LABORATÓRIO IV

ROTEAMENTO DINÂMICO

Redes de Computadores – Da Teoria à Prática com Netkit

Laboratório IV – Roteamento dinâmico

Objetivos do laboratório

- Entender como redes se ajustam dinamicamente
- Configurar um roteador para operar dinamicamente
- Entender o protocolo RIP

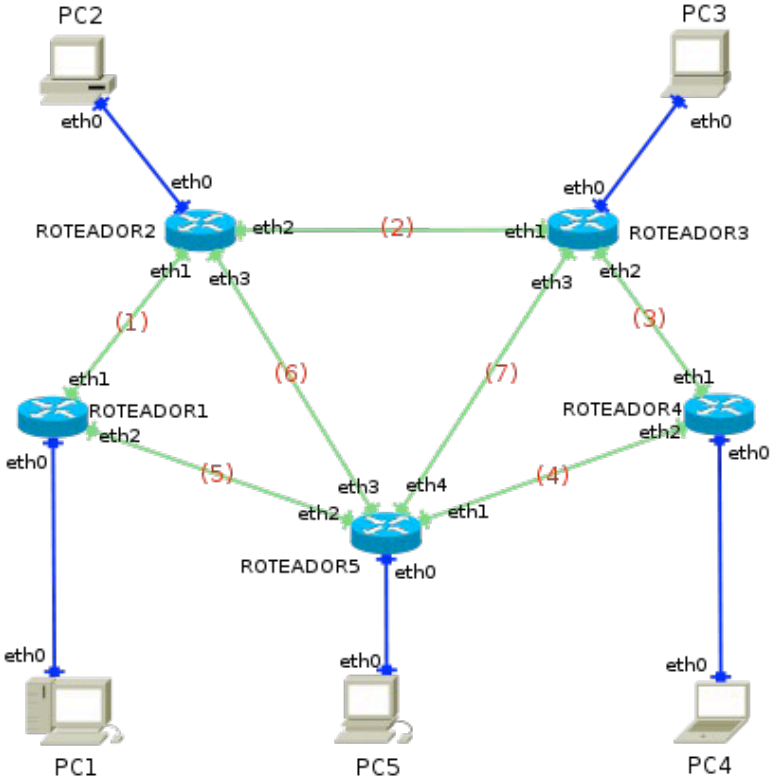
Antes de baixar este cenário, verifique o sistema de arquivos que irá utilizar. Caso você utilize a versão com dois sistemas de arquivos, utilize a versão padrão para está atividade. Mais detalhes no site, ou no tutorial de instalação.

Cenário sendo reproduzido

A figura abaixo representa a topologia de rede sendo estudada. Diferente das outras redes, esta precisa ser um pouco maior para demonstrar o funcionamento do protocolo RIP. Cada maquina se conecta a um único roteador. Os roteadores estão ligados entre sí de forma a permitir que exista pelo menos 2 caminhos para qualquer outro roteador.

Se um roteador cair, exceto a máquina ligada exclusivamente àquele roteador, todas as demais máquinas deverão estar em funcionamento e todas as rotas precisarão ser recalculadas rapidamente para que o funcionamento da rede não seja prejudicado.

Os números em vermelho indicam os enlaces representando uma subrede. Veja que existe um padrão estratégico na definição dos IPs.



LISTA DE INTERFACES DE REDE

```
PC1 - eth0 - 192.168.1.1/24
PC2 - eth0 - 192.168.2.1/24
PC3 - eth0 - 192.168.3.1/24
PC4 - eth0 - 192.168.4.1/24
PC5 - eth0 - 192.168.5.1/24

ROTEADOR1 - eth0 - 192.168.1.100/24
          - eth1 - 10.0.1.1/29
          - eth2 - 10.0.5.1/29

ROTEADOR2 - eth0 - 192.168.2.100/24
          - eth1 - 10.0.1.2/29
          - eth2 - 10.0.2.2/29
          - eth3 - 10.0.6.2/29

ROTEADOR3 - eth0 - 192.168.3.100/24
          - eth1 - 10.0.2.3/29
          - eth2 - 10.0.3.3/29
          - eth3 - 10.0.7.3/29

ROTEADOR4 - eth0 - 192.168.4.100/24
          - eth1 - 10.0.3.4/29
          - eth2 - 10.0.4.4/29

ROTEADOR5 - eth0 - 192.168.5.100/24
          - eth1 - 10.0.4.5/29
          - eth2 - 10.0.5.5/29
          - eth3 - 10.0.6.5/29
          - eth4 - 10.0.7.5/29
```

