

MiniProjeto: Medição Ativa em Redes com o Iperf - ADS29009

Jéssica Gomes Carrico E Leonardo Ludvig Silva

IFSC - Campus São José

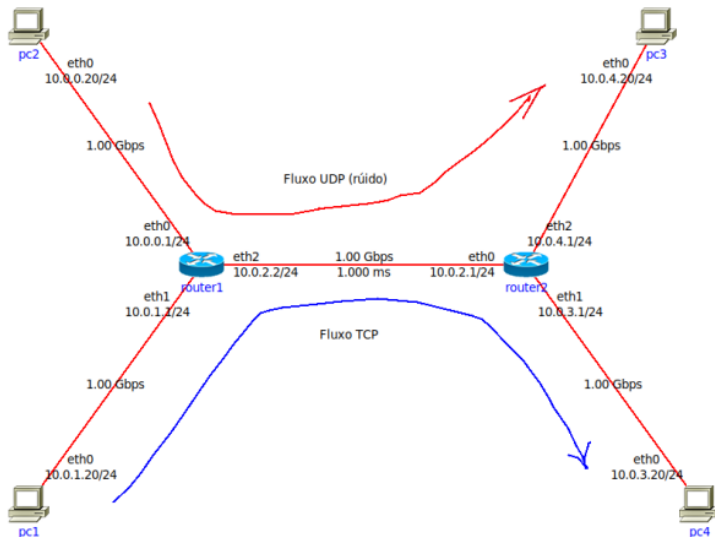
Abril de 2025

Investigar, por meio de medições ativas com o **iperf**, como a vazão de uma conexão TCP é afetada por:

- O **tamanho do buffer de envio TCP**;
- O **atraso de rede** (delay);
- E verificar se há **interação entre os fatores**.

- Emulador **Imunes**;
- Topologia com 2 PCs clientes, 2 servidores e 2 roteadores;
- Geração de tráfego TCP com **iperf**;
- Delay simulado com **vlink**;
- Script de automação em **Python**.

Cenário



Cenário da rede no IMUNES.

- **Buffer de envio TCP:**

- 64 KB (baixa)
- 208 KB (alta)

- **Delay de rede:**

- 10 ms (baixa)
- 100 ms (alta)

Execução do Experimento

- Cada combinação executada **8 vezes**;
- Ruído UDP gerado paralelamente;
- Vazão média extraída e registrada;
- Resultados armazenados em CSV.
- Intervalo de confiança de 95% calculado no Excel;

Script Python - Parte 1/3

```
1 def run_command(cmd):
2     print(f"\nExecutando: {cmd}")
3     subprocess.run(cmd, shell=True)
4
5 def get_ip(host):
6     result = subprocess.check_output(
7         f"sudo himage {host} ip -4 addr show eth0", shell=True
8     ).decode()
9     for line in result.splitlines():
10         if "inet" in line:
11             return line.strip().split()[1].split('/')[0]
12     return None
13
14 def get_eid(file: str) -> str:
15     output = subprocess.check_output(
16         f"sudo imunes -b {file} | grep 'ID'", shell=True
17     ).decode("utf-8")
18     eid = output.split("\n")[-2].split(" = ")[1]
19     return eid
20
21 def gera_ruido_udp(destino, duracao_total):
22     elapsed = 0
23     while elapsed < duracao_total:
24         dur = random.randint(2, 6)
25         taxa = random.randint(1, 10)
26         print(f"[ UDP ] {taxa} Mbps por {dur} s")
27         run_command("sudo himage {pci} iperf -u -c {destino} -t {dur} -b {taxa}M")
28         time.sleep(dur)
29         elapsed += dur
```



