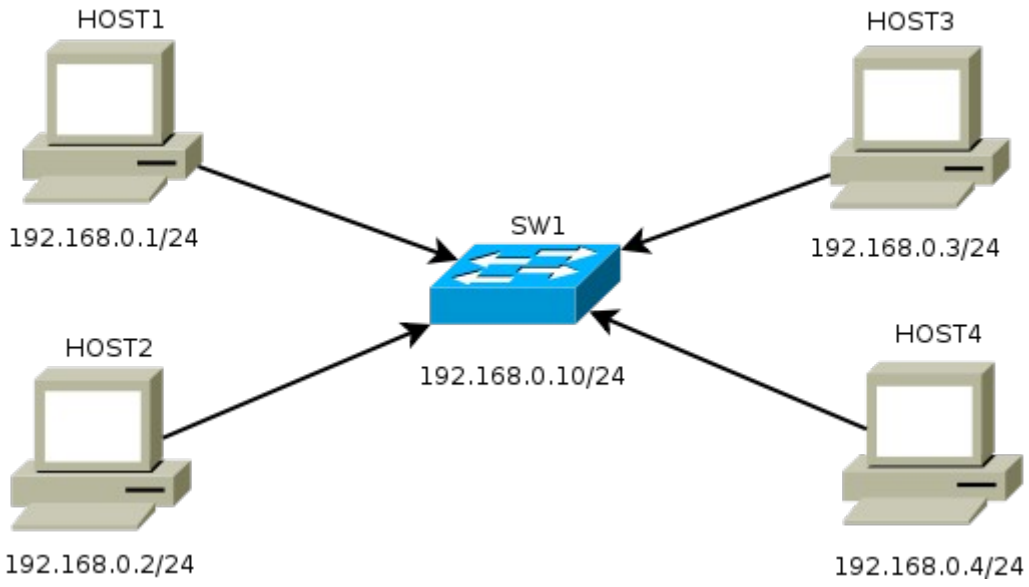
Laboratório II – Nossa rede ganhou um switch.

Objetivos do laboratório

- Entender a diferença de uma rede de difusão para uma rede ponto a ponto
- Aprender a montar uma 802.1 D Ethernet Bridge
- Estudar a composição dos pacotes, desmontando os no wireshark

Cenário sendo reproduzido

A figura abaixo representa a topologia de rede sendo emulada. De modo similar ao laboratório anterior, estamos ligando 4 computadores nomeados host1, host2, host3 e host4 respectivamente, a um switch. Como não dispomos de um equipamento virtual equivalente, iremos utilizar uma quinta máquina linux com o pacote bridge-utils como um soft-switch. Essa nossa máquina será chamada SW1.



Conhecimentos de rede que você irá adquirir

Ao completar este lab você perceberá que não é muito diferente configurar uma rede com switch, de uma rede com hub. Verá que o endereço mac tem um papel importante no funcionamento de um switch. Reforçará seus conhecimentos também com o teste de redes via ICMP.
Mais importante ainda, perceberá que o switch é um aperfeiçoamento do hub pois permite conexões ponto a ponto, reduzindo a quantidade de colisões de rede.

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.

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

1. [real] Salve o arquivo netkit_lab02.tar.gz na sua pasta de labs. (/home/seu_nome/nklabs).

2. [real] Use o comando:
   ```
   [seu_nome@suamaquina ~]$ tar -xf netkit_lab02.tar.gz
   ```
   Ele irá criar a pasta lab01 dentro da sua pasta nklabs.

3. [real] Use o comando a seguir:
   ```
   [seu_nome@suamaquina ~]$ lstart -d /home/seu_nome/nklabs/lab02
   ```

O comando **lstart** irá iniciar o laboratório virtual, ativando o console de 4 máquinas virtuais nomeadas HOST1, HOST2, HOST3 e HOST4 e também o SW1 (switch).

Você poderá suprimir o parâmetro -d se mudar para a pasta lab02 antes. Mas verifique com o comando pwd se a pasta corrente é lab02 antes de iniciar o lab. Caso inicie o lab em pasta incorreta sem o parametro -d, você poderá ter um lab estranho em execução e diversos arquivo para apagar.

4. [real] Use o comando vlist para listar as máquinas virtuais. Você terá como saída algo assim:

```
[seu_nome@sua_maquina lab01]$ vlist
USER            VHOST               PID       SIZE  INTERFACES
seu_nome        SW1                9629      11604  eth0 @ CaboAzul, eth1 @
CaboVermelho, eth2 @ CaboVerde, eth3 @ CaboAmarelo
seu_nome        HOST1             11162      11604  eth0 @ CaboAzul
seu_nome        HOST3             12183      11604  eth0 @ CaboVerde
seu_nome        HOST4             12199      11604  eth0 @ CaboAmarelo
seu_nome        HOST2             12230      11604  eth0 @ CaboVermelho

Total virtual machines:       5    (you),        5    (all users).
Total consumed memory:    58020 KB (you),    58020 KB (all users).
```

Agora você irá executar uma série de comandos, observe os resultados mostrados na tela e formule uma hipótese antes de ser apresentado à explicação exata. Alguns comandos aparentemente podem não fazer nada, travarem ou exibir mensagens de erros. Isso é normal e deve acontecer se a sequencia de passos for executada corretamente.

