

## MIECOM 12/13

### Redes de Computadores I

#### Trabalho nº 0 - Introdução ao Simulador/Emulador IMUNES/CORE

---

#### Objectivos:

- Ambientação à ferramenta (simulador/emulador) *IMUNES/CORE*
- Configuração de *routers*
- Diagnósticos de conectividade

#### **I. Ambientação à ferramenta *CORE***

**I.a)** O CORE (Common Open Research Emulator) funciona **apenas** em FreeBSD e Linux (Ubuntu) com um kernel alterado para ter mais do que uma stack TCP/IP. Uma alternativa à instalação nativa de um sistema consiste em usar uma máquina virtual. Veja em <http://cs.itd.nrl.navy.mil/work/core/> como obter um ambiente CORE adequado. Deve começar por instalar um VMWare Player para depois descarregar e descompactar a imagem do CORE no disco local. A imagem depois de descompactada ocupa aproximadamente 2GB.

**I.b)** Teste o interface gráfico *core*:

<code>core</code> (sem argumentos)	– executa o ambiente de desenvolvimento gráfico do CORE
<code>core &lt;file.imn&gt;</code>	– carrega o ficheiro <file.imn> no ambiente gráfico CORE

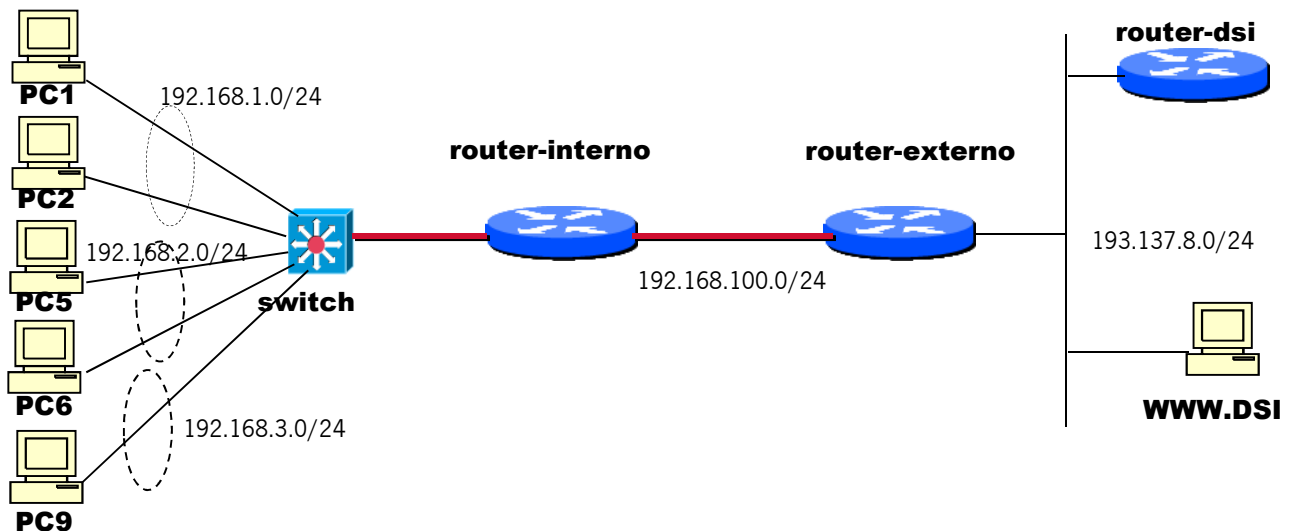
Se ficar sem rede, depois de executar o *core* tente recuperá-la com os seguintes comandos:

<code>/sbin/dhclient -r</code>	– liberta explicitamente o endereço anteriormente obtido por DHCP
<code>/sbin/dhclient</code>	– pede um novo endereço por DHCP
<code>/sbin/ifconfig -a</code>	– verifica se o interface ficou devidamente configurado

#### **II. Criação de uma pequena rede laboratorial para testes de conectividade**

Neste primeiro exercício (não será usado para avaliação, mas sim apenas para familiarização com a ferramenta) pretende-se emular no CORE uma pequena rede laboratorial (por exemplo a rede do LAP3). A interface gráfica *core* será usada para desenhar a topologia e configurar os links e os endereços. A configuração será depois feita manualmente, equipamento a equipamento, imitando tanto quanto possível a rede real. O exercício termina com o diagnóstico de conectividade.

Na figura seguinte está representado o esquema lógico da rede que queremos emular. A rede é formada por dois routers CISCO (router-interno e router-externo) interligados entre si por uma ligação FastEthernet a 100Mbps. Um dos routers (router-externo) tem uma ligação FastEthernet à rede do DSI onde também podemos encontrar o servidor WWW do DSI. As estações de trabalho estão ligadas a um switch e divididas por três redes IP distintas (192.168.1.0, 192.168.2.0 e 192.168.3.0).



### **Passo 1:** Criação da topologia no emulador *CORE* (Modo de Edição)

O CORE, no modo de edição, permite desenhar topologias. Existem vários objectos que podem ser usados: *ligações*, *hub*, *switch*, *router*, *pc*, *host*, tomada de saída, etc. Tanto o *pc* como o *host* representam sistemas terminais, do tipo computador pessoal ou servidor. A única diferença entre ambos é que o *pc* não executa nenhum processo no arranque enquanto o *host* arranca logo com alguns servidores, tais como o *inetd* e o *portmap*. Existem ainda três equipamentos de interligação, que operam a diferentes níveis da pilha protocolar: o *router* (nível 3), o *switch* (nível 2) e o *hub* (nível 1). Naturalmente que só o *router* usa endereços IP e faz encaminhamento.

