

Universidade do Minho
Mestrado Integrado em Engenharia Informática (MIEI)
Redes de Computadores
TP3: Camada de Ligação Lógica: Ethernet e Protocolo ARP
(Parte I)

1. Objectivos

O objectivo deste trabalho é estudar, de uma forma genérica, a camada de ligação lógica, focando o uso da tecnologia Ethernet e o protocolo ARP (*Address Resolution Protocol*).

O protocolo ARP, descrito na RFC 826 (<http://tools.ietf.org/html/rfc826.html>), é usado pelos equipamentos em rede para efetuar o mapeamento entre os endereços de rede e os endereços de uma tecnologia de ligação de dados. Desta forma, o protocolo ARP permite determinar, por exemplo, qual o endereço Ethernet que corresponde a um endereço IP particular.

2. Introdução

Um dos conceitos mais importantes de uma pilha protocolar estruturada em camadas é que cada camada fornece serviços às camadas superiores e usa os serviços disponibilizados pelas camadas inferiores. Por exemplo, a camada de ligação lógica oferece os seus serviços à camada de rede e através dela às camadas superiores (transporte e aplicação) e utiliza, por sua vez, os serviços da camada de ligação física.

O serviço mais básico prestado pela camada de ligação lógica é a **transferência de dados** de um nó para os nós imediatamente adjacentes na topologia da rede. No nó de origem cada unidade protocolar da dados (PDU) de nível de rede¹ é colocado dentro da trama de nível de ligação, sendo depois enviado através da camada física para o nó destino. No destino, o nó recebe a trama do nível físico, extrai o pacote IP (datagrama) da trama recebida e entrega-o ao nível de rede para ser processado.

Outros serviços que um protocolo do nível de ligação lógica pode fornecer são: controlo de acesso ao meio, entrega fiável de dados, controlo de fluxo e controlo de erros (detecção e correção). Estes serviços podem ser oferecidos por outros níveis da pilha protocolar, por exemplo, o nível de transporte com o protocolo TCP². A principal diferença é que no nível de ligação estes serviços são prestados na ligação entre nós adjacentes enquanto no nível de transporte são prestados fim-a-fim. Neste caso, uma ligação fim-a-fim envolve normalmente a travessia de um percurso na rede que passa por múltiplos nós intermédios.

Detecção e Correção de Erros

A detecção e correção de erros é outro exemplo de uma funcionalidade de serviço que pode ser prestada nos vários níveis da pilha protocolar.

¹ E.g., Datagrama IP

² *Transmission Control Protocol*, protocolo de transporte fiável usado na Internet.

Genericamente a detecção e correção de erros ao nível de ligação lógica, bastante mais sofisticada que nos níveis protocolares superiores, consegue detectar e corrigir erros de um bit e alguns erros com vários bits. O mecanismo de detecção mais comum é baseado num bloco de bits (B) criado pelo originador, que é uma função f da informação presente na trama a ser transmitida. Esse bloco de bits é acrescentado à trama original antes desta ser transmitida. O receptor ao receber a trama, utiliza a mesma função f e obtém, por sua vez, o bloco de bits ($B1$). Nessa altura, o receptor compara B com $B1$. Se não forem iguais significa que a trama tem erros e deve ser descartada. Se forem iguais a trama é considerada correta.

Existem diversos métodos de detecção e correção de erros com menor ou maior complexidade. O método de detecção CRC (*Cyclic Redundancy Check*) usa o princípio enunciado acima, em que o bloco $B1$ deve ser zero, atendendo a que a adição do bloco B à trama original a tornou divisível por f . Este método, facilmente implementado em hardware, é usado em muitos protocolos de ligação lógica, nomeadamente em redes Ethernet e WiFi. O WiFi é a designação usada para a ligação em rede local sem fios, usada normalmente como sinónimo das normas IEEE 802.11a/b/g/n.

Protocolos de Acesso de Controlo de Ligação

Dois tipos de ligações comuns numa rede são as ligações ponto-a-ponto e as ligações multiponto, em particular, de difusão³. Uma ligação ponto-a-ponto envolve um nó emissor num extremo da ligação e um nó receptor no outro extremo. Ligações de difusão envolvem vários nós que enviam e recebem através de um meio de difusão partilhado. Numa ligação de difusão, quando um nó envia uma trama todos os outros nós recebem essa trama. Exemplo de ligações de difusão são as redes locais baseadas em Ethernet ou redes sem fios (e.g., Wi-Fi)⁴.

