Dynamic Network Monitoring

Karolina Chochrek, Bartłomiej Kołacz, Karol Makara, Damian Matusz

Plan prezentacji

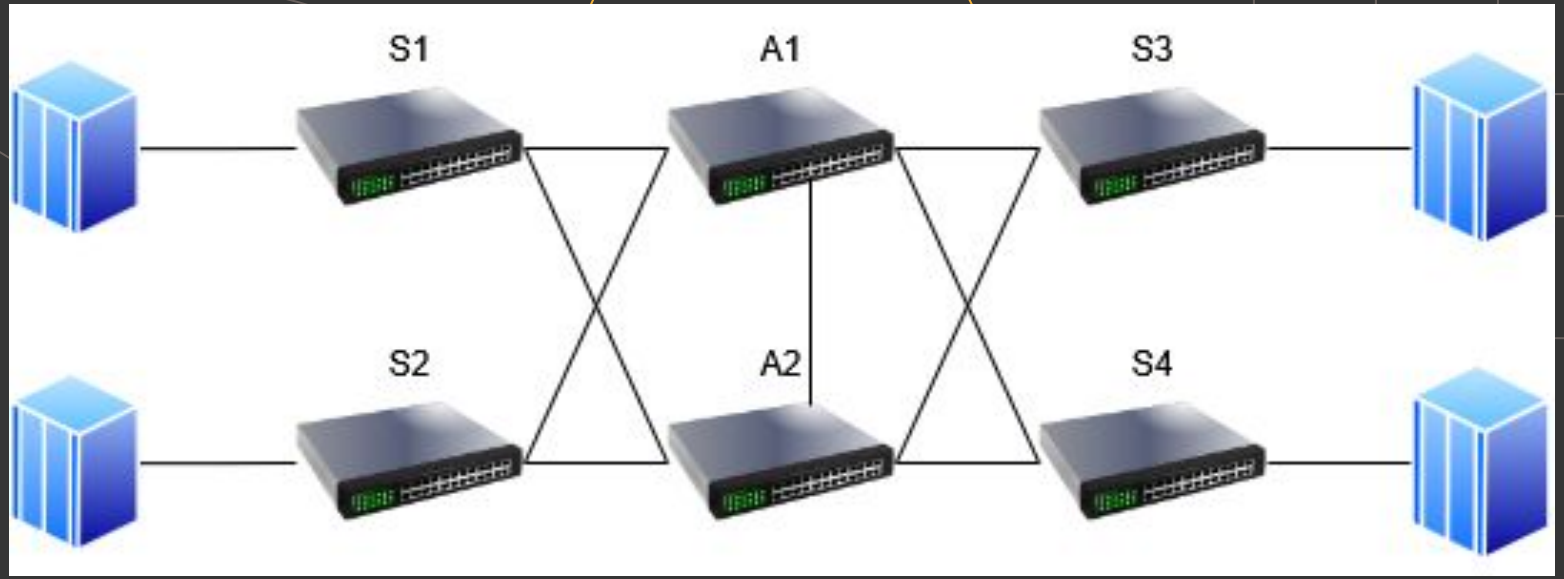
- Temat projektu
- Topologia testowa
- Generator ruchu
- Algorytm

Temat projektu

Stworzenie systemu do dynamicznego pomiaru ruchu na przełącznikach OpenFlow, poprzez zmianę częstotliowści pomiarów.

Celem jest ograniczenie narzutu na płaszczyznę sterowania, przy jednoczesnej wysokiej dokładności pomiarów w momentach wzmożonego ruchu.

