1. TRANSMISSÃO DE DADOS DIGITAIS

1. Sinais são fenómenos físicos com capacidade de se propagarem, ou seja, percorrerem uma certa distância, sendo **que para diferentes tipos de ondas há diferentes meios ideais de propagação.** *Exemplos: Luz – fibra ótica; Corrente Elétrica – cobre (metais); radiação Eletromagnética - propaga-se melhor no vazio, mas também atravessa materiais.*
    - Os sinais eletromagnéticos surgem aquando duma variação de corrente elétrica, sendo caraterizados pela sua **frequência,** uma vez que esta condiciona as propriedades do sinal. *Exemplos: baixas frequências: propagação no espaço muito limitada; rádio frequência: propagação em todas as direções; Micro-ondas: propagação começa a assemelhar-se à da luz.*
    - Associado aos sinais estão vários fenómenos tais como atenuação, uma vez que à medida que o sinal se afasta do emissor vai perdendo força. Este fenómeno é intensificado pelo ruido (tanto natural como artificial). Para uma boa transmissão é necessário um S/N **(signal noise ratio)** adequado. A atenuação/ propagação média de 1 sinal não é a mesma para todas as frequências – Largura de banda, esta corresponde **à faixa de frequência na qual um sistema tem uma resposta em frequência aproximadamente plana.**

2. Sinais digitais são **produzidos por aplicação de técnicas de codificação.** Exemplos: NRZ-L (não há retornos automáticos para zero); NRZ-M (0 e 1 são representados por contracurvas); NRZS (por transições de nível). Por norma um sinal digital é **produzido por variações bruscas entre patamares bem, gerados diretamente dos dados a transmitir e só podem ser usados se a frequência zero for suportada.** *Exemplo: no espaço de tempo que decorre entre as variações bruscas o sinal mantém-se estável num patamar, se o meio de transmissão não suportar a frequência zero esses patamares não são transmitidos e o sinal aparece distorcido.*

3. Sinais analógicos são usados quando o meio de transmissão impõe limites bem definidos quanto a valores de frequência (banda canal) e forma do sinal (sinusoidal). São **caraterizados por variações contínuas,** ao contrário dos sinais digitais. São **usados quando não é possível recorrer a sinais digitais.** A **produção destes sinais não pode ser feita diretamente a partir dos dados,** sendo que é feita através de modulação de sinal. Esta por sua vez é conseguida através da conjugação de varias técnicas: **FSK** – alteração de frequência; **ASK** – alteração da amplitude; **PSK** – Alteração da fase.

4. Numa comunicação em rede (network comunication) qualquer **participante é simultaneamente emissor e recetor**, sendo que estes são designados de nós de rede. Numa rede de computadores, qualquer nó deve ter a possibilidade de transmitir dados a outro nó.
    - Rede de broadcast: rede simples, com a **missão de receber os dados que são fornecidos pelo nó de origem e fazer esses dados chegarem ao nó de destino** indicado pelo primeiro. Estas redes usam um meio de transmissão comum para conseguir este objetivo, o que trás alguns inconvenientes, tais como **a necessidade de controlo de acesso ao meio (MAC),** uma **baixa eficiência** sob tráfego elevado, **falta de segurança**.
    - Rede de comutação: as redes de comutação são bastante mais **complexas**, baseiam-se num **conjunto de nós intermédios** com várias ligações entre si, que têm o papel de tomar decisões de encaminhamento em função do endereço de destino dos dados.
    - As redes impõem aos seus nos um **limite quanto ao volume máximo de dados que podem enviar em cada emissão**. Por norma inferior às necessidades das aplicações, o que obriga a **dividir a informação a enviar em partes mais pequenas** designadas por pacotes. A cada pacote vai ser acrescentada informação de controlo antes dos dados (cabeçalho de controlo / control header) e em muitos casos também após o fim dos dados (cauda/ tail).

