Redes de Computadores

(RCOMP – 2015/2016)

Pilha de protocolos TCP/IP.
IPv4; ARP; UDP; BOOTP/DHCP; ICMP; TCP e IGMP.

A camada IP

A pilha de protocolos normalmente designada por TCP/IP exerce atualmente um domínio quase total nas comunicações por computador assegurando deste modo o diálogo direto entre quase todos os tipos de equipamentos ligados por rede.

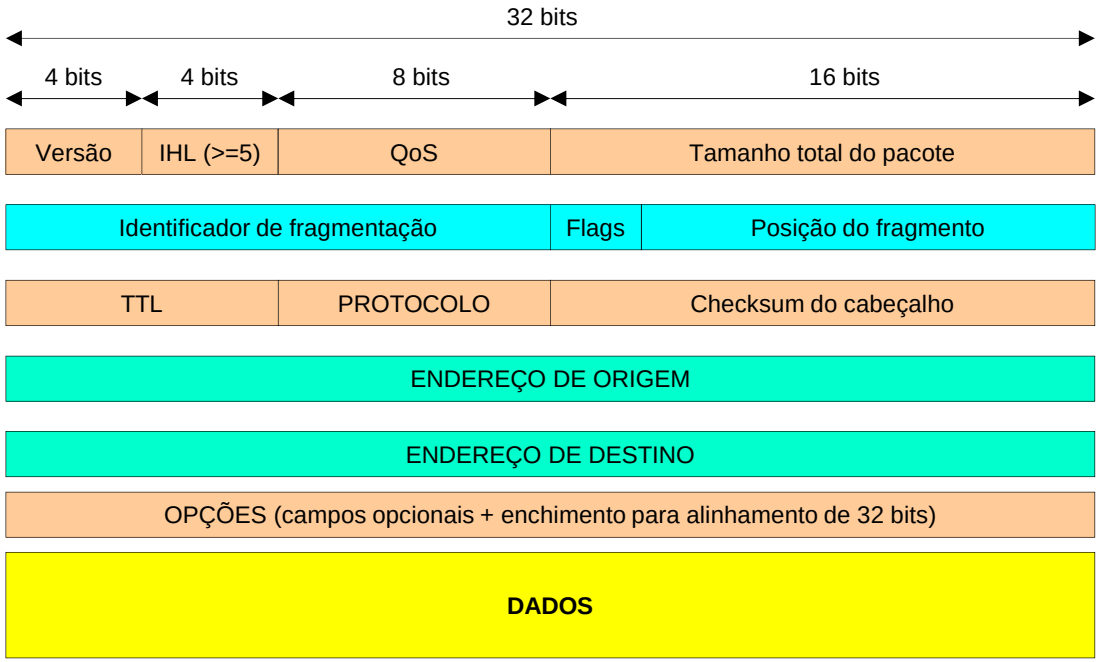
Numa altura em que várias tecnologias proprietárias concorriam entre si e o modelo OSI não se conseguia afirmar, o protocolo IP acabou por ser imposto por força da crescente adesão dos próprios utilizadores. O facto de o IP não ser um protocolo novo significa que já tinha sofrido um processo evolutivo derivado da sua utilização prática.

Principais características do protocolo IP:

>> Apresenta apenas as funcionalidades estritamente necessárias.
>> Definição de um formato de dados (Pacote IP).
>> Definição de endereços de rede, e dentro de cada rede, endereços de nó.
>> Tempo de vida dos pacotes. Identificador para multiplexagem da dados.
>> Fragmentação e reagrupamento.
>> Deteção de erros apenas no cabeçalho. Parâmetros QoS.

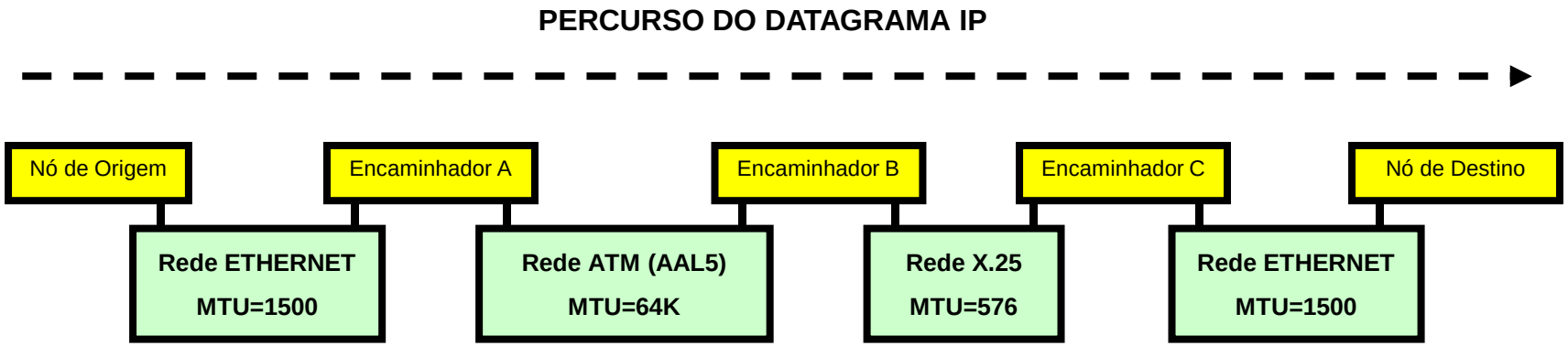
DATAGRAMAS IP

Um dos aspetos importantes de um protocolo com o objetivo de garantir a interligação universal é a definição de um formato para os pacotes que transportam os dados (*datagrama*). Uma vez definido dificilmente poderá ser alterado, neste campo o IP beneficiou de alguma maturidade que já tinha na altura em que a sua utilização se começou a generalizar.
Os *datagramas* IP podem ter até 64 Kbytes de comprimento, o cabeçalho segue uma organização (alinhamento) de 32 bits:



Tamanho dos DATAGRAMAS IP

Embora os *datagramas* IP possam ter até 64 Kbytes de comprimento, eles têm de ser transportados com recurso a tecnologias de nível 2 que podem não suportar esse volume de dados em cada PDU. O volume de dados que cada PDU de nível 2 pode transportar é conhecido por MTU (*Maximum Transmission Unit*), por exemplo numa trama ETHERNET-II o MTU é 1500 octetos.
O problema torna-se complicado porque o percurso de um determinado *datagrama* pode envolver muitos tipos diferentes de tecnologias de nível 2 com diferentes valores de MTU.



Existem duas abordagens para resolver este problema: a fragmentação e o PMTUD (*Path Maximum Transmission Unit Discovery*).

