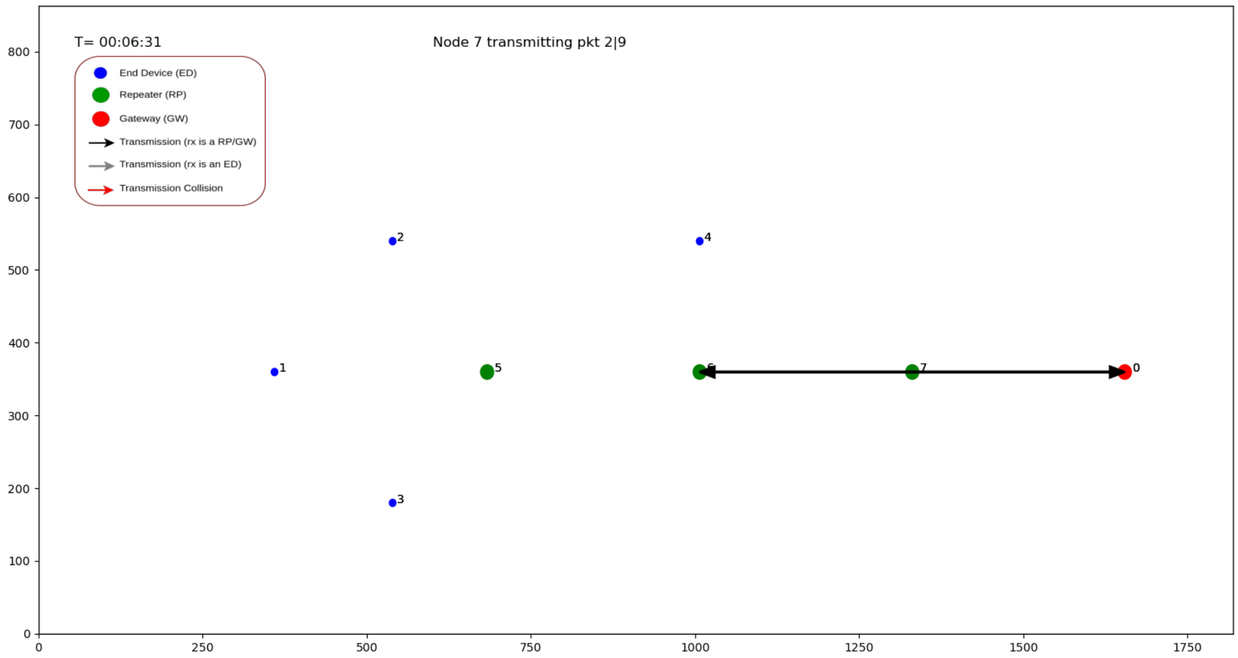
Following are two sets of such real-time plots generated by two simulations. It can be clearly observed that there are fewer collisions in Simulation-1 compared to Simulation-2 because its packet transmitting period is set with a higher mean value.

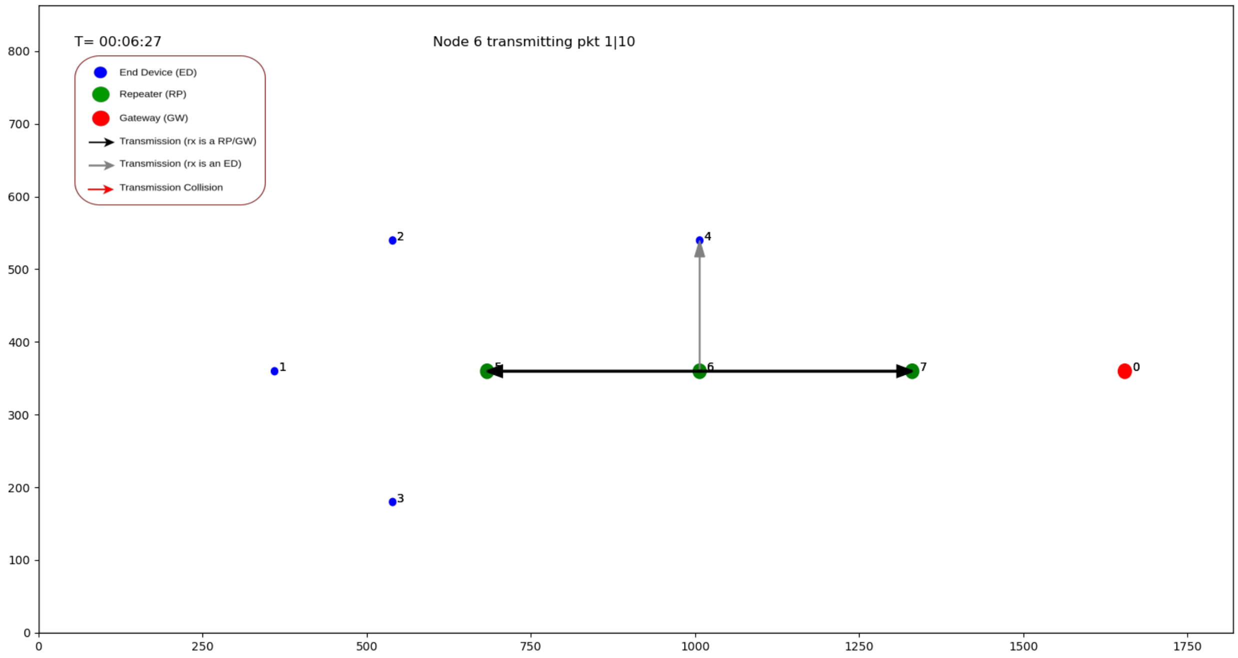
Simulation 1

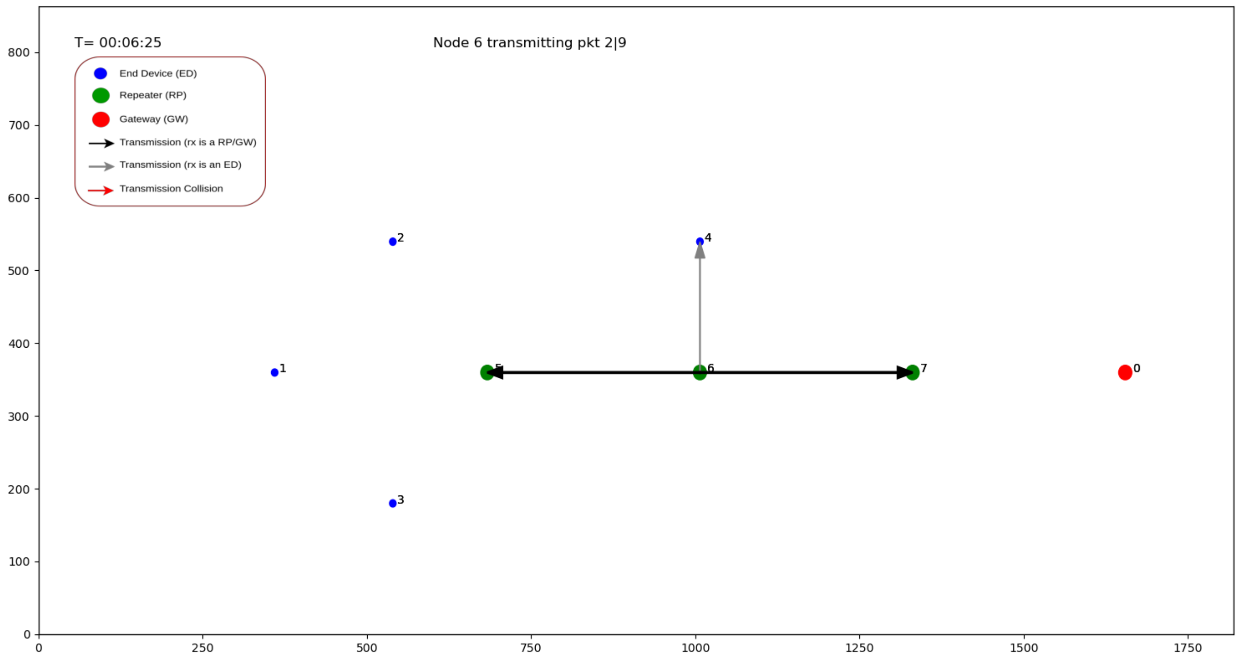


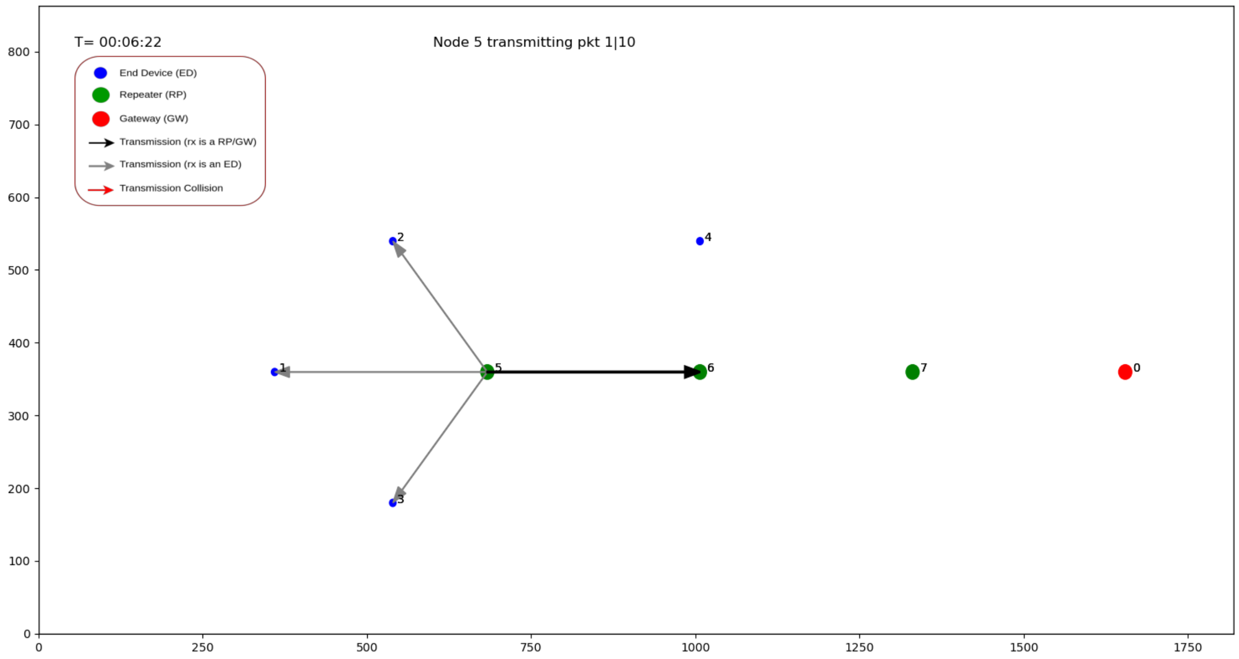


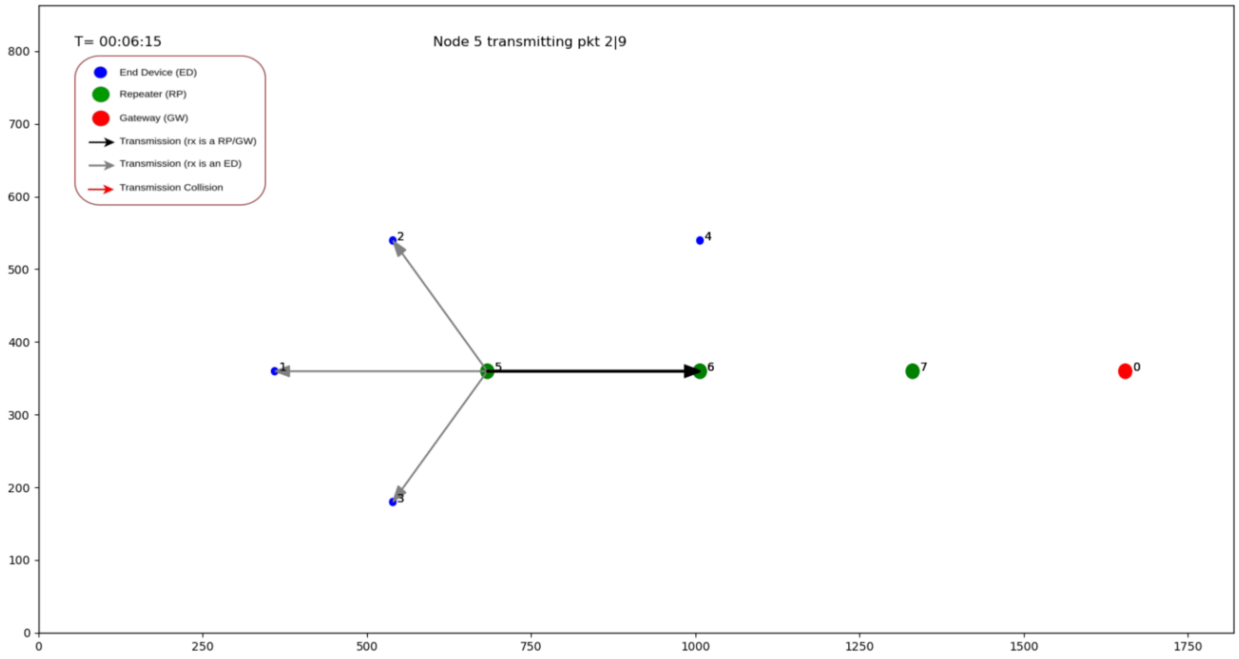

T= 00:06:49
Node 7 transmitting pkt 1|10
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

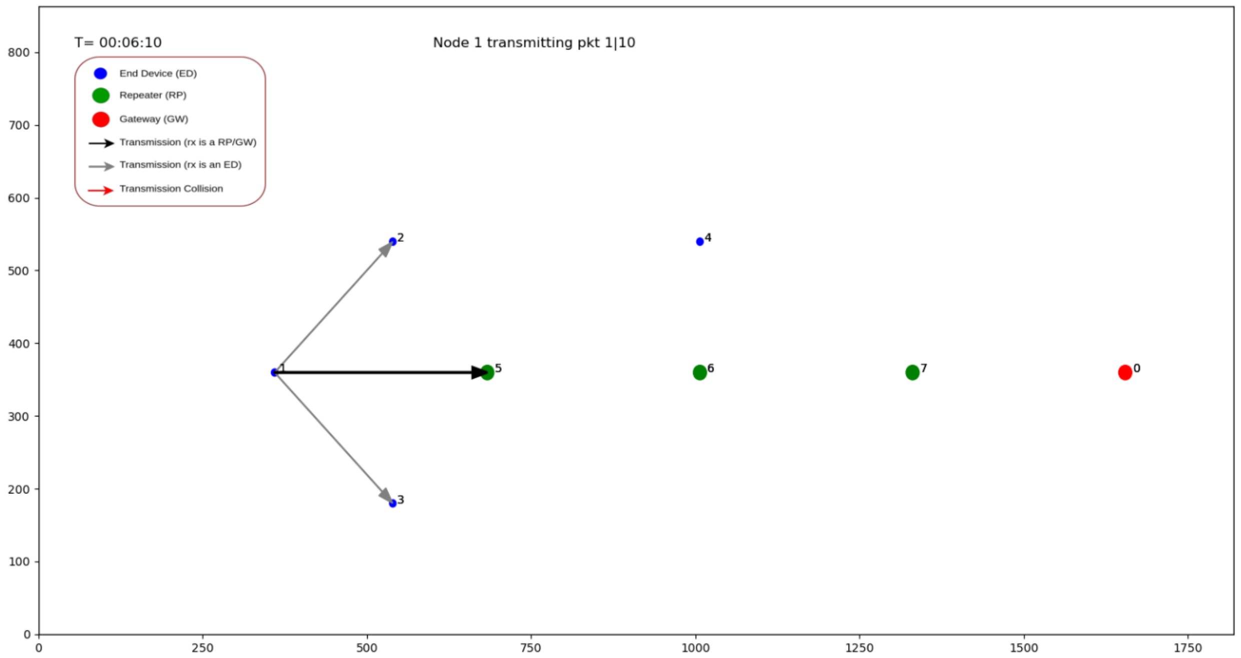

T= 00:06:31
Node 7 transmitting pkt 2|9
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

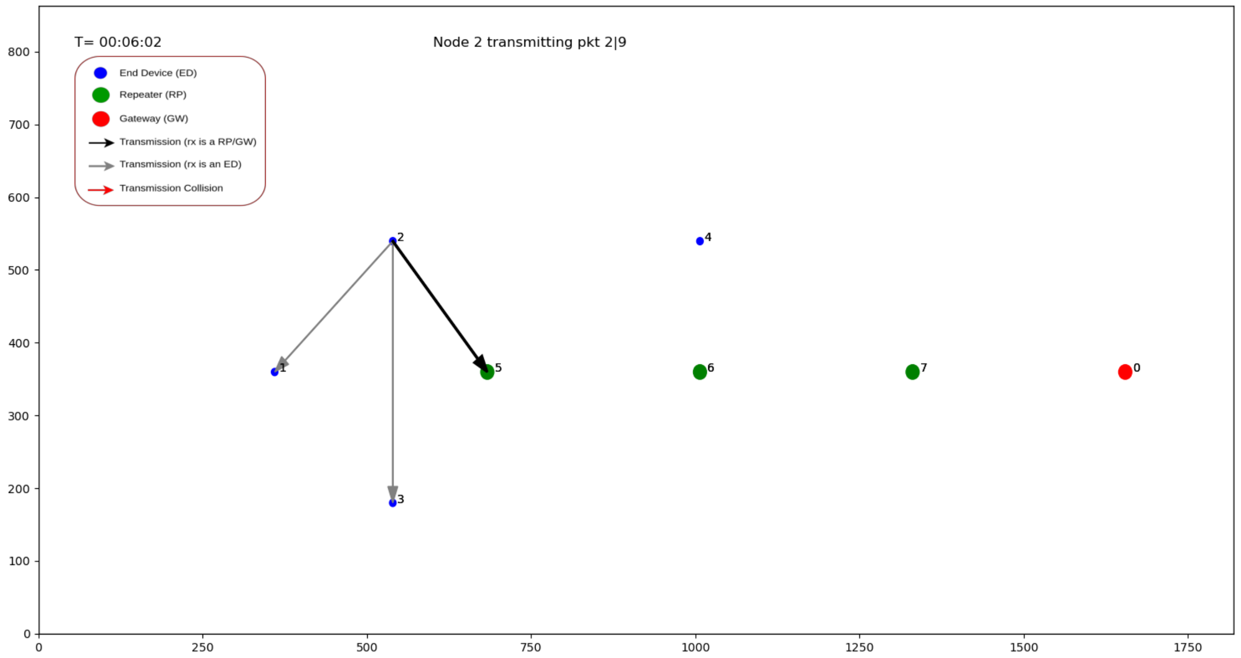

T= 00:06:27
Node 6 transmitting pkt 1|10
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

