



MiniProjeto : Medição Ativa em Redes com o Iperf

Professor: Prof. Eraldo Silveira e Silva

<https://www.sj.ifsc.edu.br>

1 Objetivo Geral

Este miniprojeto tem como objetivos aplicar conceitos de medição ativa em um cenário de redes usando:

- a ferramenta iperf;
- a ferramenta mininet;
- automação de tarefas via script python e comandos do simulador mininet;
- conceitos de intervalo de confiança e
- projeto fatorial 2^k e boas práticas de projeto de experimentos.

Nota

Três cenários serão propostos a fim de serem divididos entre grupos de alunos (duplas).

O cenário base comum é:

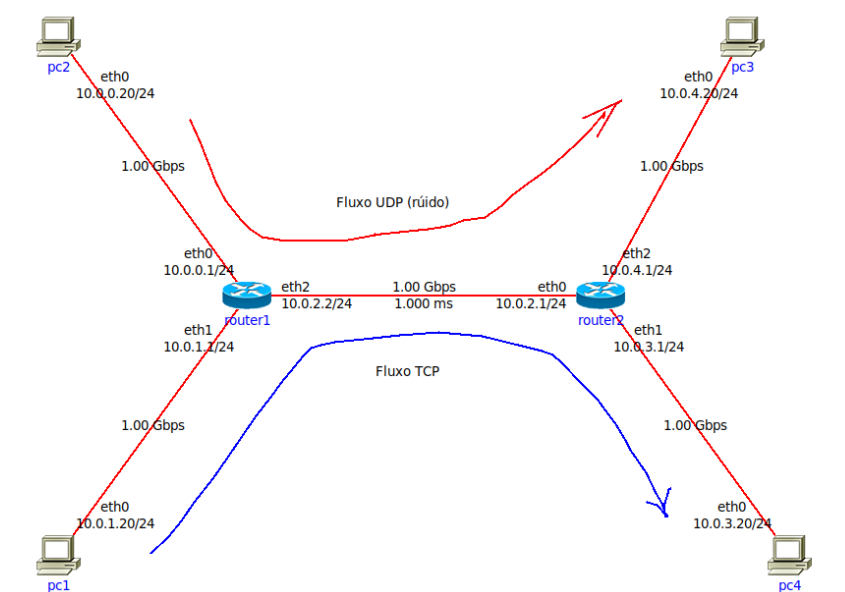


Figura 1: Legenda descritiva da figura

2 Ferramentas Imunes

Para a automação da execução do experimento recomenda-se usar as ferramentas disponíveis no imunes:

- **himage**: comando para execução de comando na máquina do cenário simulado a partir da máquina hospedeira;
- **vlink**: comando para mudar configurações de link no cenário simulado a partir da máquina hospedeira.
- **hcp**: comando para cópia de arquivos entre máquinas simuladas e hospedeiro.

Exemplo 1: executar o iperf servidor na máquina pc2:

```
1 sudo himage pc2@i2520 iperf -s
```

O identificador i2520 identifica o ID do experimento que pode ser visto na tela do Imunes, na barra de baixo.

Exemplo 2: mudar a configuração BER (Bit Error Rate) do link router2:pc2

```
2 sudo vlink -BER 1000000 router2:pc2@i2520
```

Notar que a modificação da configuração do link não aparece na interface gráfica do imunes.

Uma visão completa de comandos do imunes para automação de experimentos pode ser vista aqui [wiki do imunes](#)

3 Descrição do Cenário 1

3.1 Objetivo

Investigar, por meio de medição ativa com `iperf`, como a vazão de uma conexão TCP é afetada por dois fatores:

- O tamanho do **buffer de envio TCP** (parâmetro `-w` no cliente);
- O **atraso de rede** (emulado com comandos de mudança de atributos de link noo imunes).

Além disso, verificar se há interação entre os fatores na determinação da vazão.

3.2 Fatores e Níveis

Fator	Nível Baixo	Nível Alto
A: Buffer de envio TCP (<code>-w</code>)	64 KB (janela efetiva de 128 KB)	208 KB (janela efetiva de 416 KB)
B: Atraso de rede (RTT)	10 ms (ida \Rightarrow RTT20ms)	100 ms (ida \Rightarrow RTT200ms)

3.3 Métrica Avaliada

- Vazão média (em Mbps), medida ao final de cada execução do `iperf`.

Configuração Experimental

- Utilize o `iperf` em modo TCP, com carga fixa: `-n 100M`.

```
3 iperf -c <IP_DO_SERVIDOR> -n 100M -w <BUFFER> -i 1 -v
```

- Adicione atraso com `vlink` do imunes:

```
4 sudo vlink -dly 30000 router2:pc2@i2520
```

3.4 Execuções e replicações

Realize todas as quatro combinações de fatores:

Execução	-w	Delay (ms)	Vazão Esperada (teórica)
1	64 KB	10	Moderada
2	64 KB	100	Baixa
3	208 KB	10	Alta
4	208 KB	100	Moderada

Os experimentos deverão ser repetidos 8 vezes sendo feita então, a média da métrica final. O intervalo de confiança para 95% deverá ser calculado.

3.5 Entregáveis

- Script Python para automação do processo e geração de tabelas e gráficos.
- Tabela com os resultados obtidos (vazão em cada combinação).
- Gráfico barras com as vazões e intervalos de confiança.
- Discussão sobre os efeitos principais e possível interação.

4 Descrição do Cenário 2

4.1 Objetivo

Avaliar, por meio de medição ativa com `iperf`, como a vazão de uma conexão TCP é afetada por dois fatores:

- O tamanho do **buffer de recepção TCP** (parâmetro `-w` no servidor);
- A **perda de pacotes**, emulada com comando `vlink`.