Num meio partilhado, se não houver controlo ou coordenação entre os nós pode haver colisões entre tramas transmitidas simultaneamente por dois ou mais nós. Quando há uma colisão de tramas é quase impossível aos receptores receberem corretamente as tramas transmitidas. Assim, um dos objectivos de um protocolo MAC (*Medium Access Protocol*) é coordenar o acesso ao meio de modo a reduzir ou eliminar a probabilidade de colisão de tramas, devendo os nós emissores envolvidos recuperar dessa situação.

Os protocolos MAC estão divididos em três categorias: protocolos de partição de canal, protocolos de passagem de ficha (*token-based*) e protocolos de acesso aleatório. Em particular, estes últimos são os mais usados nas redes locais.

Endereços MAC

A nível de ligação lógica, e em particular nas redes locais, os sistemas interligados são identificados por um endereço MAC. Um endereço MAC tem 48 bits de comprimento e é normalmente escrito em formato hexadecimal, por exemplo, 1A-23-F9-CD-06-9B. O endereço MAC é atribuído pelo fabricante da NIC (Network Interface Card) e não muda quando o nó emissor muda de rede. Daí ser também designado como endereço físico. Pelo contrário, um endereço IP é um endereço lógico, i.e. muda consoante a rede IP de acesso.

Normalmente, um nó terminal ou de interligação possui tantos endereços MAC quantas interfaces de rede ativas. Por exemplo, um *router* (apesar de operar sobre pacotes IP) tem também vários endereços MAC, um por cada interface de rede que interliga.

³ *Broadcast*, no original em inglês

⁴ *Wireless LAN*

Quando um nó quer enviar uma trama na rede local insere os endereços MAC de origem e destino na trama. Numa rede local de difusão, Ethernet ou WiFi, todos os nós da rede local recebem a trama. Cada nó receptor verifica se o endereço do destino MAC é igual ao seu. Em caso afirmativo, o campo de dados da trama (*payload*) é extraído e passado para o nível de rede; senão, a trama é descartada. Há uma exceção: se o endereço destino for FF-FF-FF-FF-FF-FF (endereço de difusão) todos os nós recebem e processam a trama.

Address Resolution Protocol

O principal objectivo do protocolo ARP (*Address Resolution Protocol*) é permitir fazer um mapeamento entre endereços do nível de rede (e.g. IP) e endereços nível de ligação lógica (MAC) por forma a possibilitar a entrega de dados entre nós adjacentes.

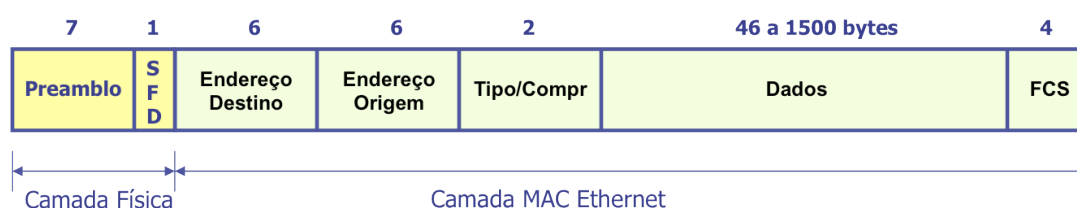
Suponha que um *host* na rede local quer enviar um datagrama IP para outro *host* na rede local. Suponha que conhece, provavelmente a partir do serviço de resolução de nomes – DNS, o endereço IP do *host* destino. Como sabe, o datagrama IP para ser enviado terá de ser entregue à camada de ligação lógica (L2) para ser encapsulado numa trama da tecnologia disponível e serializado para transmissão. A questão que se coloca é saber qual o endereço MAC destino a usar para enviar a trama que encapsula o datagrama IP, i.e. o *host* de origem vai ter de determinar o endereço MAC correspondente. Assim, sempre que necessário, o protocolo ARP permite obter o endereço MAC pretendido, através do uso das primitivas *arp-request* e *arp-reply*. Por cada resposta ARP recebida, e por questões de eficiência, cada nó da rede mantém uma tabela ARP (*cache*) que contém a correspondência entre endereços IP e os endereços MAC da rede local.

Note que o protocolo ARP tem um âmbito de operação restrito à rede local. Quando o destino IP é remoto, o protocolo ARP é usado para determinar o endereço MAC do *router* que está na mesma rede local, que, por sua vez, tem possibilidade de determinar qual o caminho que o datagrama IP deve seguir.