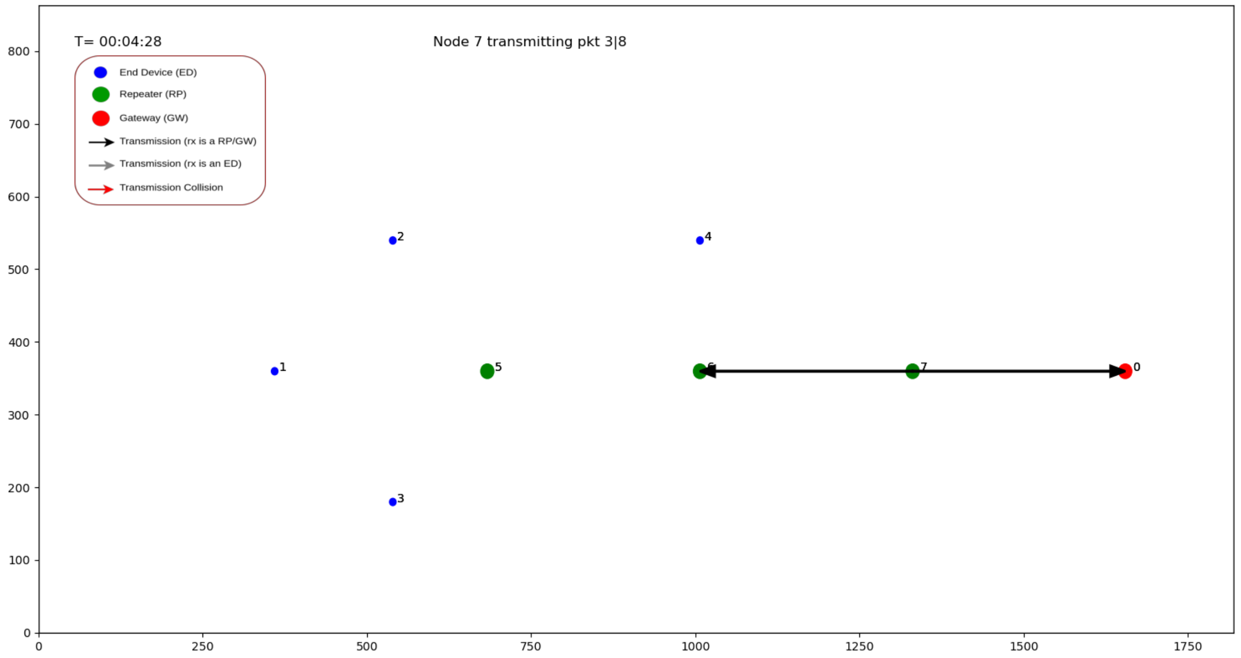

T= 00:06:25
Node 6 transmitting pkt 2|9
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750
1
2
3
4
5
6
7
0

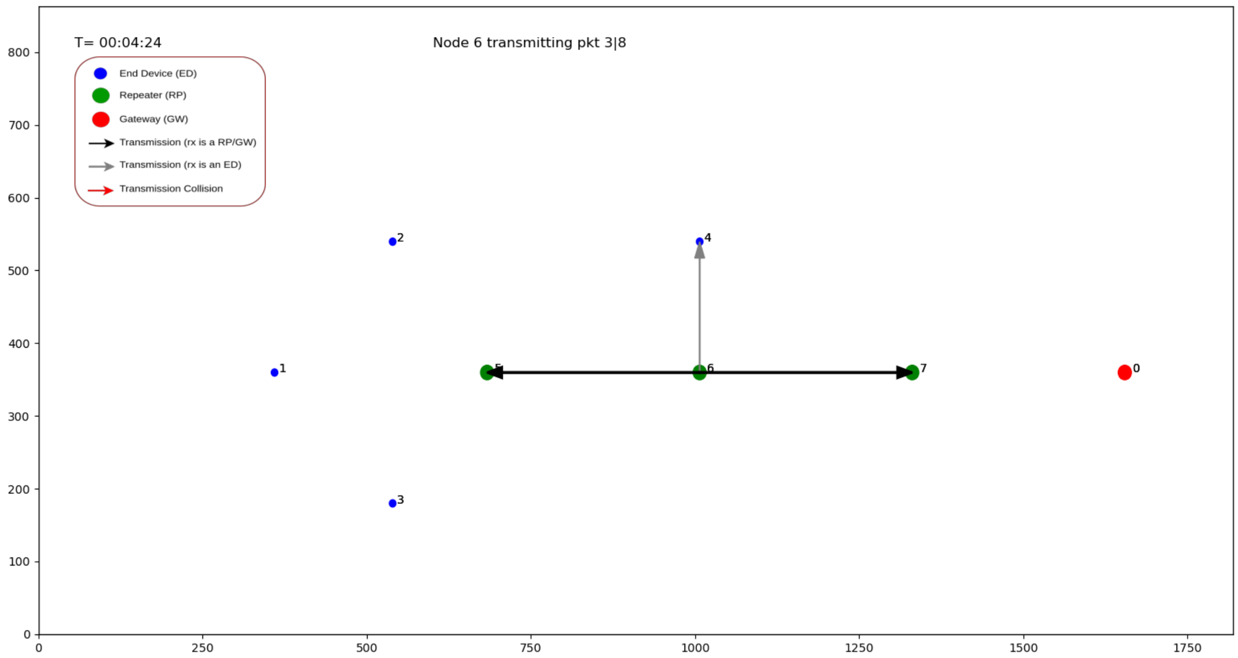

T= 00:06:22
Node 5 transmitting pkt 1|10
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750
1
2
3
4
5
6
7
0

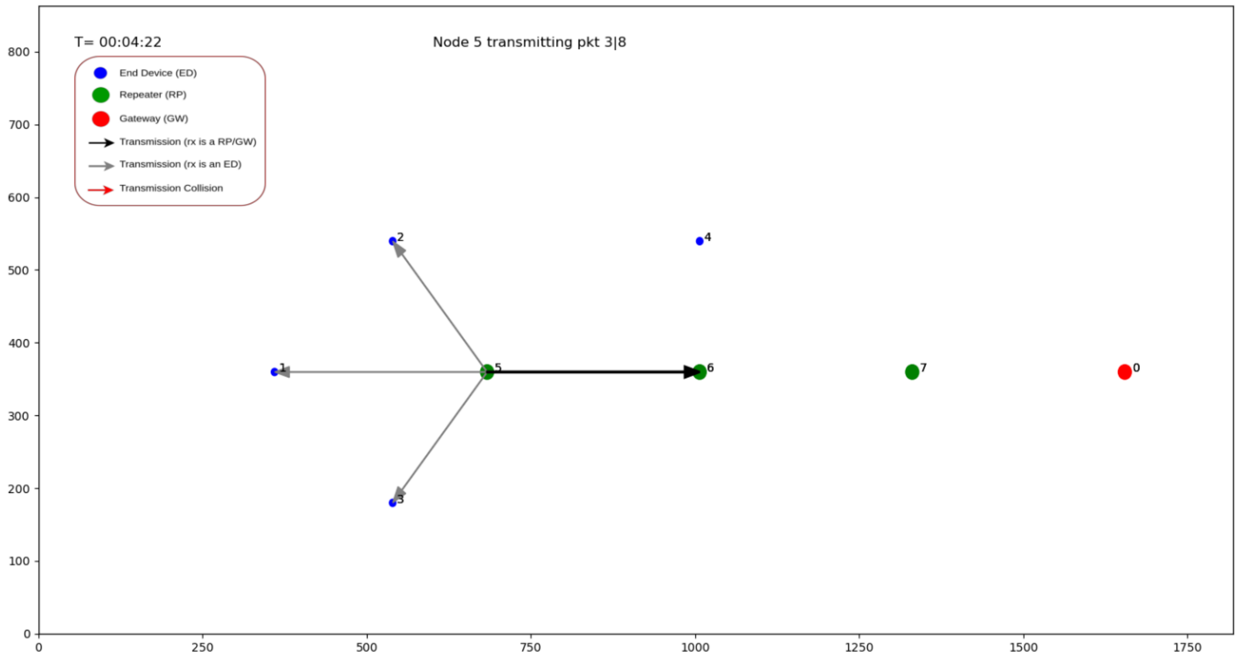

T= 00:06:15
Node 5 transmitting pkt 2|9
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750
1
2
3
4
5
6
7
0

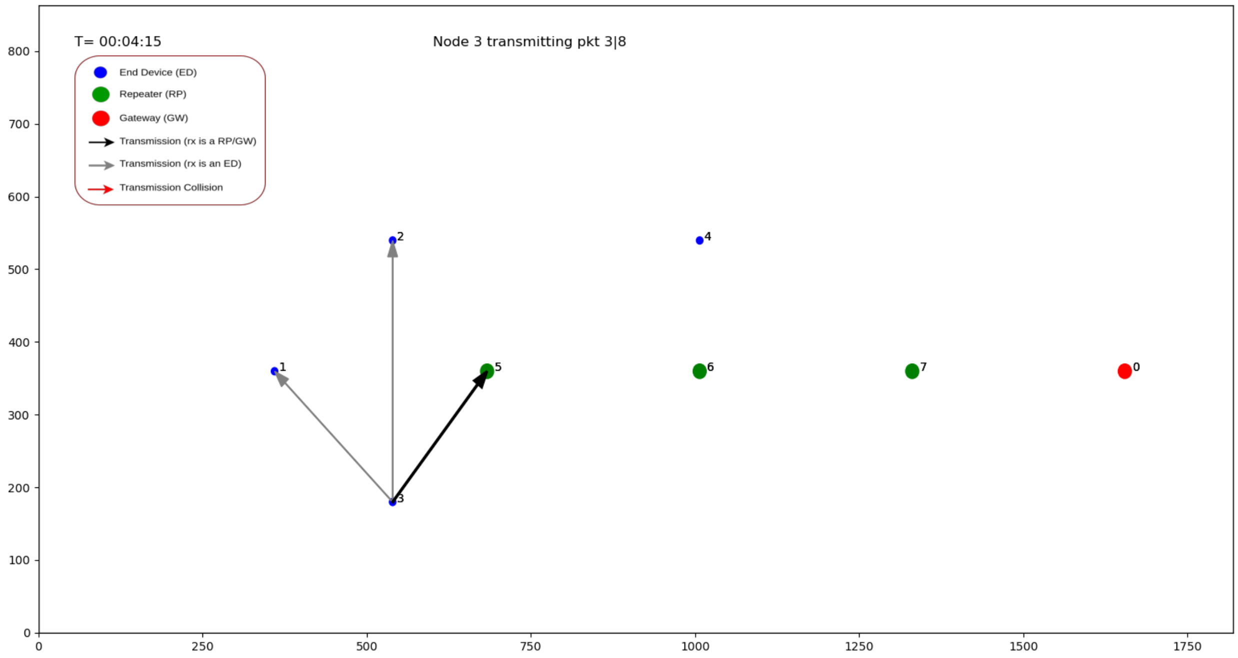

T= 00:06:10
Node 1 transmitting pkt 1|10
● End Device (ED)
● Repeater (RP)
● Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

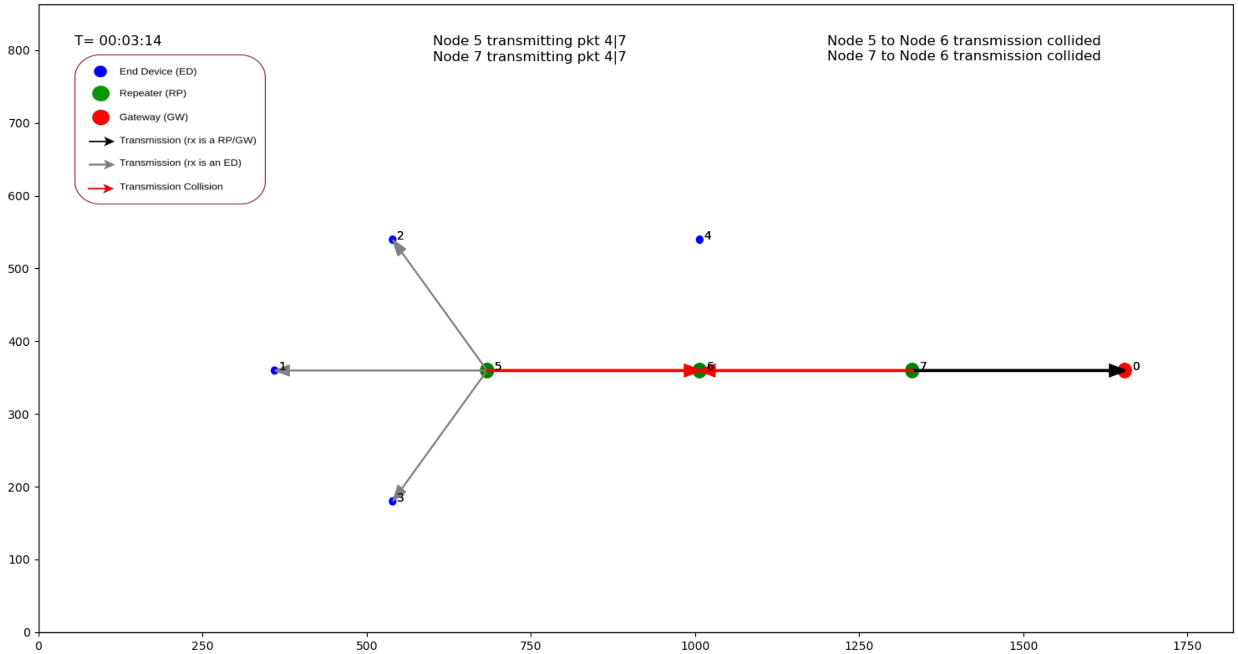

T= 00:06:02
Node 2 transmitting pkt 2|9
● End Device (ED)
● Repeater (RP)
● Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

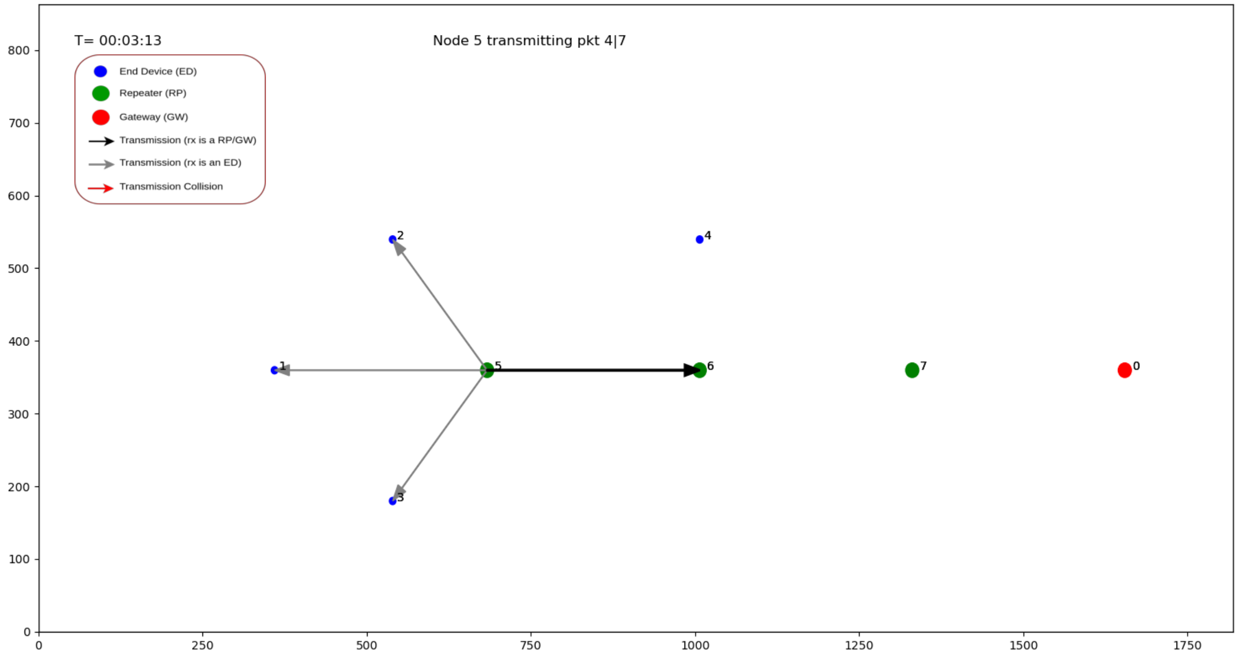

T= 00:04:28
Node 7 transmitting pkt 3|8
● End Device (ED)
● Repeater (RP)
● Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

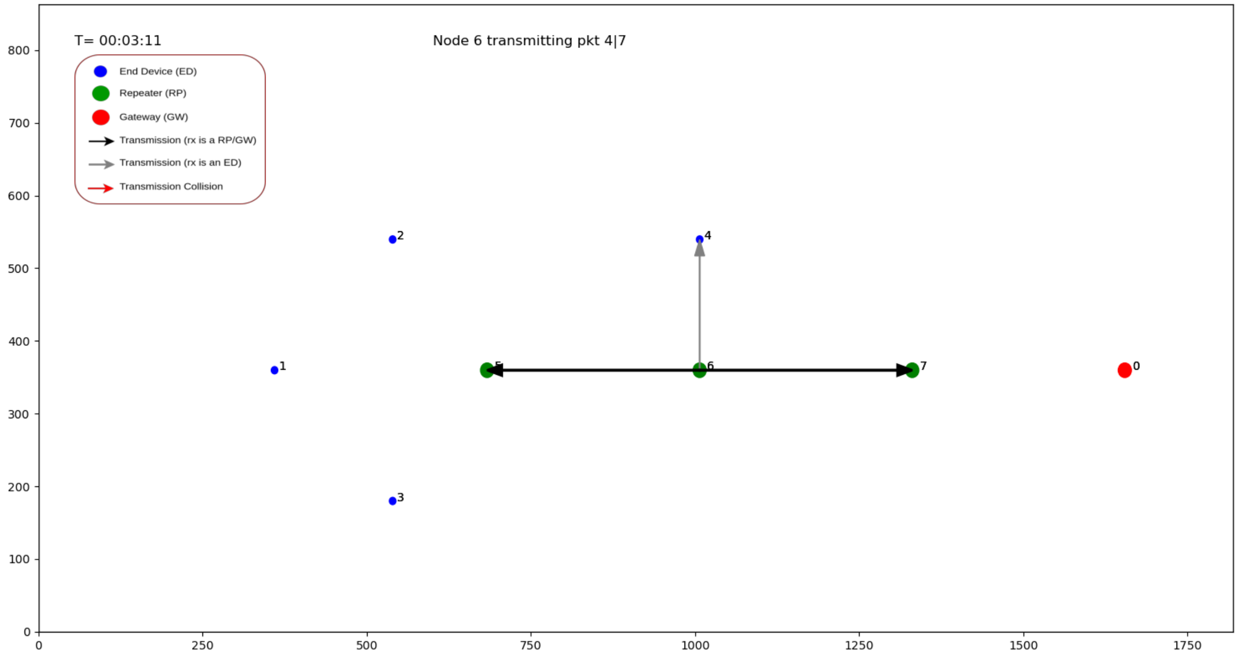

T= 00:04:24
Node 6 transmitting pkt 3|8
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

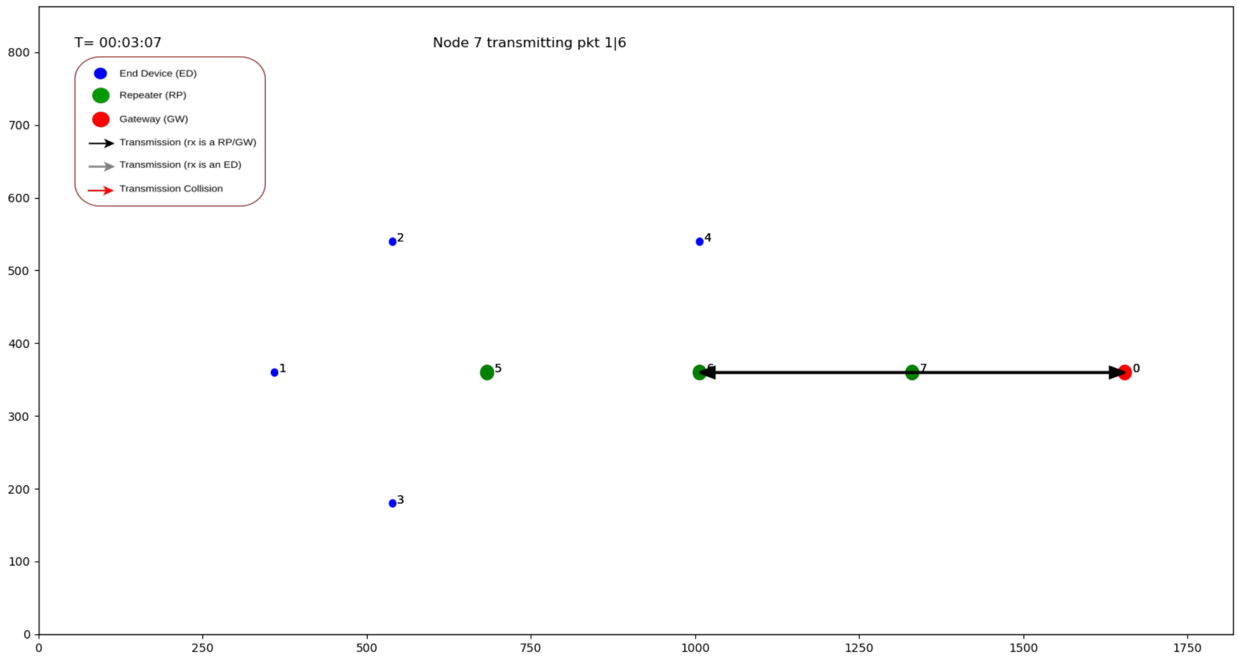

T= 00:04:22
Node 5 transmitting pkt 3|8
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

