U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Licenciatura em Engenharia Informática e Computação

L.EIC025 - Redes de Computadores (RC)

2021/2022 | 1º Semestre

APLICAÇÃO FTP + CONFIGURAÇÃO DA REDE

2º TRABALHO LABORATORIAL

Fernando Rego | up201905951@edu.fe.up.pt

Miguel Amorim | up201907756@edu.fe.up.pt

Tomás Vicente | up201904609@edu.fe.up.pt

Introdução

Este relatório, surgindo no contexto da unidade curricular de Redes de Computadores da Licenciatura em Engenharia Informática e Computação, detalha a implementação de uma aplicação de download usando o protocolo FTP sobre uma rede local feita no laboratório.

Este projeto tem como objetivo uma implementação correta do protocolo FTP para realizar transferências de ficheiros e de uma configuração de um *switch* e de um *router* no âmbito da criação de uma rede local de computadores.

Parte 1 - Aplicação de Download

Arquitetura da Aplicação de Download

Podemos dividir a aplicação de download em três partes: Argumentos, Conexão e Download do ficheiro

Argumentos

Nesta parte é feito a verificação e processamento do argumento (URL) passado na execução do programa para uma instância da estrutura parse_info. Esta estrutura guarda os seguintes campos:

- Scheme
- Username
- Password
- Hostname
- URL Path

Após o parse do URL para preencher todos os campos anteriores, é feita a obtenção do IP do Host utilizando a função getIP() que tem como um dos seus argumentos o Hostname. Na existência de algum erro durante este processo, o programa irá mostrar uma mensagem de erro e então termina.

Conexão

Neste módulo começamos pela criação de um socket para iniciar a ligação através do endereço de IP fornecido através da função connectToSocket().

Para a comunicação são utilizadas as funções sendCommand() que através do descritor de ficheiro do socket envia um comando em forma de string para o servidor. Para a leitura de respostas são utilizadas as funções read_response() e readPasvResponse() que fazem a leitura das respostas do servidor. No caso da read_response() faz-se a leitura para um buffer da resposta do servidor, o qual é usado para verificar o código de resposta. Na função readPasvResponse() é feito o processamento da resposta do servidor após o envio do comando pasv. A função obtém um endereço de IP e uma porta através da mensagem de resposta.

Através destas funções previamente referidas e após ser realizada a conexão com o socket é feito o login. O login é feito através do envio do comando 'user' onde irá associado o username e com o comando '*pass*', associado com a password. Caso ocorra algum erro durante o login o programa acaba por terminar, mostrando uma mensagem de erro ao utilizador.

Download

Antes de começar o download propriamente dito, é enviado ao servidor o comando ‘pasv’, pedido ao servidor para que transfira dados em modo passivo. A resposta a este comando é lida e processada através da função readPasvResponse() que obtém o endereço IP e a porta que são utilizados numa nova ligação devido à entrada do servidor em modo passivo.

Para inicializar o download é enviado o comando 'retr’ com o caminho para o ficheiro obtido no URL dado pelo utilizador. O programa chama a função downloadFile() que guarda para um ficheiro (o nome do ficheiro é obtido também através do URL) os dados enviados pelo servidor a partir do novo socket criado.

Relatório de um Download bem-sucedido

Quando o processo é bem-sucedido (código de retorno de sucesso), é mostrada uma mensagem de sucesso:

Parte 2 - Análise e Configuração da Rede

Configuração do IP

Nesta experiência foi possível perceber os mecanismos subjacentes ao funcionamento do ARP (*Address Resolution Protocol*). Em particular, compreendemos o seu papel na conversão de endereços da camada de rede (concretamente IPv4) em endereços da camada de ligação de dados (MAC). Inspecionando os pacotes referentes a este protocolo recorrendo ao *Wireshark* identificámos a relação entre os endereços (IP e MAC) de origem e de destino. Concretamente, constatámos que, no envio de um pacote *ARP request*, o endereço MAC de destino contém o valor 0.0.0.0, sendo esse valor substituído pelo verdadeiro endereço aquando da receção do pacote *ARP response*. Desta forma, tornou-se evidente que, visto que o papel do ARP é justamente identificar o endereço MAC correspondente ao endereço IP de destino, o valor **0.0.0.0** representa que o valor é desconhecido e pretendido naquele momento.