DATAGRAMAS IP - Fragmentação

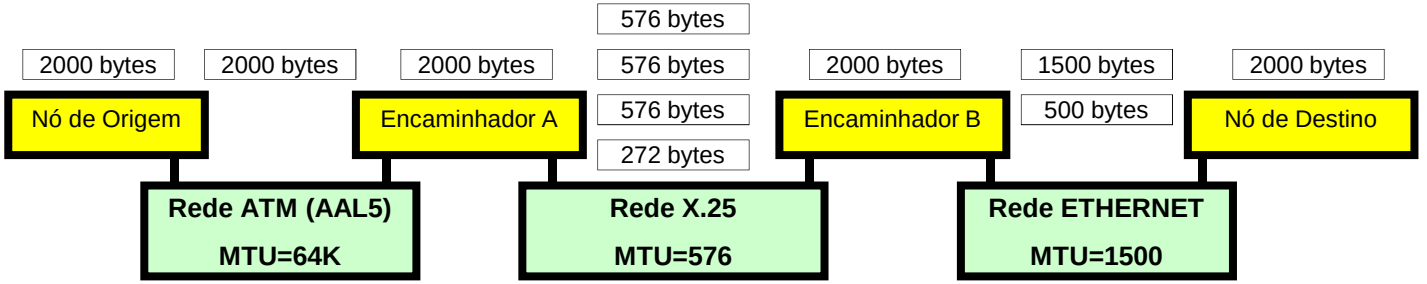
Identificador de fragmentação	*Flags*	Posição do fragmento

A solução teórica mais completa é a fragmentação que é diretamente suportada. Para fragmentar um *datagrama* é gerado um "identificador de fragmentação" único que servirá para identificar os fragmentos como pertencentes a um dado *datagrama*. O campo "posição do fragmento" serve para indicar a posição do fragmento no *datagrama* original. O primeiro bit do campo *flags* serve para indicar que existem mais fragmentos (valor 1) o valor 0 indica que se trata do último fragmento.

À primeira vista a fragmentação é a solução ideal porque deste modo os valores de MTU são sempre aproveitados ao máximo (menor *overhead*).

Na prática contudo, a fragmentação sobrecarrega bastante os nós de rede, em especial os nós que recebem os fragmentos e têm de fazer o reagrupamento.

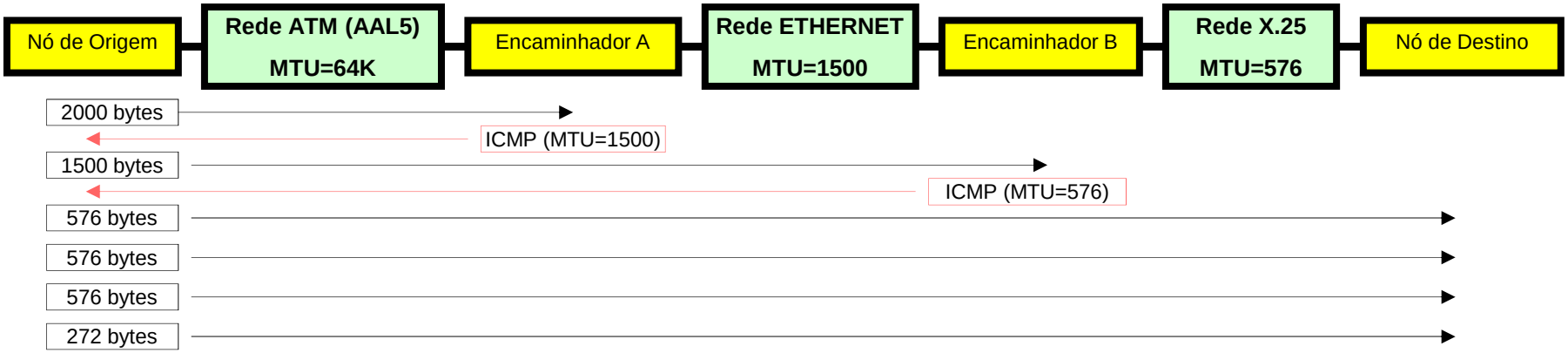
Este problema é especialmente desfavorável nos encaminhadores que devem apresentar atrasos de propagação (trânsito) o mais reduzidos possível.



PMTUD (*Path Maximum Transmission Unit Discovery*)

Uma vez que a aplicação prática da fragmentação apresenta grandes problemas de desempenho, foi desenvolvida uma técnica alternativa cuja aplicação se generalizou. A solução é simples: o segundo bit do campo *flags* do cabeçalho IP é colocado com o valor um, serve para indicar que o *datagrama* não pode ser fragmentado (DF).

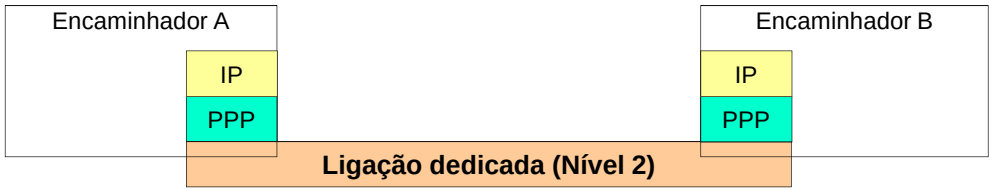
Quando um encaminhador recebe um *datagrama* destes com tamanho superior ao MTU seguinte ignora-o e devolve uma mensagem de erro ICMP "*Destination Unreachable*" como o código "*fragmentation needed and DF set*", indicando ainda o valor do MTU que causou o erro (RFC1191). Usando estas mensagens o nó de origem determina constantemente o PMTU associado a esse caminho.



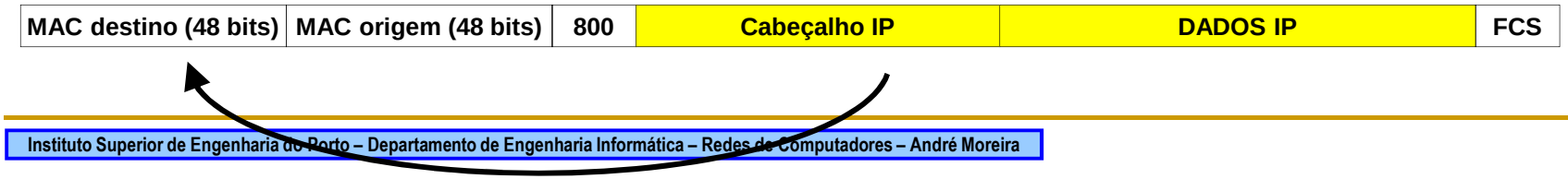
Transporte de *datagramas* IP

A pilha de protocolos TCP/IP recorre a um serviço externo de transporte de pacotes (tramas de nível 2), para assegurar a transferência dos *datagramas* IP entre encaminhadores.

Se o serviço externo de nível 2 for do tipo ligação dedicada, ou seja uma rede com apenas dois nós o endereçamento é implícito e normalmente usa-se o protocolo PPP para controlo das transferências da pacotes.