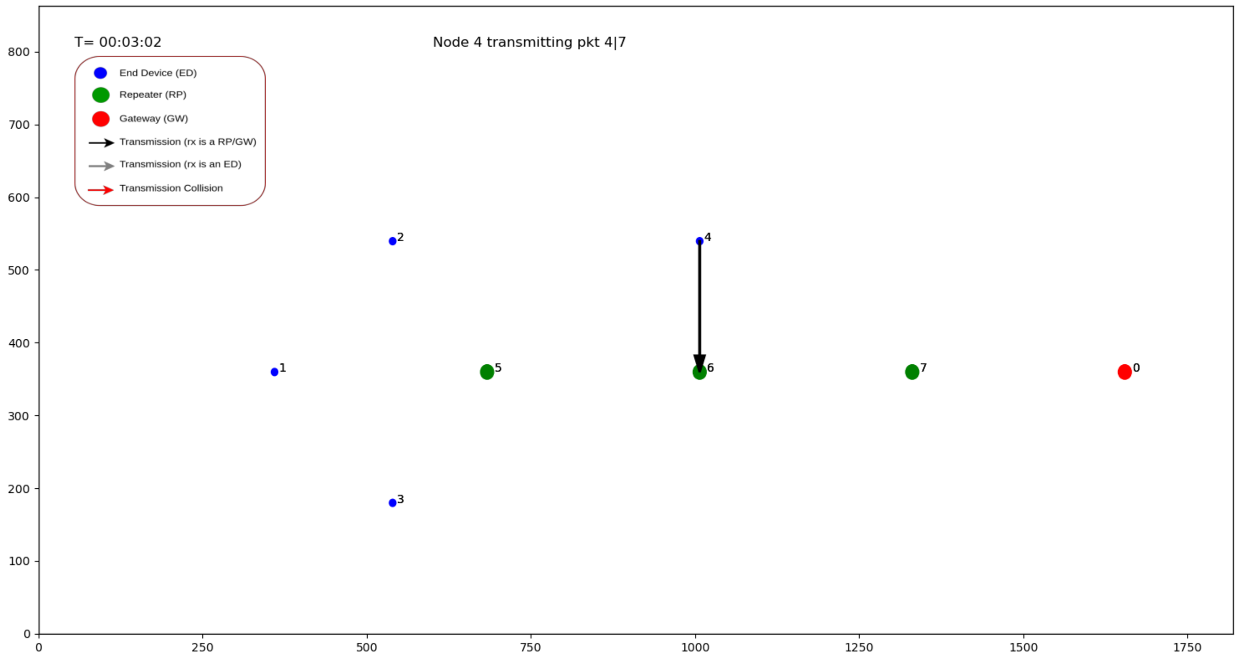

T= 00:04:15
Node 3 transmitting pkt 3|8
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

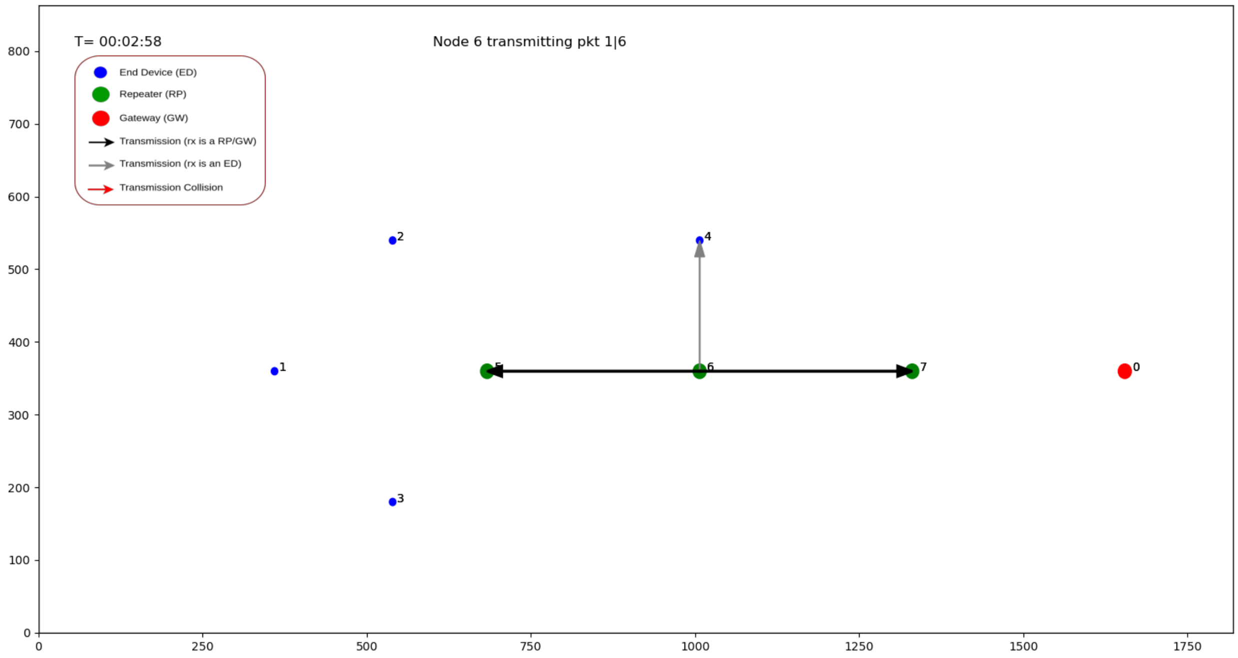

T= 00:03:14
Node 5 transmitting pkt 4|7
Node 7 transmitting pkt 4|7
Node 5 to Node 6 transmission collided
Node 7 to Node 6 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

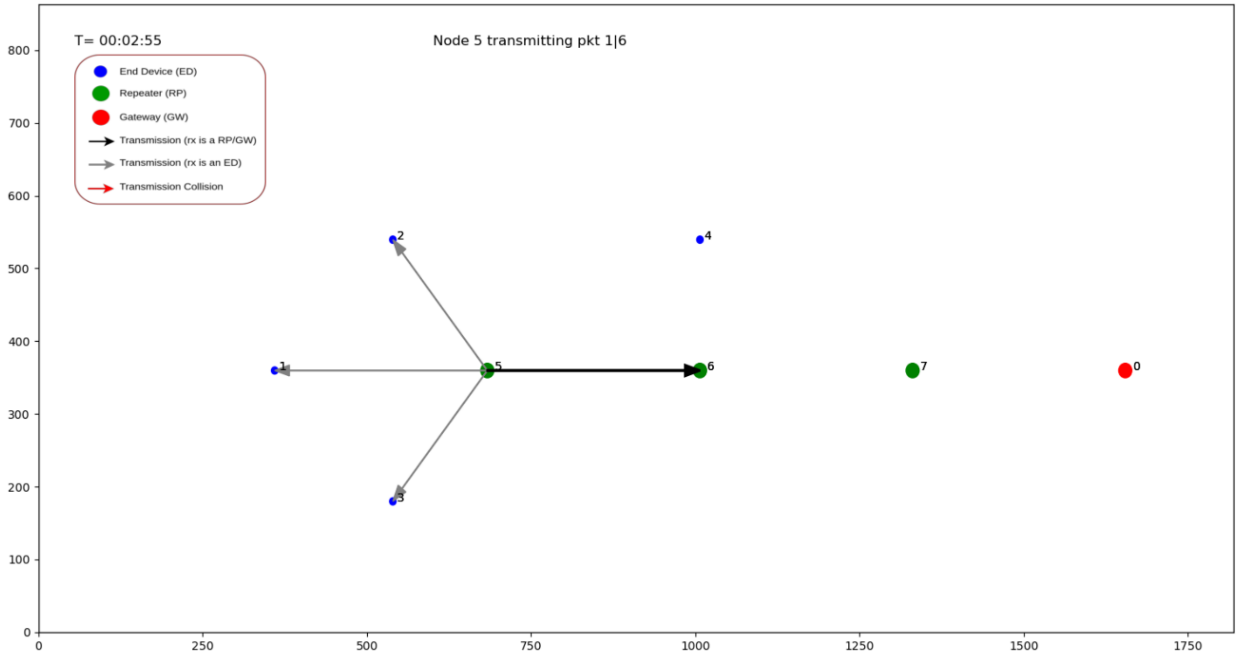

T= 00:03:13
Node 5 transmitting pkt 4|7
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

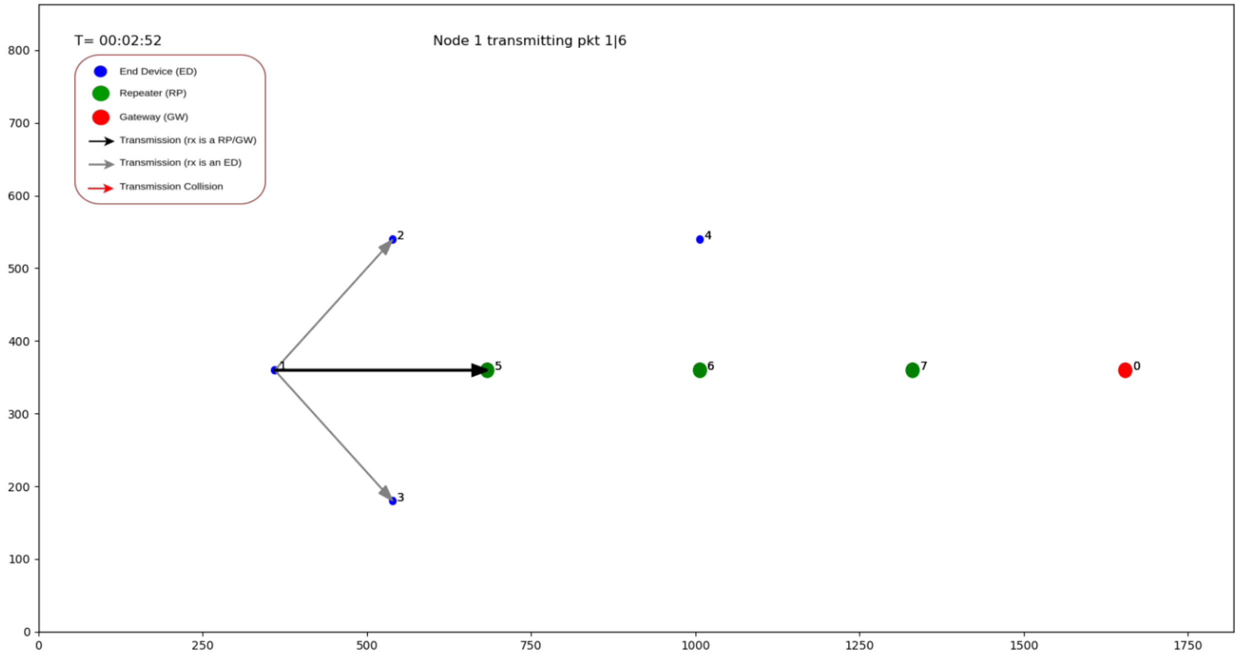

T= 00:03:11
Node 6 transmitting pkt 4|7
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

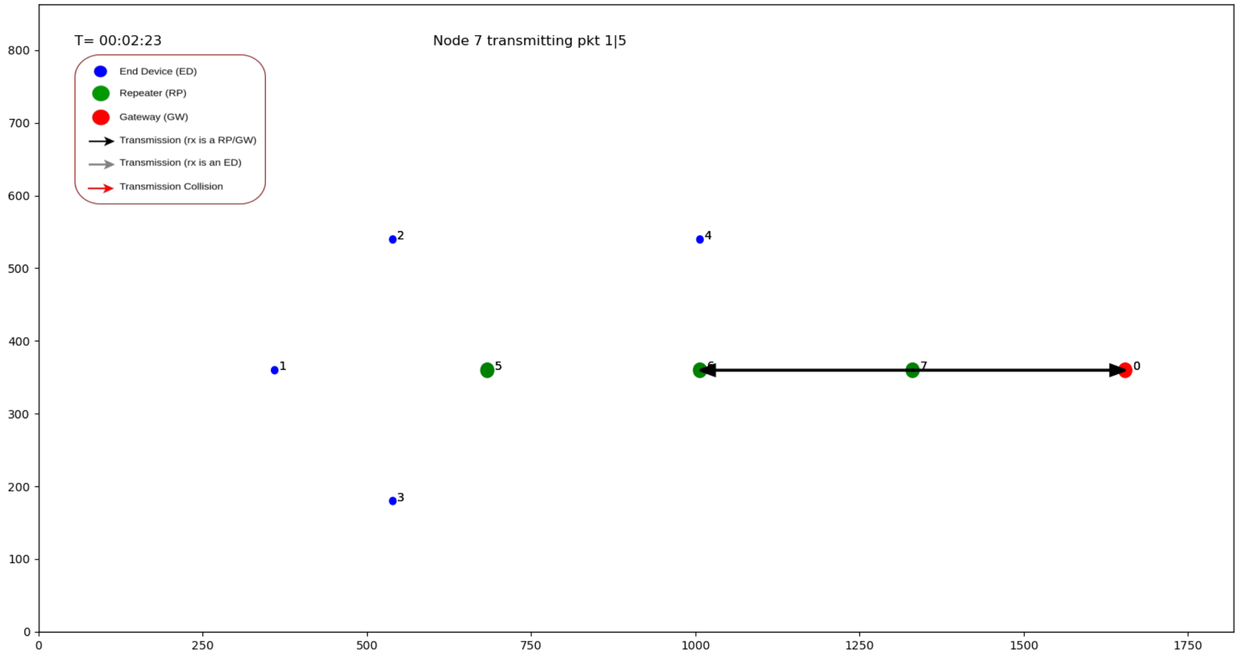

T= 00:03:07
Node 7 transmitting pkt 1|6
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

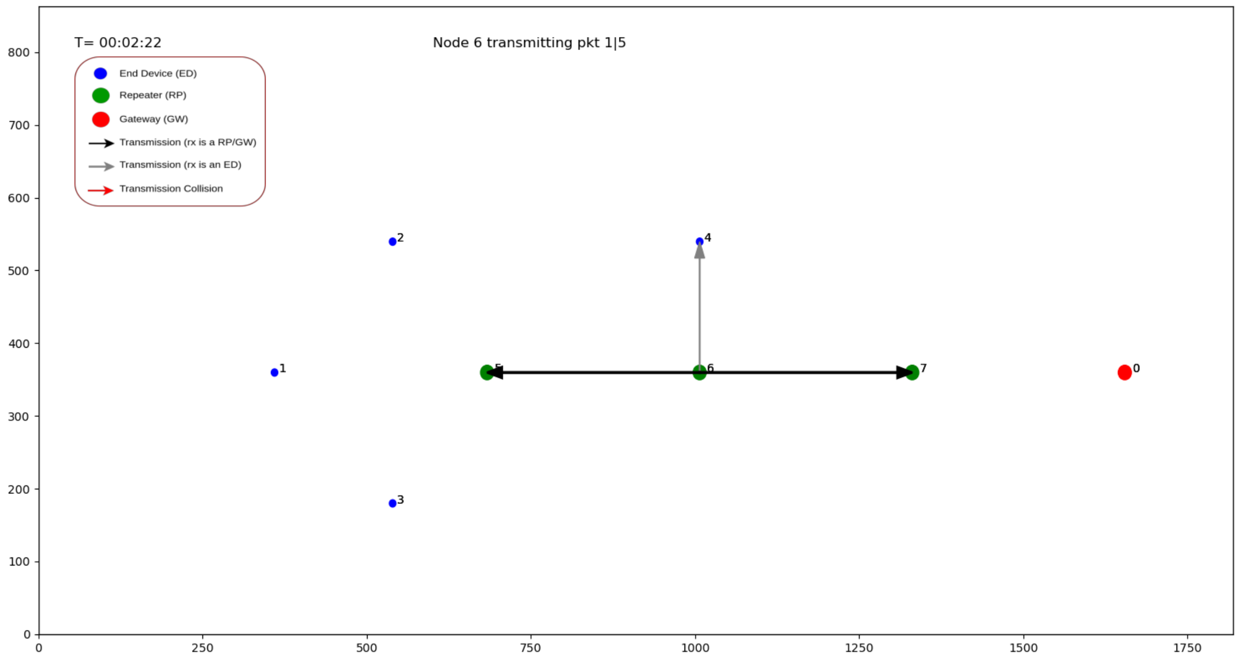

T= 00:03:02
Node 4 transmitting pkt 4|7
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:02:58
Node 6 transmitting pkt 1|6
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

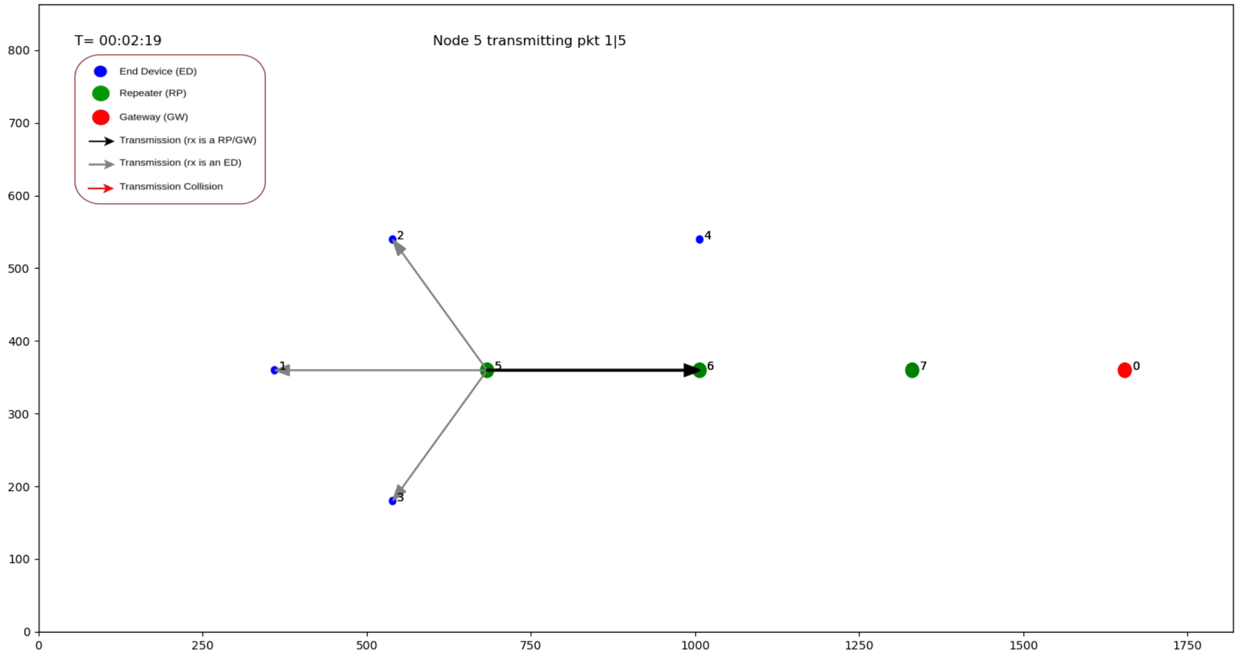

T= 00:02:55
Node 5 transmitting pkt 1|6
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:02:52
Node 1 transmitting pkt 1|6
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