Script Python - Parte 2/3

```
1 def salva_csv(buffer, delay, repeticao, output):
2     with open("resultadosComRepeticao.csv", "a") as f:
3         f.write(f"Buffer={buffer},Delay={delay},Repeticao={repeticao}\n")
4         f.write(output)
5         f.write("\n")
6
7 file = "/home/aluno/labADS/topologiaLAB1.imn"
8 run_command("sudo imunes -b /home/aluno/labADS/topologiaLAB1.imn")
9
10 buffers = [64000, 208000]
11 delays = ['10000', '100000']
12 repeticoes = 8
13
14 exec_id = get_eid(file)
15 pc1 = f"pc1@{exec_id}"
16 pc2 = f"pc2@{exec_id}"
17 pc3 = f"pc3@{exec_id}"
18 pc4 = f"pc4@{exec_id}"
19
20 router1 = "router1"
21 router2 = "router2"
22 pc3_ip = get_ip(pc3)
23 pc4_ip = get_ip(pc4)
24
25 run_command(f"sudo himage {pc3} iperf -s -u &")
26 run_command(f"sudo himage {pc4} iperf -s &")
27 time.sleep(2)
```


Script Python - Parte 3/3

```
1 for buffer in buffers:
2     for delay in delays:
3         for i in range(repeticoes):
4             print(f"\n=== Teste {i+1} | Buffer: {buffer} | Delay: {delay} ===")
5             run_command(f"sudo vlink -e {exec_id} -dly {delay} router1:router2")
6             output = subprocess.check_output(
7                 f"sudo himage {pc2} iperf -c {pc4_ip} -n 100M -w {buffer} -i 1 -y C"
8                 ,
9                 shell=True
10            ).decode("utf-8")
11            salva_csv(buffer, delay, i + 1, output)
12            gera_ruido_udp(pc3_ip, 10)
13            time.sleep(2)
```

Cálculo do Intervalo de Confiança (IC)

Para cada configuração, o intervalo de confiança para a vazão média foi calculado com base em 8 repetições, utilizando a fórmula:

$$IC = \bar{x} \pm t_{n-1, \alpha/2} \cdot \frac{s}{\sqrt{n}}$$

Onde:

- \bar{x} : média das repetições
- s : desvio padrão amostral
- n : número de repetições ($n = 8$)
- $t_{n-1, \alpha/2}$: valor crítico da distribuição t de Student

Observação:

- Usa-se a distribuição t de Student pois $n \leq 30$
- $t = 2,365$

Resultados - Vazão com IC

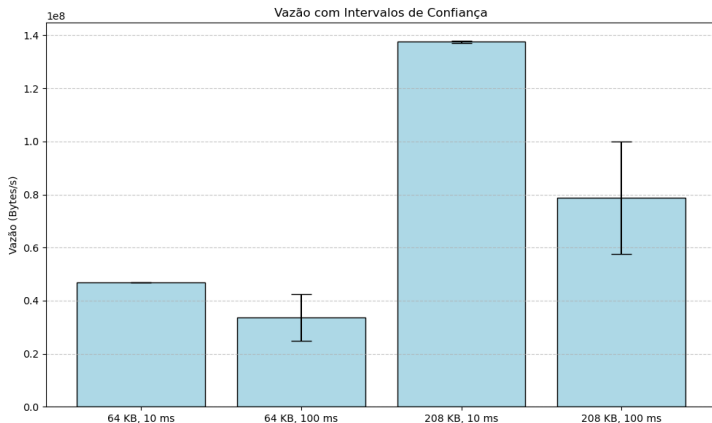


Gráfico de barras com média de vazão e intervalo de confiança.

Vazões Médias e Intervalos de Confiança

Execução	Vazão Média (Mbps)	IC Inferior (Mbps)	IC Superior (Mbps)
64 KB, 10 ms	46,81	46,74	46,89
64 KB, 100 ms	33,61	24,83	42,40
208 KB, 10 ms	137,51	137,21	137,81
208 KB, 100 ms	78,73	57,59	99,86

Cada valor representa a média de 8 repetições para a respectiva combinação de buffer e atraso.

Efeito do Buffer:

- Buffers maiores aumentam a vazão.

Efeito do Delay:

- Maior delay reduz a vazão.

Interação:

- O efeito do buffer é mais expressivo com menor atraso.