Se tratar de uma tecnologia multiponto (rede comutada ou de *broadcast*), é necessário estabelecer uma equivalência entre os endereços de nível 2 e os endereços de nível 3. Quando um *datagrama* IP é colocado no interior de uma trama de nível 2 (encapsulamento) é necessário determinar o endereço de destino (nível 2) para a trama.



ARP – *Address Resolution Protocol*

O protocolo ARP tem como objetivo assegurar a ligação entre o endereçamento IP de 32 bits e o endereçamento local de uma rede de nível 2 multiponto, tipicamente 802.3 com endereços de 48 bits.

A camada ARP gere uma tabela de equivalências entre endereços IP dos nós IP da rede local e os endereços MAC respetivos.

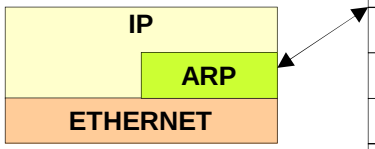


TABELA ARP	
IP	MAC
192.168.10.5	00:0A:B3:45:34:AB
192.168.10.34	00:0A:B3:50:54:14
192.168.10.240	00:0E:F3:50:00:FA

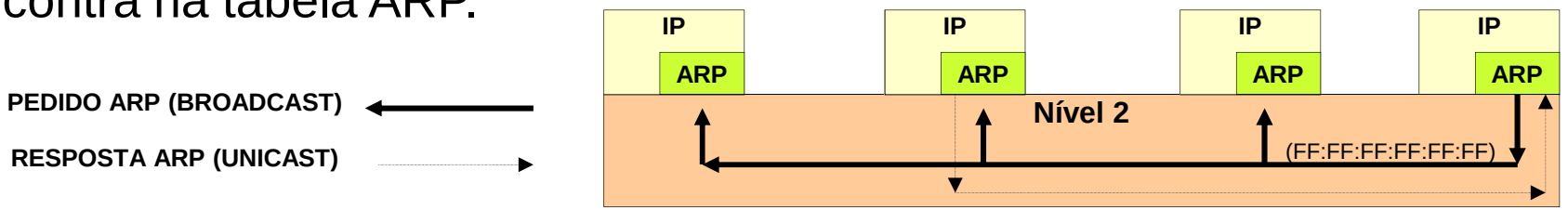
A gestão desta tabela é totalmente dinâmica, cada linha tem um tempo de vida de apenas alguns segundos, após esse tempo é eliminada.

Quando é necessário enviar um *datagrama* IP a um nó local, o endereço de destino (IP) é pesquisado na tabela para obter o respetivo endereço MAC de destino para colocar na trama (encapsulamento no nível 2).

Quando é necessário um endereço que não se encontra na tabela, então é usado o protocolo ARP propriamente dito para determinar esse endereço que será depois adicionado à tabela.

Protocolo ARP

O protocolo ARP é usado quando é necessário um endereço de nível 2 (MAC) correspondente a um dado endereço de nível 3 (IP), e este não se encontra na tabela ARP.



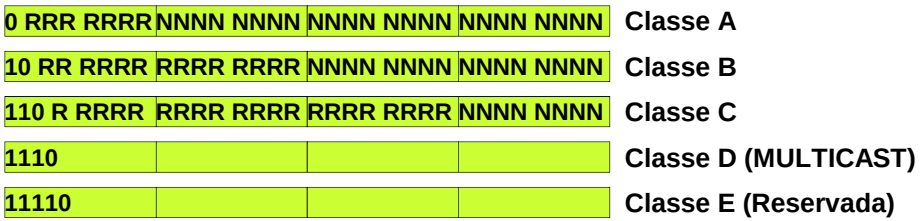
A camada ARP começa por enviar um PEDIDO ARP que contém o endereço IP pretendido, este pedido é enviado em BROADCAST, por isso todos os nós da rede recebem.

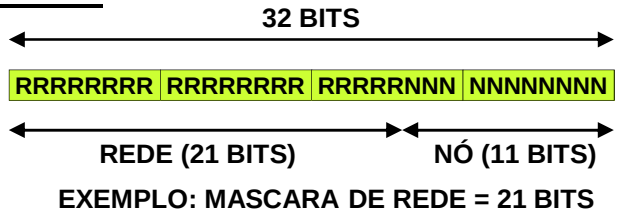
Todas as camadas ARP estão à escuta destes pedidos, quando os recebem comparam o seu próprio endereço IP com o que consta no pedido, se forem iguais enviam a RESPOSTA ARP que contém o endereço MAC pretendido.

Ao receber a resposta, o nó que desencadeou o processo, adiciona os novos dados à sua tabela ARP, estes novos dados serão válidos apenas durante algum tempo.

Endereçamento IP

O protocolo IP, como protocolo de rede que é, define não apenas endereços de nó, mas também endereços de rede.





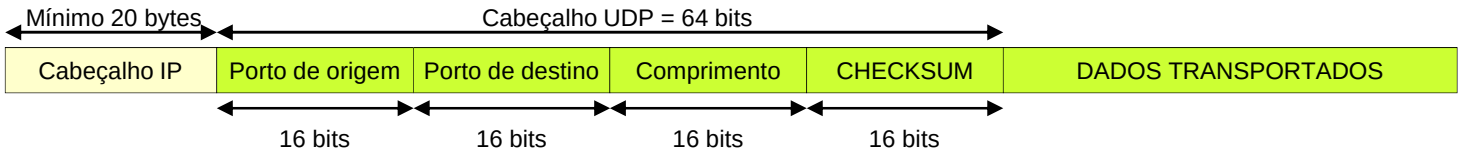
Um endereço IP (32 bits) é constituído por uma N bits de rede (mascara de rede) seguidos de (32-N) bits de nó.

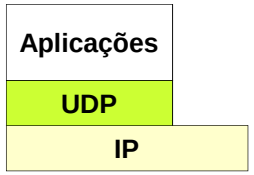
Um endereço IP (32 bits) identifica univocamente e universalmente (*Internet*) um nó e ao mesmo tempo identifica a rede onde esse nó se encontra.

Os nós de uma mesma rede IP têm endereços IP (32 bits) com os bits de rede exatamente iguais e os bits de nó obrigatoriamente diferentes. O endereço com 1 em todos os bits de nó significa *broadcast* na rede.