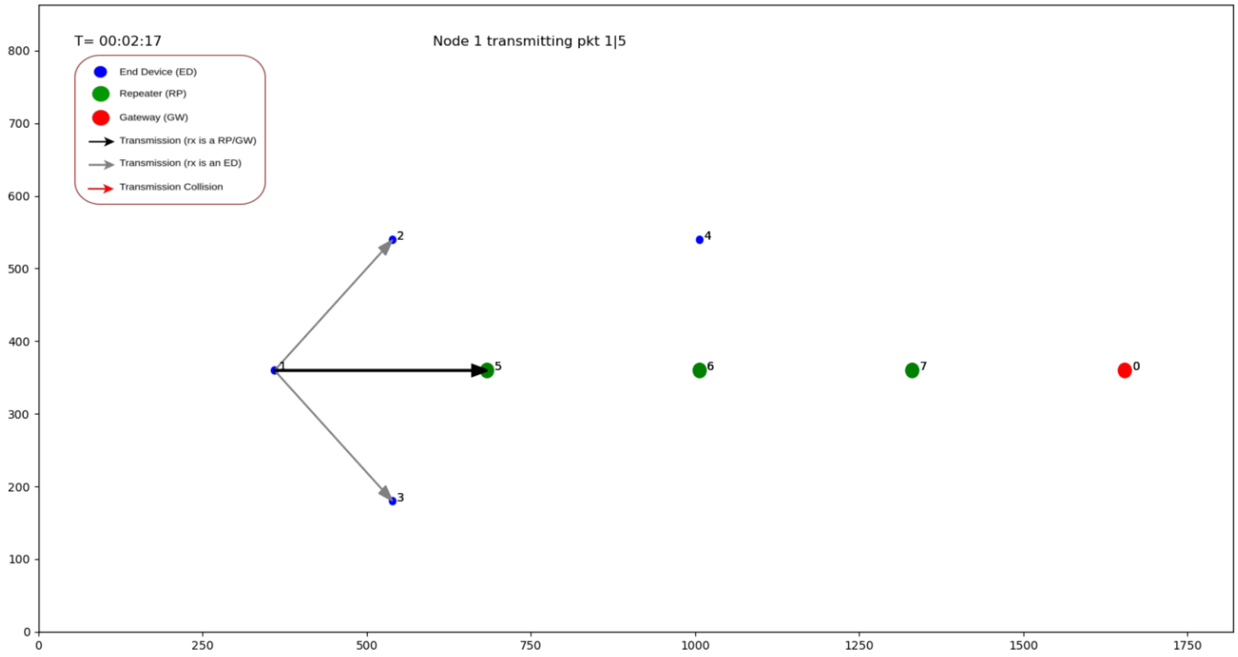

T= 00:02:23
Node 7 transmitting pkt 1|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750



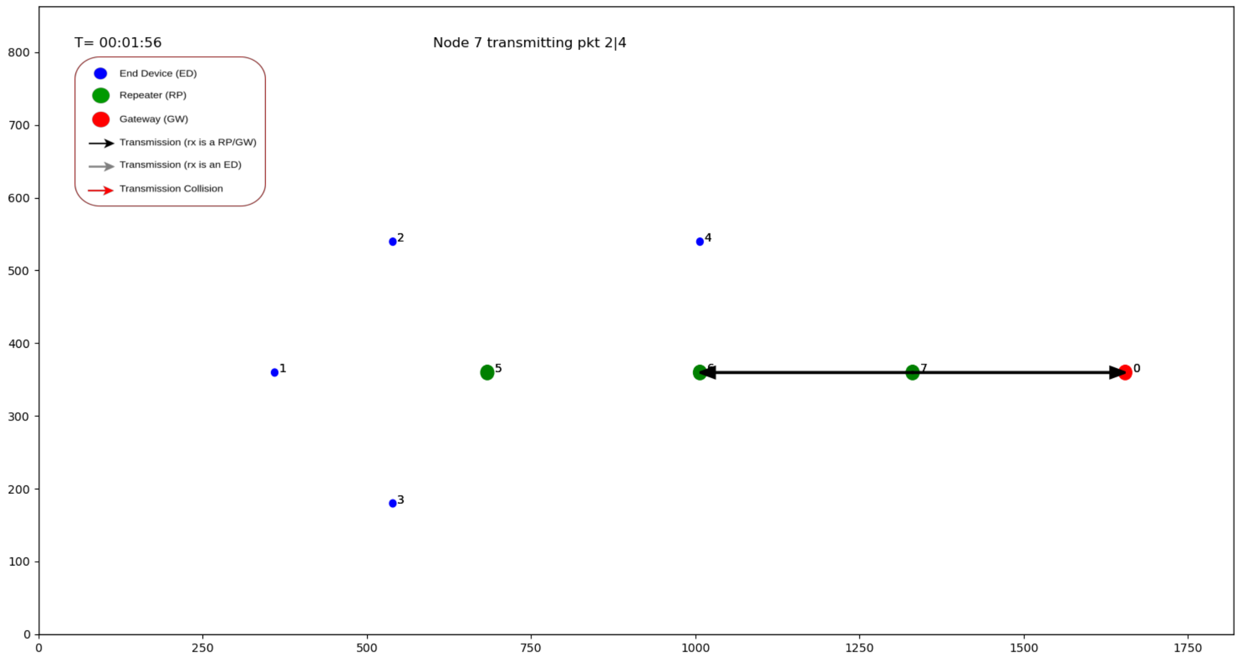
T= 00:02:22
Node 6 transmitting pkt 1|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

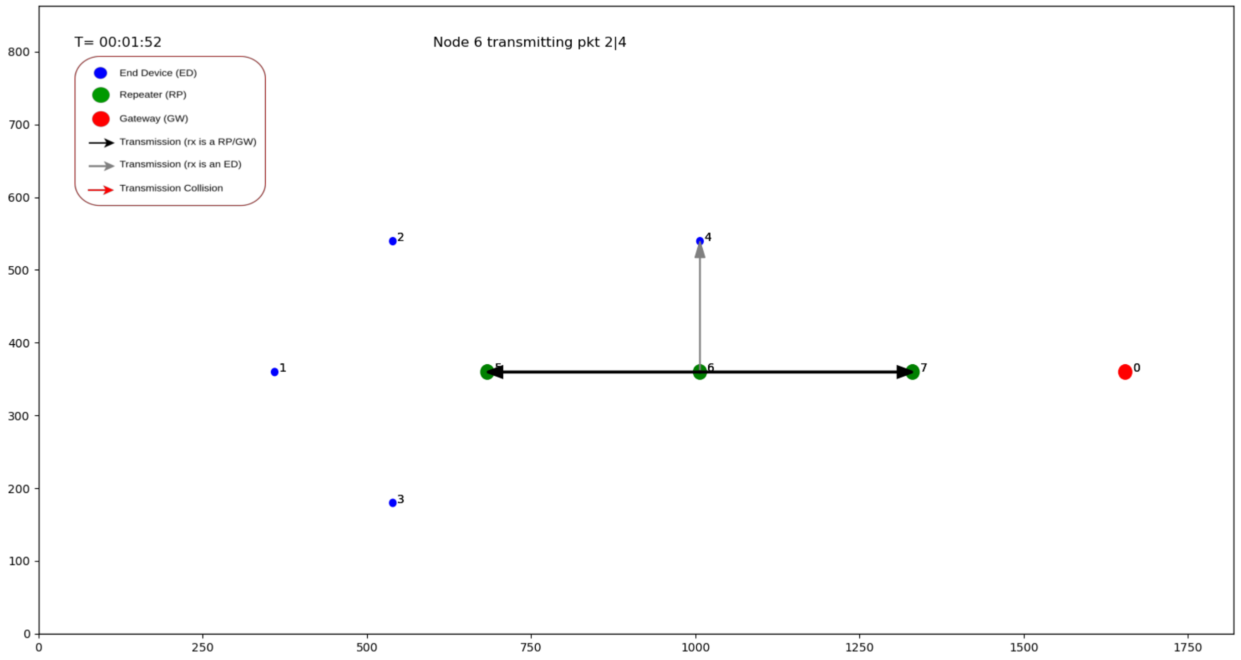


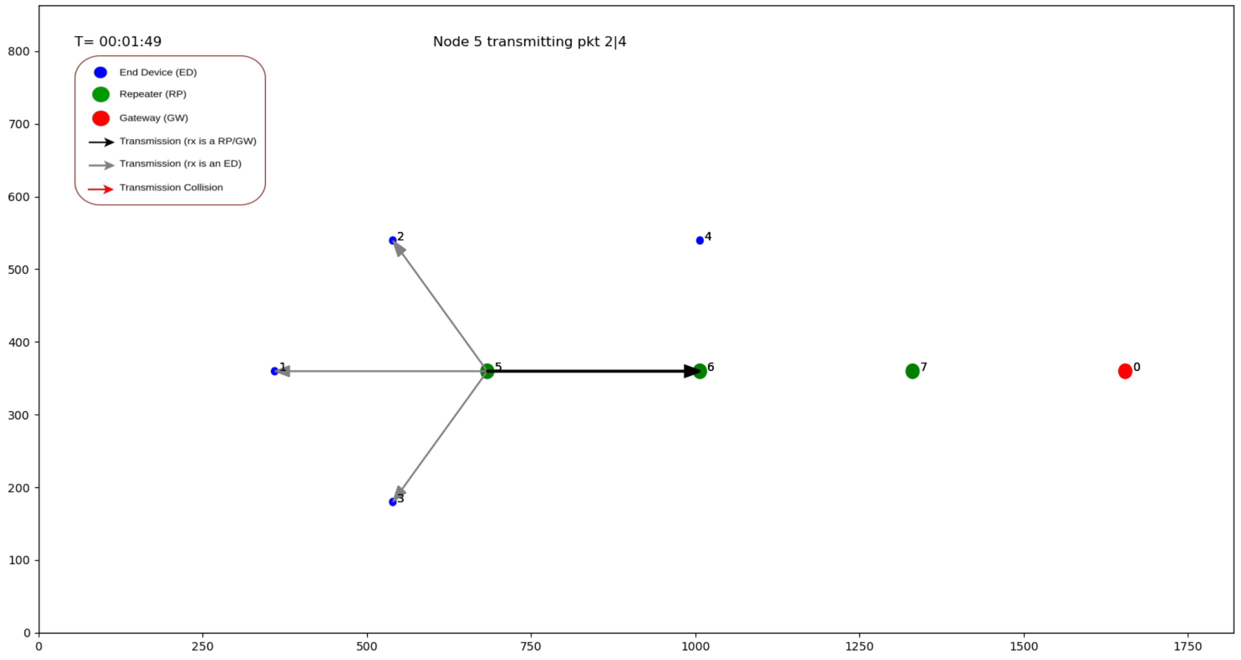
T= 00:02:19
Node 5 transmitting pkt 1|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

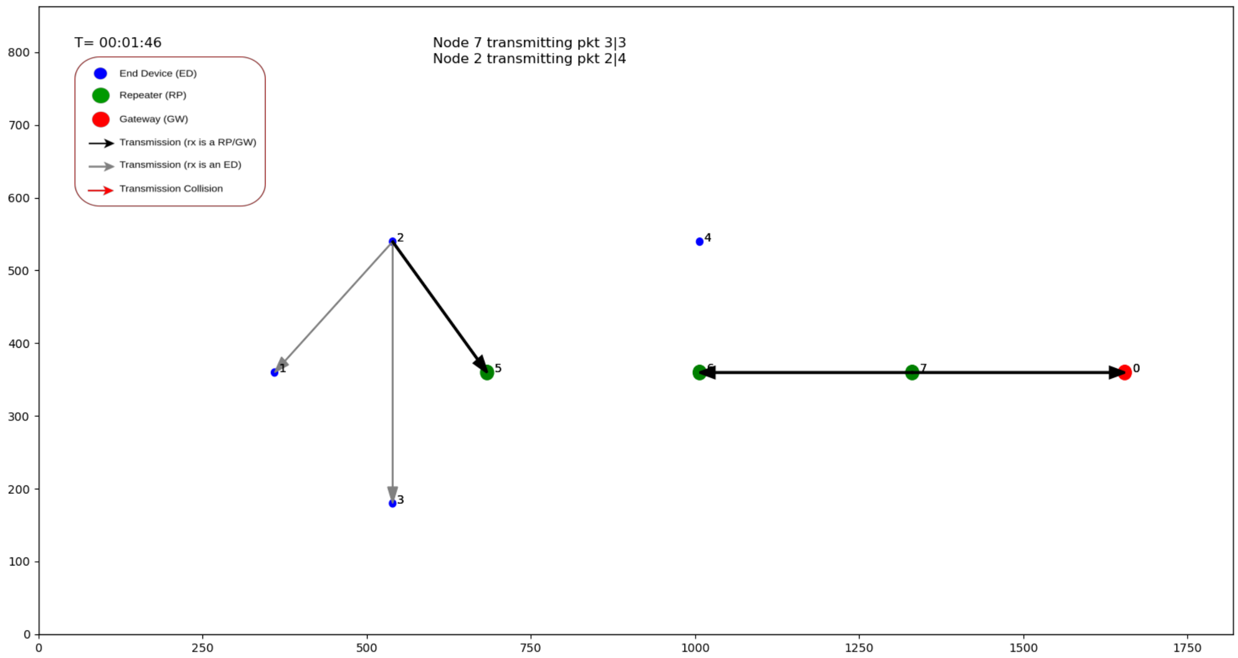


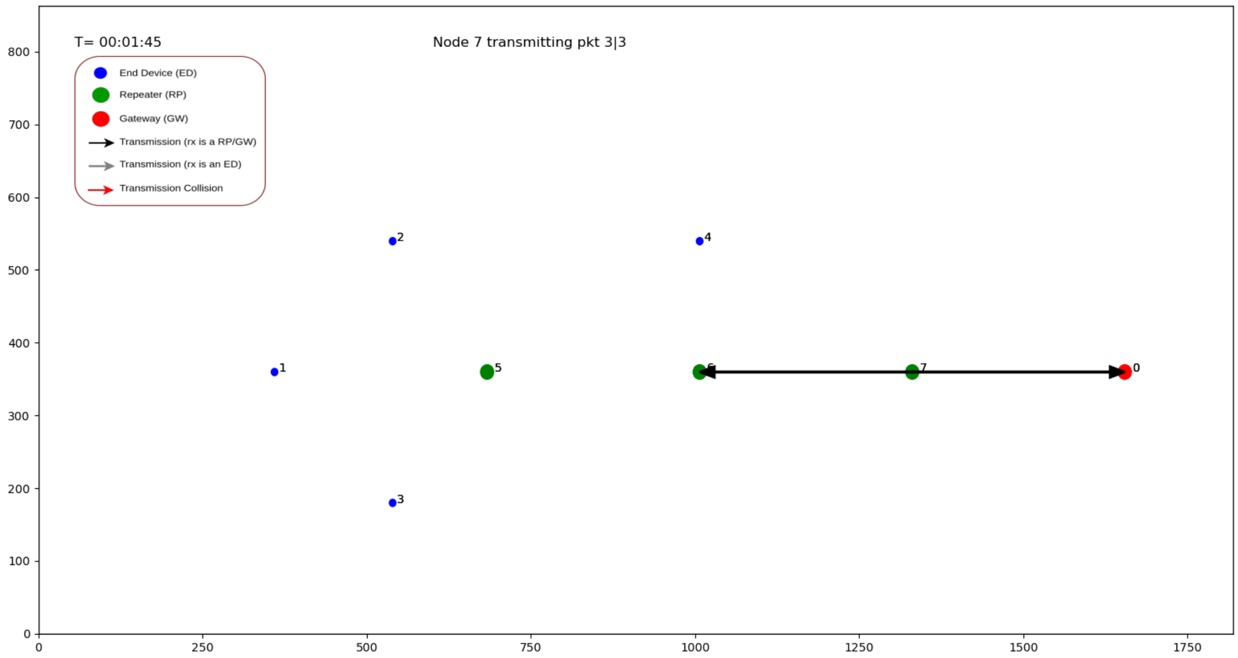
T= 00:02:17
Node 1 transmitting pkt 1|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

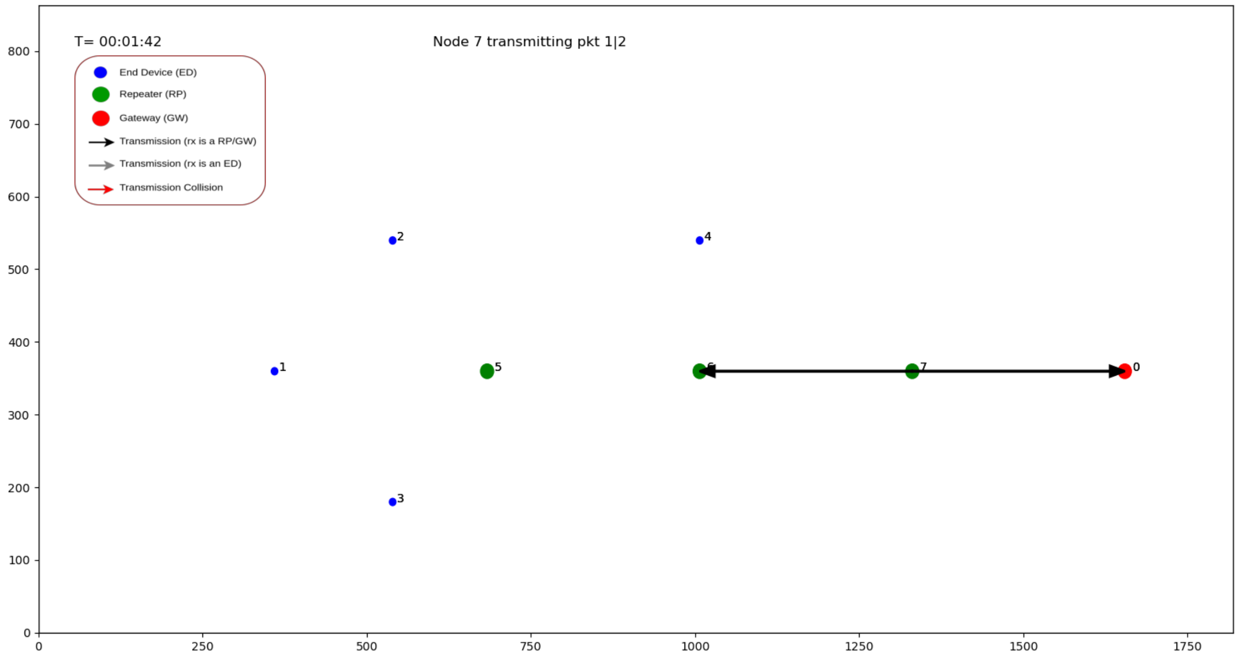

T= 00:01:56
Node 7 transmitting pkt 2|4
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

