



**Escola de Engenharia**  
**Curso de Bacharelado em Engenharia de Sistemas**

**DCC218 - Introdução a Sistemas Computacionais**  
**Relatório do TP 1 - Parte 1**

Raphael Henrique Braga Leivas - 2020028101

Belo Horizonte  
13 de outubro de 2025

## SUMÁRIO

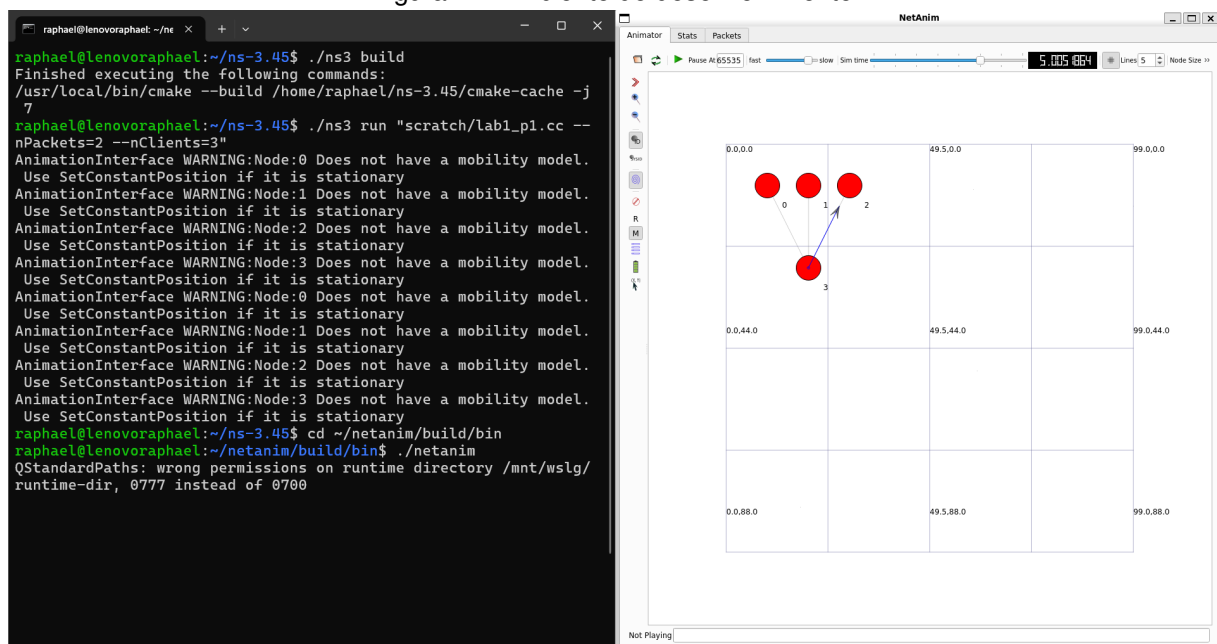
1	AMBIENTE DE DESENVOLVIMENTO . . . . .	3
2	PARTE 1: COMUNICAÇÃO PONTO-A-PONTO . . . . .	4
3	PARTE 2: REDE ETHERNET . . . . .	7
4	PARTE 3: REDE WIFI . . . . .	9
5	CONCLUSÃO . . . . .	11

## 1 AMBIENTE DE DESENVOLVIMENTO

O projeto é desenvolvido em um notebook Lenovo com processador Intel-Core i5 de 11ª geração, com arquitetura x64. A frequência do processador é 2.4 GHz e possui 4 cores. O processador possui 8 GB de memória RAM disponível. O sistema operacional do computador é Windows 11, mas o ambiente NS-3 é instalado através de WSL (Windows Subsystem for Linux) Ubuntu 22.04.1 LTS.

O ambiente NS-3 é instalado na pasta `home` do WSL. A Figura 1 ilustra o ambiente de desenvolvimento utilizado no projeto.

Figura 1 – Ambiente de desenvolvimento.



Fonte: Elaboração própria.

