

Laboratório Casa 01

Pedro Henrique Dornelas Almeida
Dept. de Engenharia Elétrica (FT-ENE)
Universidade de Brasília (UnB)
Brasília, Brasil
phdornelas.almeida@gmail.com

Resumo—Este relatório tem como objetivo mostrar um experimento realizado no NS-3, as ferramentas envolvidas para análise de resultados e os passos envolvidos no processo.

Index Terms—NS-3, WireShark, Dumbbell, TraceMetrics, Atraso, Perda de Pacotes.

I. OBJETIVOS

O objetivo deste experimento é para ambientar ao ambiente de simulação NS-3, bem como o uso de outras ferramentas envolvidas, como o FlowMonitor, o WireShark, o TraceMetrics e o PyVis. Outro objetivo é conseguir construir diferentes topologias usando as classes, conceitos envolvidos ao NS-3, para então extrair informações como *throughput*, atraso e perda de pacotes.

II. INTRODUÇÃO TEÓRICA

Primeiramente, deve-se explorar as ferramentas que serão utilizadas no experimento. Utilizaremos o TraceMetrics, o PyVis, o WireShark e o FlowMonitor, que serão brevemente resumidas abaixo.

A. TraceMetrics

Esta ferramenta é um analisador de arquivos de rastreamento para o NS-3. O objetivo principal dele é fazer uma análise rápida do arquivo do rastreamento produzido pelas simulações do NS-3 e calcular métricas úteis para pesquisa e medição de desempenho. Essa ferramenta é necessária porque uma simulação de pesquisa pode gerar um arquivo de rastreamento com milhares de linhas, tornando-se difícil de analisar uma por uma, manualmente. Por isso, esta ferramenta é muito útil para pessoas que realizam vários experimentos. Esta ferramenta é open-source, desenvolvida em Java.

B. PyVis

O PyVis é uma biblioteca em Python que permite a visualização de dados de rede, de topologias, grafos, e redes iterativas. Assim é possível visualizar a rede quando se escreve o código de uma topologia, incluindo taxas de transmissão, sentido dos dados, dar zoom em alguma parte que se deseja observar com detalhes. Também é possível pausar a simulação e observar os pontos de interesse.

C. WireShark

O WireShark é uma ferramenta que permite a análise detalhada de pacotes, tráfego de rede, protocolos. Ele permite a utilização de uma interface gráfica bem intuitiva, com possibilidades de montar inúmeros filtros para se obter as informações que desejam. Ele é um sniffer, ou capturador, e é usado bastante por administradores de rede, em estudos de redes, para analisar desempenho, segurança, dentre várias outras aplicações.

D. FlowMonitor

Por sua vez, o FlowMonitor é um módulo utilizado no NS-3 que trás flexibilidade aos sistemas que se deseja mensurar, medir a performance de uma rede, de protocolos. O FlowMonitor coleta, analisa e monitora o tráfego dos dados dentro de uma simulação, gerando vários arquivos que permite ao desenvolvedor obter análises bem completas sobre a simulação feita, podendo gerar gráficos, histogramas, dentre outros formatos de dados.

III. DESCRIÇÃO DAS ATIVIDADES E ANÁLISES

A. Configuração do ambiente de simulação

Para realizar o experimento primeiro foi necessário construir uma topologia dumbbell com as seguintes características:

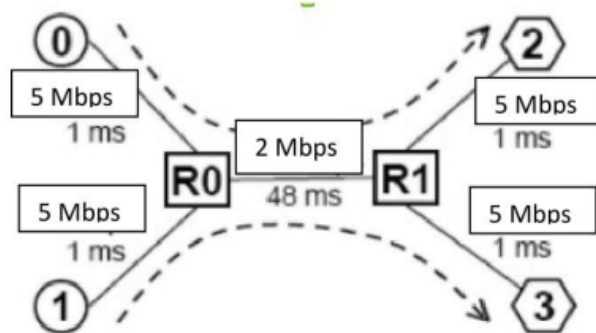


Figura 1. Topologia Dumbbell

Note da topologia que temos diferentes taxas de transmissão para os usuários finais e entre os roteadores, o que podemos pensar neste momento que causará algum tipo de gargalo caso os usuários finais utilizem mais da banda fornecida entre os roteadores.