Conhecimentos de rede que você irá adquirir

Você entenderá como funciona a internet, com milhares de equipamentos sendo ligados e desligados continuamente. O RIP foi o protocolo usado nos primeiros anos da internet, quando ainda era a ARPANET. Mais tarde foi substituído por outros mais sofisticados, com algoritmos de roteamento mais avançados, mas o RIP pode ainda ser usado para redes internas.

Você verá também que a rede é onipresente e dinâmica e modificações na estrutura não paralisam por completo a rede. Imagine se para trocar um roteador tivesse que parar a internet globalmente? O meio tem capacidade de recalcular rotas alternativas e restabelecer a comunicação.

Antes de continuar, é importante lembrar que você deve ter feito a instalação do software **Wireshark** que será utilizado neste lab, portanto use os comandos apt-get install wireshark (distribuições debian) ou urpmi wireshark (mandriva) para instalar este software, caso o mesmo não esteja instalado.

Devemos lembrar que, os comandos marcados com a tag [real] deverão ser executados no console real. Os demais comandos serão executados dentro das máquinas virtuais. Sempre que exigido a instrução pedirá uma máquina virtual específica.

Execução do laboratório

Importante: Antes de executar este lab, você desejará se prepara com os seguintes requisitos: Este lab requer diversas janelas. Use um ambiente de trabalho com vários espaços, preferencialmente 4 deles. Gnome, Kde, Xfce tem quatro espaços por padrão. Use um deles ou configure seu ambiente preferido para quatro espaços. Os dumps deste laboratório são trabalhosos de gerar.

1. [real] Salve o arquivo netkit_lab04.tar.gz na sua pasta de labs. (/home/seu_nome/nklabs).

2. [real] Acesse a pasta nklabs a partir do terminal

3. [real] Use o comando:

```
[seu_nome@suamaquina ~]$ tar -xf netkit_lab04.tar.gz
```

Será criada a pasta lab04 dentro da sua pasta nklabs.

4. [real] Use o comando a seguir:

```
[seu_nome@suamaquina ~]$ lstart -d /home/seu_nome/nklabs/lab04
```

Desta vez serão iniciadas 10 máquinas virtuais. Entretanto, diferente de qualquer lab

anterior, teremos 15 janelas em execução, sendo duas janelas por roteador, sendo a principal apenas o nome da máquina e a segunda, recebe o nome adicional *Virtual Console #1*.

5. [real] Organize suas janelas de modo a localizar qualquer uma delas rapidamente. A sugestão é enviar as 5 janelas dos Pcs para o espaço de trabalho 2, 5 janelas de roteadores para o espaço 3, e as últimas 5 para o espaço de trabalho 4.

As interfaces de redes estão todas configuradas, bem como os gateways dos hosts. As tabelas dos roteadores estão limpas exceto suas rotas padrão definidas pelas próprias interfaces de rede.

6. Tente executar um ping do PC1 para o PC5. (A resposta esperada é destination net unreachable). A mesma resposta é esperada de qualquer para qualquer PC.

7. Apenas as ligações de enlaces estão disponíveis. Por exemplo, se você executar um ping do roteador3 para o roteador4 através do enlace (3) ele irá responder. (A partir do ROTEADOR3, ping 10.0.3.4)

8. Confira a tabela de roteamento do ROTEADOR1 com o comando route. Anote em um rascunho quantas entradas tem em sua tabela de roteamento.

9. Consulte as tabelas de roteamento dos demais roteadores.

Você poderia neste momento tentar efetuar o roteamento manualmente, acrescentando rotas estáticas. Para escolher as rotas, poderia executar algoritmo de Djisktra para obter o melhor caminho baseado no número de hops.