Topologia



Generator ruchu

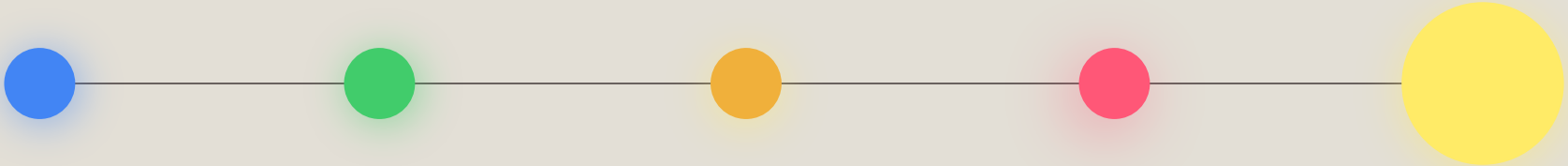
```python
def start_generators():
    setup_script()
    print("=== STARTING GENERATORS ===")

    processes = []

    for i, h in enumerate(HOSTS):
        targets = ["10.0.0.{}".format(j+1) for j in range(len(HOSTS)) if HOSTS[j] != h]
        target_str = ",".join(targets)

        pid = get_host_pid(h)
        if not pid:
            continue

        cmd = ["mnexec", "-a", pid, "python3", SCAPY_SCRIPT_PATH, target_str, str(PORT)]

        print("[{} PID={}] -> Targets: {}".format(h, pid, targets))

        devnull = open(os.devnull, 'w')
        p = subprocess.Popen(cmd, stdout=devnull, stderr=devnull)
        processes.append(p)

    print("=== Started {} generators in background ===".format(len(processes)))
    print("Press Ctrl+C to stop the generators...")

    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        print("\nShutting down generators...")
        for p in processes:
            p.terminate()
```

```python
SCAPY_CODE = r"""
import sys, random
from scapy.all import *

targets = sys.argv[1].split(',')
my_port = int(sys.argv[2])
payload = b"X" * 1400

my_iface = [i for i in get_if_list() if "-eth0" in i][0]

conf.verb = 0
sock = conf.L2socket(iface=my_iface)

pkts = [Ether(dst="ff:ff:ff:ff:ff:ff")/IP(dst=t)/UDP(dport=my_port)/Raw(load=payload) for t in targets]

print("Start high-speed traffic on %s" % my_iface)

while True:
    try:
        sock.send(random.choice(pkts))
        time.sleep(0.0001) # around 33 Mb/s
    except:
        pass
"""
```

Algorytm

```
Ustaw CZAS_ODCZYTU_NISKI_RUCH = 5 sekund
Ustaw CZAS_ODCZYTU_WYSOKI_RUCH = 1 sekunda
Ustaw PROG_NISKI_RUCH = 10 Mb/s
Ustaw PROG_WYSOKI_RUCH = 20 Mb/s
Ustaw STAN_SIECI = "NORMALNY"
Ustaw OKNO_ANALIZY = 60 sekund
Ustaw CZAS_ODCZYTU = CZAS_ODCZYTU_NISKI_RUCH
TABLICA_RAW = pusty zbiór

PĘTLA nieskończona:
  Dla każdego przełącznika agregującego w sieci:
    DANE_CHWILOWE = ODCZYTAJ_STATYSTYKI(Si)
    Dodaj DANE_CHWILOWE do TABLICA_RAW
  Czekaj(CZAS_ODCZYTU)

  Jeżeli minął czas OKNO_ANALIZY:
    SREDNIA_PRZEPUSTOWOSC =
      OBLICZ_SREDNIA_WAZONA(TABLICA_RAW)
    WYCZYŚĆ TABLICA_RAW

    Jeżeli STAN_SIECI == "NORMALNY" oraz SREDNIA_PRZEPUSTOWOSC > PROG_WYSOKI_RUCH:
      STAN_SIECI = "KRYTYCZNY"
      CZAS_ODCZYTU = CZAS_ODCZYTU_WYSOKI_RUCH

    Jeżeli STAN_SIECI == "KRYTYCZNY" oraz SREDNIA_PRZEPUSTOWOSC < PROG_NISKI_RUCH:
      STAN_SIECI = "NORMALNY"
      CZAS_ODCZYTU = CZAS_ODCZYTU_NISKI_RUCH

Koniec pętli
```

Prezentacja live



Dokumentacja w repozytorium

https://github.com/KarolMakara/dynamic-network-monitoring/blob/main/README.md

Dziękujemy!