No que toca ao utilitário *ping*, observámos a sua relação com os pacotes do ICMP (*Internet Control Message Protocol*) e verificámos a função deste protocolo no diagnóstico de possíveis erros na rede. Tal como esperado, detetou-se a correspondência entre os endereços (IP e MAC) de origem e de destino.

De modo geral, compreendemos o funcionamento de tramas *Ethernet* e dos seus campos. Especificamente, reconhecemos os diferentes valores no campo *EtherType* que permitem distinguir entre os diferentes protocolos (ARP com 0x0806 e IP com 0x0800) e a ligação entre os tamanhos do

cabeçalho (14 *bytes*), dos dados (variando conforme o protocolo) e de verificação (4 *bytes*) e o tamanho total da trama.

Finalmente, estudámos a finalidade da *loopback interface* e observámos a sua utilidade para deteção de erros de configuração da rede. Adicionalmente, percebemos o seu funcionamento como *interface* virtual e de que forma é utilizada para comunicação entre dois processos de um mesmo computador dado que o tráfico que lhe chega é imediatamente reenviado para a *stack* de rede como se se tratasse de tráfico proveniente do exterior.

Virtual LANS

Com este guião, realizámos e compreendemos os passos necessários para a configuração de uma VLAN (*Virtual Local Area Network*). Neste processo de configuração, compreendemos as ligações físicas necessárias entre as *interfaces* (através das portas Ethernet) de cada um dos computadores e a *switch*, bem como a necessidade de adicionar as respetivas portas na configuração da *switch*. Encontra-se em anexo essa sequência ordenada de passos.

Referente aos *broadcast domains*, concluímos tratarem-se de divisões lógicas de uma rede onde todos os dispositivos podem comunicar entre si através de *broadcast*. Para ser possível identificá-los, ativámos as respostas a pedidos de *broadcast*, desativando a flag *echo-ignore-broadcast*. Correndo o comand ping e observando os pacotes, foi possível detetar dois domínios (um em cada uma das VLAN), tal como esperado. De notar que um desses domínios contém apenas um computador, pelo que os seus pedidos não foram respondidos. Foram, portanto, identificados os domínios **172.16.30.255** e **172.16.31.155**, correspondentes às sub-redes VLAN 30 e VLAN 31, respetivamente.

Configuração do Router

Nesta experiência, compreendemos o esquema de configuração de um *router* comercial, acrescentando rotas estáticas e configurando NAT (*Network Address Translation*). Nas rotas, percebemos a significado das linhas **IP ROUTE PREFIX MASK ADDR** e entendemos o seu significado no contexto da rede e de ligação entre dispositivos. Em relação ao mecanismo de NAT, concluímos que se trata de um processo de substituição pelo *router* de endereços privados por endereços públicos, por forma a garantir a comunicação com o exterior. Assim, o *router* torna-se responsável por encaminhar os pacotes para o endereço correto, em ambas as direções. Pesquisámos acerca da sua origem e compreendemos a sua utilidade em lidar com a gama limitada de endereços possíveis usando IPv4.

Adicionalmente, entendemos os passos necessários no processo de configuração de serviços de DNS e a sua relação com a descodificação de *hostnames* em endereços. Percebemos o significado dos domínios de pesquisa configurados pela linha **SEARCH DOMAIN**, bem como os servidores DNS adicionados com **NAMESERVER ADDR.**

Conseguimos identificar as rotas existentes na nossa máquina local, nomeadamente aquelas necessárias para a comunicação com o *default gateway* (*router*). Durante a aula remota, foram também visíveis aquelas que asseguram o funcionamento da VPN.

Na transmissão da informação de DNS, identificámos o envio e receção de pacotes UDP (contendo a informação) e ICMP (contendo informações de confirmação), tal como seria esperado

numa transmissão deste tipo. Estes pacotes continham o endereço correspondente ao *hostname* fornecido e tinham como origem o nosso endereço e como destino o endereço do servidor de DNS.

Configuração do Router (Laboratório)

Para a experiência final, foi necessário interligar as componentes da rede para que funcionasse como um todo. Assim, adicionámos rotas que permitissem aos pacotes provenientes do tux3 chegar ao router (e à VLAN 31) através do tux4 e aos pacotes provenientes do router e do tux2 chegar ao tux3. Adicionou-se também a rota ao *default gateway* (router) para que pudessem aceder à internet. As rotas adicionadas encontram-se em anexo.