    - Entre a informação de controlo existente no cabeçalho encontra-se o endereço do nó de destino que servirá para a rede fazer chegar o pacote ao nó correto. O endereço do nó de origem também se encontra no cabeçalho, serve para a rede ou o nó de destino saberem como responder ao pacote, ou simplesmente **dizerem "recebido" (ACK).**

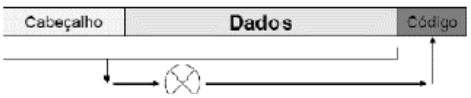
2. COMUNICAÇÃO EM REDE

1. Comutação de pacotes com circuitos virtuais: Numa rede de comutação, **os pacotes** mesmo pertencendo todos à mesma transação**, são encaminhados pelos nós intermédios de uma forma independente uns dos outros,** chegando ao destino por vezes **desordenados**.

2. Circuito virtual é um **caminho entre nó de origem e nó de destino** que é definido antes de se começar a enviar pacotes com dados. O processo começa pelo **nó de origem pede à rede para criar um circuito virtual com o nó de destino**, cujo endereço é indicado. Posteriormente os **nós intermédios da rede definem o caminho e associam-lhe um identificador**. A rede devolve o **identificador do circuito virtual ao nó de origem** que pode começar a enviar pacotes, sendo que agora não coloca nos

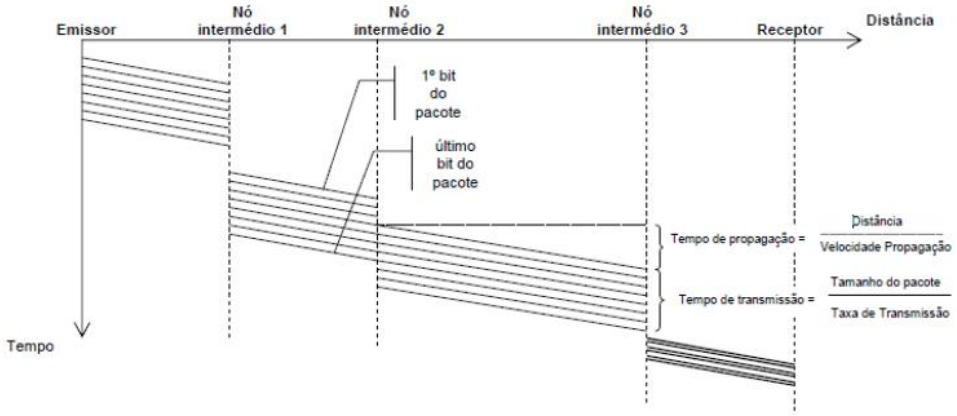
cabeçalhos dos pacotes o endereço do nó de destino, mas sim o identificador do circuito virtual. Como os **nós intermédios encaminham os pacotes segundo o circuito virtual pré-estabelecido**, por isso todos os pacotes seguem o mesmo caminho.

- Os erros de transmissão podem ser detetados se o **emissor acrescentar ao pacote um código de validação**. O código é **produzido por uma função apropriada que recebe o conteúdo do pacote e produz um código** que representa esse conteúdo.



- O objetivo da função é que **qualquer** pequena **alteração nos dados de entrada leve à produção de um código diferente**. Desta forma o código produzido no emissor é enviado juntamente com o pacote, isso dá ao recetor a possibilidade de repetir o processo e confrontar os códigos. *Exemplos: diferentes - houve erro de transmissão; iguais – grande probabilidade de não haver erro algum.*
- FEC (Forward Error Correction) - funções que produzem códigos que são autocorretores, apenas justificados em casos especiais.