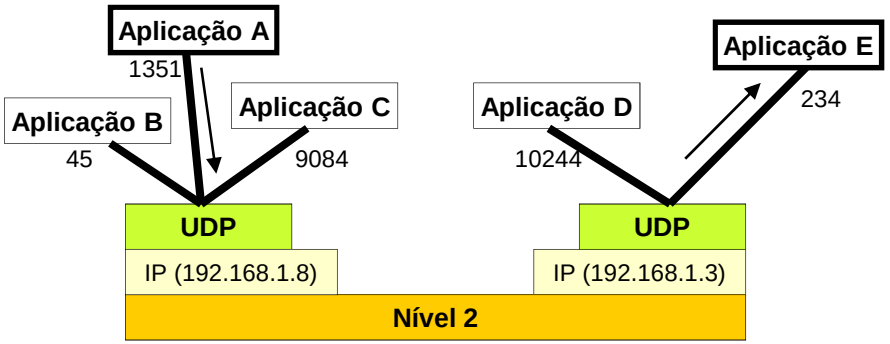
Protocolo UDP (*User Datagram Protocol*)

O protocolo IP não apresenta as funcionalidades mínimas para ser usado diretamente por aplicações de rede de uso geral. Um problema menor é a ausência de deteção de erros nos dados, mais grave é a ausência de um mecanismo de identificação de aplicações. O protocolo UDP implementa estas duas funcionalidades.





Os números de porto servem para identificar aplicações individuais no interior de um nó. Uma vez que em cada nó podem existir centenas de aplicações de rede esta identificação é fundamental.



EXEMPLO

Quando a "Aplicação A" pretende enviar dados para a "Aplicação E" não é suficiente especificar o destino "192.168.1.3". Também é necessário o número de porto, ou seja "**192.168.1.3:234**".

A origem destes dados será "192.168.1.8:1351" e o destino será "192.168.1.3:234". Se a "Aplicação E" enviar uma resposta, ela será enviada para "**192.168.1.8:1351**".

Deste modo as aplicações comunicam entre si sem interferir com terceiros.

Números de porto e servidores

No contexto do modelo cliente-servidor, os números de porto assumem um papel importante. Uma vez que os números de porto servem para identificar aplicações, o cliente necessita de conhecer não apenas o endereço IP da máquina onde está o servidor, mas também o número de porto que o servidor está a usar. Por exemplo “192.200.10.6:200” para indicar que o servidor está no porto 200 do endereço “192.200.10.6”.

Para simplificar a utilização das redes são definidos números de porto normalizados para cada tipo de serviço, os mais importantes abaixo de 1024.

Quando o utilizador não indica à aplicação cliente qual é o número de porto do servidor, a aplicação cliente assume que se pretende usar o porto “normal” para esse tipo de serviço.

Por exemplo um *browser* que é capaz de suportar diversos protocolos observa a parte inicial do URL para saber que porto usar.

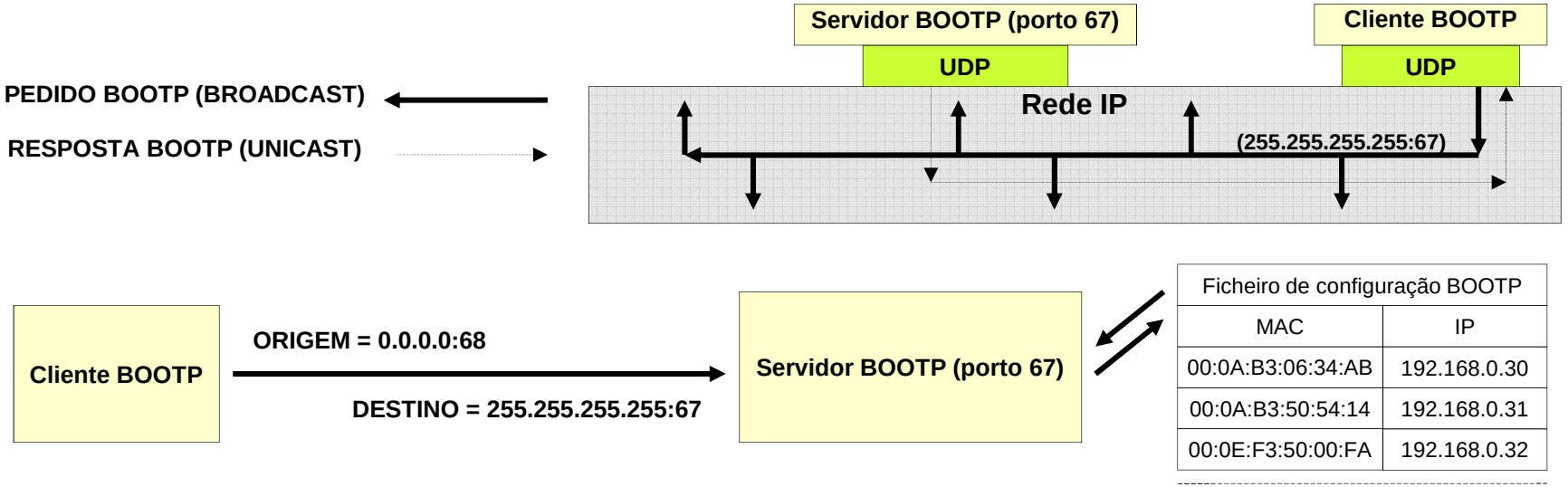
http://192.168.10.1 significa porto 80 no nó 192.168.10.1

https://192.168.10.1 significa porto 443 no nó 192.168.10.1

O ficheiro */etc/services* existente nos sistemas tipo Unix contém uma lista com os principais números de porto normalizados.

Protocolo BOOTP

O protocolo BOOTP (*BOOTstrap Protocol*) é um exemplo de um serviço UDP, no caso, o servidor recebe *datagramas* UDP no porto 67. Como tal os clientes BOOTP sabem que devem enviar os seus pedidos para o número de porto 67.
O objetivo deste protocolo é obter informação para configuração IP do nó. Assim sendo, o cliente nada conhece sobre a rede a que está ligado, nem sequer o seu próprio endereço IP.
Como o cliente não sabe a que rede está ligado, não pode usar o endereço de *broadcast* correspondente, em vez disso usa o endereço "255.255.255.255" que significa *broadcast* na rede local (seja ela qual for).



BOOTP dinâmico

O protocolo BOOTP cumpre totalmente as funções para que foi desenvolvido, fornecendo aos clientes todo o tipo de informações de configuração de que necessitam para funcionar e permitindo ainda acrescentar outras informações. Contudo é necessário que cada endereço MAC seja registado manualmente no ficheiro de configuração.

Muitas vezes é desejável que novas máquinas que se ligam à rede tenham automaticamente um endereço IP atribuído sem necessidade de nenhum ato de administração manual.

Os servidores BOOTP podem realizar esta tarefa, o problema é que não conseguem determinar quando é que um dado nó deixa de necessitar do endereço que lhe foi atribuído, ou seja esta atribuição é permanente.
Assim para o servidor BOOTP funcionar em modo dinâmico, é necessária uma quantidade de endereços IP igual ao número total de nós que potencialmente pode ser ligado à rede.
O ideal seria necessitar de uma quantidade de endereços IP igual ao número máximo de nós ligados simultaneamente.