Antes da transmissão de pacotes são visíveis os endereços (IP e MAC) transmitidos nas mensagens ARP para comunicação entre os diferentes tux, bem como nos pacotes ICMP, sendo frequentes os endereços do intermediário (tux4).

Acerca da *forwarding table*, concluímos tratar-se de uma tabela que faz corresponder *interfaces* de destino a pacotes recebidos para que sejam encaminhados para o dispositivo correto. Assim, é constituída pelos endereços MAC das interfaces responsáveis por encaminhar cada um dos pacotes.

Conclusões

Este projeto permitiu-nos compreender melhor os conceitos teóricos relacionados sobre o protocolo FTP e programação utilizando sockets bem como sobre a configuração de uma rede de computadores através das experiências feita nas aulas.

A criação da aplicação de download aliada à configuração continua da rede local permitiu-nos, perceber os mecanismos que operam sobre a organização de redes e subredes (fisicas e virtuais), bem como encaminhamento de pacotes, tendo contribuído para cimentar o que tínhamos aprendido.

Referências

Anexos

Comandos de Configuração

- Experiência 1

Tux3:

> ifconfig eth0 up
> ifconfig eth0 172.16.30.1/24

Tux4:

> ifconfig eth0 up
> ifconfig eth0 172.16.30.254/24

- Experiência 2 (continuação da experiência anterior)

Tux2:

> ifconfig eth0 up
> ifconfig eth0 172.16.31.1/24

Login GtKterm ligado ao switch:

> enable
> password: *******

Criação da vlan30 e criação das portas associadas:

> configure terminal
> vlan 30
> end
> show vlan id 30
>
> configure terminal
> interface fastethernet 0/1
> switchport mode access
> switchport access vlan 30
> end
>
> configure terminal
> interface fastethernet 0/2
> switchport mode access
> switchport access vlan 30
> end

Criação da vlan31 e criação da porta associada:

> configure terminal
> vlan 31
> end
> show vlan id 31
>
> configure terminal
> interface fastethernet 0/3
> switchport mode access
> switchport access vlan 31
> end

- Experiência 4 (continuação das experiências anteriores)

Tux4:

> ifconfig eth1 up
> ifconfig eth1 172.16.30.253/24

Login GtKterm ligado ao switch:

> configure terminal
> interface fastethernet 0/4
> switchport mode access
> switchport access vlan 31
> end

Para verificações:

> show vlan

Enable IP forwarding and Disable ECMP echo ignore broadcast

> echo 1 > /proc/sys/net/ipv4/ip_forward
> echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

Configuração das rotas no Tux3 e no Tux2 de forma a ambos se conectarem:

Tux3:

> ip route add 172.16.31.0/24 via 172.16.30.254

Tux2:

> ip route add 172.16.30.0/24 via 172.16.31.253

Verificação que a interface GE0 do Router Cisco estava conectada ao Switch e que a interface GE1 Router Cisco estava conectada ao Router.

Configuração do Router Cisco com NAT:

> conf t
> interface gigabitethernet 0/0 *
> ip address 172.16.31.254 255.255.255.0
> no shutdown

> ip nat inside
> exit

> interface gigabitethernet 0/1*
> ip address 172.16.1.39 255.255.255.0
> no shutdown
> ip nat outside
> exit

> ip nat pool ovrld 172.16.1.39 172.16.1.39 prefix 24
> ip nat inside source list 1 pool ovrld overload
> access-list 1 permit 172.16.30.0 0.0.0.7
> access-list 1 permit 172.16.31.0 0.0.0.7

> ip route 0.0.0.0 0.0.0.0 172.16.1.254
> ip route 172.16.30.0 255.255.255.0 172.16.31.253
> end

Para verificar comunicação a partir do Cisco Router:

> ping [tuxes]
> ping 172.16.1.254
> ping 104.17.113.188 (internet)

Para finalizar configuramos a gateway default dos tuxes:

Tux2:

> ip route add default via 172.16.31.254

Tux4:

> ip route add default via 172.16.31.254

Tux3:

> ip route add default via 172.16.30.254

Logs Wireshark

Tux2

Tux3

Tux4