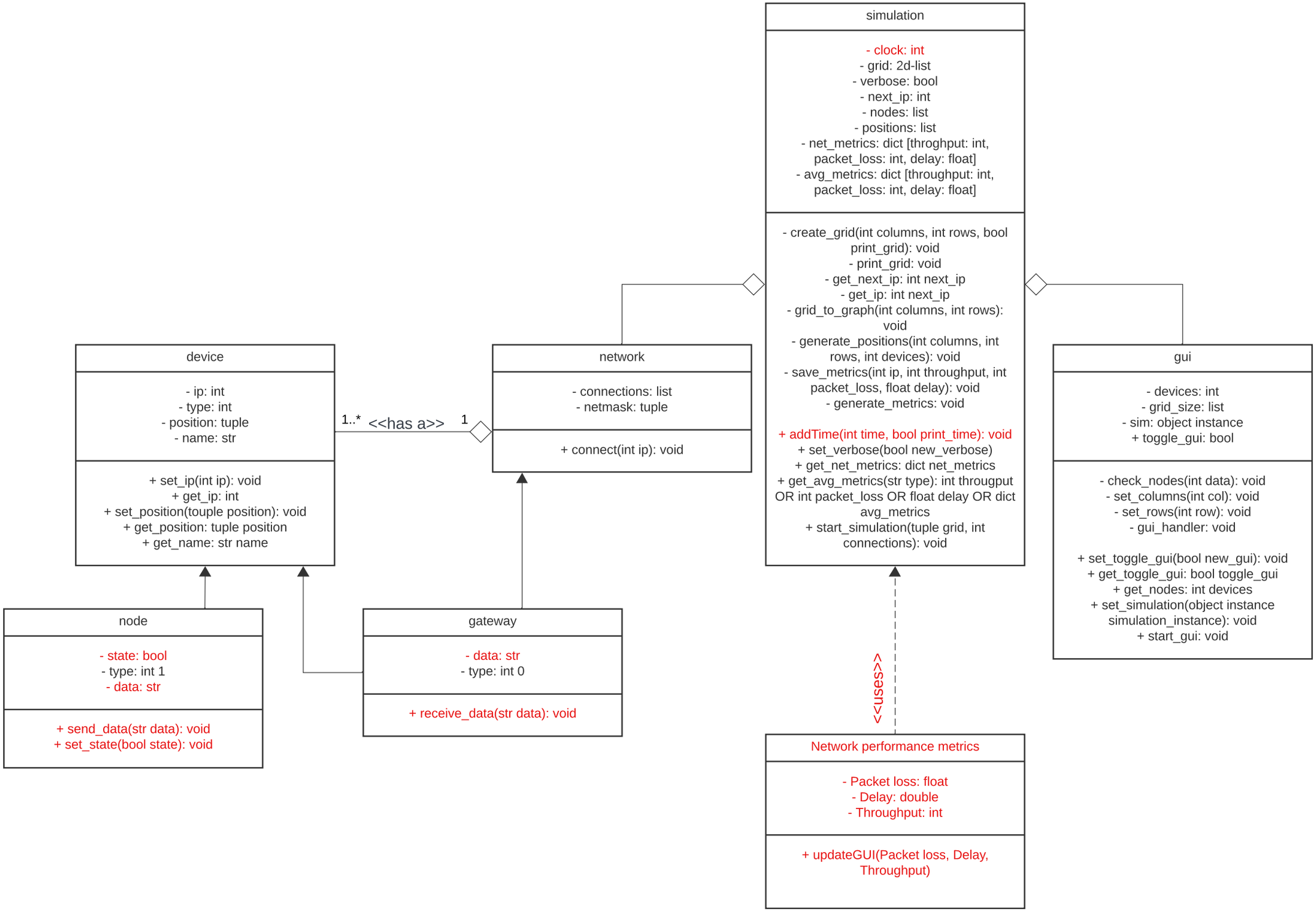
simulation
- clock: int
- grid: 2d-list
- verbose: bool
- next_ip: int
- nodes: list
- positions: list
- net_metrics: dict [throghput: int, packet_loss: int, delay: float]
- avg_metrics: dict [throughput: int, packet_loss: int, delay: float]
- create_grid(int columns, int rows, bool print_grid): void
- print_grid: void
- get_next_ip: int next_ip
- get_ip: int next_ip
- grid_to_graph(int columns, int rows): void
- generate_positions(int columns, int rows, int devices): void
- save_metrics(int ip, int throughput, int packet_loss, float delay): void
- generate_metrics: void
+ addTime(int time, bool print_time): void
+ set_verbose(bool new_verbose)
+ get_net_metrics: dict net_metrics
+ get_avg_metrics(str type): int througput OR int packet_loss OR float delay OR dict avg_metrics
+ start_simulation(tuple grid, int connections): void
device
- ip: int
- type: int
- position: tuple
- name: str
+ set_ip(int ip): void
+ get_ip: int
+ set_position(touple position): void
+ get_position: tuple position
+ get_name: str name
1..*
<<has a>>
1
network
- connections: list
- netmask: tuple
+ connect(int ip): void
gui
- devices: int
- grid_size: list
- sim: object instance
+ toggle_gui: bool
- check_nodes(int data): void
- set_columns(int col): void
- set_rows(int row): void
- gui_handler: void
+ set_toggle_gui(bool new_gui): void
+ get_toggle_gui: bool toggle_gui
+ get_nodes: int devices
+ set_simulation(object instance simulation_instance): void
+ start_gui: void
node
- state: bool
- type: int 1
- data: str
+ send_data(str data): void
+ set_state(bool state): void
gateway
- data: str
- type: int 0
+ receive_data(str data): void
<<uses>>
Network performance metrics
- Packet loss: float
- Delay: double
- Throughput: int
+ updateGUI(Packet loss, Delay, Throughput)