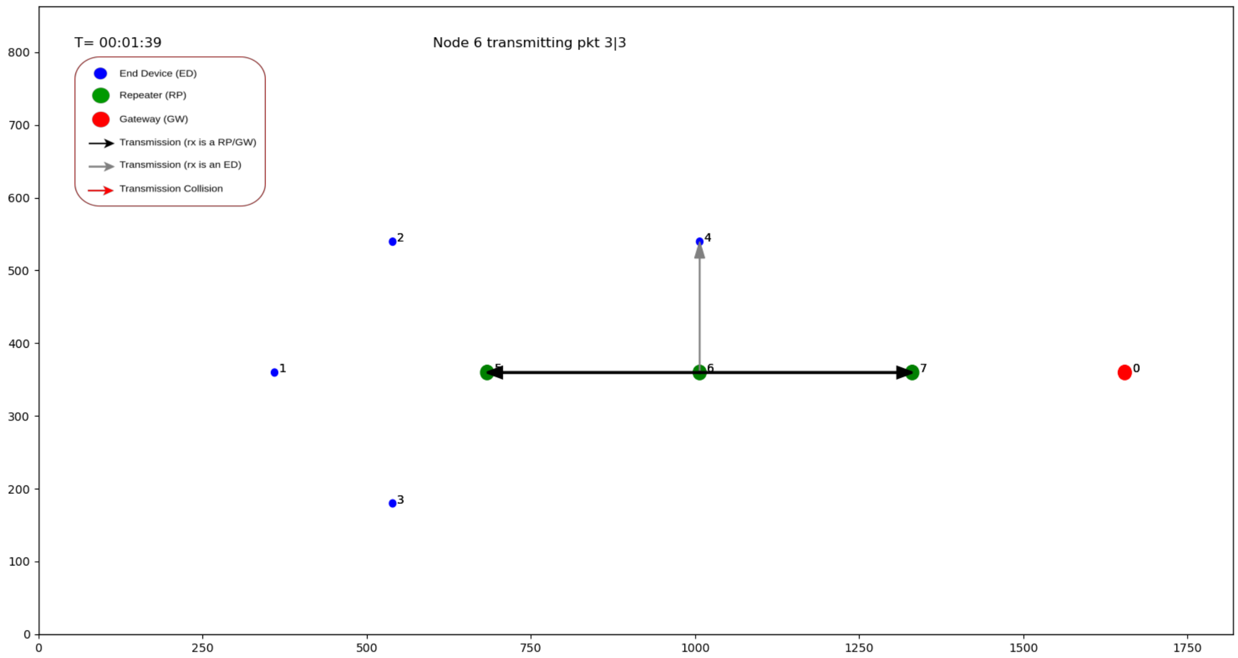

T= 00:01:52
Node 6 transmitting pkt 2|4
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

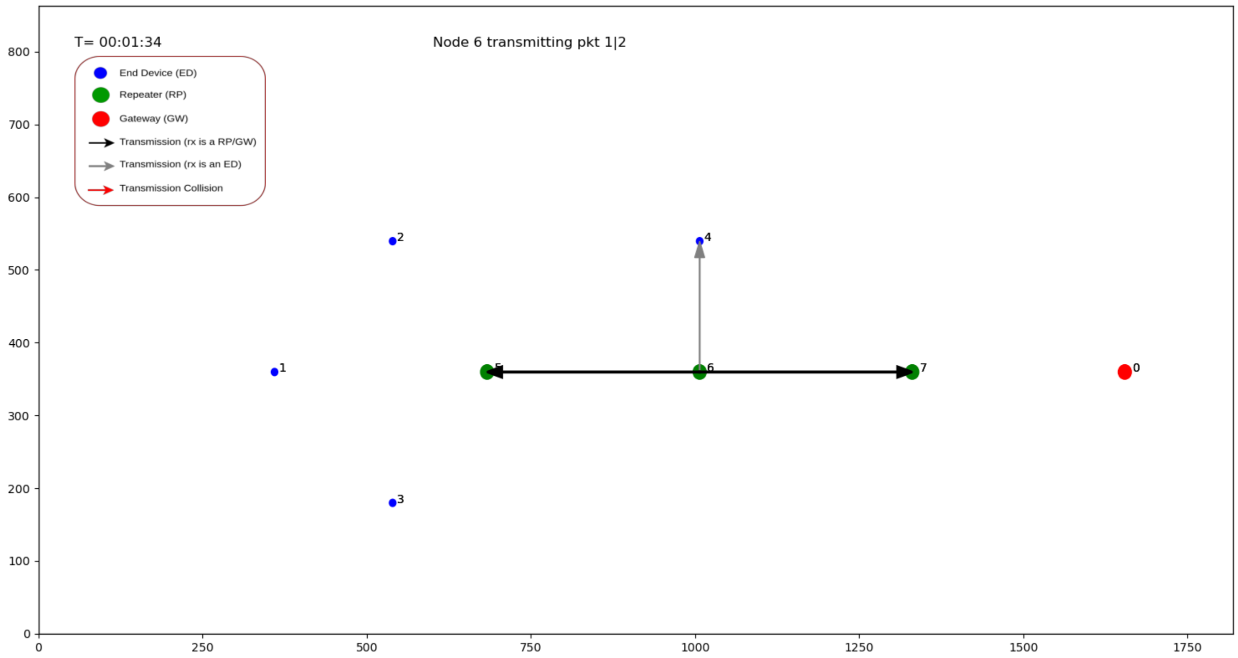

T= 00:01:49
Node 5 transmitting pkt 2|4
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

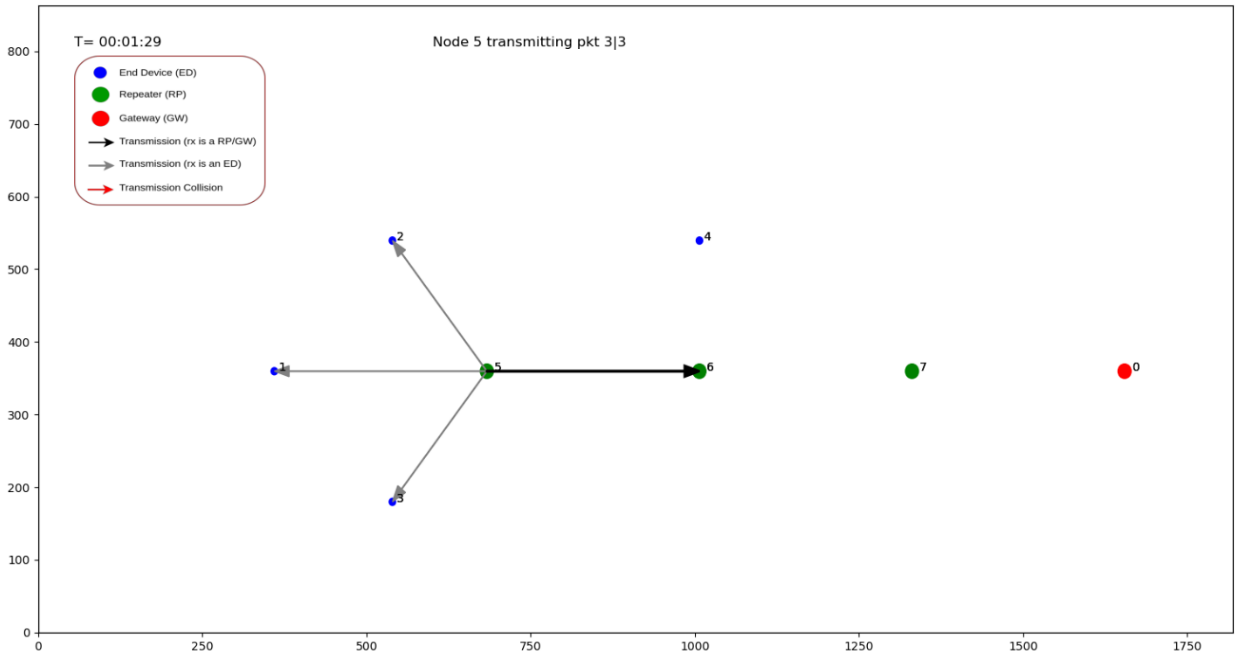

T= 00:01:46
Node 7 transmitting pkt 3|3
Node 2 transmitting pkt 2|4
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

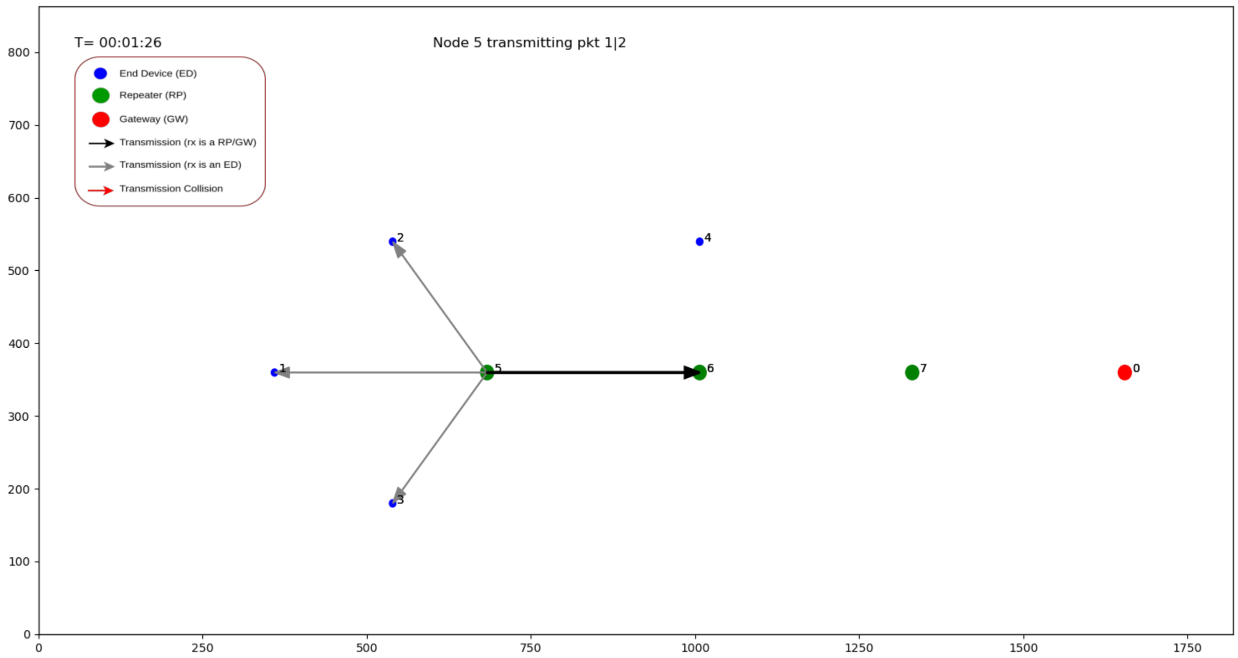

T= 00:01:45
Node 7 transmitting pkt 3|3
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

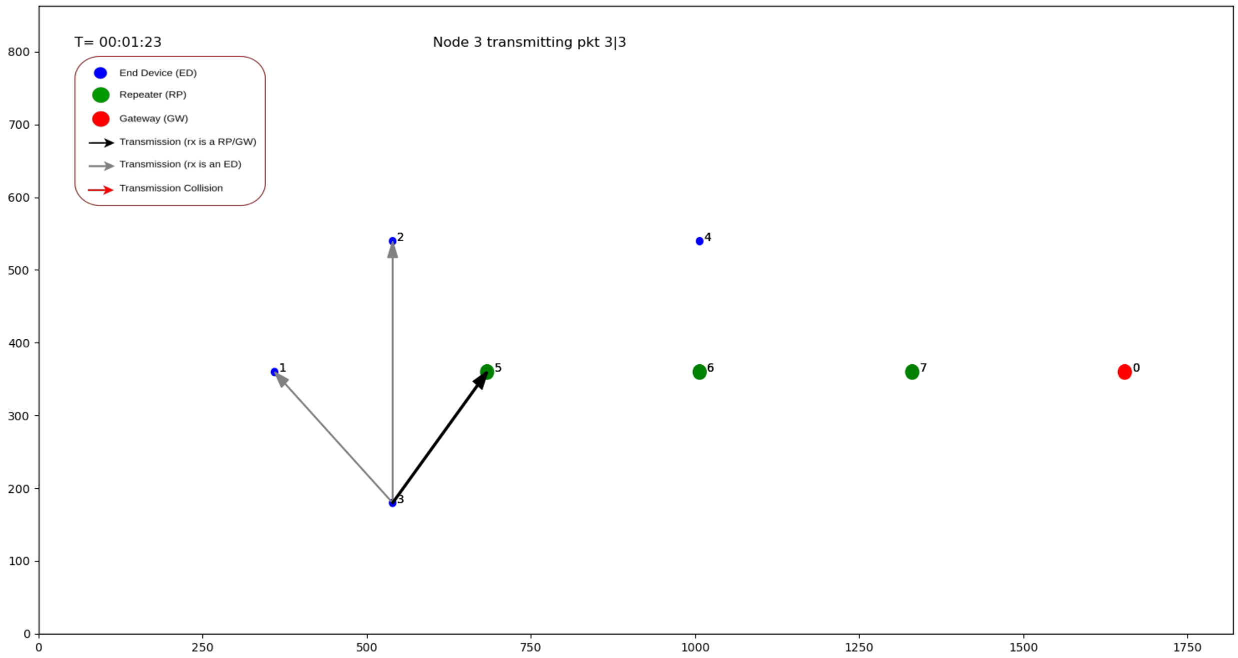

T= 00:01:42
Node 7 transmitting pkt 1|2
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
800
700
600
500
400
300
200
100
0
250
500
750
1000
1250
1500
1750

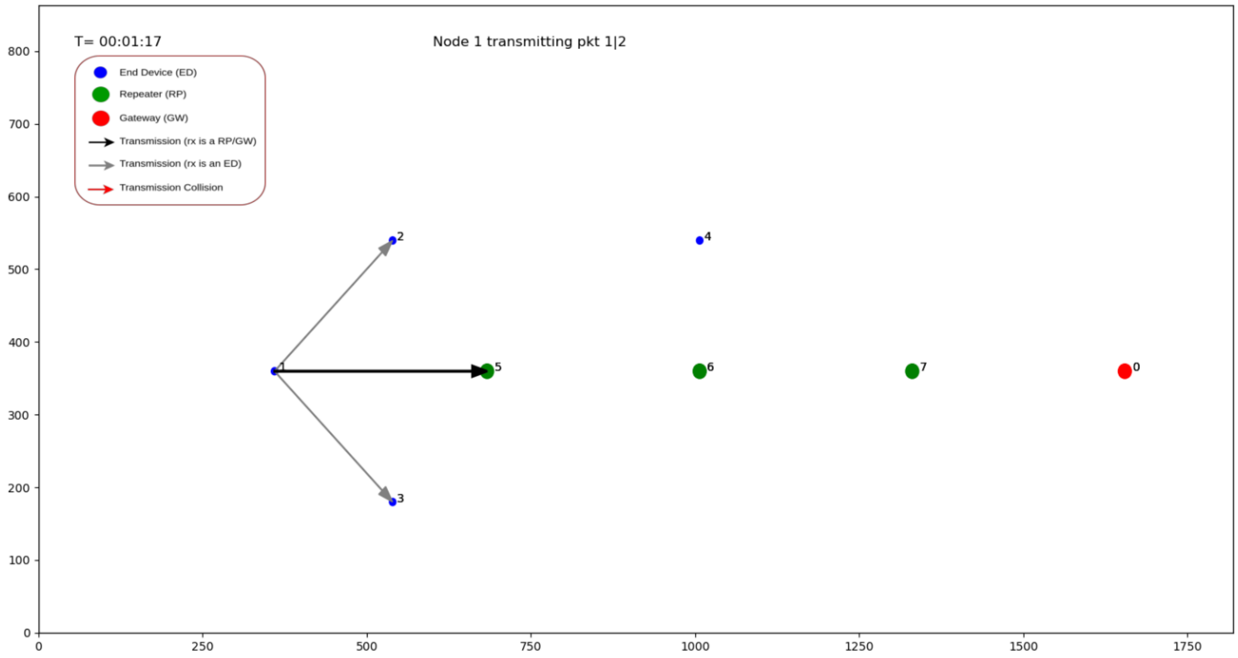

T= 00:01:39
Node 6 transmitting pkt 3|3
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

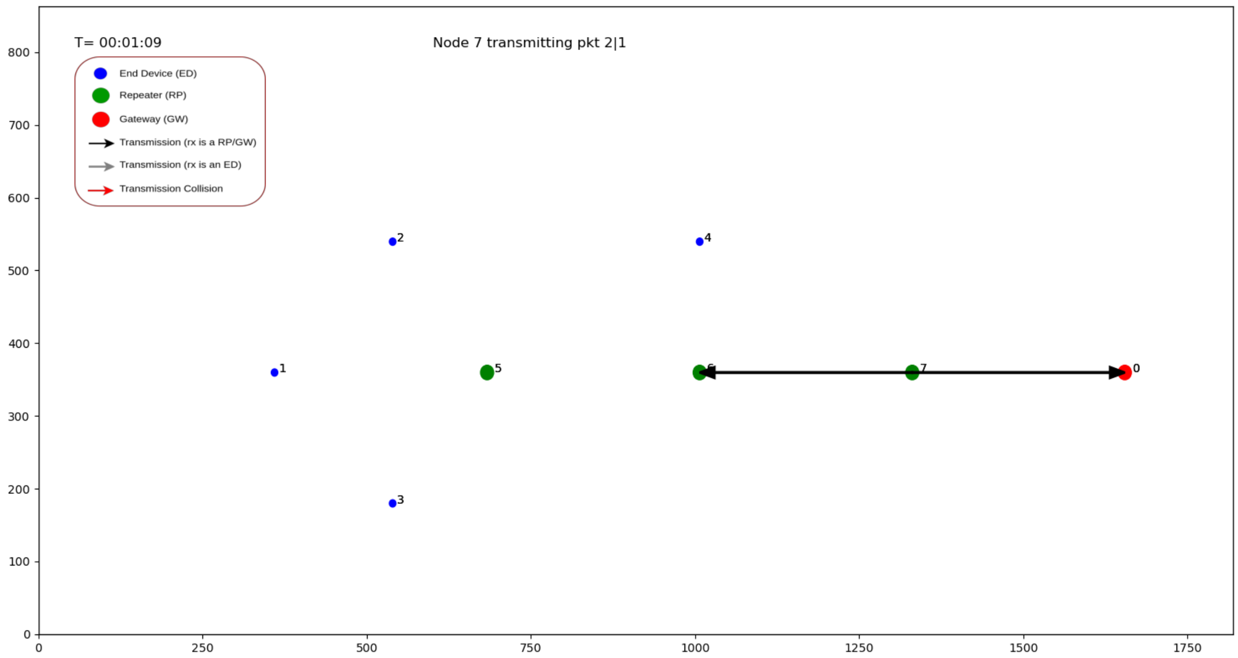

T= 00:01:34
Node 6 transmitting pkt 1|2
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

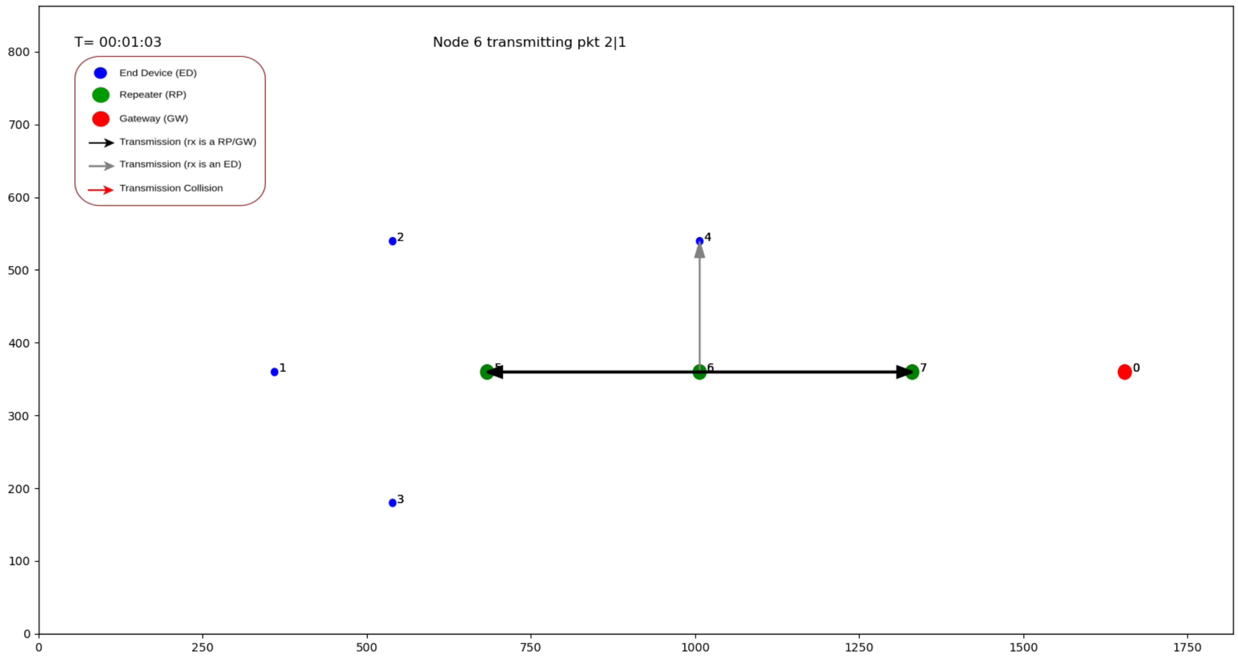

T= 00:01:29
Node 5 transmitting pkt 3|3
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