Além disso, verificar se há interação entre os fatores na determinação da vazão.

4.2 Fatores e Níveis

Fator	Nível Baixo	Nível Alto
A: Buffer de recepção TCP (-w)	64 KB (janela efetiva ã 128 KB)	208 KB (janela efetiva ã 416 KB)
B: Perda de pacotes (tc loss)	0.0%	1.0%

4.3 Métrica Avaliada

- Vazão média (em Mbps), medida no cliente ao final da execução.

4.4 Configuração Experimental

- No servidor:

```
5 iperf -s -w <BUFFER>
```

- No cliente:

```
6 iperf -c <IP_DO_SERVIDOR> -n 100M -i 1
```

- Configurar as perdas com o comando vlink. Exemplo:

```
7 vlink -BER 1000000 router1:pc1
```

4.5 Execuções

Realize todas as quatro combinações de fatores:

Execução	Buffer no servidor	Perda	Vazão Esperada
1	64 KB	0.0%	Boa
2	64 KB	1.0%	Redução acentuada
3	208 KB	0.0%	Melhor ainda
4	208 KB	1.0%	Redução moderada

Os experimentos deverão ser repetidos 8 vezes sendo feita então, a média da métrica final. O intervalo de confiança para 95% deverá ser calculado.

4.6 Entregáveis

- Script Python de tarefas automatizadas comentado
- Tabela com os resultados obtidos (vazão em cada combinação).
- Gráfico de barras com as vazões e o intervalo de confiança.
- Discussão sobre os efeitos principais e possível interação.

5 Descrição do Cenário 3

5.1 Objetivo

Avaliar, por meio de medição ativa com `iperf`, como a vazão de uma conexão TCP é afetada por dois fatores:

- O número de **fluxos TCP paralelos** (parâmetro `-P` no cliente);
- O **retardo de rede**, emulado com o comando `vlink`.

Além disso, verificar se há interação entre os fatores na determinação da vazão.

5.2 Fatores e Níveis

Fator	Níveis Baixo	Níveis Alto
A: Paralelismo TCP (<code>-P</code>)	1 fluxo	4 fluxos
B: Retardo de rede (RTT)	10 ms (ida \Rightarrow RTT20ms)	100 ms (ida \Rightarrow RTT200ms)

5.3 Métrica Avaliada

- Vazão total (em Mbps), somando todos os fluxos, medida no cliente ao final da execução.

5.4 Configuração Experimental

- No servidor:

```
8 iperf -s
```

- No cliente:

```
9 iperf -c <IP_DO_SERVIDOR> -n 100M -P <N> -i 1
```

- Para simular retardo:

```
10 sudo vlink -dly 30000 router2:pc2@i2520
```

5.5 Execuções

Realize todas as quatro combinações de fatores:

Execução	Fluxos (<code>-P</code>)	Delay (ms)
1	1	10
2	1	100
3	4	10
4	4	100

Os experimentos deverão ser repetidos 8 vezes sendo feita então, a média da métrica final. O intervalo de confiança para 95% deverá ser calculado.

5.6 Entregáveis

- Script Python para automação do processo e geração de tabelas e gráficos.
- Tabela com os resultados obtidos (vazão em cada combinação).
- Gráfico barras com as vazões.
- Discussão sobre os efeitos principais e possível interação.

6 Automação de Tarefas e Uso do Simulador Imunes

ESQUELETO SCRIPT DE REFERÊNCIA

Como será necessário repetir o experimento para 8 configurações diferentes, recomenda-se a construção de um script python.

Notar que os dados foram armazenados em um arquivo de dados.csv. Alternativas para facilitar o posterior processamento seria: Armazenar os dados em arquivos que identifiquem o experimento: Exemplo> dados-CUBIC-BER100000-r0 Onde r0 identifica a replicação 0 do experimento Armazenar todos os dados em um arquivo com append. Neste caso, a identificação do experimento deve ser acrescentada em cada linha de saída para facilitar o posterior processamento.

- Inicia todos fluxos repetindo para cada combinação de fator:

```
11 import subprocess
12
13 ##### inicializacao de dados
14
15 alg = ['cubic', 'reno']
16 BER = ['100000', '1000000']
17 e2e_delay = ['10000', '100000']
18
19 repeticao = 8
20
21 cmd_iperf_client="sudo himage pc1@i3f30 iperf -c 10.0.0.21 -y C -Z
    " + alg[0] + " > dados-" + alg[0] + ".csv"
22
23 ##### execucao dos experimentos
24
25 #para cada uma das repeticoes faca
26 for rep in range(repeticao):
27     for proto in alg:
28         for ber in BER:
29             for e2e in e2e_delay:
30                 subprocess.run("sudo vlink -BER " + ber + " pc1:pc2@i3f30
    ", shell=True)
31
32 #configure o link usando o comando vlink
33
34 #execute o iperf cliente salvando os dados em arquivo separado (?)
35 #print(cmd_iperf_client)
36 subprocess.run(cmd_iperf_client, shell=True)
37
```

```
38 OBSERVAÇ ES
39 É interessante resetar qualquer memória de configuração de sessões:
40 sysctl -w net.ipv4.tcp_no_metrics_save=1
```

Referências

- [1] IPERF. Ferramenta para medição ativa de desempenho em redes TCP/UDP. Disponível em: <https://iperf.fr>. Acesso em: abr. 2025.
- [2] TANENBAUM, A. S.; WETHERALL, D. J. *Redes de Computadores*. 5. ed. Pearson, 2011.
- [3] IMUNES – Integrated Multiprotocol Network Emulator/Simulator. Repositório oficial GitHub. Disponível em: <https://github.com/imunes/imunes>. Acesso em: abr. 2025.