As aplicações que rodaram nessa simulação tratam de 2 duas transferências de dados sob UDP, em que o nó 1 seria um cliente e o nó 2 seria o servidor e os dados vão fluir do nó 0 para o nó 2 passando pelo *backbone*. A outra aplicação irá rodar como cliente no nó 1 e como servidor o nó 3. Assim teremos os nossos fluxos de pacotes para as 2 diferentes aplicações.

Dos parâmetros da simulação temos que o tempo de simulação utilizado foi de 100 segundos, e as taxas de transmissão permitidas em cada link estão na figura acima (Fig. 1). As métricas a serem estudadas serão informações de taxa da aplicação, *throughput*, histogramas de atraso e *jitter* de cada fluxo. Tamanho médio da fila nos nós 0, 1 e R0.

B. QUESTÃO 1

Na questão 1 a topologia acima foi seguida para realizar a simulação. Após rodar o código construído para esta simulação no modo *-vis*, foi possível obter a seguinte visualização:

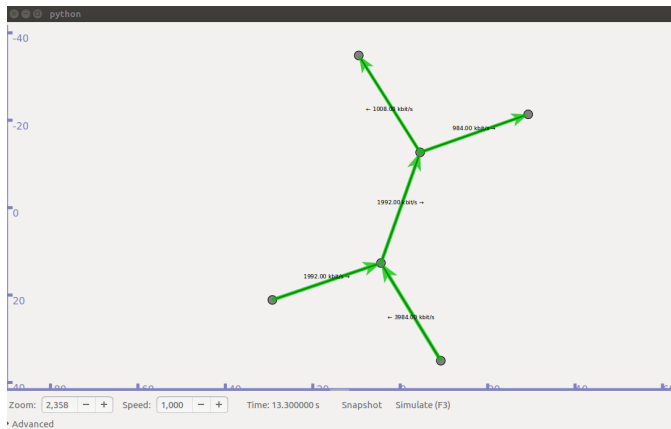


Figura 2. Questão 1 –vis

Também foi possível analisar a saída obtida no terminal:

```
ns3@ns3-VirtualBox: ~/ns-allinone-3.33/ns-3.33
Pacotes na fila: 99->100
Flow 1 (192.168.0.1:49153 -> 192.171.0.2:4444)
Tx Pacotes: 16650
Tx Bytes: 24941760
Tx Taxa de bits media: 1.99745e+06 bps
Rx Pacotes: 8296
Rx Bytes: 12427408
Rx Taxa de bits media: 995846 bps
Throughput medio (simulacao): 994193 bps
Atraso medio: 0.729222 s
Jitter medio: 2.02604 ms
Numero de pacotes perdidos: 8288
Flow 2 (192.169.0.1:49153 -> 192.172.0.2:4444)
Tx Pacotes: 33300
Tx Bytes: 49883400
Tx Taxa de bits media: 3.99479e+06 bps
Rx Pacotes: 8344
Rx Bytes: 12499312
Rx Taxa de bits media: 1.00173e+06 bps
Throughput medio (simulacao): 999945 bps
Atraso medio: 1.29251 s
Jitter medio: 3.4484 ms
Numero de pacotes perdidos: 24607
ns3@ns3-VirtualBox:~/ns-allinone-3.33/ns-3.33$
```

Figura 3. Questão 1 - Terminal

Pela figura anterior (Fig. 3) foi possível identificar os fluxos, o fluxo 1 foi o fluxo que parte do n0 até o n2, e o fluxo 2 parte do n1 até o n3. Nesta figura também é possível ver a taxa da aplicação, o *throughput*, o atraso total e o número de

pacotes para cada um dos fluxos de maneira bem intuitiva. Para o fluxo 1:

- Taxa da aplicação: 1992,00 kbps
- Throughput: 994kbps
- Atraso: 0,729222s
- Número de pacotes perdidos: 8288
- Jitter: 2,02604ms

Para o fluxo 2:

- Taxa da aplicação: 3997,00 kbps
- Throughput: 999kbps
- Atraso: 1,29251s
- Número de pacotes perdidos: 24607
- Jitter: 3,4484ms

Em seguida, deve-se analisar o *throughput* gerado pelo wireshark para cada um dos fluxos. Neste momento faremos uma filtragem pelo IP de origem e gerar o gráfico:

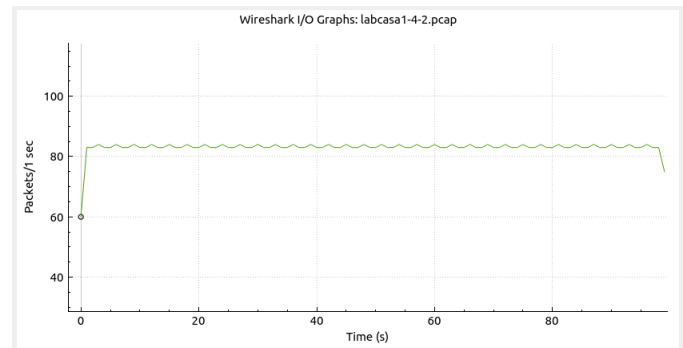


Figura 4. Fluxo 1 - Throughput

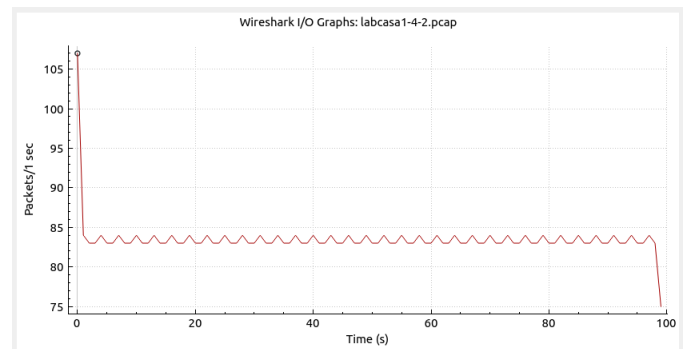


Figura 5. Fluxo 2 - Throughput

É possível também juntar os dois gráficos e ver como é a relação dos dois fluxos que compartilham o mesmo link para chegar ao seu destino:

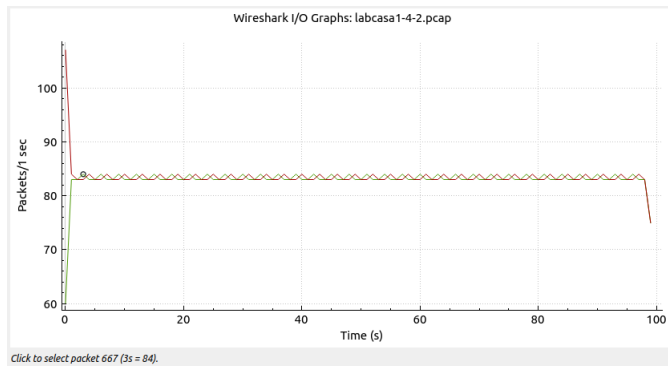


Figura 6. Fluxo 1 e Fluxo 2 - Throughput

O próximo passo é utilizar o TraceMetrics para obter o tamanho médio da fila nos nós n0, n1 e R0. Foi possível obter o seguinte:

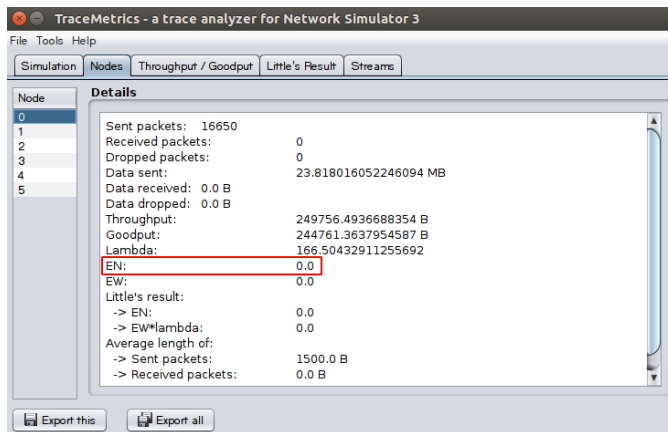


Figura 7. Tamanho médio da fila - Nó 0

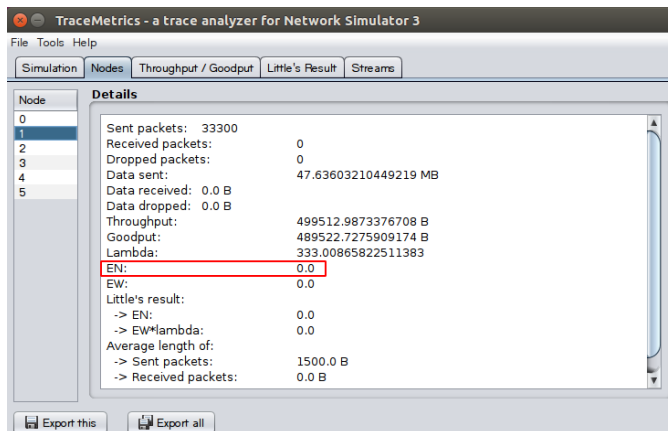


Figura 8. Tamanho médio da fila - Nó 1

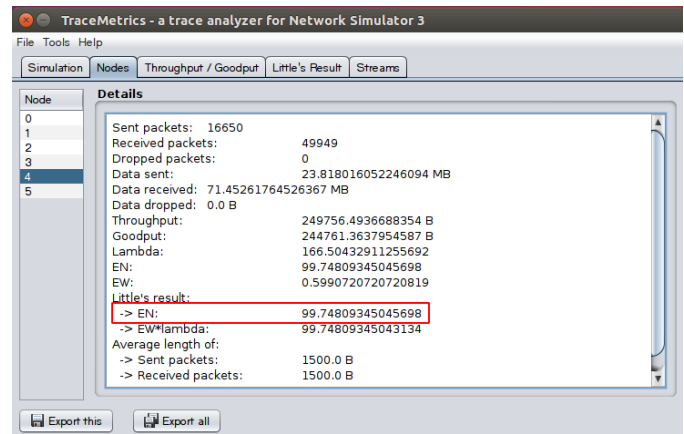


Figura 9. Tamanho médio da fila - R0

É possível notar que nos nós 0 e 1 o tamanho médio da fila é de 0. Isto ocorre pelo fato de que a taxa da aplicação é menor que a taxa de transmissão do link que liga esses nós ao R0. Por sua vez, o R0 tem o tamanho médio da fila de aproximadamente 99,74 pacotes, isso se deve ao fato de que o fluxo de pacotes que chega nele é maior que a banda do link entre R0 e R1, o que acaba aumentando a fila. Pode-se perceber também, que o tamanho da fila máximo é de 100 pacotes, pelo código usado, então o R0 fica a maior parte do tempo com sua fila bastante ocupada, o que também era de se esperar olhando a figura 2, que mostra que a taxa de transmissão do n0+n1 é maior que a taxa que R0 consegue transmitir.

Neste momento é possível ver as diferenças de resultados das métricas analisadas durante o experimento entre os fluxos 1 e 2. Então, baseando nos resultados do FlowMonitor, os gráficos gerados e os resultados do TraceMetrics, pode-se explicitar as diferenças para:

- **Taxa da aplicação:** neste quesito, é possível ver que a taxa de aplicação para o fluxo 1 foi de aproximadamente 2Mbps(1997kbps), já para o fluxo 2 foi de aproximadamente 4Mbps(3994kbps). Esta diferença se deve ao fato do intervalo diferente entre o envio dos pacotes que foi colocado para cada um dos nós dessas aplicações(nó 0 e nó 1, respectivamente).
- **Throughput:** neste atributo é possível ver por meio dos gráficos gerados pelo Wireshark, que o *Throughput* do Fluxo 1 começa mais baixo que o do Fluxo 2, porém, eles se juntam perto de aproximadamente 84 pacotes por segundo quando a fila enche. Este acontecimento se deve ao fato de que a taxa de aplicação do fluxo 1 é menor que o do fluxo 2, e enquanto a fila não está cheia, os pacotes fluem sem retenção, por este motivo, o *throughput* do fluxo 1 aumenta até a fila encher, pois continua a enviar os pacotes, até ele não conseguir enviar mais sem retenção pelo tamanho da fila ter estourado. Por este mesmo motivo, o *throughput* do fluxo 2 diminui, pois quando ele começa a colocar os pacotes na rede, e como ele tem a taxa de transmissão maior, ele já enche a fila, e então o seu *throughput* diminui para se compensar com o

fluxo 1, e então os dois se equiparam dividindo a banda disponível em R0.

- **Atraso médio:** o atraso médio tem muito a ver com a fila que os pacotes enfrentam até chegar ao seu destino. Assim, o fluxo 1 tem o atraso médio menor que o fluxo 2, isto porque a taxa de aplicação é menor, logo, menos pacotes enfrentam a fila, o que reduz o atraso médio fim a fim do fluxo 1, o drop de pacotes da fila também afeta o atraso médio, neste caso, como o fluxo 1 envia menos pacotes, também perde-se menos pacotes na fila. Já no fluxo 2, como mais pacotes entram na fila, consequentemente tem que esperar mais para serem enviados, e também o número de pacotes que ocorre drop aumenta, isso aumenta o atraso fim a fim para este fluxo, o que faz sentido ser maior que o fluxo 1
- **Número de pacotes perdidos:** para o fluxo 1, pode-se reparar que a perda de pacotes está em cerca de 50% (8288/16650), já para o fluxo 2 a perda está em cerca de 73% (24607/33300). Isto ocorre porque como o fluxo 2 tem uma taxa da aplicação maior, ele também dropa mais pacotes quando a fila está cheia. Da mesma forma, como o fluxo 1 tem a taxa da aplicação menor, ele também dropa menos pacotes proporcionalmente.

C. QUESTÃO 2

Para a questão 2, alternativas como internet via satélite vão ser exploradas, de forma que um link entre backbone tenha delay bem maior que o comum. Para realizar este experimento a topologia dumbbell da figura 1 ainda vai ser mantida, fazendo a transferência de um fluxo UDP entre os clientes e servidores, somente sendo alterado o delay no link entre R0 e R1, ou link de backbone. Será utilizado um delay de 250ms para a simulação.

Ajustando o parâmetro de delay é possível ver que a simulação roda como o desejado:

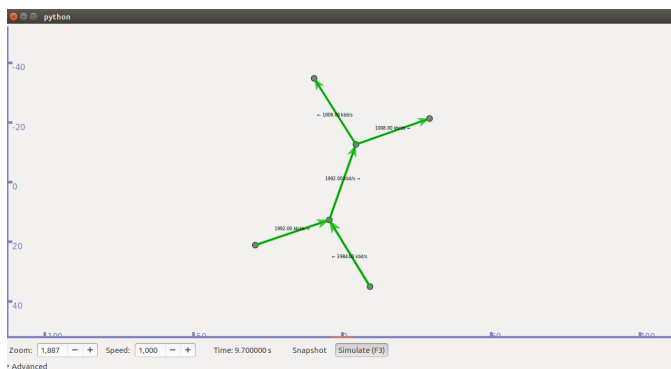


Figura 10. Questão 2 –vis

Também foi possível analisar a saída obtida no terminal:

```
ns3@ns3-VirtualBox: ~/ns-allinone-3.33/ns-3.33
Pacotes na fila: 99->100
Flow 1 (192.168.0.1:49153 -> 192.171.0.2:4444)
Tx Pacotes: 16650
Tx Bytes: 24941700
Tx Taxa de bits media: 1.99745e+06 bps
Rx Pacotes: 8279
Rx Bytes: 12401942
Rx Taxa de bits media: 995840 bps
Throughput medio (simulacao): 992155 bps
Atraso medio: 0.931172 s
Jitter medio: 2.02948 ms
Numero de pacotes perdidos: 8288
Flow 2 (192.169.0.1:49153 -> 192.172.0.2:4444)
Tx Pacotes: 33300
Tx Bytes: 49883400
Tx Taxa de bits media: 3.99479e+06 bps
Rx Pacotes: 8328
Rx Bytes: 12475344
Rx Taxa de bits media: 1.00173e+06 bps
Throughput medio (simulacao): 998028 bps
Atraso medio: 1.49405 s
Jitter medio: 3.45358 ms
Numero de pacotes perdidos: 24607
ns3@ns3-VirtualBox: ~/ns-allinone-3.33/ns-3.33$
```

Figura 11. Questão 2 - Terminal

Pela figura anterior (Fig. 11) foi possível identificar os fluxos, o fluxo 1 foi o fluxo que parte do n0 até o n2, e o fluxo 2 parte do n1 até o n3. Nesta figura também é possível ver a taxa da aplicação, o throughput, o atraso total e o número de pacotes para cada um dos fluxos de maneira bem intuitiva. Para o fluxo 1:

- Taxa da aplicação: 1997,00 kbps
- Throughput: 992kbps
- Atraso: 0,931172s
- Número de pacotes perdidos: 8288
- Jitter: 2,02948ms

Para o fluxo 2:

- Taxa da aplicação: 3994,00 kbps
- Throughput: 998kbps
- Atraso: 1,49405s
- Número de pacotes perdidos: 24607
- Jitter: 3,45358ms

Em seguida, deve-se analisar o *throughput* gerado pelo wireshark para cada um dos fluxos. Neste momento faremos uma filtragem pelo IP de origem e gerar o gráfico:

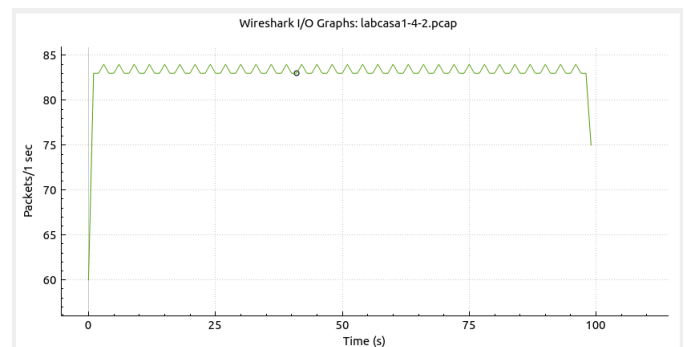


Figura 12. Fluxo 1 - Throughput

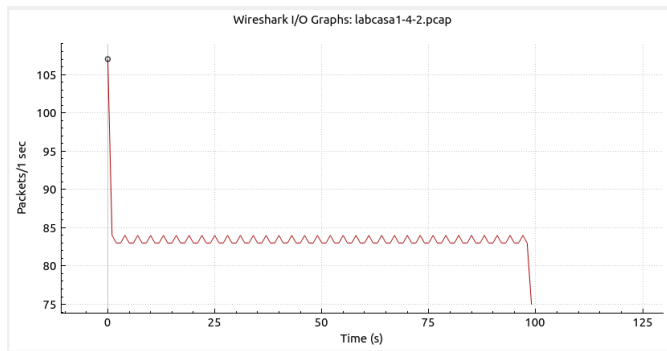


Figura 13. Fluxo 2 - Throughput

É possível também juntar os dois gráficos e ver como é a relação dos dois fluxos que compartilham o mesmo link para chegar ao seu destino:

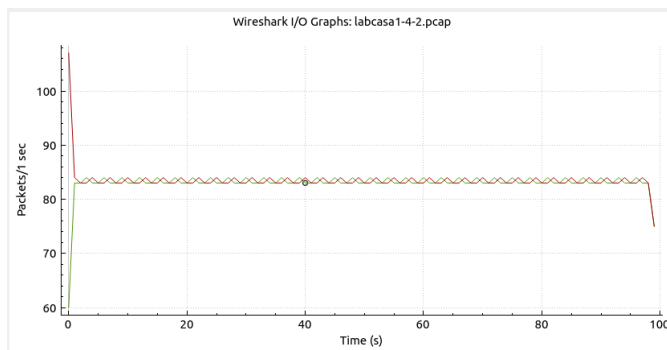


Figura 14. Fluxo 1 e Fluxo 2 - Throughput

O próximo passo é utilizar o TraceMetrics para obter o tamanho médio da fila nos nós n0, n1 e R0. Foi possível obter o seguinte:

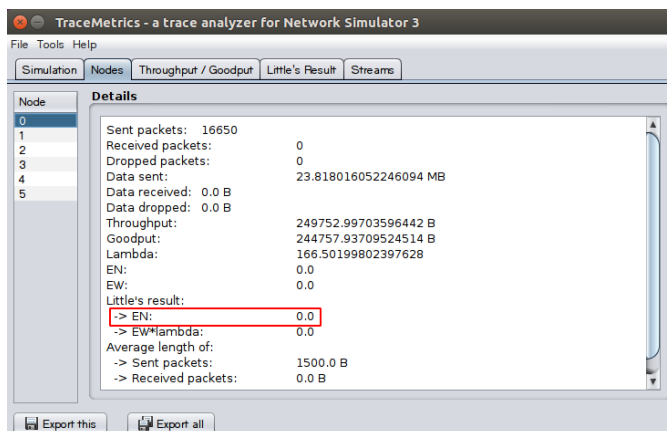


Figura 15. Tamanho médio da fila - Nó 0

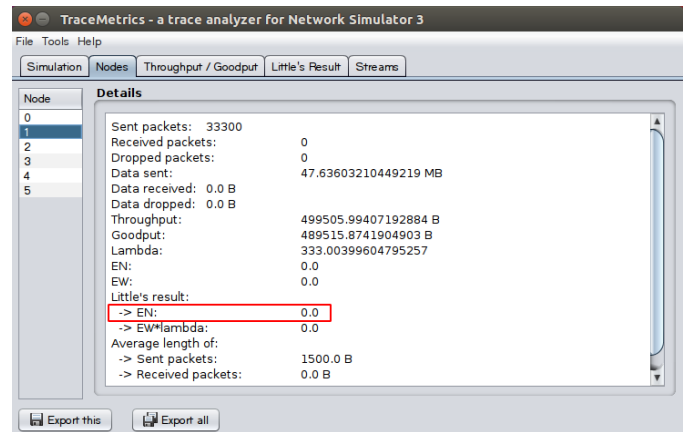


Figura 16. Tamanho médio da fila - Nó 1

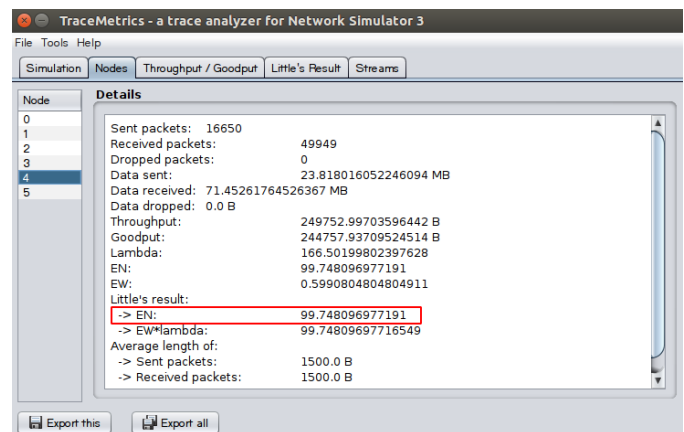


Figura 17. Tamanho médio da fila - R0

Assim como na questão 1, é possível notar que nos nós 0 e 1 o tamanho médio da fila é de 0. Isto ocorre pelo fato de que a taxa da aplicação é menor que a taxa de transmissão do link que liga esses nós ao R0. Por sua vez, o R0 tem o tamanho médio da fila de aproximadamente 99,74 pacotes, isso se deve ao fato de que o fluxo de pacotes que chega nele é maior que a banda do link entre R0 e R1, o que acaba aumentando a fila. Pode-se perceber também, que o tamanho da fila máximo é de 100 pacotes, pelo código usado, então o R0 fica a maior parte do tempo com sua fila bastante ocupada, o que também era de se esperar olhando a figura 10, isso mostra que a taxa de transmissão do n0+n1 é maior que a taxa que R0 consegue transmitir. Note que a análise da fila não muda pelo delay, já que a fila permanece quase sempre cheia, o delay continua permitindo com que essa situação ocorra novamente.

Neste momento é possível ver as diferenças de resultados das métricas analisadas durante o experimento entre os fluxos 1 da questão 1 e do fluxo 1 da questão 2. Logo, baseando nos resultados do FlowMonitor, os gráficos gerados e os resultados do TraceMetrics, pode-se explicitar as diferenças para:

- **Taxa da aplicação:** A taxa da aplicação do fluxo 1 da questão 1 e da questão 2 não se alteram. Isto ocorre pois

o delay não é alterado no link entre n0 e R0. Assim, a taxa da aplicação se mantém aproximadamente a mesma.

- **Throughput:** Para a vazão, também não é alterada para o fluxo 1 da questão 1 e questão 2, pois a vazão depende muito da capacidade do link entre R0 e R1 no que diz respeito a banda do link. Logo, o throughput também não vai ter alterado.