Essencialmente, é adotado um procedimento de duas etapas:

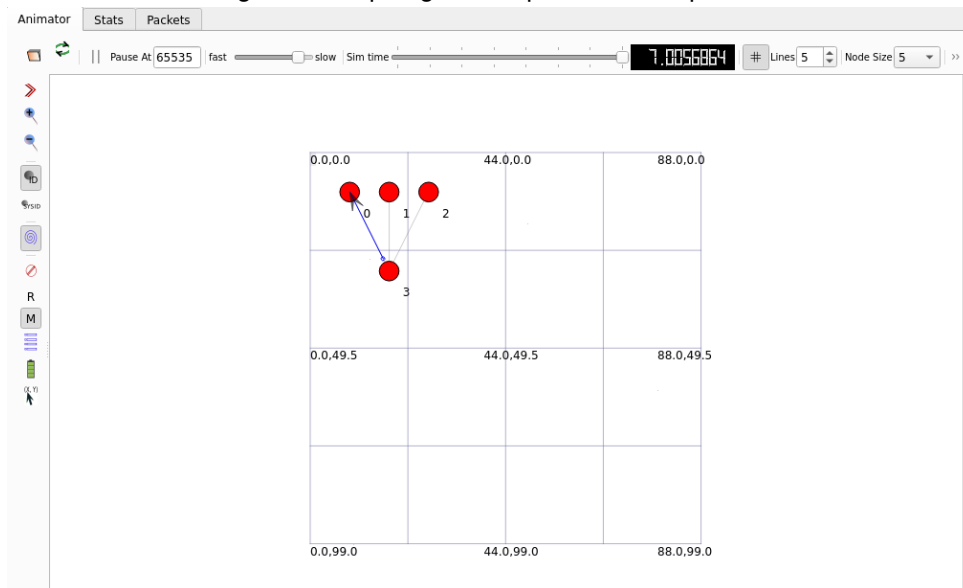
1. O comando `./ns3 build`, seguido de `./ns3 run` com o programa e argumentos de linha de comando desejados, é usado para executar o script C++ com as funções do NS-3. A saída dos comandos será um arquivo XML;
2. O comando `./netanim` é usado para visualizar o arquivo XML exportado na etapa 1, obtendo a animação exibida no lado direito da Figura 1.

## 2 PARTE 1: COMUNICAÇÃO PONTO-A-PONTO

O código completo da Parte 1 pode ser visto no seguinte link: <[https://github.com/RaphaelLeivas/redes-2025-2/blob/main/Lab1\\_Raphael\\_Leivas/Part1/lab1\\_p1.cc](https://github.com/RaphaelLeivas/redes-2025-2/blob/main/Lab1_Raphael_Leivas/Part1/lab1_p1.cc)>.

Para a parte 1, o objetivo é estabelecer conexões ponto-a-ponto entre clientes e um servidor. Usando `nPackets = 2` e `nClients = 3` nos argumentos da linha de comando, é obtido a topologia exibida na Figura 2.

Figura 2 – Topologia do experimento da parte 1.



Fonte: Elaboração própria.

A Figura 3 mostra os envios dos pacotes dos nós clientes para o nó servidor ao longo do tempo, com três clientes e cada um enviando dois pacotes.

