Analyseur d’IP LAN

Ce script Python analyse un réseau local pour trouver les adresses IP actives. Il utilise la commande `ping` pour vérifier si un hôte est joignable et utilise le multithreading pour accélérer le processus d’analyse. Un sémaphore limite le nombre de threads concurrents pour éviter de surcharger le système. Le script prend une adresse réseau (par exemple, “192.168.1.0/24”) en entrée et affiche si chaque adresse IP du réseau est active ou inactive.

```
import subprocess
import ipaddress
import threading
import os

MAX_THREADS = 255  # Nombre maximal de threads à utiliser

def is_host_up(host):
    """
    Vérifie si un hôte est actif en utilisant ping.
    Retourne True si l'hôte est actif, False sinon.
    """
    try:
        # -c 1: Envoyer seulement 1 paquet
        # -W 1: Attendre 1 seconde une réponse
        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):
    """
    Analyse une seule adresse IP et affiche son statut.
    """
    if is_host_up(ip_str):
        print(f"{ip_str} est active")
    else:
        print(f"{ip_str} est inactive")

def scan_network(network):
```

```
    """
    Analyse un réseau pour les hôtes actifs en utilisant des threads, en limitant le nombre de threads concurr
    """
    print(f"Analyse du réseau : {network}")
    threads = []
    semaphore = threading.Semaphore(MAX_THREADS)  # Limite le nombre de threads concurrents

    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"  # Changer ceci pour votre réseau
    scan_network(network_to_scan)
```