- **Atraso médio:** Neste quesito vemos a maior diferença entre a questão 1 e 2 para o fluxo 1. O atraso médio aumentou em aproximadamente 200ms, o que realmente era de se esperar, no fluxo 1 da questão 1 o atraso médio foi de 0,7292s, já no fluxo 1 da questão 2 o atraso médio foi de 0,9311. Note que o aumento foi como dissemos de aproximadamente 200ms, isto ocorreu pelo fato de aumentarmos o delay de 48ms para 250ms, o que daria uma diferença de 202ms.
- **Número de pacotes perdidos:** Para o número de pacotes perdidos, o que mais importa neste caso é o tamanho médio da fila associado com a capacidade do link em banda. Estes dois fatores não se alteraram com o delay, pelo fato de que a fila continuou sendo 'pequena' para a quantidade de pacotes que se deseja enviar, comparado com a importância do delay neste caso. Note que o delay somente altera o quanto um pacote demora para chegar ao outro lado do link, e não na saída da fila do R0. Por isso os outros fatores também não se alteram.

IV. CONCLUSÃO

Durante todo o experimento foi possível aumentar o conhecimento sobre as ferramentas envolvidas no processo, NS-3, PyVis, TraceMetrics e FlowMonitor. Foi possível aprender como montar novas topologias, links de acesso, bem como alterar suas propriedades no NS-3. Também foi necessário estudar os arquivos de saída do NS-3, aprender a como usá-los para obter análises fiéis aos comportamentos das redes, nós e demais componentes, isto associado com as outras ferramentas necessárias para isto.

Em suma, foi possível estudar os conceitos necessários para o começo da disciplina de Avaliação de Desempenho de Redes e Sistemas no que diz respeito a parte prática.

REFERÊNCIAS

- [1] *Roteiro do Lab. Casa 01*, <https://aprender3.unb.br/mod/resource/view.php?id=679477>
- [2] J. F. Kurose e K. W. Ross, *Computer Networks: A Top-Down Approach*. (5th ed.). Pearson Addison-Wesley, 2009.
- [3] Documentação NS3: <https://www.nsnam.org/documentation/>
- [4] Documentação FlowMonitor <https://www.nsnam.org/docs/models/html/flow-monitor.html>
- [5] Documentação WireShark <https://www.wireshark.org/docs/>