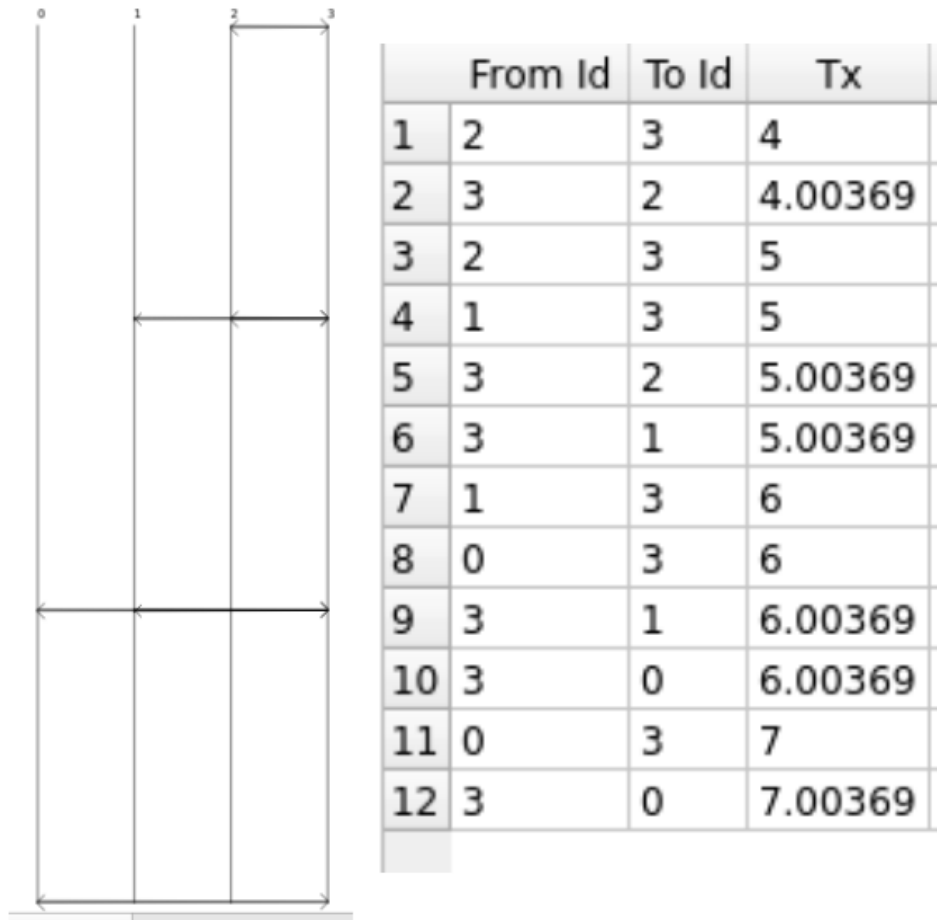
É possível observar que o tempo de propagação é muito menor do que o tempo de espera para envio dos pacotes. O tempo de RTT é de 3.69 ms, enquanto 1 segundo é esperado para enviar cada pacote. Em cada envio, o cliente envia para o nó 3 (servidor), e o servidor responde de volta para o nó cliente (0 a 2) que o requisitou.

Para tornar o gráfico mais didático, é possível aumentar o tempo de propagação para 200 ms e reduzir a largura de banda para 5 kbps, obtendo o resultado da Figura 4. Vemos que o pacote gasta mais tempo para sair do cliente e chegar ao servidor, e que o servidor demora 1.88 segundos a enviar a resposta ao cliente. Esse valor de atraso é dado pela expressão que vimos em sala de aula:

$$T_{prop} + \frac{tamanho}{banda} = 0.2 + \frac{1024 \cdot 8 \text{ bits}}{5 \cdot 10^3 \text{ kbps}} = 1.83 \text{ s}$$

uma vez que o pacote enviado tem 1024 bytes de tamanho. Vemos que os valores calculado e teórico são condizentes.

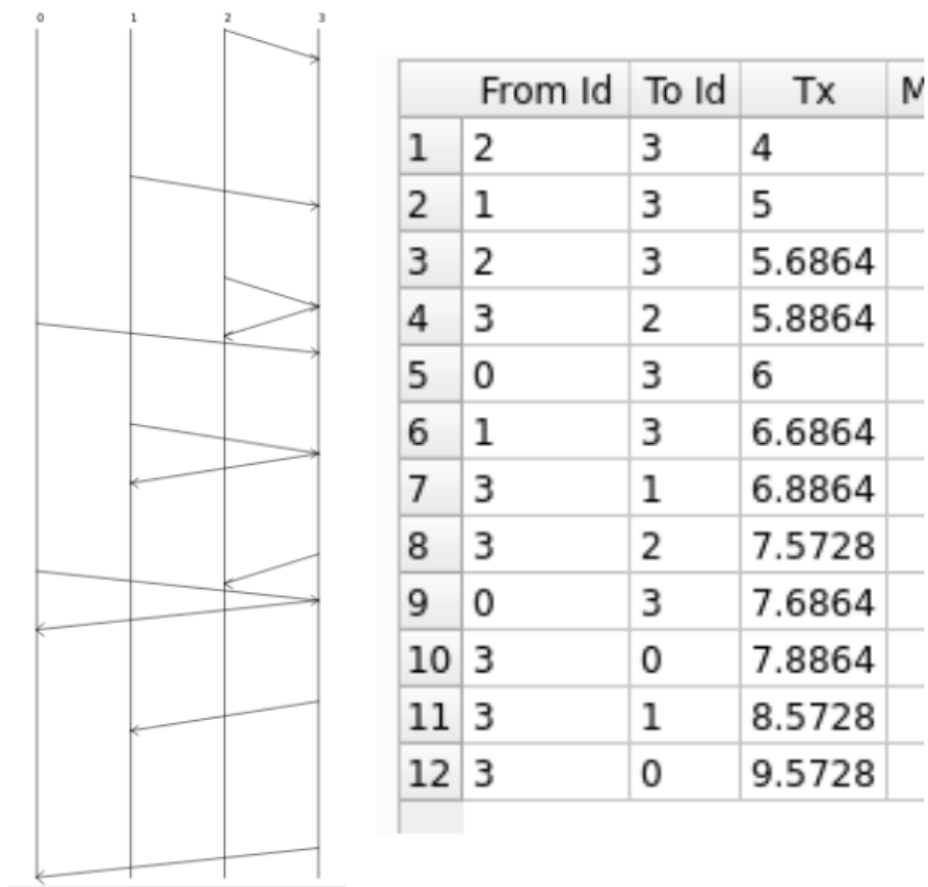
Figura 3 – Resultado do experimento da parte 1.