Protocolo DHCP (*Dynamic Host Configuration Protocol*)

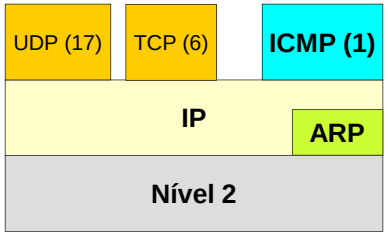
O protocolo DHCP não é mais do que uma extensão do protocolo BOOTP, ao qual é adicionado o conceito de aluguer do endereço (lease) por determinado tempo. Quando um cliente recebe um endereço via DHCP tem de controlar o tempo de aluguer e, se pretender continuar a usar o endereço, tem de renovar o pedido antes que o aluguer se esgote.

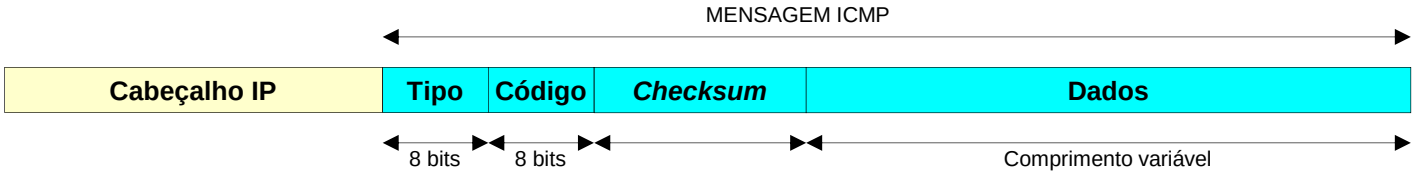
Uma vez esgotado o tempo de aluguer o servidor DHCP tem a liberdade de fornecer esse endereço IP a outro cliente. Na prática os servidores DHCP tentam manter sempre o IP de cada cliente, apenas quando se esgota a gama disponível é que os endereços são reutilizados para clientes diferentes.

Os próprios clientes DHCP tentam manter o mesmo IP e quando arrancam enviam ao servidor um pedido de renovação com o IP que tinham anteriormente.

Protocolo ICMP

O protocolo ICMP (*Internet Control Message Protocol*) é um protocolo auxiliar de controlo que permite realizar a notificação de vários tipos de situações anómalas relacionadas com o protocolo IP e ainda desencadear alguns tipos de operações de manutenção.



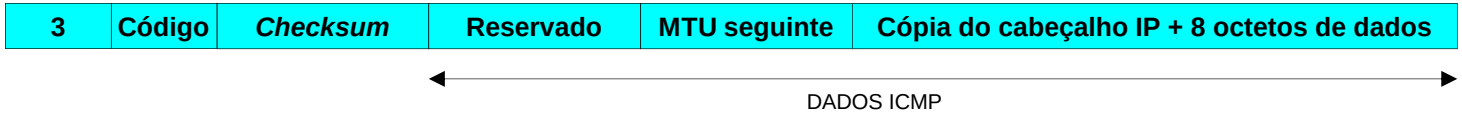


O campo "Tipo" identifica o tipo de mensagem, para cada tipo poderão existir vários valores possíveis para o campo "Código". O campo *Checksum* serve para detetar erros na mensagem ICMP e o campo "Dados" contém elementos dependentes do tipo de mensagem ICMP.

Por exemplo a mensagem ICMP tipo 0 significa "*echo reply*", é usada em conjunto com o tipo 8 ("*echo request*") e servem para teste de conectividade sendo usadas pelo bem conhecido comando "*ping*".
Em ambos os casos o código deverá ser zero, os dados são constituídos por um identificador e um número de sequência, ambos com 16 bits, seguidos de um padrão de bis que o requisitante pode colocar e no pedido e será devolvido na resposta.

Mensagem ICMP “Destino inatingível”

A mensagem ICMP de tipo 3 (*Destination Unreachable*) é enviada ao nó de origem quando um *datagrama* IP não pode ser entregue no endereço de destino pretendido. O campo “Código” é usado para indicar a razão dessa falha.



O campo “MTU seguinte” é usado apenas para o código 4. Em qualquer caso é devolvida na mensagem ICMP uma cópia da parte inicial do *datagrama* IP que causou o problema.

Alguns códigos “Destino inatingível”	
Código	**Significado**
0	NETWORK UNREACHABLE – significa que o DATAGRAMA não chegou à rede de destino.
1	HOST UNREACHABLE – significa que o DATAGRAMA chegou à rede de destino, mas não pode ser entregue no nó de destino.
2	PROTOCOL UNREACHABLE – significa que o DATAGRAMA IP chegou ao nó de destino, mas esse nó não tem o protocolo indicado.
3	PORT UNREACHABLE – o DATAGRAMA IP chegou ao nó de destino, mas não existe nenhuma aplicação no número porto de destino indicado.
4	O DATAGRAMA necessita de ser fragmentado e a *flag* DF está ativa. O campo “MTU seguinte” contém o valor do MTU que causou o problema.
5	Foi usada a opção IP “source route” e falhou

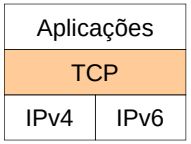
Outras mensagens ICMP importantes

Mensagem tipo 4 (*Source Quench*) – aviso gerado por um nó saturado, pede que o fluxo de dados seja reduzido.

Mensagem tipo 5 (*Redirect*) – indica ao nó de origem que está a usa o encaminhador errado para chegar ao destino pretendido e como tal deve corrigir o encaminhamento, para esse efeito os primeiros 32 bits da zona de dados da mensagem ICMP contêm o endereço IP do encaminhador correto. O nó de origem pode usar esta informação para alterar a sua tabela de encaminhamento.

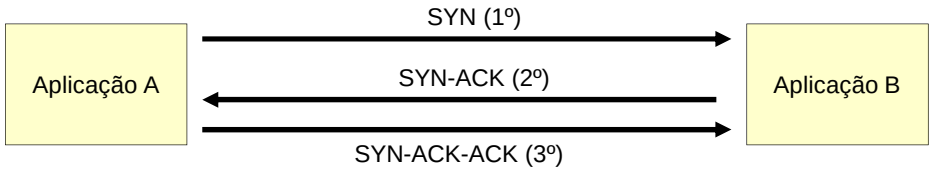
Mensagem tipo 11 (*Time Exceeded*) – indica que o TTL chegou a zero (código=0) ou que o tempo máximo de reagrupamento de um *datagrama* fragmentado se esgotou (código=1). Os primeiros 32 bits de dados não são usados, e segue-se o cabeçalho IP mais 64 bits (copiados do *datagrama* que causou o erro). O código 0 é usado pelo comando *traceroute*, gerando erros sucessivos nos vários encaminhadores fica a saber-se o caminho que os dados seguem.