3. Atrasos na rede: pode existir um **atraso entre o instante em que um pacote de dados começa a ser emitido e o instante em que é recebido.** Existem 3 razões para isto aconteça:
   - Atraso de propagação proporcional à distância: os sinais não se propagam com velocidade infinita.
   - Tempo de transmissão: **a emissão/receção de um pacote de dados não é um processo instantâneo**, demora algum tempo, que será tanto maior quanto maior for o volume de dados e depende ainda da taxa de transmissão**, quanto maior for a taxa menor será o tempo necessário.**
   - FIFO (política de fila de espera): política seguida pelos pacotes recebidos nos nós intermédios para serem processados, sendo que em caso de tráfego elevado podem ficar retidos.
   - Store and foward: Retransmissão após armazenamento integral dos pacotes.
   - Cut-throught: Retransmissão antes de ter terminado a receção dos pacotes.



4. Controlo de fluxo: o **objetivo** é regular o fluxo de dados entre emissor e recetor para **evitar um overflow no recetor.** Na técnica conhecida por "stop & wait" **o emissor tem de aguardar um sinal (ACK)** do recetor antes de enviar o pacote seguinte. Tal torna-se inconveniente, devido à existência de atrasos de propagação, resultando assim numa transmissão lenta e de eficiência reduzida. A **solução passa pela utilização do protocolo de janela deslizante,** no qual o emissor em vez de aguardar pelo ACK de um pacote antes de enviar o seguinte, **pode** desde logo **enviar uma série de w pacotes em que w é o tamanho da janela e é um parâmetro configurável.** Apesar de após enviar W pacotes o emissor ter de aguardar, por cada ACK que chega torna-se possível emitir mais um pacote.
   - Ligação full-duplex: condições ideais, ou seja, **um nº w de pacotes apropriado, de forma a que não existam paragens de transmissão.**

5. Controlo de erros: tem como objetivo corrigir erros detetados, no entanto apesar de se poder recorrer a mecanismos autocorretores (FEC), na maioria das situações usa-se a **retransmissão dos dados** (BEC – Backward Error Correction; ARQ – Automatic Repeat Request). Passam a existir duas **respostas possíveis por parte do recetor**: ACK e NACK (significa que foi detetado um erro e como tal o pacote em questão deverá ser retransmitido).
   - Quando se usa o protocolo de janela deslizante o controlo de erros por retransmissão é conhecido por ARQ Contínuo.
   - **O erro pode ser mínimo, mas o pacote terá que ser retransmitido na íntegra. Logo, quanto maior for o tamanho do pacote, maior será o volume de informação a retransmitir e maior será o seu impacto na eficiência da transmissão.**
   - **Quanto maior for o pacote maior é a probabilidade de ocorrer um erro.**

3. ARQUITETURAS E PILHAS DE PROTOCOLOS

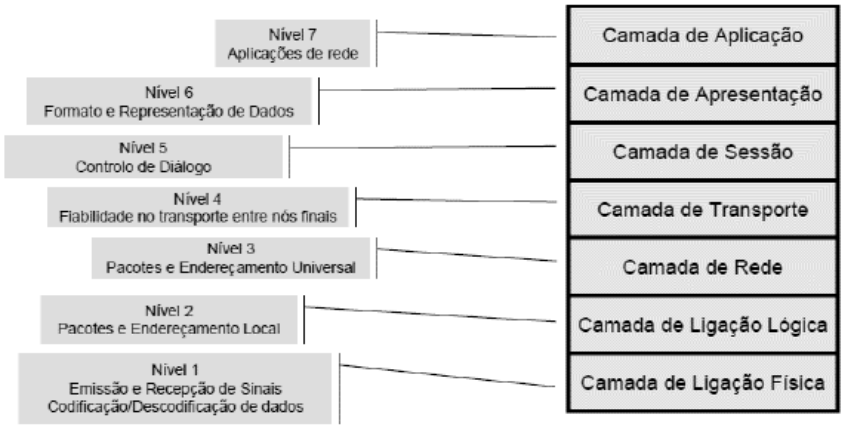
1. Camadas: A comunicação entre aplicações residentes em sistemas fisicamente afastados é um processo complicado porque envolve muitos problemas que têm de ser resolvidos. **Devido a esta complexidade, adotou-se uma estratégia de módulos sucessivos, normalmente designados de camadas.** Cada camada resolve uma parte dos problemas.