5. Use o comando **ifconfig** em cada um dos HOSTS. Você perceberá que a interface de rede de todas as máquinas estão ativas.
6. Execute o comando **ifconfig** no SW1. Você verá que há diversas interfaces de redes levantadas, a saber eth0, eth1, eth2, eth3, lo e br0
7. No SW1, mude para a pasta /hosthome/
8. Execute o comando **tcpdump -i br0 -w lab2_sw1.pcap**
9. Nos hosts 3 e 4, faça o **tcpdump** sobre a eth0, salvando os como **lab2_hostX.pcap** (onde X é o número do host). Não se esqueça de mudar para a pasta **hosthome**.
10. A partir do HOST1, faça um ping nos Hosts3 e 4 (com os comandos **ping 192.168.0.3** e **ping 192.168.0.4**). Cancele o ping com após 3 ou 4 saltos.

11. A partir do HOST2, faça um ping apenas no Host3.
12. Vá até os Hosts 3 e 4, e ao switch, e usando a combinação de teclas Ctrl + C, cancele os tcpdumps.
13. Em sua pasta home (/home/seu_nome/) deverá existir os arquivos lab2_host3.pcap, lab2_host4.pcap e lab2_sw1.pcap. Inicie o software wireshark e abra estes arquivos. Estude seu conteúdo.

Experimente

1. Faça um ping para o ip configurado no switch. Você perceberá que obterá resposta. Isso não aconteceria se fosse um hardware de um switch layer 2 comum.
2. A partir do HOST1, faça **traceroute 192.168.1.3**. Você verá apenas uma linha contendo o endereço do Host3. Isso acontece porque o nosso soft switch está totalmente transparente (como estaria o dispositivo real). Se fosse um roteador simples, você veria um salto (hop) adicional na lista mostrando o caminho (rota) percorrido pelo pacote.

Formule as teorias

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

1. Explique as diferenças nos arquivos de captura dos hosts deste lab e do lab anterior. .
2. Baseado na sua explicação acima, quais são as vantagens do uso da topologia indicada em relação à segurança da informação? E ao desempenho?
3. Usando seu conhecimento, de acordo com o modelo OSI, explique como o switch funciona.

Aprendendo um pouco sobre linux

Além do laboratório lab2, na página você tem acesso ao lab2b. Este lab não configura o switch para que você tenha oportunidade de aprender a fazê-lo. O lab2 contem o arquivo /lab2/SW1/root/switch-me.sh que é um shell bash script com os comandos mais simples para montar o soft switch.

Nós conseguimos montar o switch utilizando um pacote chamado **bridge-utils**. Uma bridge é uma ligação direta e transparente que conecta duas LANs. Se o endereçamento entre as máquinas das duas LANs forem compatíveis, elas poderão se comunicar diretamente, como se estivessem sobre o mesmo hub/hard-switch.

Um efeito colateral do bridge utils é que analisa os quadros e verifica o endereço mac destino dos pacotes. Dessa forma, ele envia o quadro apenas ao endereço mac destino, pela porta correta, exatamente como um hardware-switch.

O software brctl é o ethernet bridge administrator, um software para controlar as bridges do computador. Uma bridge é um conjunto de interfaces de redes transformadas em portas de switch.

O script linha à linha:

1. **brctl addbr br0 >** Nessa primeira o parametro addbr permite criarmos uma interface de rede virtual chamada br0, que representa a bridge.
2. **brctl addif br0 eth0 >** O comando addif, adiciona uma interface de rede à bridge, como uma porta. A eth0 adicionada a br0 vira a porta 1.
3. **brctl addif br0 eth1 >** Adicionamos a nossa segunda porta à bridge.
4. **brctl addif br0 eth2 >** Adicionamos a nossa terceira porta à bridge.
5. **brctl addif br0 eth3 >** Adicionamos a nossa quarta porta à bridge.
6. ifconfig eth0 0.0.0.0 > Configuramos a interface de rede eth0 para o endereço de rede 0.0.0.0. Isso deixa-a sem nenhum ip e livre para o controle da bridge.
7. ifconfig eth1 0.0.0.0 > Repetimos o mesmo com a segunda....
8. ifconfig eth2 0.0.0.0 > Com a terceira
9. ifconfig eth3 0.0.0.0 > e com a quarta porta.
10. ifconfig br0 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255 up > Finalmente levantamos a interface de rede br0 que interconecta as 4 interfaces físicas como se fossem portas de um switch.

Dica: após executar o script ou passo a passo estes comandos, Você pode usar **brctl show** para mostrar quais portas estão associadas á quais bridges.

Cada placa de rede ethernet (cabo) no linux recebe tipicamente o apelido de ethX, onde X é um inteiro positivo. Então para montar o lab em questão, nosso switch deveria ter no mínimo interfaces físicas de rede (4 placas pci por exemplo).

A velocidade do seu switch, se não houver outros elementos limitantes, é dado pela velocidade suportada pelo cabo e pelas placas de rede utilizadas. Podemos então usar essa configuração para uma rede de 10, 100 ou 1000Mpbs de acordo com estes fatores.