Protocolo TCP

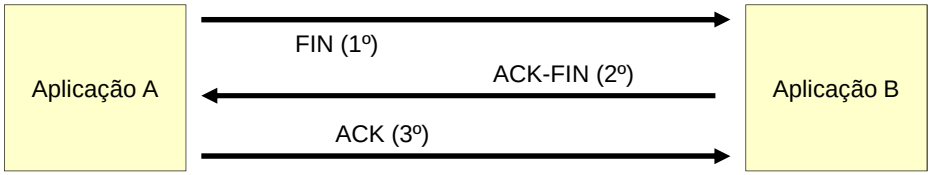


Ao contrário do protocolo UDP que é muito simples, o *Transmission Control Protocol* (TCP) é comparativamente bastante complexo devido ao conjunto de funcionalidades disponibilizadas. Usando um simples e não fiável serviço de *datagramas* do IPv4 ou IPv6, o TCP proporciona um serviço de transferência de dados em fluxo através de uma ligação lógica. A operação do TCP utiliza controlo de erros e de fluxo baseado a ARQ continuo que garante a total ausência de erros.

Estabelecimento de ligação – para que a comunicação seja possível em TCP é necessário ter uma ligação lógica, para o efeito um dos intervenientes envia uma mensagem SYN, que terá como resposta um SYN-ACK, finalmente é enviado o SYN-ACK-ACK.



Finalização de ligação – qualquer dos dois nós pode requerer o fim da ligação. Para esse efeito envia o comando FIN. O “parceiro” pode responder apenas com ACK, nesse caso a ligação fica meio aberta. Ou pode responder com um ACK-FIN (finalização com 3 envios):



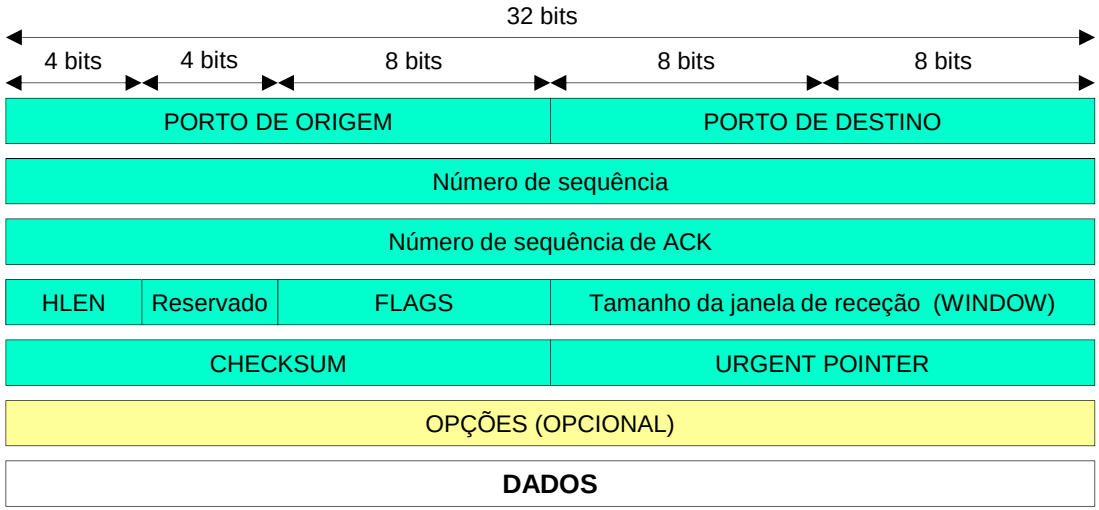
TCP - Segmentos

A informação de controlo e os dados do protocolo TCP são divididos em partes capazes de serem colocadas no interior de pacotes IP, conhecidas por segmentos TCP:

O parâmetro MSS (*maximum segment size*) indica o tamanho máximo que o segmento TCP pode ter em função do MTU determinado e eventualmente o *Path MTU*.

Em cabeçalhos simples (sem opções):

IPv4: MSS = MTU - 40
IPv6: MSS = MTU - 60

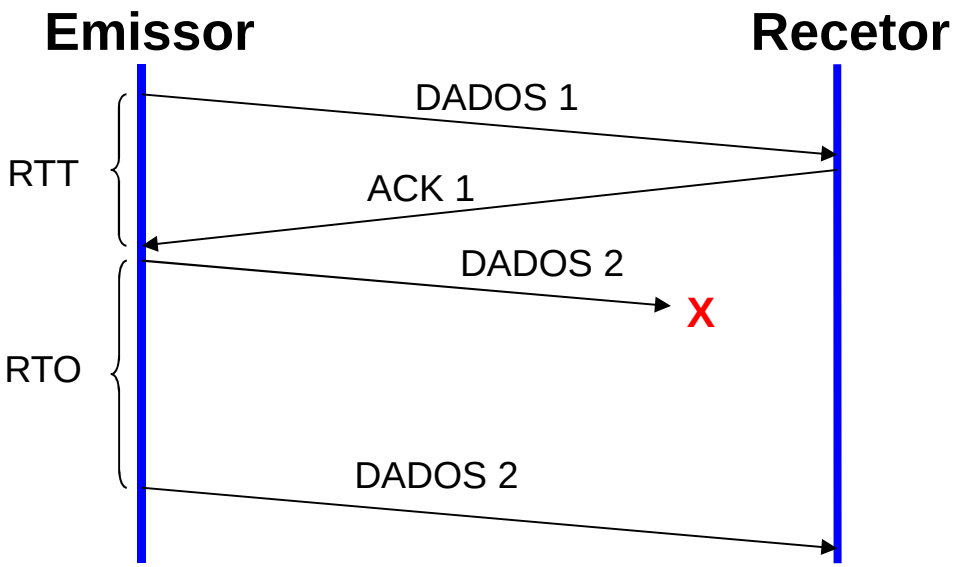


Os portos de origem e destino TCP são totalmente independentes dos portos UDP pois trata-se de duas camadas independentes no nível 4.
O campo *HLEN* especifica o tamanho do cabeçalho TCP em unidades de 32 bits, pode variar de 5 a 15 devido às opções. O campo *CHECKSUM* permite detetar erros no cabeçalho e dados, engloba ainda partes do cabeçalho IP. A forma de cálculo do *CHECKSUM* é diferente conforme o encapsulamento seja feito em IPv4 ou IPv6.
O campo *FLAGS* é composto por vários bits que identificam comandos tais como SYN; ACK; FIN e RST. O campo *OPÇÕES* pode ser usado com diversos objetivos, por exemplo informar o parceiro de qual o valor de MSS que deve usar.

TCP - RTO (*Retransmission TimeOut*)

Quando um nó emite dados TCP, aguarda que o respetivo ACK seja devolvido. Se após algum tempo o ACK não chega, emite novamente os mesmos dados. O tempo máximo que se aguarda à espera do ACK designa-se RTO (*Retransmission TimeOut*).