2. Modelo ou arquitetura: forma como as camadas estão organizadas e interagem entre si.

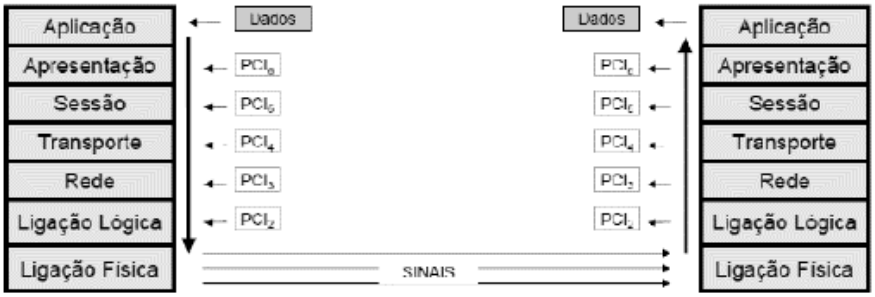
3. Arquiteturas proprietárias: as redes de computadores começaram a surgir espontaneamente no início dos anos 70. Nessa altura **cada fabricante desenvolveu o seu próprio sistema fechado, seguindo uma cultura de patentes para evitar que o mesmo fosse copiado por outros.** Estas arquiteturas são conhecidas por arquiteturas proprietárias.

4. Protocolos e pilha de protocolos: um conjunto de regras de uma dada camada é designado de protocolo, ao conjunto de protocolos de uma dada camada diz-se de pilha.

5. Modelo de referência OSI (open systems interconnection): sistema de normalização das arquiteturas de rede **desenvolvido pela ISO** (International Organization for Standardization).



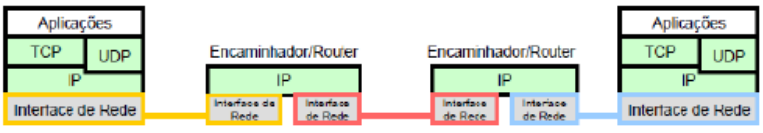
   - Camadas: as camadas sucessivas da pilha **interagem entre si segundo um modelo de prestação de serviço no sentido descendente através de pontos de acesso** (SAP).
   - Relação entre camadas: cada camada **usa os serviços prestados pela camada imediatamente abaixo** e **acrescenta-lhes novas funcionalidades e características, as quais obrigam à adição de informação de controlo** (PCI – Protocol Control Information).
   - (PCI – Protocol Control Information): o PCI é adicionado aos dados (SDU – Service Data Unit) que vêm da camada superior. Em cada camada, o conjunto **PCI mais SDU é designado por PDU** (Protocol Data Unit).
   - Protocolos: as **interações diretas ocorrem apenas entre camadas sucessivas e no nível físico**. No entanto, **as camadas do mesmo nível, residentes em nós de rede diferentes comunicam entre si usando o PCI dessa camada.**



   - Apesar de todos os objetivos do modelo OSI não terem sido alcançados, este foi um passo muito importante porque comporta um conjunto de **normas**, **nomenclatura**, **técnicas** e **ideias** que passaram a ser um ponto de referência para qualquer discussão na área das redes de computadores.

6. Arquitetura IEEE 802 (ISO 8802): A maioria das tecnologias de rede local estão normalizadas pelo **IEEE** e pela **ISO**, estas tecnologias correspondem aos níveis **1** e **2** do **MR-OSI**. *Exemplos: as redes Ethernet têm, por exemplo o identificador IEEE 802.3 (ISO 8802-3);* redes Ethernet a *100 Mbps são definidas na norma 802.3u e as redes Ethernet a 1 Gbps na norma 802.3z (aditamentos identificados por letras minúsculas).* A maioria das implementações de rede não usa a camada LLC **(camada de ligação lógica)** e interagem diretamente com a camada MAC **(camada de controlo de acesso ao meio).**
   - Camada MAC: assegura a transferência de pacotes de dados a nível local, endereçamento de nó e deteção de erros.