Fonte: Elaboração própria.

Na pasta compactada no Moodle, está presente um teste com 5 clientes e cada um enviando 4 pacotes, bem como o print com as linhas de comando e arquivo txt.

Figura 4 – Resultado do experimento da parte 1 - tempo de propagação 200 ms e largura de banda 5 kbps.



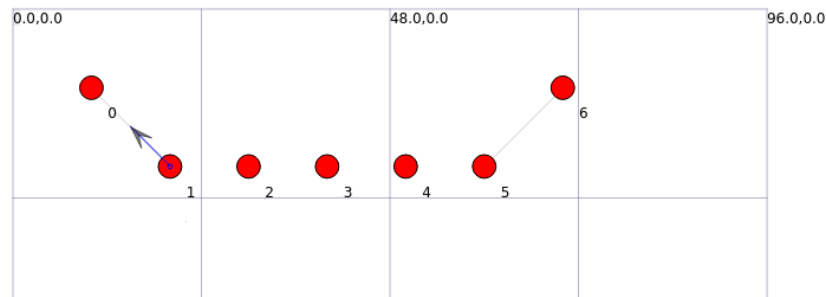
Fonte: Elaboração própria.

### 3 PARTE 2: REDE ETHERNET

O código completo da Parte 2 pode ser visto no seguinte link: <[https://github.com/RaphaelLeivas/redes-2025-2/blob/main/Lab1\\_Raphael\\_Leivas/Part2/lab1\\_p2.cc](https://github.com/RaphaelLeivas/redes-2025-2/blob/main/Lab1_Raphael_Leivas/Part2/lab1_p2.cc)>.

A topologia implementada na parte 2 está exibida na Figura 5. Basicamente, o nó 0 quer se comunicar com o servidor (nó 6). Para isso, ele usa 4 nós CSMA intermediários que enviam a requisição ao servidor final, representando o meio Ethernet. Nós CSMA utilizam o protocolo Carrier-Sense Multiple-Access (CSMA) que reduz o risco de colisões entre os pacotes enviados simultaneamente por vários nós no mesmo meio.

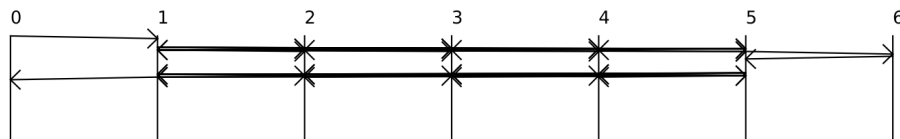
Figura 5 – Topologia da parte 2.



Fonte: Elaboração própria.

Utilizando 4 nós CSMA intermediários enviando 10 pacotes, a sequência de envio dos pacotes em função do tempo está exibida na Figura 6.

Figura 6 – Envio dos pacotes na topologia da parte 2.

