

# Simulação de uma rede Ethernet

Antonio Carlos Salzvedel Furtado Junior e Tiago Rodrigo Kepe  
GRR20080946,GRR20084630

12 de novembro de 2010

Universidade Federal do Paraná  
CI058 Redes I Bacharelado em Ciência da Computação  
For: Luiz Carlos Pessoa Albini

---

## 1 A simulação

Nosso trabalho começou com a criação do script `ethernet.tcl`, que deve ser executado pelo programa NS. Este script receberá dois parâmetros da linha de comando, o segundo deles é uma semente para o gerador de números aleatórios. Nós usamos a aleatoriedade para escolher, entre as 70 máquinas disponíveis, 10 origens e 2 destinos. Esta escolha pode estar intercalada, ou seja, um destino pode também ser uma origem.

O primeiro parâmetro é o número de mensagens por segundo. Como usamos o gerador de tráfego presente no NS2, chamado CBR, este parâmetro serve para definir o intervalo entre o envio de mensagens. Por exemplo, para um número de mensagens igual a 100, o intervalo seria:

$$mensagens \rightarrow 100$$

$$intervalo = 1/100 \text{ segundos}$$

Detalhes sobre a rede foram todos definidos como constantes internas, eles estão inclusos em uma estrutura chamada (`opt`). Algumas constantes foram definidas de acordo com a especificação do trabalho, outras de acordo com o comportamento esperado do ns para uma rede ethernet. Elas podem ser facilmente modificadas. Aqui estão as principais:

- Duração do envio de pacotes  $\rightarrow$  entre 1.5 e 181.5 segundos = 180 segundos
- Número de máquinas  $\rightarrow$  70
- Número de switches  $\rightarrow$  11
- Tamanho dos pacotes  $\rightarrow$  1000

- Número de origens  $\rightarrow 10$
- Número de destinos  $\rightarrow 2$

Os links entre máquinas e switches e entre os switches intermediários e o pai são todos duais, e seguem especificações do trabalho.

## 1.1 Conexões

Depois de escolhermos aleatoriamente máquinas origens e destinos, conectamos metade das origens a um destino, e a outra metade a outro destino, de tal forma que as origens só enviarão pacotes aos destinos selecionados. Como a escolha foi aleatória, não há nenhum problema.

## 2 Análise e geração de gráficos

Com base no arquivo de trace gerado pelo NS2 (`/dev/shm/simulacao/mensagens_por_segundo.tr`), dois scripts para análise de dados foram feitos. O primeiro deles é o `entrega.awk`, responsável por gerar a taxa de entrega. Ele recebe apenas o tipo de evento ( `enfila`, `drop`, etc. ) e a ID da mensagem. Fazemos também com que ele receba apenas linhas únicas de entrada, para contar apenas envios iniciais de cada mensagem.

Se o `entrega.awk` receber da entrada um evento de enfileiramento (+), então ele incrementa o número de mensagens enviadas. Caso a mensagem seja um `drop` (d), incrementamos o número de mensagens perdidas. Se dividirmos o número de mensagens perdidas pelo número de enviadas, obtemos a taxa de erro, como o inverso da taxa de erros é a taxa de acertos, então:

$$\begin{aligned} e &\leftarrow \text{enviadas} \\ d &\leftarrow \text{perdidas} \\ \text{acertos} &\leftarrow 1 - (e/d) \end{aligned}$$

O `entrega.awk` retornará a taxa de acertos em porcentagem ao final da sua execução.

O segundo script criado foi o `latencia.c`, responsável pelo retorno da latência média. Neste caso precisamos saber o o tempo de envio e chegada de cada mensagem. Utilizamos dois vetores, um para guardar os tempos de envio e outro para os tempos de chegada de cada mensagem. O tempo de envio é o tempo do primeiro enfileiramento da mensagem com determinado ID. O tempo de recebimento é tempo do evento recebe quando o destino da mensagem é igual ao nó de chegada, assim evitamos contar o recebimento em switches. É retornada então uma média da diferença de todos os tempos.

Eram necessárias várias simulações, três para cada taxa de mensagens por segundo (1,10,100,1000,10000). Criamos o script `taxaentregalatencia.sh` para lidar com isso. Para cada taxa acima, ele vai executar nossos outros scripts 3 vezes, vai então calcular as médias destas saídas e deve concatenar estes resultados para dois arquivos, o `entrega.out` e o `latencia.out`. O `entrega.out` contém as taxas de envio e as médias de entrega, o `latencia.out` é semelhante, só que contém as médias de latência.

A última tarefa do Shell Script criado é executar outros dois scripts do GNUPlot, o entrega.gnu e o latencia.gnu. Com base nos arquivos de saída mencionados ele gerará gráficos de taxa de entrega e latência no formato PNG. Os gráficos são incluídos neste relatório.

### 3 Modo de execução

```
make  
./taxaentregalatencia.sh
```

### 4 Gráficos