Ethernet

Ethernet é uma tecnologia de rede local bastante popular, havendo normas (*standards*) que permitem que a rede opere sobre diferentes meios de transmissão, topologias físicas e débitos de transmissão (tipicamente de 10Mbps a 10Gbps). A tecnologia Ethernet implementa um método de controlo de acesso ao meio que será detalhado nas aulas teóricas, e usa um formato de trama simples que inclui campos de controlo e um campo de dados.

Um trama Ethernet tem exactamente seis campos: (i) um campo para uma sequência de bits específica chamado *preâmbulo* (que o *host* destino utiliza para sincronizar o seu relógio com o relógio do *host* de origem e, assim, determinar quando começa a trama); (ii) o endereço MAC destino; (iii) o endereço MAC origem; (iv) um campo que indica o tipo de dados que a trama encapsula; (v) o campo de dados (*payload*); e (vi) o campo FCS (*Frame Check Sequence*) para o código de detecção de erros (CRC-32).



Interligação de Redes Locais

As redes locais são interligadas através de repetidores (*hubs*), pontes (*bridges*) ou comutadores (*switches*).

Os *hubs* são dispositivos de interligação que operam a nível físico, i.e. repetem o sinal que chega através de uma porta de entrada para todas as outras portas.

Os *switches*, tal como as *bridges*, são dispositivos do nível de ligação lógica, processando tramas do nível de ligação. Um *switch*, com a ajuda de uma tabela de comutação, mantém para cada endereço MAC a indicação da interface de saída. Assim, quando chega uma trama Ethernet a uma interface é comutada de imediato para a interface apropriada. O preenchimento da tabela é feito através de um mecanismo de auto-aprendizagem. Quando chega uma trama a uma das suas interfaces, o *switch* examina o endereço de origem da trama e acrescenta uma entrada na tabela com o endereço MAC correspondente. Quando chega uma trama que o *switch* não consegue comutar com base na tabela de comutação difunde-a através de todas as suas interfaces.

Por sua vez os *routers*, que serão estudados em detalhe mais adiante, funcionam ao nível de rede encaminhando pacotes IP (ou datagramas IP) com base no endereço IP destino, i.e., de maneira parecida à forma como os *switches* lidam com os tramas. Para esse efeito, os *routers* utilizam uma tabela de encaminhamento que é atualizada manualmente com rotas estáticas ou automaticamente através da utilização de protocolos de encaminhamento tais como o OSPF (*Open Shortest Path First*).

As entradas da tabela de comutação de um *switch* têm um tempo de vida pré-definido após o qual são removidas se não chegarem tramas que refresquem essas entradas.

3. Captura e análise de Tramas Ethernet

A captura e análise de tramas Ethernet será efectuada usando a aplicação Wireshark.

Assegure-se que utiliza a ligação com fios, i.e., a ligação à rede Ethernet da sala de aula.

Assegure-se que a *cache* do seu browser está vazia e está ligado em rede através da interface Ethernet.

Ative o Wireshark na sua máquina nativa.

No seu *browser*, aceda ao URL <http://miei.di.uminho.pt>.

Pare a captura do Wireshark.

Obtenha o número de ordem da sequência de bytes capturada (coluna da esquerda na janela do Wireshark) correspondente à mensagem HTTP GET enviada pelo seu computador para o servidor Web, bem como o começo da respectiva mensagem HTTP Response proveniente do servidor.

No sentido de proceder à análise do tráfego, selecione a trama Ethernet que contém a mensagem HTTP GET. Recorde-se que a mensagem GET do HTTP está no interior de um segmento TCP que é transportado num datagrama IP que, por sua vez, está encapsulado no campo de dados de uma trama Ethernet. Expanda a informação Ethernet II e observe o conteúdo da trama Ethernet (cabeçalho e dados (*payload*)).

Responda às perguntas seguintes com base no conteúdo da trama Ethernet que contém a mensagem HTTP GET.

Sempre que aplicável, deve incluir a impressão dos dados relativa ao pacote capturado (ou parte dele) necessária para fundamentar a resposta à questão colocada. Para imprimir um pacote, use File-→Print, escolha *Selected packet only* e *Packet summary line*, ou use qualquer outro método que lhe pareça adequado para a captura desses dados. Selecione o mínimo detalhe necessário para responder à pergunta.