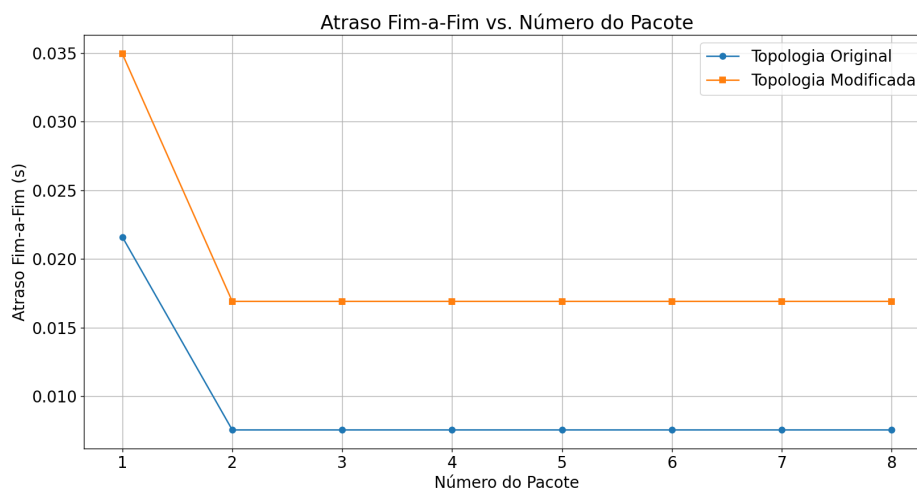


Fonte: Elaboração própria.

O gráfico do atraso de fim-a-fim de uma requisição versus o número do pacote está exibido na Figura 7. A topologia original é a presente no exemplo `second.cc` da documentação do NS-3, enquanto a topologia modificada é a exibida na Figura 5 desenvolvida na parte 2. O gráfico foi obtido a partir do seguinte procedimento:

1. Modifica os códigos para que eles salvem os atrasos em um arquivo txt com as funções prontas de `TxTrace` e `RxTrace`;
2. Carrega os arquivos txt em um arquivo Python e faz os plots com a biblioteca `matplotlib`.

Figura 7 – Atraso fim-a-fim em função do número do pacote para cada uma das topologias.



Fonte: Elaboração própria.

É possível observar que o atraso na topologia original é menor do que na modificada. Isso é conforme esperado, uma vez que a topologia modificada possui um nó a mais, que aumenta o tempo de propagação. O fato de o aumento no atraso ser constante ao longo dos pacotes reforça essa hipótese.

Além disso, o atraso do primeiro pacote é maior do que os demais. Analisando os arquivos PCAP gerados para ambas topologias, vemos que isso é causado pelo tempo de inicialização necessário para estabelecer a conexão, como a criação do socket UDP e conexão com a porta do servidor. A Figura 8 ilustra esse efeito nos arquivos PCAP. Temos um atraso de 0.0349 segundos no primeiro pacote, e 0.0169 segundos nos demais.

Figura 8 – Atraso do primeiro pacote nos arquivos PCAP.

```
raphael@lenovoraphael:~/ns-3.45$ tcpdump -r first_p2p_net-trace-0-1.pcap
reading from file first_p2p_net-trace-0-1.pcap, link-type PPP (PPP), snapshot length 65535
21:00:02.000000 IP 10.1.1.1.49153 > 10.1.3.2.discard: UDP, length 1024
21:00:02.034941 IP 10.1.3.2.discard > 10.1.1.1.49153: UDP, length 1024
21:00:03.000000 IP 10.1.1.1.49153 > 10.1.3.2.discard: UDP, length 1024
21:00:03.016916 IP 10.1.3.2.discard > 10.1.1.1.49153: UDP, length 1024
21:00:04.000000 IP 10.1.1.1.49153 > 10.1.3.2.discard: UDP, length 1024
21:00:04.016916 IP 10.1.3.2.discard > 10.1.1.1.49153: UDP, length 1024
```

Fonte: Elaboração própria.