Usaremos as janelas extras para executar o tcpdump nos roteadores e capturar os pacotes de negociação das rotas para poder visualizá-los no wireshark posteriormente. Atenção às interfaces observadas de cada comando, para poder observas pacotes em várias subredes.

10. Vá ao Virtual Console #1 do ROTEADOR1 e use o comando **tcpdump -i eth1 -v -n -s 1600 -w /homehost/lab4rip1.pcap**.

11. Vá ao Virtual Console #1 do ROTEADOR2 e use o comando **tcpdump -i eth2 -v -n -s 1600 -w /homehost/lab4rip2.pcap**.

12. Vá ao Virtual Console #1 do ROTEADOR3 e use o comando **tcpdump -i eth3 -v -n -s 1600 -w /homehost/lab4rip3.pcap.**

13. Vá ao Virtual Console #1 do ROTEADOR4 e use o comando **tcpdump -i eth2 -v -n -s 1600 -w /homehost/lab4rip4.pcap**.

14. Vá ao Virtual Console #1 do ROTEADOR5 e use o comando **tcpdump -i eth2 -v -n -s 1600 -w /homehost/lab4rip5.pcap.**

Você está se perguntando se poderia enviar estes tcpdumps para background com o “&” e usar um terminal só? Poderia sim, assim como poderia em vez de enviar para um arquivo de captura, exibir a saída em tela, ou utilizar uma ferramenta de monitoria, top, iotop ou outra mais sofisticada e específica para um determinado propósito. Por isso que sugerimos que você acostume-se a trabalhar com várias janelas, você pode gostar da ideia. Prometemos não fazer isso com você de novo.

Atenção às diferenças nos comandos acima destacados em vermelho! Os parâmetros usados do tcpdump são:
v para acrescentar mais detalhes na saída, como a contagem de pacotes
n para não resolver os nomes onde possível, mantendo os números de ip
s para definir o tamanho de captura do pacote, pois usualmente o tcpdump corta um pedaço do pacote
-w para escrever a saída num arquivo de captura ao invés da saída do console.

15. Vá ao roteador1 e inicie o daemon zebra com o comando **/etc/init.d/zebra start**.

16. Repita o procedimento com os demais roteadores.

17. Tente executar novamente o ping até o PC5 (192.168.5.1) a partir do PC1.

18. Se obtiver sucesso, vá aos Virtuais Console #1 de cada roteador e pressione **Ctrl + C** para interromper a captura. Nós iremos estudar os pacotes do wireshark posteriormente. Caso não tenha sucesso aguarde alguns segundos e tente novamente. Em geral o tempo de digitar cada comando nos roteadores é suficiente para a propagação das tabelas de roteamento.

19. Em cada roteador, use o comando route e compare a tabela de roteamento com a que você anotou (quantidade de entradas).

Observe com atenção também que o comando route mostra a métrica, que em resumo é a quantidade de saltos esperados para atingir aquele destino.

20. Use o comando **tracert 192.168.5.1** no PC1 para determinar a rota descoberta. Repita o comando para encontrar os pc's 2, 3 e 4.

Algumas vezes o traceroute mostrará alguns * * *. Isso é porque o protocolo RIP está recalculando rotas e não mostra a informação de retorno. Quando as rotas estiverem estáveis ele mostrará todos os passos. Não há diferença entre os comandos traceroute e tracert. São as mesmas aplicações com nomes diferentes. Algumas distribuições de Linux têm apenas um, ou outro, ou ambos.

Nossa intenção agora é demonstrar o que acontece quando a comunicação é interrompida. Vamos supor que o cabo do roteador5 rompeu. Para simular este

comportamento, nós iremos desativar a porta eth2 do roteador e aguardar alguns instantes até o roteamento estabilizar.

21. Verifique que todos os PCs conseguem se comunicar neste momento.

22. No roteador5, execute o comando **ifconfig eth2 down**.

23. Aguarde 3 minutos! Esse é o tempo configurado por padrão para o timeout da rota que foi desativada.

24. Tente a partir do PC1 e PC5 executar um ping para o PC5. Será normal não responder novamente. Tente executar alguns traceroutes de 1 para 5 e de 5 para 1.