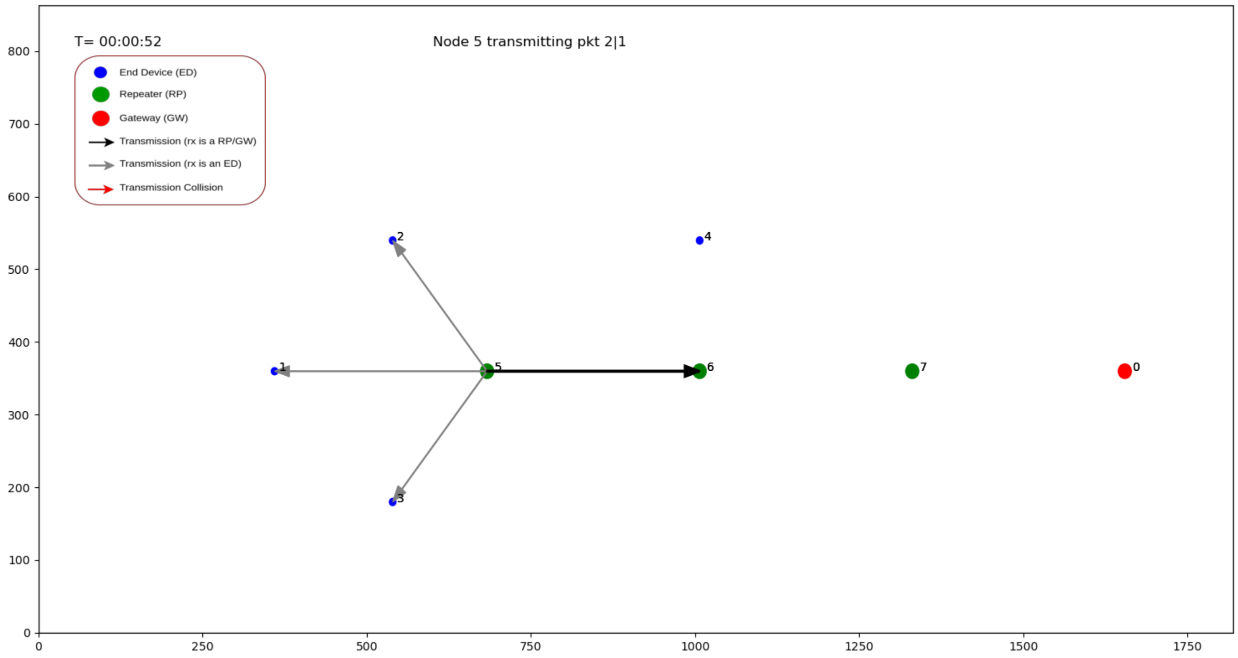

T= 00:01:26
Node 5 transmitting pkt 1|2
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:01:23
Node 3 transmitting pkt 3|3
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

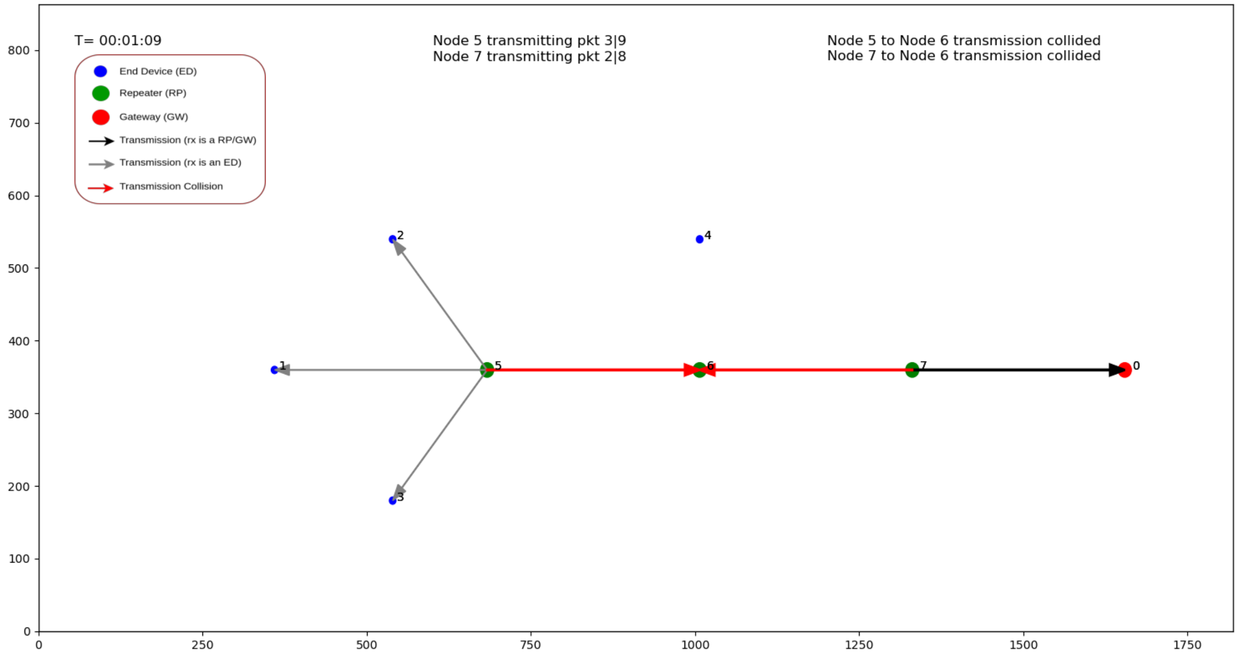

T= 00:01:17
Node 1 transmitting pkt 1|2
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:01:09
Node 7 transmitting pkt 2|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

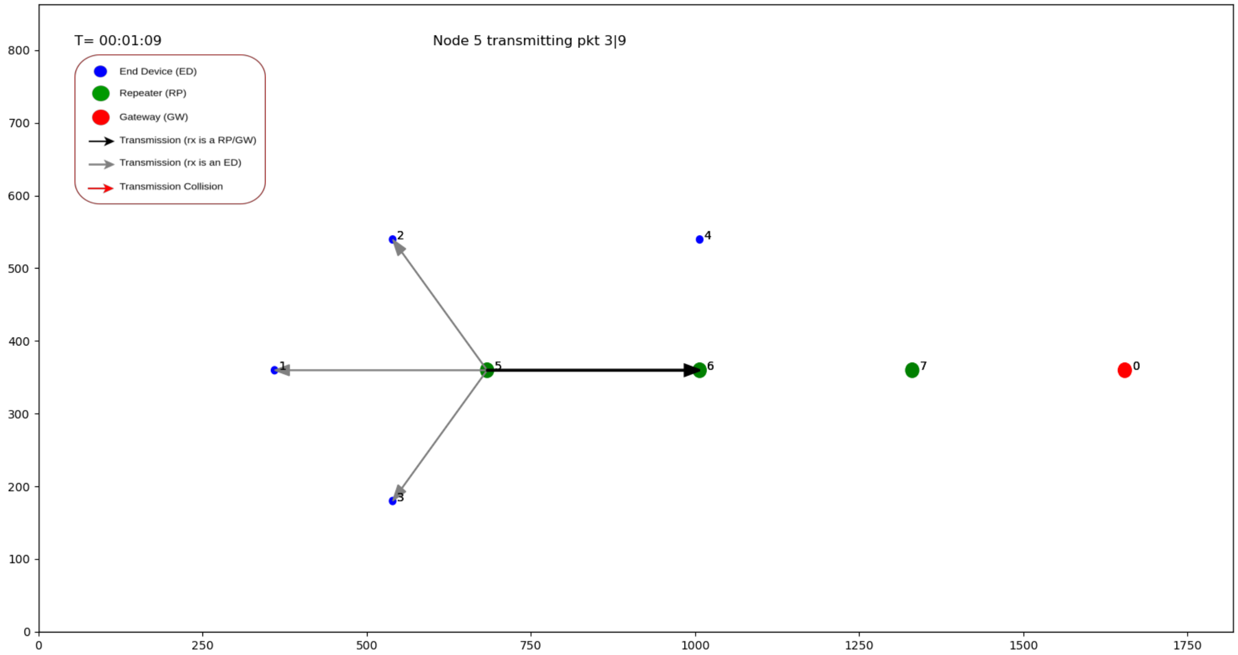

T= 00:01:03
Node 6 transmitting pkt 2|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

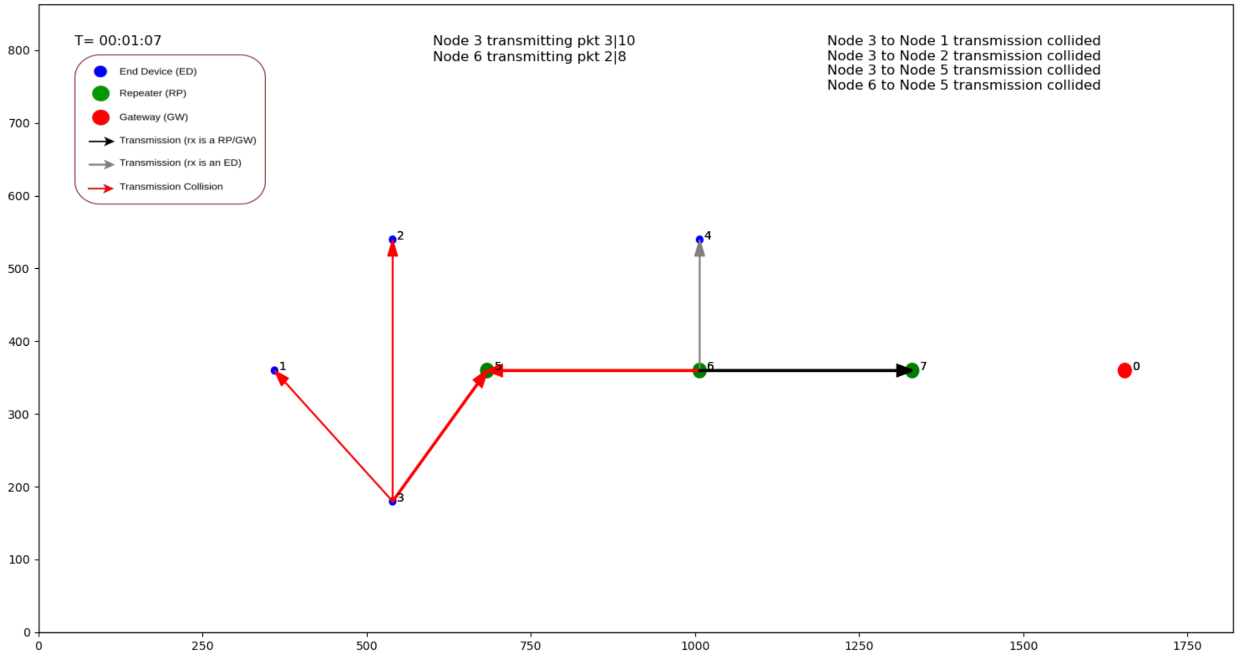

T= 00:00:52
Node 5 transmitting pkt 2|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

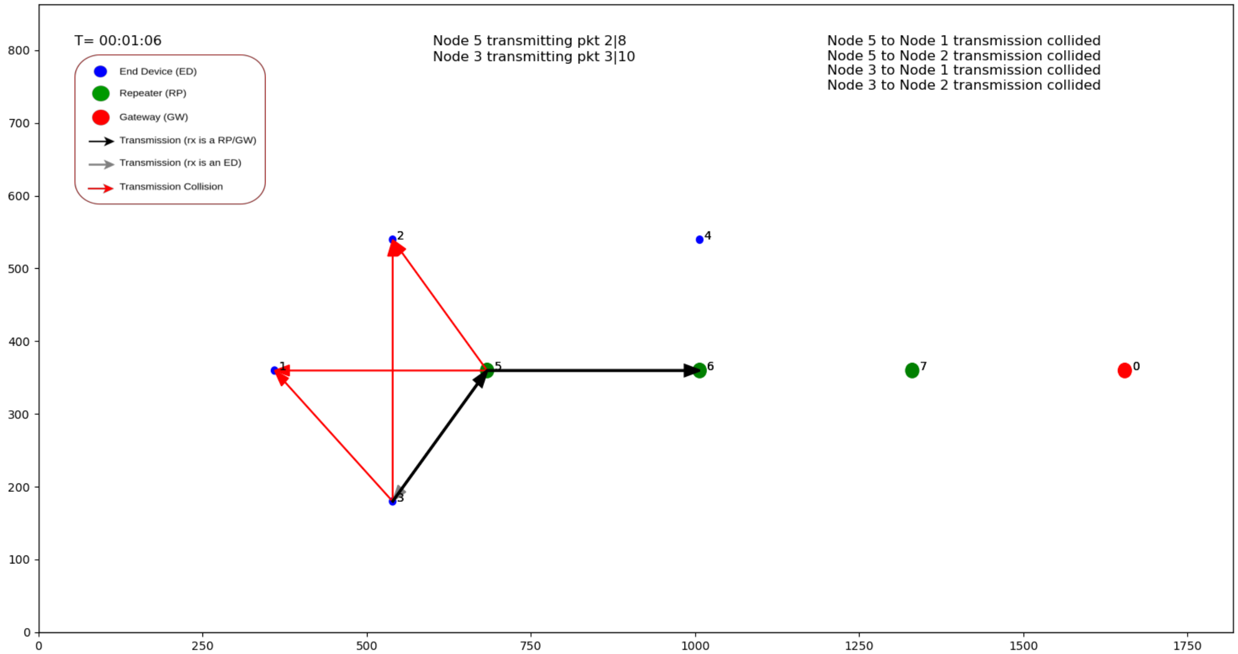
Simulation 2

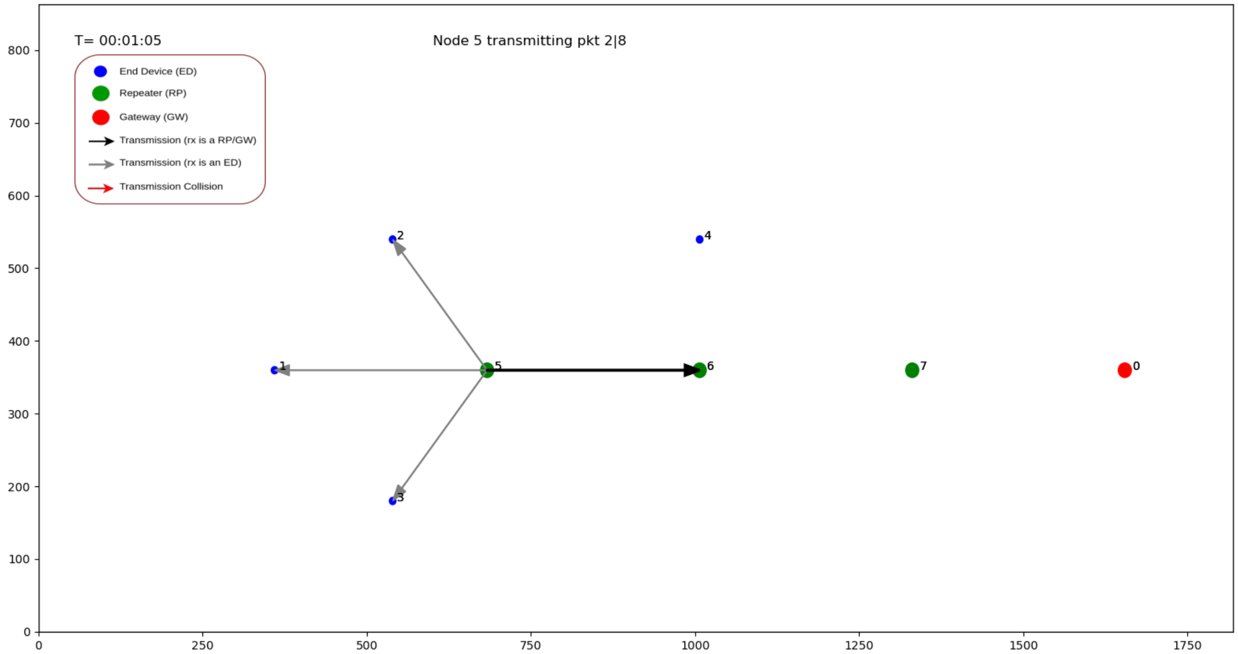


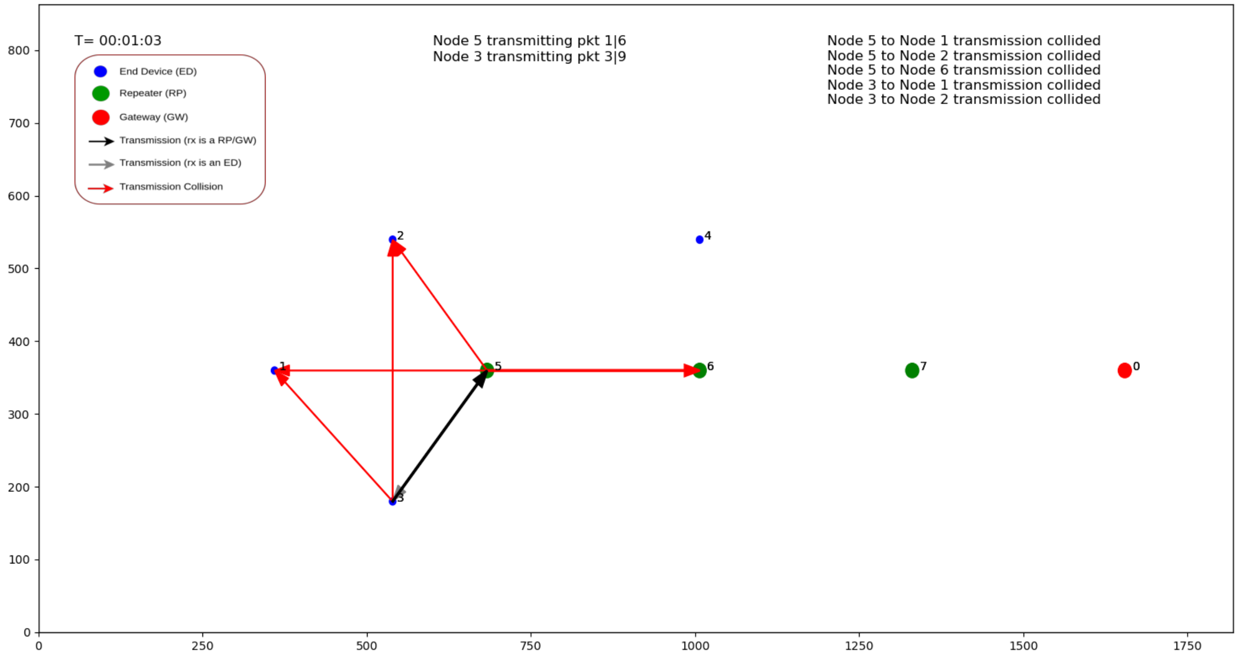
T= 00:01:09
Node 5 transmitting pkt 3|9
Node 7 transmitting pkt 2|8
Node 5 to Node 6 transmission collided
Node 7 to Node 6 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