Há apenas um objecto para representar ligações, mas que pode ser parametrizado para diferentes cenários de utilização. Os parâmetros configuráveis são por exemplo a largura de banda disponível e a taxa de erros. É ainda possível ligar a topologia virtual com a topologia real, usando o objecto *tomada de saída*. Este objecto, com a forma de uma ficha RJ45, pode ser associado a um interface físico do computador, dando-lhe exactamente o mesmo nome que o *device real* tem no sistema operativo. Supondo que o computador tem duas placas de rede (*ln0* e *le0*) podem-se incluir duas tomadas na topologia virtual com o mesmo nome. Os dispositivos da topologia virtual que aí forem ligados, devem também usar os mesmos endereços IP que o dispositivo real.

**II.a)** Crie no *Core* uma topologia semelhante à apresentada na figura. Sempre no modo de edição, ajuste os endereços IP de todos os interfaces e a largura de banda das ligações.

### **Passo 2:** Configuração dos equipamentos (Modo de Execução)

Uma vez terminada a topologia, passa-se ao modo de execução escolhendo a opção “Experiment -> Start”.

Esta opção faz despoletar a criação de múltiplas imagens (stacks virtuais), uma por cada objecto representado. É preciso algum cuidado com os erros nesta fase, porque podem implicar ter de sair da aplicação para limpeza das stacks virtuais.

No modo de execução, estão disponíveis novas acções associadas a alguns objectos, nomeadamente ao *pc*, *host*, e *router*. Seleccionando um desses equipamentos com o botão do rato do lado direito, obtém-se um menu com as opções disponíveis. No caso dos *pc* e *host* é apenas possível abrir uma shell (bash ou sh) no nó respectivo para neles executar comandos. No caso do router, além da shell normal, pode-se executar uma *vtys* e entrar no modo de configuração do Quagga. Os comandos que se podem usar são muito semelhantes aos do IOS da CISCO e estão bem documentados no **manual do Quagga**.

## Login no router

Para fazer login no router basta abrir uma *vtys/* (interface de comandos do daemon quagga). O modo privilegiado distingue-se pela prompt **#**. Pode-se mudar de modo com os comandos **enable** e **disable**.

```
Router>
Router>enable
Router#
Router#disable
Router>
Router>exit
```

## Modos de configuração

Os diferentes modos de configuração do router são identificados pela prompt, o que pode determinar, em cada instante, os comandos que podem ou não ser executados. Para alterar um interface, por exemplo, passa-se primeiro ao modo de configuração e depois identifica-se o interface a configurar:

```
Router#config terminal
Router(config)#
Router(config)#interface ?
Router(config)#interface eth0
Router(config-if)# ip address 192.168.90.90/24
Router(config-if)# interface eth1
Router(config-if)# ip address 192.168.89.89/24
Router(config-if)# exit
Router(config)# exit
```

Para configurar o encaminhamento dinâmico, o prompt muda também de acordo com o novo modo. Pode-se sair do modo de configuração, qualquer que ele seja, com o control-Z:

```
Router(config)#router rip
Router(config-router)# ^Z
Router#
```

## Ajuda em linha

Pode-se usar um **?** para obter ajuda em qualquer circunstância. Quando colado a algum prefixo, surgem todos os comandos com esse prefixo. Quando isolado mostra formas de continuar o comando. Isto é uma função muito útil nos CISCO e também no Quagga. Também se pode usar o TAB para completar a escrita e setas para cima e para baixo para navegar na lista de comandos:

```
Router#?
Router#c?
Router#configure ?
```

## Obtenção de informação sobre o router

```
Router#show version
Router#show memory
Router#show interfaces
Router#show ip route
Router#show running-config
Router#show startup-config
```

**Gravar a configuração em memória**

```
Router#configure
Router#service integrated-vtysh-config
Router#^Z
Router#write mem

OU

Router#copy running-config startup-config
```

**Comandos de anulação e outros**

Habitualmente, um comando pode ser anulado com “no” no início:

```
Router(config)# no router rip
Router(config)# interface eth0
Router(config-if)# shutdown
Router(config-if)# no shutdown
```

Testar conectividade

```
Router#ping ?
Router#traceroute ?
```

Para mais informações consultar o **manual do Quagga online**.

**II.b) Desactive o encaminhamento dinâmico em todos os routers e adicione manualmente as rotas** (encaminhamento estático) necessárias para garantir a conectividade IPv4 entre qualquer sistema e o servidor WWW.DSI.UMINHO.PT. Teste a conectividade com os comandos *ping* e *traceroute*. (entre outros)

**Passo 3:** Junção de todas as configurações testadas no ficheiro ex1.imn (Modo Execução)

Antes de terminar a execução é necessário preservar as configurações introduzidas em cada equipamento. O fim da execução determina também o fim dos ficheiros de configuração associados a cada nó, que na realidade existem apenas numa área temporária. A visão do *filesystem* difere de nó para nó, por mapeamento de algumas directorias noutras, mas esse mapeamento desfaz-se mal termine a execução. Infelizmente o File->Save do Core não consegue coleccionar a configuração introduzida em cada nó e limita-se a guardar aquilo que foi criado no modo de edição. É pois necessário preservar o resto manualmente.

**III. Caso de estudo 2: Endereçamento e Encaminhamento Estático**

**III.a)** Construa a topologia usada no exercício teórico-prático nº5 (Nível de REDE) ambiente gráfico do CORE.

**III.b)** De acordo com o plano de endereçamento IPv4 elaborado no exercício teórico-prático nº5 atribua endereços e configure todas as interfaces de rede presentes.

**III.c)** Configure estaticamente o encaminhamento IP de forma a garantir a conectividade entre todas as redes. Teste a configuração.