```
# 2024 Â© Idan Hazay
# Import libraries

import threading
from PyQt6.QtCore import pyqtSignal, QThread

class ReceiveThread(QThread):
        # Define a signal to emit data received from recv_data
        reply_received = pyqtSignal(bytes)

        def __init__(self, network):
            super().__init__()
            self.running = True  # Add a flag to control the thread loop
            self._pause_event = threading.Event()  # Event to manage pausing
            self._pause_event.set()  # Initially, the thread is not paused
            self.network = network

        def run(self):
            while self.running:
                # Wait for the thread to be resumed if paused
                self._pause_event.wait()

                # Simulate receiving data
                reply = self.network.recv_data()  # Assume this method exists and returns bytes
                if reply:
                    self.reply_received.emit(reply)  # Emit the received reply to the main thread

        def pause(self):
            self._pause_event.clear()

        def resume(self):
            self._pause_event.set()
```