LAN IP Scanner

This Python script scans a local network for active IP addresses. It uses the `ping` command to check if a host is reachable and employs multithreading to speed up the scanning process. A semaphore limits the number of concurrent threads to avoid overwhelming the system. The script takes a network address (e.g., “192.168.1.0/24”) as input and prints whether each IP address in the network is up or down.

```python
import subprocess
import ipaddress
import threading
import os

MAX_THREADS = 255  # Maximum number of threads to use

def is_host_up(host):
    """
    Checks if a host is up using ping.
    Returns True if the host is up, False otherwise.
    """
    try:
        # -c 1: Send only 1 packet
        # -W 1: Wait 1 second for a response
        subprocess.check_output(["ping", "-c", "1", "-W", "1", host], timeout=1)
        return True
    except subprocess.CalledProcessError:
        return False
    except subprocess.TimeoutExpired:
        return False

def scan_ip(ip_str):
    """
    Scans a single IP address and prints its status.
    """
    if is_host_up(ip_str):
        print(f"{ip_str} is up")
    else:
        print(f"{ip_str} is down")

def scan_network(network):
    """
    Scans a network for live hosts using threads, limiting the number of concurrent threads.
    """
    print(f"Scanning network: {network}")
    threads = []
    semaphore = threading.Semaphore(MAX_THREADS)  # Limit the number of concurrent threads

    def scan_ip_with_semaphore(ip_str):
        semaphore.acquire()
        try:
            scan_ip(ip_str)
        finally:
            semaphore.release()

    for ip in ipaddress.IPv4Network(network):
        ip_str = str(ip)
        thread = threading.Thread(target=scan_ip_with_semaphore, args=(ip_str,))
        threads.append(thread)
        thread.start()

    for thread in threads:
        thread.join()

if __name__ == "__main__":
    network_to_scan = "192.168.1.0/24"  # Change this to your network
    scan_network(network_to_scan)
```