

Simulando ICMP Echo request Uma implementação em Ruby

Diego Pinto da Jornada
diego.jornada@acad.pucrs.br

Matthias Nunes
matthias.nunes@acad.pucrs.br

1. IMPLEMENTAÇÃO

O simulador foi desenvolvido na linguagem *Ruby*. Sua estrutura consiste em um conjunto de interfaces que podem ser nodos ou pertencerem a roteadores, e essas mesmas pertencem a uma rede que são ligadas através dos roteadores.

A interface possui todas as funções principais que são especializadas quando se trata de um nodo, mantendo o mesmo nome para que todas as interfaces sejam tratadas da mesma maneira, independente de ser um nodo ou pertencer a um roteador. Existem as funções de *ARP* e de *ICMP* para executar os protocolos, por exemplo `arp_request` e `icmp_reply`.

Além disso existe uma função para enviar mensagem, que é a que chama as outras, tanto para um interface de roteador ou um nodo.

2. COMO UTILIZAR O SIMULADOR

Para utilizar o simulador é necessário ter o *Ruby* instalado.

O formato do arquivo de entrada deve ser como no exemplo fornecido:

```
#NODE
n1,00:00:00:00:00:01,192.168.0.2,5,192.168.0.1
n2,00:00:00:00:00:02,192.168.0.3,5,192.168.0.1
n3,00:00:00:00:00:03,192.168.1.2,5,192.168.1.1
n4,00:00:00:00:00:04,192.168.1.3,5,192.168.1.1
#ROUTER
r1,2,00:00:00:00:00:05,192.168.0.1,5,00:00:00:00:00:06,\
192.168.1.1,5
#ROUTERTABLE
r1,192.168.0.0,0.0.0.0,0
r1,192.168.1.0,0.0.0.0,1
```

Para executar o programa execute o comando:

```
ruby app.rb <topologia> <nodo1> <nodo2> <mensagem>
```

3. LIMITAÇÕES E DIFICULDADES

O nosso simulador não consegue fragmentar os pacotes. O *MTU* não é considerado e todos os pacotes são enviados como se a rede comportasse o tamanho.

A maior dificuldade durante a implementação foi para debugar quando a mensagem não ia, ou os dados iam errados. Trabalhar com um grafo muito grande de objetos de tipos diferentes pode deixar as coisas confusas.

4. EXEMPLO

Segue um exemplo executado utilizando a mesma topologia usada na seção 1. Ela possui 4 nodos e 1 roteador.

Segue o resultado:

```
ruby app.rb topologia.txt n1 n3 hello

n1 box n1 : ARP - Who has 192.168.0.1? Tell 192.168.0.2;
r1 => n1 : ARP - 192.168.0.1 is at 00:00:00:00:00:05;
n1 => r1 : ICMP - Echo (ping) request (src=192.168.0.2
dst=192.168.1.2 ttl=8 data=hello);
r1 box r1 : ARP - Who has 192.168.1.2? Tell 192.168.1.1;
n3 => r1 : ARP - 192.168.1.2 is at 00:00:00:00:00:03;
r1 => n3 : ICMP - Echo (ping) request (src=192.168.0.2
dst=192.168.1.2 ttl=7 data=hello);
n3 rbox n3 : Received hello;
n3 => r1 : ICMP - Echo (ping) reply (src=192.168.1.2
dst=192.168.0.2 ttl=8 data=hello);
r1 => n1 : ICMP - Echo (ping) reply (src=192.168.1.2
dst=192.168.0.2 ttl=7 data=hello);
n1 rbox n1 : Received hello;
```