1. Anote os endereços MAC de origem e de destino da trama capturada.
2. Identifique a que sistemas se referem. Justifique.
3. Qual o valor hexadecimal do campo `Type` da trama Ethernet? O que significa?
4. Quantos bytes são usados desde o início da trama até ao caractere ASCII "G" do método HTTP GET? Calcule e indique, em percentagem, a sobrecarga (*overhead*) introduzida pela pilha protocolar no envio do HTTP GET.
5. Em ligações com fios pouco susceptíveis a erros, nem sempre as NICs geram o código de detecção de erros. Através de visualização direta de uma trama capturada, verifique se o campo FCS está visível, i.e., se está a ser utilizado. Aceda à opção Edit/Preferences/Protocols/Ethernet e indique que é assumido o uso do campo FCS. Verifique qual o valor hexadecimal desse campo na trama capturada. Que conclui? Reponha a configuração original.

A seguir responda às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP.

6. Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.
7. Qual é o endereço MAC do destino? A que sistema corresponde?
8. Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.

4. Protocolo ARP

Nesta secção, pretende-se analisar a operação do protocolo ARP.

Verifique o conteúdo da *cache* ARP do seu computador.

- **Windows.** Digite `arp` ou `c:\windows\system32\arp` na linha de comando.
- **Linux/Unix.** O executável para o comando `arp` pode estar em vários locais. É habitual estar em `/sbin/arp` (Linux), `/usr/sbin/arp` ou `/usr/etc/arp` (para outras variantes de Unix). O comando `arp` sem argumentos ou com a opção `-a` mostra o conteúdo da *cache* do seu computador (consultar `man arp`).

9. Observe o conteúdo da tabela ARP. Diga o que significa cada uma das colunas?

No sentido de observar o envio e recepção de mensagens ARP, é conveniente apagar o conteúdo da *cache* ARP. Caso contrario, é provável que a associação entre endereços IP e MAC já exista em *cache*.

- **Windows.** O comando `arp -d *` apaga a *cache* ARP. A flag `-d` indica a operação de remoção e o `*` (*wildcard*) refere-se para todas as entradas da tabela.
- **Linux/Unix.** O comando `arp -d *` apaga a *cache* ARP. Este comando requiere privilégios de `root`).

Para observar o protocolo ARP em operação, apague novamente a *cache* ARP e assegure-se que o *cache* do browser está vazia.

Inicie a captura de tráfego com o Wireshark, e aceda a <http://cesium.di.uminho.pt>. Pode simultaneamente tentar efectuar *ping* para um *host* da sala de aula (e.g. *ping 192.168.100.xxx*). Pare a captura de tráfego e tente localizar o tráfego ARP.

Se necessário, limite os protocolos visíveis apenas a protocolos abaixo do nível IP. Para tal, seleccione *Analyze->Enabled Protocols* e remova a selecção da opção IPv4 e IPv6.

Responda às seguintes perguntas:

10. Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido ARP (*ARP Request*)? Como interpreta e justifica o endereço destino usado?
11. Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?
12. Qual o valor do campo ARP *opcode*? O que especifica? Se necessário, consulte a RFC do protocolo ARP <http://tools.ietf.org/html/rfc826.html>.
13. Identifique que tipo de endereços estão contidos na mensagem ARP? Que conclui?
14. Explícite que tipo de pedido ou pergunta é feita pelo *host* de origem?
15. Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.
 - a. Qual o valor do campo ARP *opcode*? O que especifica?
 - b. Em que posição da mensagem ARP está a resposta ao pedido ARP?

5. ARP numa topologia CORE

No emulador CORE prepare uma topologia com 3 *routers* em que n1 liga a n2 e este a n3.

16. Com auxílio do comando `ifconfig` obtenha os endereços Ethernet das interfaces dos diversos *routers*.
17. Usando o comando `arp` obtenha as *caches arp* dos diversos sistemas.
18. Faça *ping* de n1 para n2. Que modificações observa nas *caches ARP* desses sistemas? Faça *ping* de n1 para n3. Consulte as *caches ARP*. Que conclui?
19. Em n1 remova a entrada correspondente a n2. Coloque uma nova entrada para n2 com endereço Ethernet inexistente. O que acontece?

Adicione agora um *switch* (n4) à rede e ligue o *router* n1, e os *hosts* n5 e n6 a esse *switch*.

20. Faça *ping* de n6 para n5. Sem consultar a tabela ARP anote a entrada que, em sua opinião, é criada na tabela ARP de n6. Verifique, justificando, se a sua interpretação sobre a operação da rede Ethernet e protocolo ARP estava correcto.

(Fim da Parte I)

Nota: Deve ser entregue apenas um relatório referente às Partes I e II.