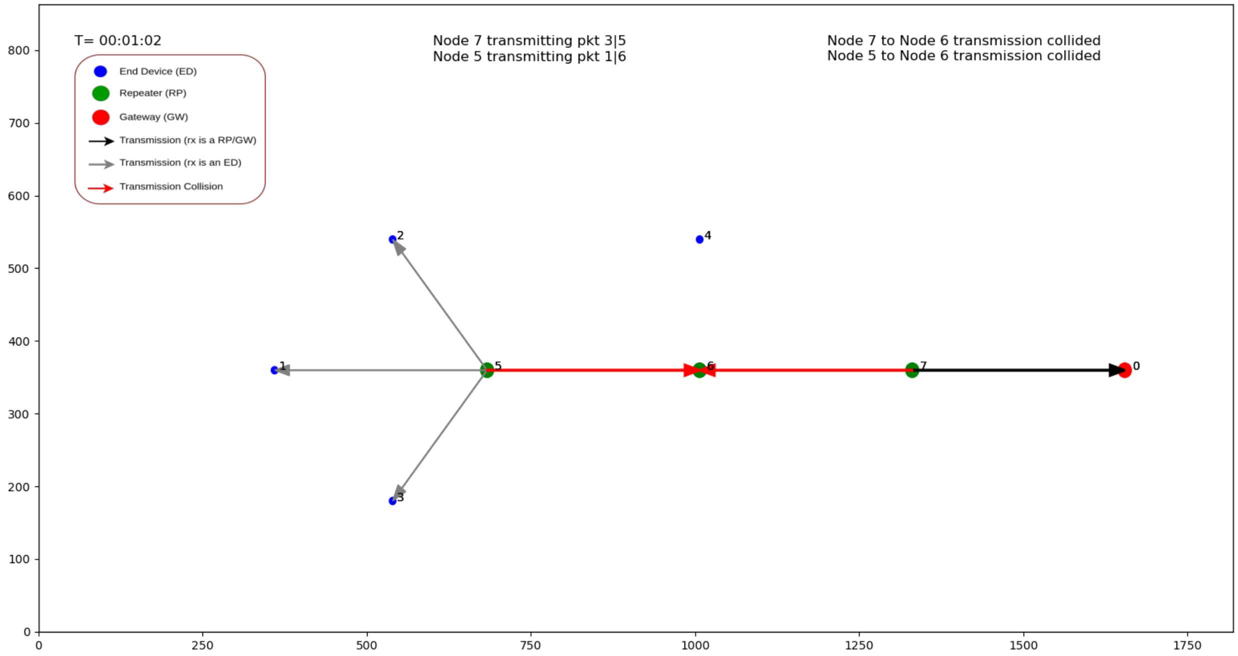


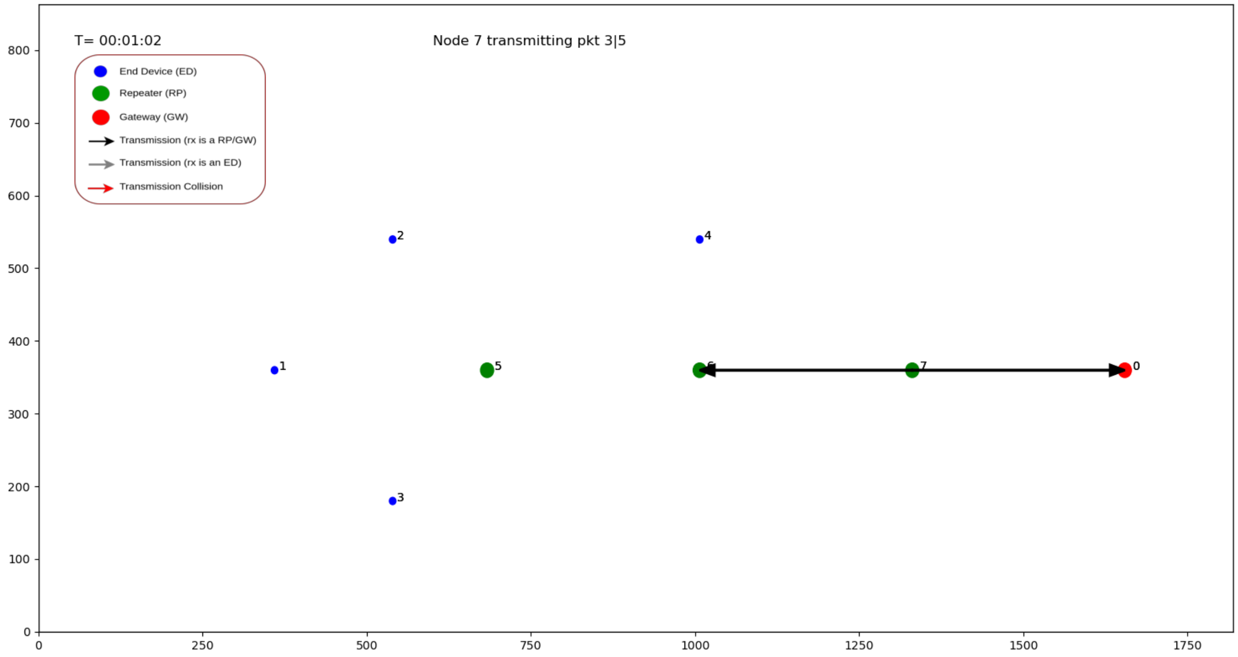

T= 00:01:07
Node 3 transmitting pkt 3|10
Node 6 transmitting pkt 2|8
Node 3 to Node 1 transmission collided
Node 3 to Node 2 transmission collided
Node 3 to Node 5 transmission collided
Node 6 to Node 5 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

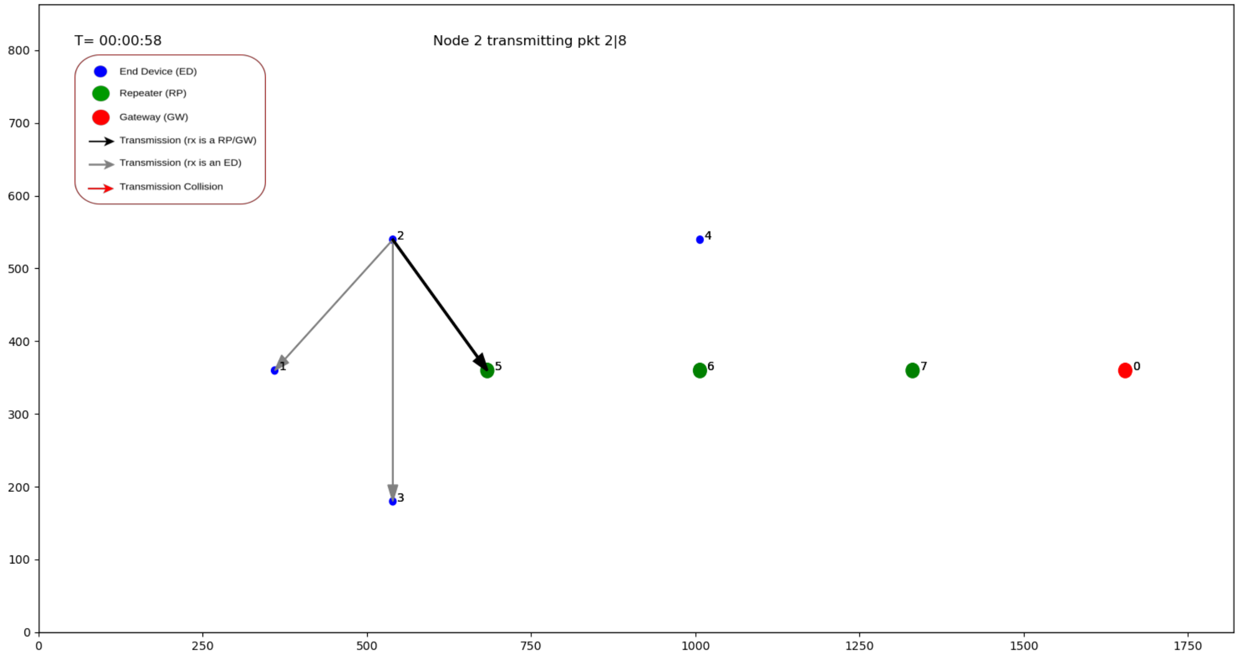

T= 00:01:06
Node 5 transmitting pkt 2|8
Node 3 transmitting pkt 3|10
Node 5 to Node 1 transmission collided
Node 5 to Node 2 transmission collided
Node 3 to Node 1 transmission collided
Node 3 to Node 2 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

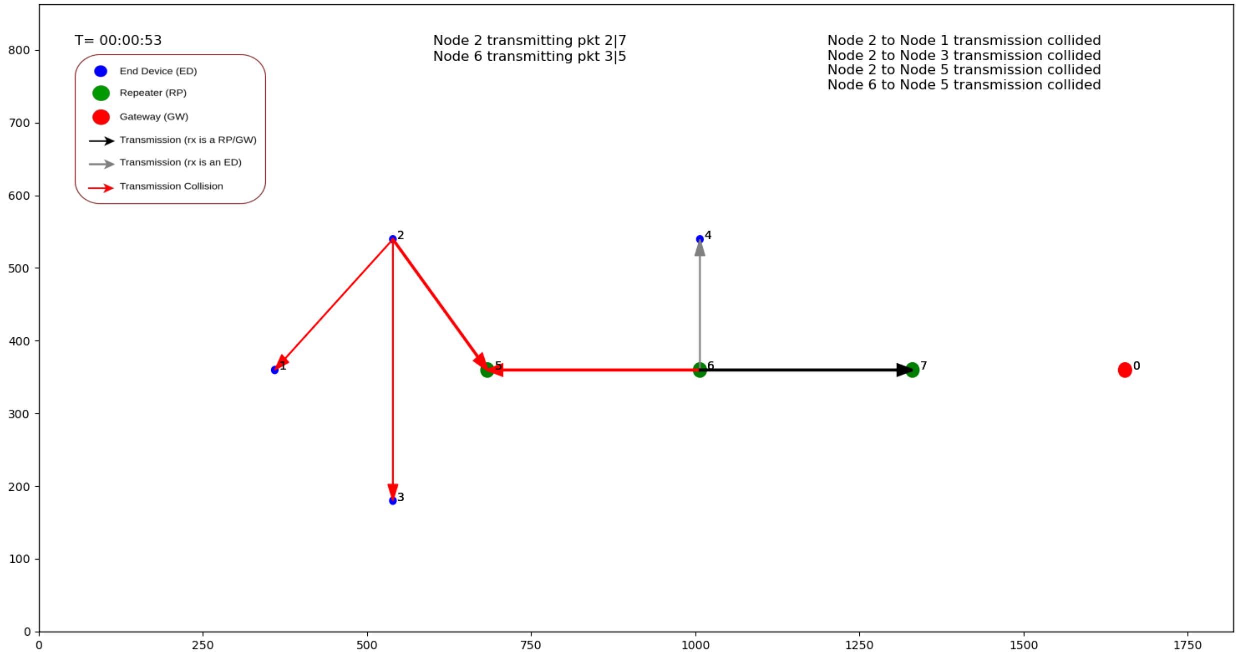

T= 00:01:05
Node 5 transmitting pkt 2|8
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

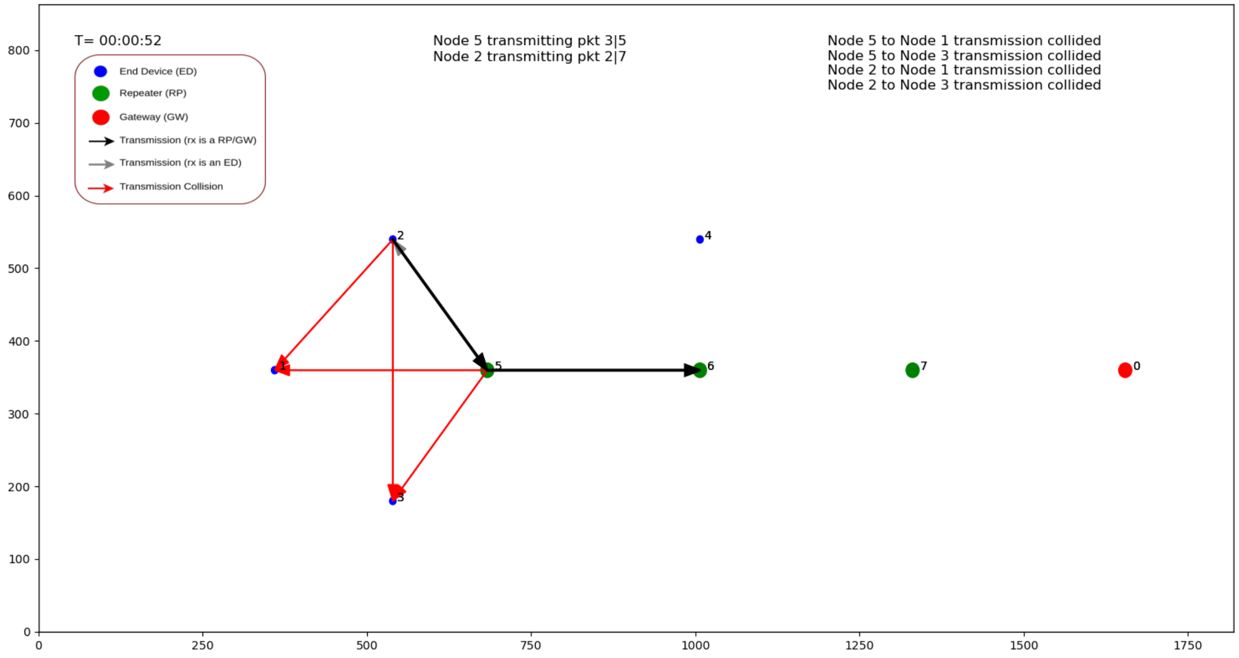

T= 00:01:03
Node 5 transmitting pkt 1|6
Node 3 transmitting pkt 3|9
Node 5 to Node 1 transmission collided
Node 5 to Node 2 transmission collided
Node 5 to Node 6 transmission collided
Node 3 to Node 1 transmission collided
Node 3 to Node 2 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

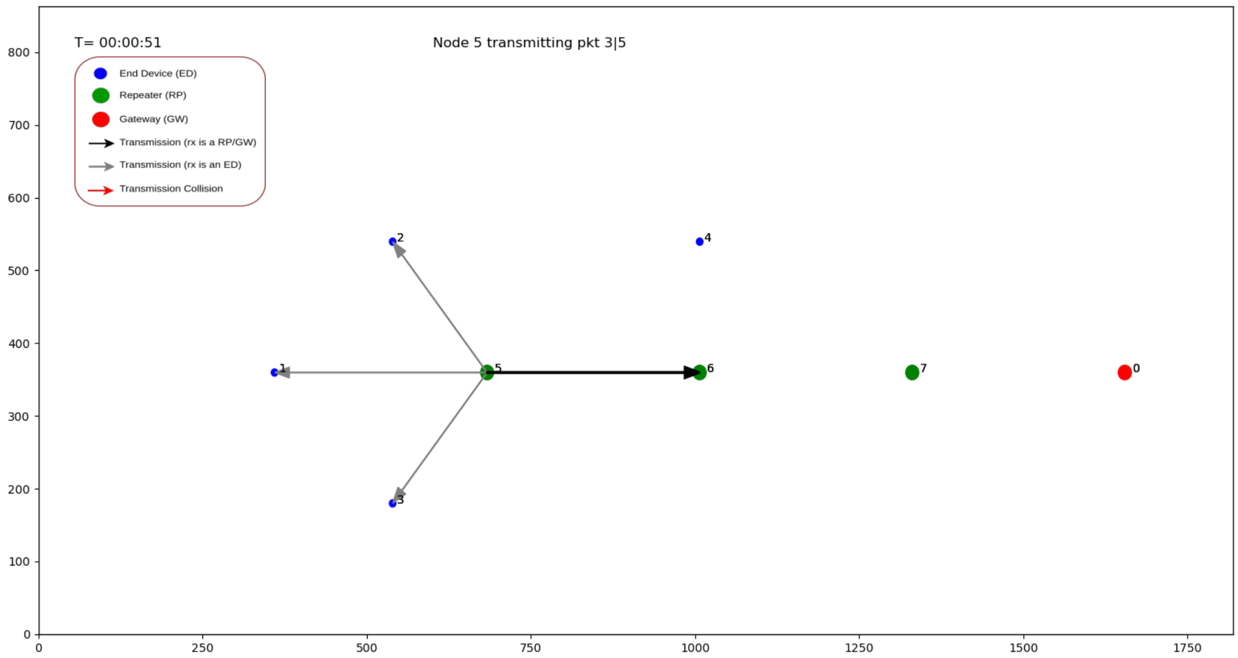

T= 00:01:02
Node 7 transmitting pkt 3|5
Node 5 transmitting pkt 1|6
Node 7 to Node 6 transmission collided
Node 5 to Node 6 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

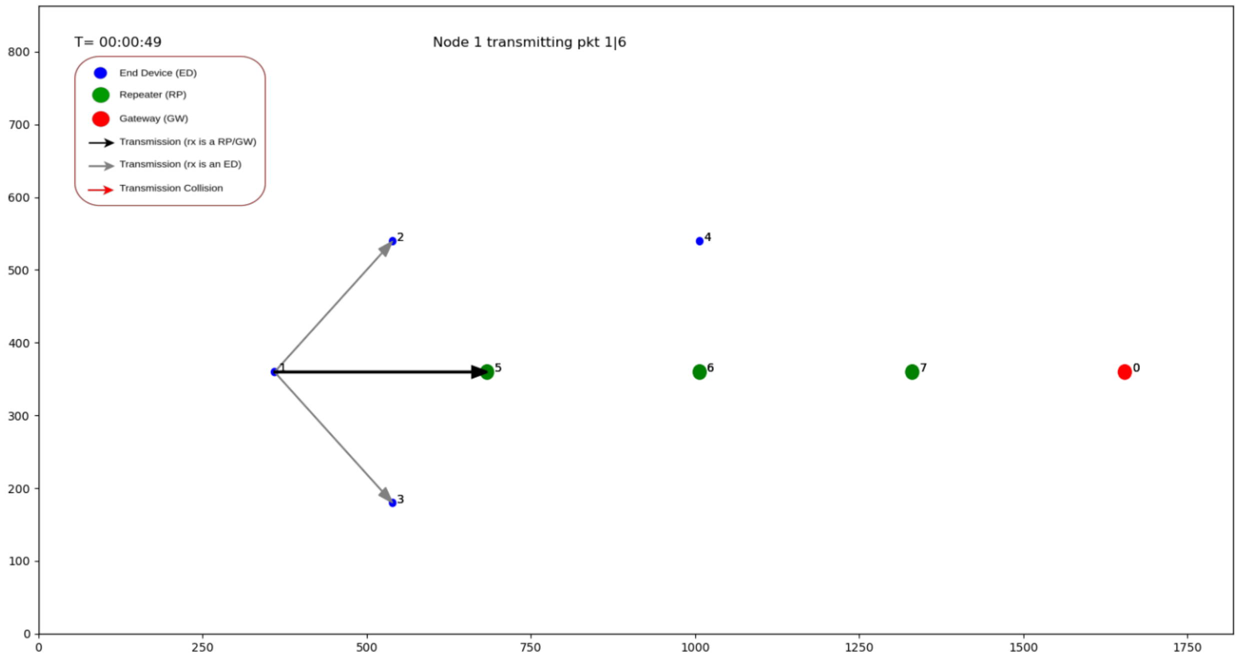

T= 00:01:02
Node 7 transmitting pkt 3|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