25. Verifique a comunicação entre os demais PCs com PING.

26. Use o comando **route** em cada um dos roteadores para consultar a nova tabela de roteamento.

Vamos acessar brevemente a interface de configuração do roteador para ver como navegar pelo software.

27. Use o comando **telnet localhost zebra** no roteador1.

28. A senha solicitada é **zebra**.

29. Pressione a tecla “?” para ver os comandos disponíveis. A tecla TAB também funciona aqui para completar comandos.

30. Dentro do zebrarot1, use o comando **show ip route**. Ele irá mostrar qual o caminho conhecido para atingir cada destino.

31. Use o comando **enable**. Isso irá transferir para o modo de usuário privilegiado.

32. Neste nível a senha é **zebraadmin**.

33. Use a tecla “?” e veja que há mais opções disponíveis agora.

34. Neste nível nós termos mais privilégios. Vamos trocar a senha de acesso de usuário. Use o comando **configure terminal**.

35. Use o comando **password teste**.

36. Use o comando **exit** (para sair do modo de configuração).

37. Use o comando **disable** para sair do modo de usuário privilegiado.

38. Use o comando **exit** novamente para sair da configuração do roteador. Se você tentar entrar novamente, verá que a senha de acesso agora, ao primeiro nível, é teste. A senha de configuração (administrativa por assim dizer), permanece a mesma.

Aqui nós vimos a configuração do software básico de roteamento. Podemos acessar a configuração do daemon ripd que cuida do protocolo rip. Os comandos são os mesmos. Vamos entrar brevemente.

39. Use o comando **telnet localhost ripd**

40. A senha é rip

41. Use **enable** para passar para o modo de configurador.

42. A senha é ripadmin

43. Use o comando **configure terminal** para ativar o modo de configuração.

44. Use o comando **router rip** e em seguida use o comando **timers basic 20 30 22**.

O comando **router rip** irá ativar o modo de configuração do protocolo rip. O comando **timers basic** irá modificar para 20 segundos para atualizar a tabela com novas rotas. 30 segundos como tempo de vida de uma rota (o quanto tempo ela permanece na lista) e 22 segundos para o "garbage collector". Ele tenta verificar se tem rotas mortas e remove-as da lista.

45. Use o comando **exit** três vezes para encerrar a configuração.

46. Acesse a pasta **/var/log/quagga** e veja o conteúdo dos arquivos ripd.log e zebra.log. (use **cat ripd.log** e **cat zebra.log**)

Antes de seguir, ignore as reclamações dos agentes SNMP. Você estudará isso posteriormente, mas basta saber que não há agentes SNMP preparados no momento, por isso dos erros.

Essa mesma tela de configuração permitiria acrescentar em um determinado roteador uma rota estática para um roteador que não estivesse com o protocolo RIP habilitado. A vantagem de usar essa configuração e não o comando route é que essa rota seria propagada corretamente para outros roteadores vizinhos com suporte ao RIP ativo.

O modo de configuração que você experimentou é muito similar aos conceitos mais avançados de roteamento dos equipamentos da empresa Cisco. Já ouviu falar em certificações CCNA? Comece aqui.

Formule as teorias

Lembrando a especificação da rede, com seus atuais conhecimentos de rede, tente explicar:

1. Estude os arquivos /etc/zebra/daemons, /etc/zebra/zebra.conf e /etc/zebra/ripd.conf e veja o que descobre sobre a configuração destes softwares.

2. Se um novo roteador RotaWeb fosse adicionado a nossa rede, ligado a uma porta livre do roteador 3 de IP 200.120.2.1/24. Explique quais seriam as modificações necessárias no laboratório (arquivos de configuração em geral, rotas estáticas) para que todos os pcs pudessem acessar a internet?

Aprendendo um pouco sobre linux

Esta série inicial de laboratórios permitiu vermos o potencial do linux para redes, podendo fazer o papel de qualquer dispositivo de redes. Conhecemos um dos softwares de roteamento, o zebra, capaz de efetuar o roteamento com o protocolo RIP. O zebra é capaz também de usar outros protocolos, como pode ser visto em seus arquivos de configuração, mas o assunto foge do escopo deste experimento