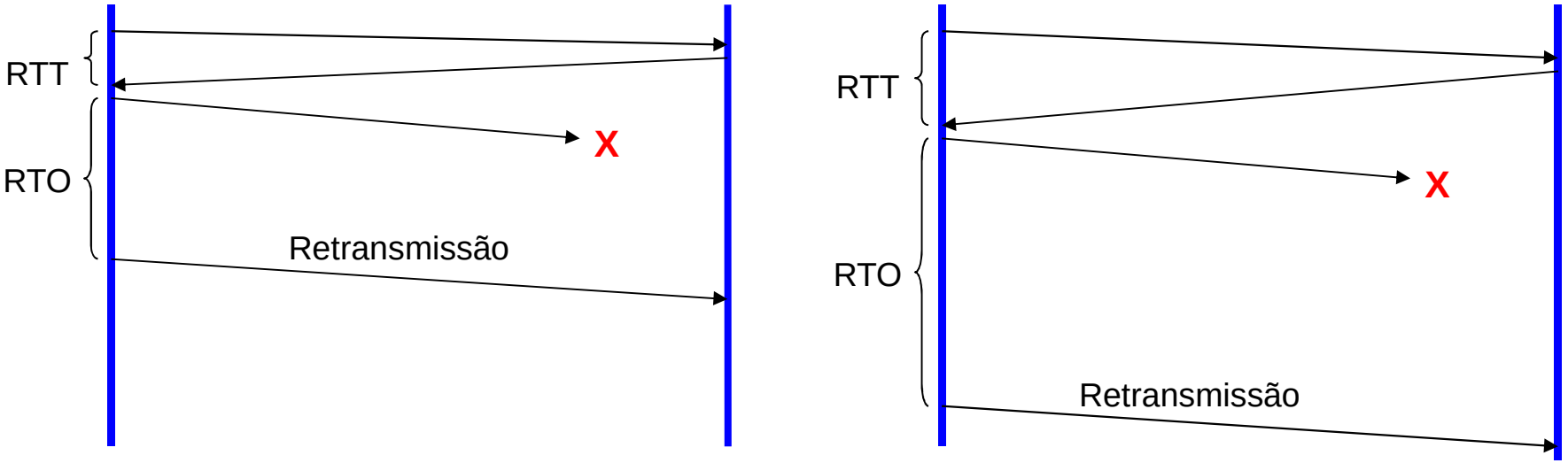
Uma vez que o nó recetor tanto pode estar na mesma rede local como no outro lado do planeta, os RTO terão de ser diferentes caso a caso.



Para adaptar de forma correta o valor do RTO ao estado da rede o TCP usa os valores de RTT (*Round-Trip Time*) que vai medindo, ou seja os tempos que decorrem entre a emissão dos dados e a receção do respetivo ACK.

TCP – Cálculo dinâmico do RTO

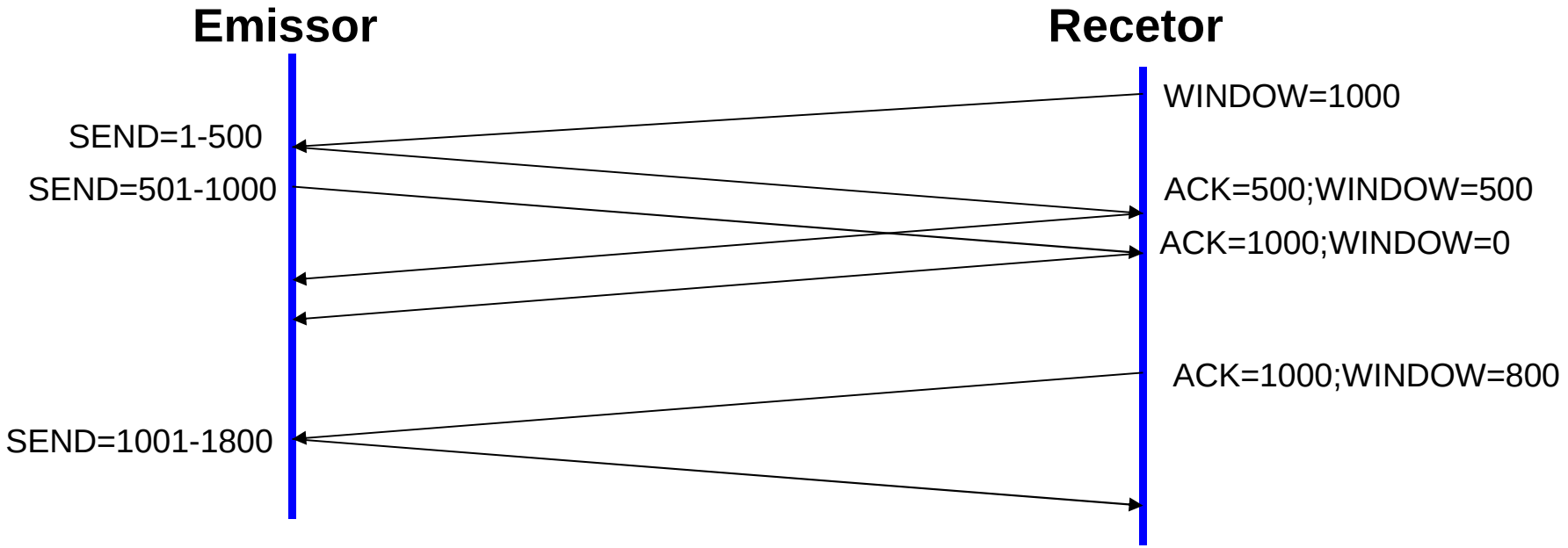
Cálculo do tempo de retransmissão (RTO) – o RTO é automaticamente adaptado às caraterísticas da ligação, começa com o valor de 3 segundos, mas quando o emissor começa a receber os ACK usa o valores do RTT. Começa por calcular RTO = 3 x RTT, usando posteriormente um algoritmo que envolve os vários RTT que vão sendo obtidos, com maior peso para os últimos e tendo em consideração as variações que ele sofre. O RTO mínimo tem o valor de 1 segundo.



A adaptação automática do RTO evita que os pequenos atrasos na rede desencadeiem a retransmissão. Quando a rede começa a ficar congestionada o RTT aumenta e o RTO é automaticamente atualizado.

TCP – Controlo de fluxo

O TCP utiliza o protocolo da janela deslizante, o recetor anuncia ao emissor o tamanho de janela disponível em número de bytes, esse tamanho de janela de receção indica quantos bytes é possível enviar sem receber o respetivo ACK.



O tamanho de janela é controlado pelo recetor indicando quantos bytes ele está disponível para receber, no exemplo acima, após o envio de 1000 bytes o emissor é forçado a aguardar, desta forma o recetor exerce controlo sobre o fluxo dos dados.

TCP – *Congestion avoidance* e *Slow Start*

O controlo de fluxo por janela deslizante é muito mais eficiente do que o *stop & wait*, em que para cada item de informação enviado tem ser recebido o ACK antes de enviar o item seguinte. Sob o ponto de vista de congestionamento o facto de existir um maior volume de informação a circular na rede sem ACK é contudo uma desvantagem.