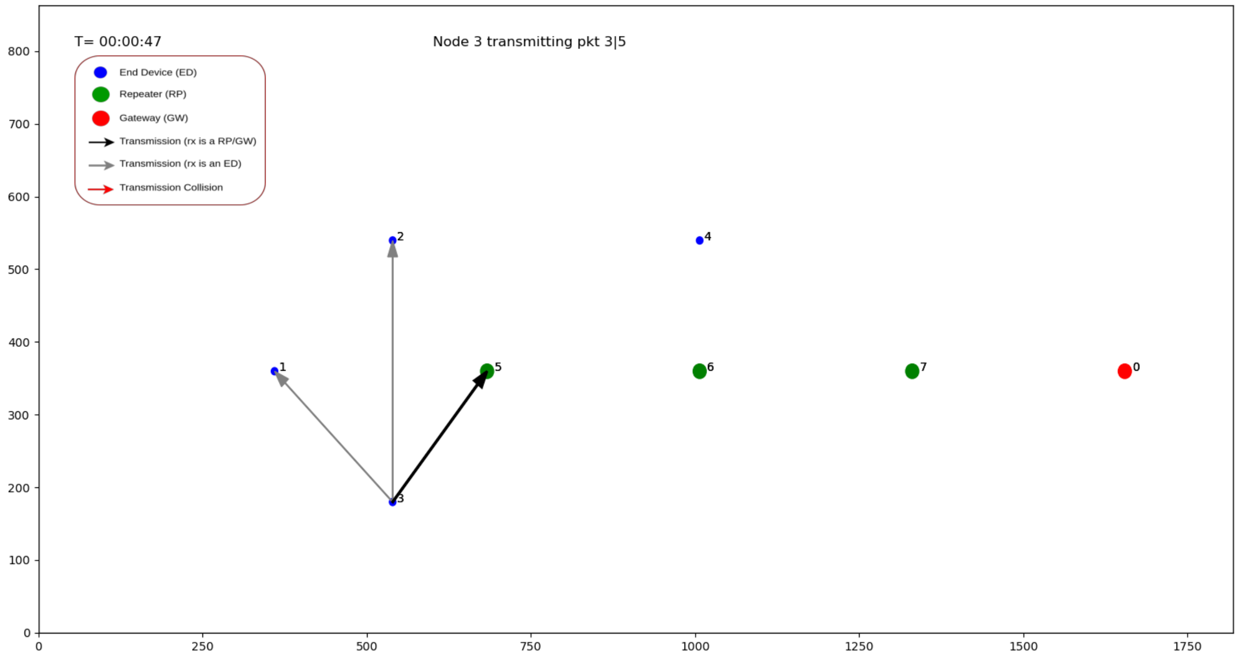

T= 00:00:58
Node 2 transmitting pkt 2|8
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

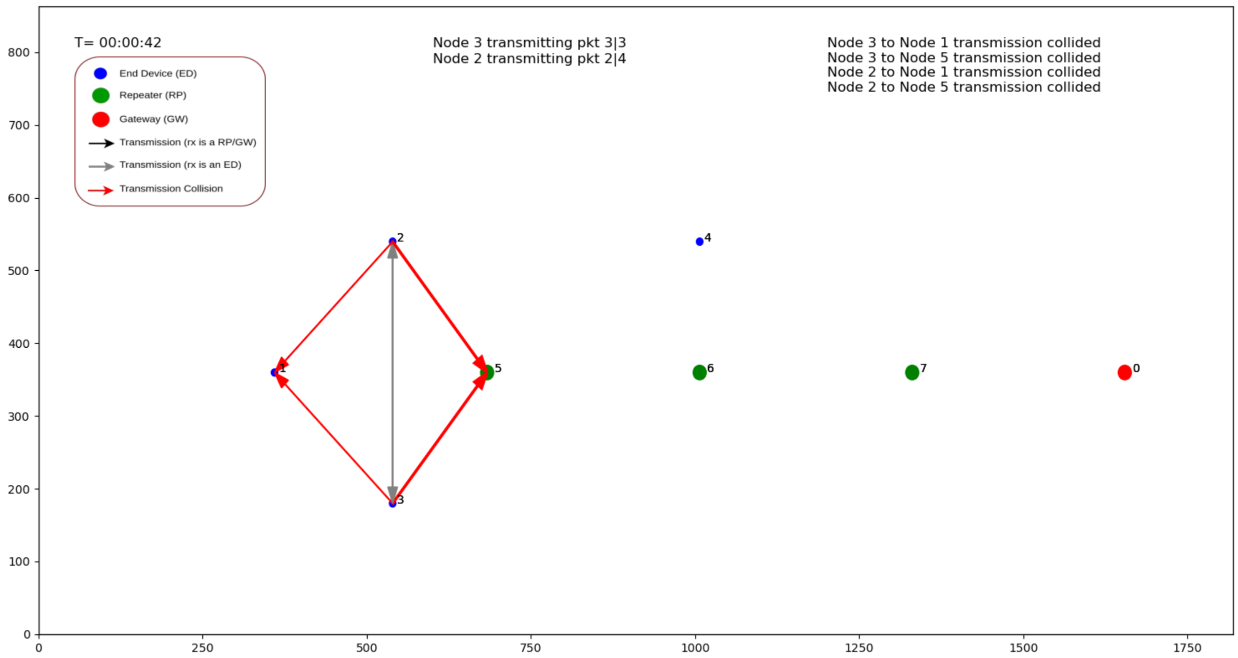

T= 00:00:53
Node 2 transmitting pkt 2|7
Node 6 transmitting pkt 3|5
Node 2 to Node 1 transmission collided
Node 2 to Node 3 transmission collided
Node 2 to Node 5 transmission collided
Node 6 to Node 5 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

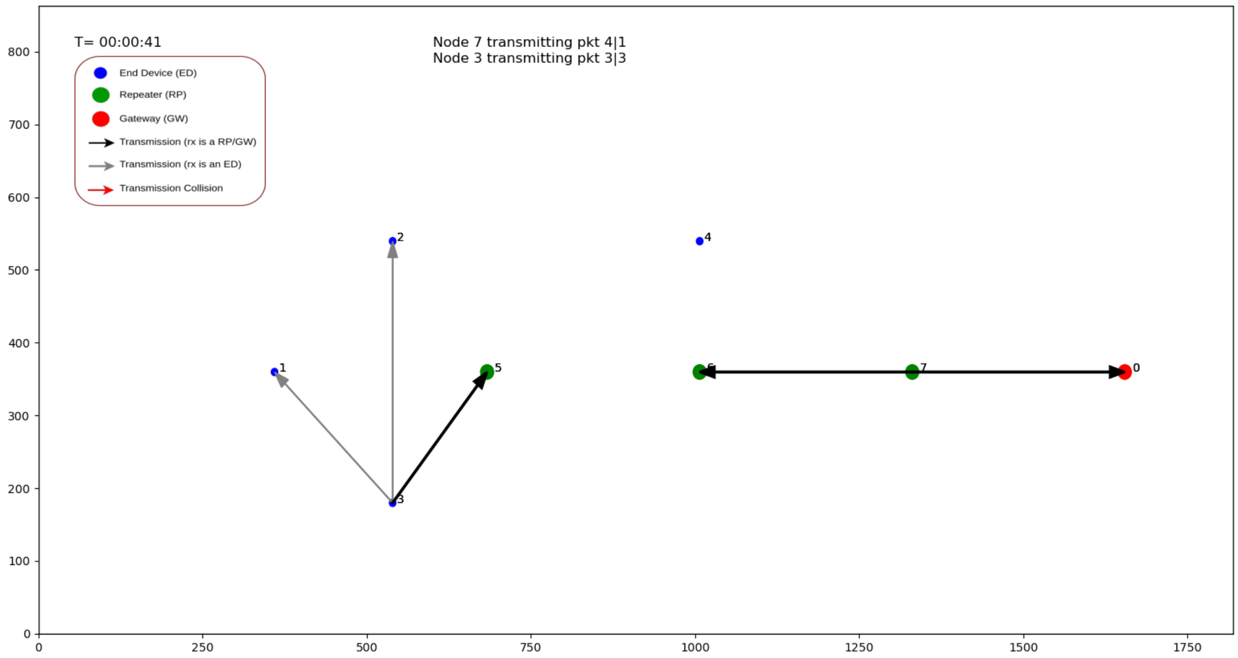

T= 00:00:52
Node 5 transmitting pkt 3|5
Node 2 transmitting pkt 2|7
Node 5 to Node 1 transmission collided
Node 5 to Node 3 transmission collided
Node 2 to Node 1 transmission collided
Node 2 to Node 3 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750

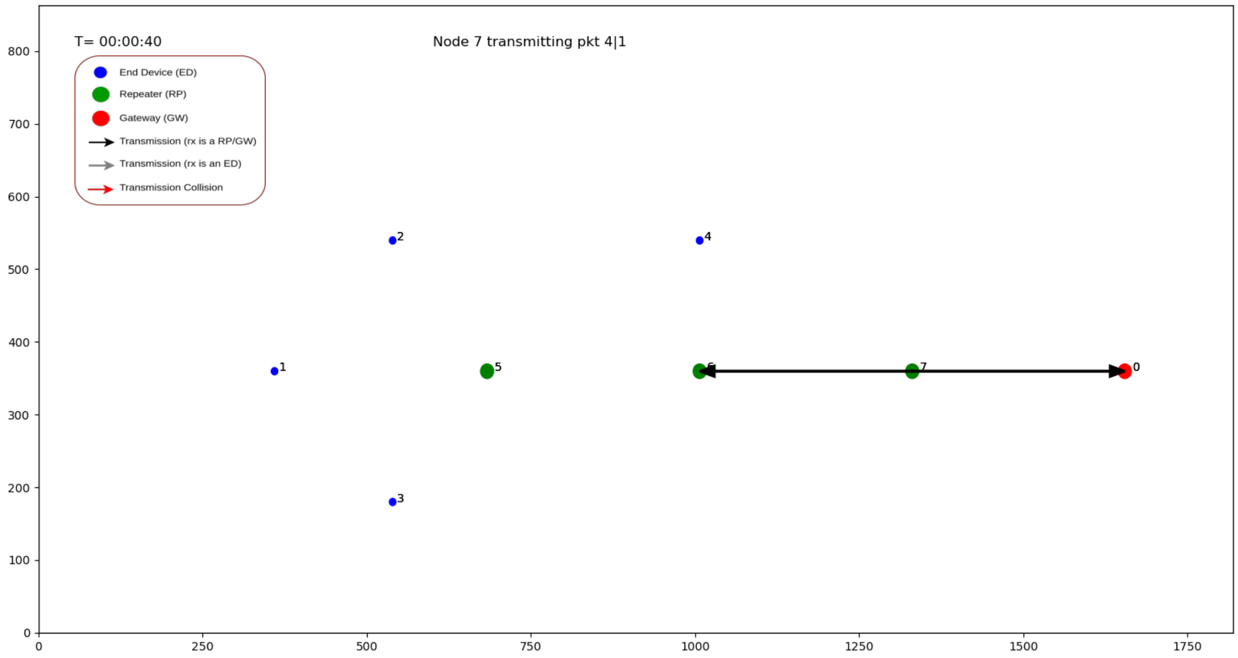

T= 00:00:51
Node 5 transmitting pkt 3|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
800
700
600
500
400
300
200
100
0
0
250
500
750
1000
1250
1500
1750

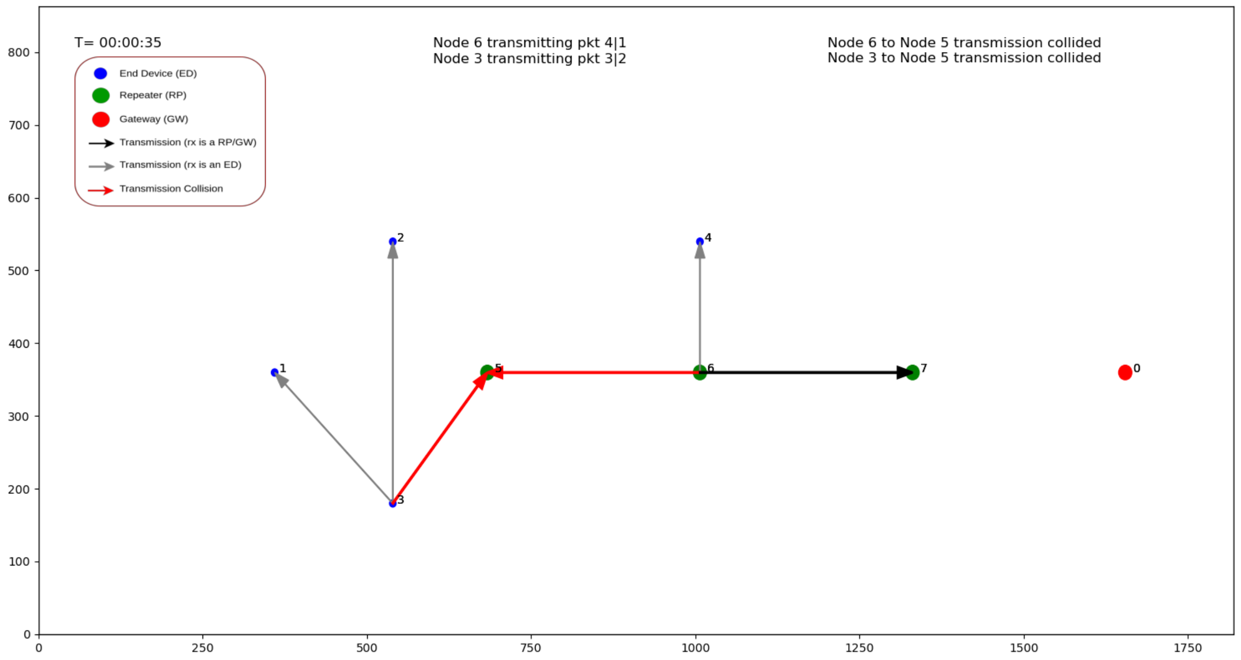

T= 00:00:49
Node 1 transmitting pkt 1|6
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
800
700
600
500
400
300
200
100
0
0
250
500
750
1000
1250
1500
1750

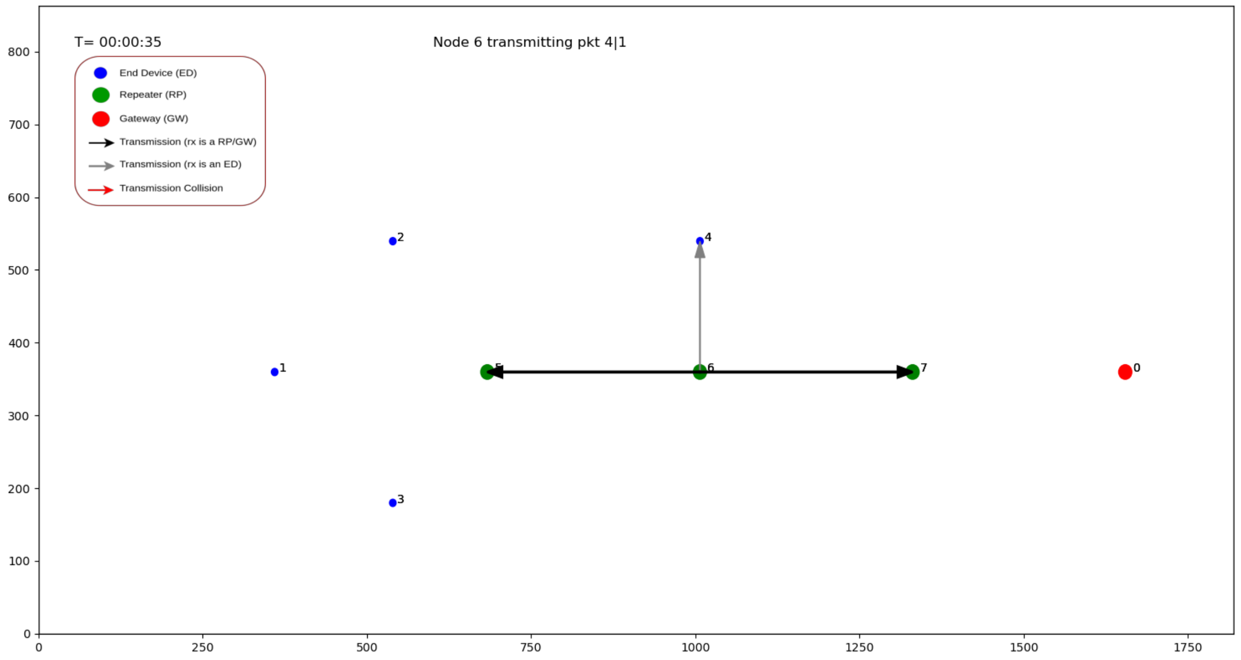

T= 00:00:47
Node 3 transmitting pkt 3|5
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
2
4
1
5
6
7
0
3
800
700
600
500
400
300
200
100
0
0
250
500
750
1000
1250
1500
1750

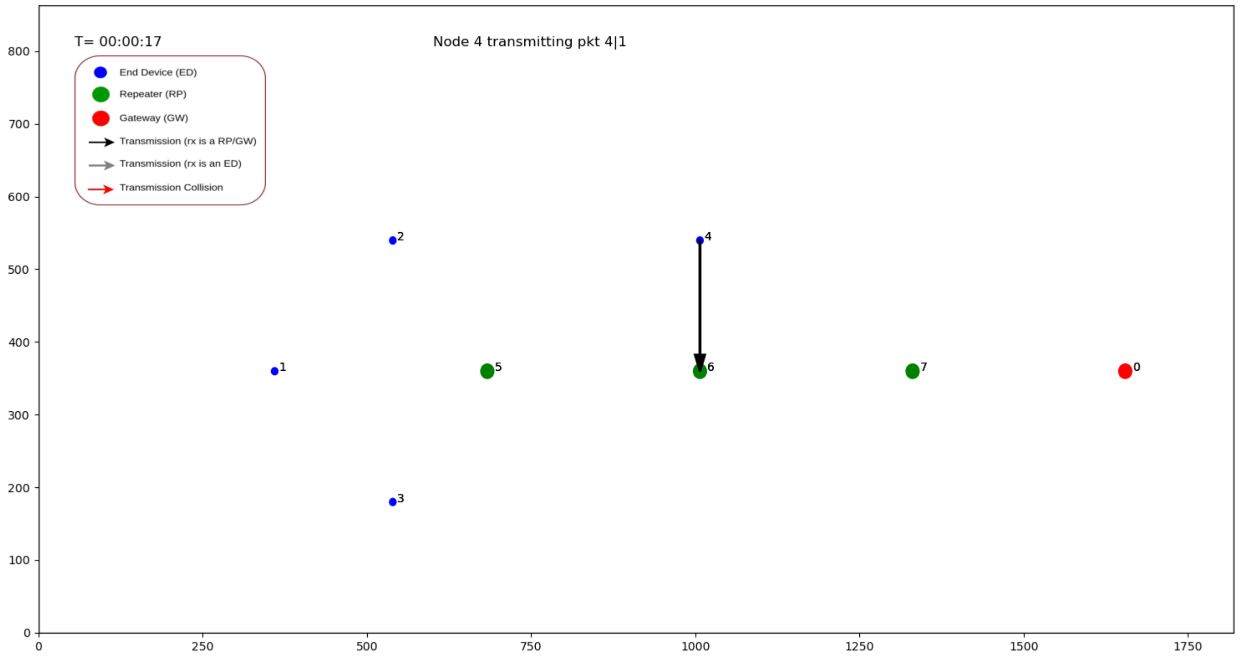

T= 00:00:42
Node 3 transmitting pkt 3|3
Node 2 transmitting pkt 2|4
Node 3 to Node 1 transmission collided
Node 3 to Node 5 transmission collided
Node 2 to Node 1 transmission collided
Node 2 to Node 5 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:00:41
Node 7 transmitting pkt 4|1
Node 3 transmitting pkt 3|3
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:00:40
Node 7 transmitting pkt 4|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
1
2
3
4
5
6
7
0
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:00:35
Node 6 transmitting pkt 4|1
Node 3 transmitting pkt 3|2
Node 6 to Node 5 transmission collided
Node 3 to Node 5 transmission collided
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:00:35
Node 6 transmitting pkt 4|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750


T= 00:00:17
Node 4 transmitting pkt 4|1
End Device (ED)
Repeater (RP)
Gateway (GW)
Transmission (rx is a RP/GW)
Transmission (rx is an ED)
Transmission Collision
0
1
2
3
4
5
6
7
0
100
200
300
400
500
600
700
800
0
250
500
750
1000
1250
1500
1750