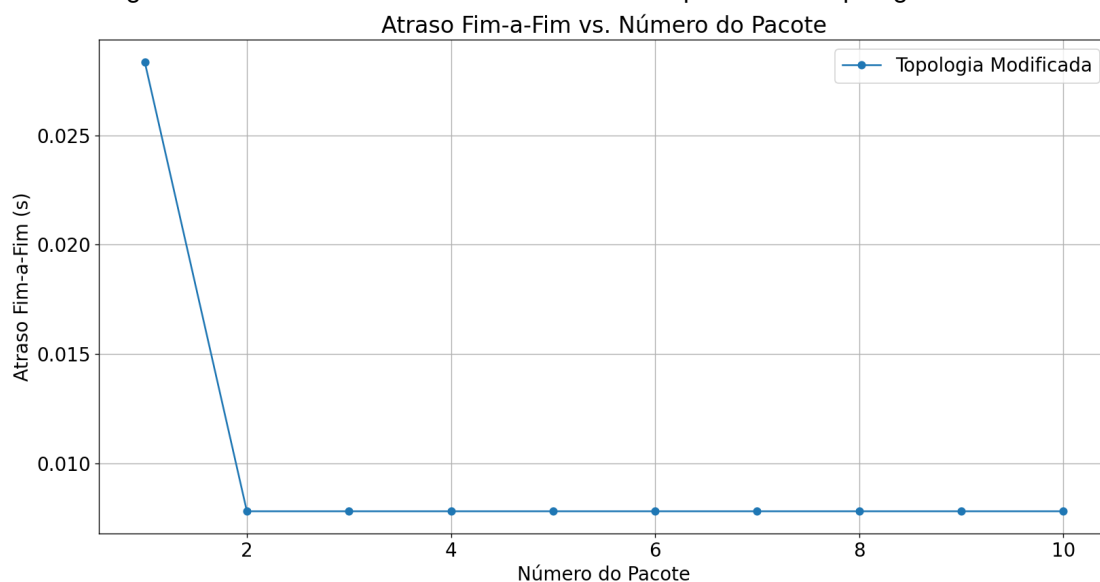


Figura 10 – Logs com timestamps de envio e recebimento dos pacotes - parte 3.

```
raphael@lenovoraphael:~/ns-3.45$ ./ns3 run "scratch/lab1_p3_alt.cc --nWifi=4 --nPackets=10"
[ 0%] Building CXX object scratch/CMakeFiles/scratch_lab1_p3_alt.dir/lab1_p3_alt.cc.o
[ 0%] Linking CXX executable /home/raphael/ns-3.45/build/scratch/ns3.45-lab1_p3_alt-default
At time +2s client sent 1024 bytes to 10.1.4.4 port 9
At time +2.01458s server received 1024 bytes from 10.1.3.1 port 49153
At time +2.01458s server sent 1024 bytes to 10.1.3.1 port 49153
At time +2.02834s client received 1024 bytes from 10.1.4.4 port 9
At time +3s client sent 1024 bytes to 10.1.4.4 port 9
At time +3.00387s server received 1024 bytes from 10.1.3.1 port 49153
At time +3.00387s server sent 1024 bytes to 10.1.3.1 port 49153
At time +3.00781s client received 1024 bytes from 10.1.4.4 port 9
At time +4s client sent 1024 bytes to 10.1.4.4 port 9
At time +4.00386s server received 1024 bytes from 10.1.3.1 port 49153
At time +4.00386s server sent 1024 bytes to 10.1.3.1 port 49153
At time +4.00781s client received 1024 bytes from 10.1.4.4 port 9
At time +5s client sent 1024 bytes to 10.1.4.4 port 9
At time +5.00386s server received 1024 bytes from 10.1.3.1 port 49153
At time +5.00386s server sent 1024 bytes to 10.1.3.1 port 49153
At time +5.00781s client received 1024 bytes from 10.1.4.4 port 9
```

Fonte: Elaboração própria.

Figura 11 – Gráfico com os atrasos de envios de pacotes na topologia com Wi-Fi.



Fonte: Elaboração própria.

## 5 CONCLUSÃO

Na parte 1, foi possível simular os tempos de transmissão e propagação que vimos em sala de aula. Em particular, verificamos como os gráficos temporais que mostram os tempos de envio e recebimento dos pacotes em forma de barras verticais variam em função do atraso de propagação, largura de banda e tamanho do pacote. Já sabíamos que essa dependência existia devido à equação teórica, mas com o NS-3 foi possível simular isso numericamente com a ferramenta.

Também foi possível na parte 1 usar uma das equações que vimos em sala de aula e comparar o valor teórico com o valor simulado, sendo ambos bastante semelhantes.

Na parte 2, aprendemos sobre o papel que nós CSMA desempenham em redes cabeadas por Ethernet, evitando colisões ao enviar pacotes ao menos tempo. Além disso, em sala de aula havíamos visto apenas do ponto de vista teórico as ações iniciais que a rede realiza ao estabelecer uma conexão (criar sockets, conectar-se à porta). Com o NS-3, vimos como isso afeta o atraso fim-a-fim do primeiro pacote, estabilizando-se nos pacotes seguintes.

Por fim, na parte 3 vimos como a mudança no meio de propagação de Ethernet para Wi-Fi afeta os atrasos nos envios de pacotes. Vimos também como todos os nós de uma rede usam o ponto de acesso (AP) da rede para comunicar-se com outra, algo que havíamos visto apenas do ponto de vista teórico em sala de aula, sendo possível verificar isso na prática com o NS-3.