Os algoritmos *Congestion avoidance* e *Slow Start* são usados em conjunto pelo TCP para atenuar estes problemas.

O *Congestion avoidance* define uma janela de congestão. Para efeitos de envio de dados é sempre considerado o menor valor entre a janela de congestão, controlada pelo emissor e a janela de receção, comunicada pelo recetor.

A janela de congestão (**cwnd**) é iniciada com o valor correspondente a um segmento (536 ou 512 bytes), é definido ainda o valor inicial do *slow start threshold size* (***ssthresh***) como 65535 bytes. Sempre que ***cwnd < ssthresh*** o emissor encontra-se num modo designado *slow start*.

Por cada ACK recebido é aumentado um segmento a *cwnd*. Em condições normais o *cwnd* atinge rapidamente o valor ssthresh saindo do modo *slow start*.

TCP – Deteção de congestionamento

Para atuar com os algoritmos *Congestion avoidance* e *Slow Start* o TCP necessita de detetar as situações de congestionamento:

- não receção de ACK (esgotamento do RTO)
- receção de ACK duplicados

A receção de um ACK duplicado pode indicar que foi feita uma retransmissão devido a ter sido atingido o RTO, mas que os dados chegaram todos ao destino. Isso aponta claramente para o congestionamento num nó intermédio. Nenhuns dados se perderem, apenas demoraram muito a atravessar a rede e isso desencadeou a retransmissão.

Infelizmente o TCP usa ACK duplos para avisar o emissor de que alguns dados foram recebidos fora de ordem, por essa razão apenas quando são recebidos triplicados, podem ser usados para assinalar claramente um situação de congestionamento.

TCP – *Congestion avoidance* e *Slow Start*

Quando o congestionamento é detetado, ssthresh é redefinido com o valor correspondente a metade do tamanho de janela atual (metade do menor valor entre cwnd e a janela anunciada pelo recetor). O novo valor para ssthresh nunca poderá ser inferior ao tamanho de 2 segmentos.

Adicionalmente se o congestionamento foi detetado pela não receção de um ACK, então o valor de cwnd é redefinido com o tamanho correspondente a um segmento, entrando por isso em *slow start*.

Por cada ACK recebido, cwnd é incrementado no valor de um segmento até atingir ssthresh, nessa altura sai do modo *slow start*. O ssthresh vai sendo ajustado para metade do tamanho da janela.

Depois de sair do modo *slow start*, o valor de cwnd continua a aumentar por cada ACK recebido, mas o valor desse aumento deixa de ser um segmento e passa a ser:

$$\frac{(tamanho - de - segmento)^2}{cwnd}$$

Enquanto cwnd vai aumentado, estes incrementos vão sendo cada vez menores.

Multicast em IPv4

Designa-se *multicast* (por oposição a *unicast*) a emissão para um endereço de destino que representa um conjunto de nós finais, quando um pacote é emitido com um endereço de destino *multicast*, todos os nós do conjunto recebem o pacote.

O *broadcast* é portanto um caso particular de *multicast*, em que o conjunto de nós são todos os nós de uma rede local. O *broadcast* recorre diretamente à tecnologia de nível 2 onde esse mesmo conceito existe (por exemplo o endereço ff:ff:ff:ff:ff:ff numa rede Ethernet).

O *multicast* é uma generalização em que se podem criar grupos arbitrários de nós, para esse efeito são usados endereços IPv4 começados pela sequência binária "1110" (224.0.0.0/4), ou seja 224.0.0.0 a 239.255.255.255. Também conhecidos por classe D.

Cada endereço *multicast* IPv4 designa um grupo de nós (grupo *multicast*), alguns estão já reservados para fins específicos, exemplos:

224.0.0.1 – "All hosts" (todos os nós da rede local, equivalente ao *broadcast*)

224.0.0.2 – "All routers" (todos os *routers* da rede local)

Multicast em IPv4 – *Multicast* Ethernet

Numa rede local *Ethernet*, o *multicast* IPv4 é diretamente suportado pela tecnologia de nível 2. Em *Ethernet* o bit menos significativo do primeiro byte do endereço Ethernet designa-se bit de *broadcast/multicast* os comutadores de nível 2 retransmitem em todos as portas os pacotes que são destinados a endereços com este bit ativo.

Os endereços *Ethernet* 01:00:5E:00:00:00 a 01:00:5E:7F:FF:FF estão reservados para *multicast* IPv4, nesse caso os 23 bits menos significativos devem ser exatamente iguais aos 23 bits menos significativos do endereço *multicast* IPv4.

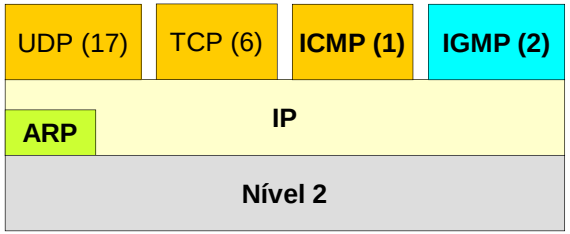
Note-se que como são usados apenas 23 bits, podem existir dois endereços IPv4 *multicast* diferentes mapeados para o mesmo endereço *ethernet multicast*.

A cada endereço IPv4 *multicast* corresponde um endereço *ethernet multicast*, por exemplo quando é emitido um pacote IPv4 com destino 224.0.0.1 (*all hosts*) ele será colocado num pacote *ethernet* com endereço de destino 01:00:5E:00:00:01.

Os comutadores de nível 2 retransmitem estes pacotes *ethernet* em todas as portas, cabe aos nós finais verificarem se pertencem ou não a esse grupo *multicast* e em função disso aceitar ou não o pacote.

IGMP – *Internet Group Management Protocol*

Tal como o ICMP opera diretamente sobre o IPv4, o objetivo principal do IGMP é gerir os grupos *multicast*.



Quando um router recebe um pacote IPv4 destinado a um endereço *multicast* IPv4 tem de saber se em cada uma das redes a que está diretamente ligado existem nós que são membros desse grupo *multicast*, pois apenas nesse caso deve retransmitir o pacote na rede.

Os routers enviam periodicamente a mensagem IGMP *Membership Query* para o endereço 224.0.0.1 (all hosts) nas redes a que estão ligados. Cada um dos nós deve responder com uma mensagem IGMP *Membership Report* para cada um dos grupos a que pertence.

Cada mensagem de resposta IGMP *Membership Report* é enviada para o endereço IPv4 *multicast* correspondente ao grupo, mas os nós não respondem de imediato, aguardam um período de tempo aleatório antes de responder. Se durante o período de espera recebem de outro nó um *Membership Report* do mesmo grupo abortam o processo e não enviam a resposta. O router só necessita de uma.