7. Arquitetura TCP/IP: A pilha de protocolos TCP/IP tem uma **origem oposta à do modelo OSI, foi desenvolvida sem grande planeamento teórico, usando uma abordagem minimalista** em que os problemas são resolvidos à medida que vão surgindo na prática, atingindo, contudo, alguns dos propósitos iniciais do referido modelo. Devido à generalização da Internet que obriga à utilização do protocolo IP **(Internet Protocol),** há uma tendência geral e irreversível de migração de todos os sistemas para a pilha TCP/IP e abandono de todos os outros protocolos resolvendo consequentemente a interligação de sistemas.
   - Constituição da pilha TCP/IP (os mais importantes): **IP (Internet Protocol)** fornece um serviço de transferência de dados independente da implementação da camada de ligação lógica que é depois usado por outros protocolos de nível superior; **UDP (User Datagram Protocolo)** protocolo de pacotes sem fiabilidade, apenas deteção de erros; **TCP (Transmission Control Protocol)** protocolo fiável, com ligação lógica, controlo de fluxo e controlo de erros, sendo o mais usado.
   - Encaminhamento IP: **A camada de rede IP usa uma qualquer tecnologia de transmissão de pacotes de nível 2** para proporcionar um **endereçamento de nó universal com 32 bits**, permitindo assim a interligação de redes de tipos diferentes.



- Endereçamento IP: introduz o conceito de **endereço de rede**, que tem como objetivo **facilitar o encaminhamento** pois passa a ser realizado rede a rede e não nó a nó como acontece no nível 2. **Por observação do endereço IP do nó de destino é possível determinar a que rede pertence.** As aplicações limitam-se a usar endereços IP juntamente com os protocolos UDP e TCP, tornando-se transparente o encaminhamento através de redes heterogéneas. **Os protocolos UDP e TCP usam números de 16 bits para etiquetar os dados** (Etiqueta: número de porto ou serviço), sabendo deste modo a que aplicações em particular devem entregar os dados.

8. Camadas multiprotocolo: é comum a coexistência de camadas paralelas numa pilha de protocolos. A existência de camadas paralelas significa que **existem fluxos de dados em paralelo que divergem (sentido ascendente) e convergem (sentido descendente) em camadas inferiores. Para que estas junções de fluxos possam ser invertidas mais tarde os dados têm de ser etiquetados para se saber a que camadas pertencem** (multiplexagem - o processo que consiste em permitir a circulação, numa ligação, de informações que provêm de diversas aplicações). Este processo repete-se sucessivamente ao longo de uma pilha de protocolos.

9. Modelo cliente-servidor: modelo de diálogo muito simples, no qual **o cliente envia um pedido ao servidor** (normalmente por ação do utilizador), tendo assim de **saber encontrar o servidor** (necessita do endereço de rede do servidor e do número de porto). **O endereço de rede é fornecido pelo utilizador, eventualmente sob a forma de um nome,** sendo que o número de porto é fixo para cada tipo de servidor. Posteriormente, **depois de receber o pedido, o servidor executa-o,** ficando o cliente à espera de uma resposta. Por fim, **depois de processar o pedido o servidor responde ao cliente,** sendo que para saber o endereço do cliente (e número de porto) basta verificar a origem do pedido.

10. Conclusão: embora tenha existido uma vasta variedade de arquiteturas proprietárias fechadas, **a expansão da internet com a sua arquitetura TCP/IP aberta veio alterar esse panorama,** sendo que o TCP/IP passou a ser obrigatório. Desta forma, o desaparecimento total das outras arquiteturas é iminente, uma vez que não é eficiente manter muitos protocolos num sistema. **O que se verifica é que as aplicações das arquiteturas proprietárias são modificadas para poderem funcionar sobre TCP/IP.** O sucesso do TCP/IP dá-se devido a este se situar em apenas duas das sete camadas do MR-OSI.

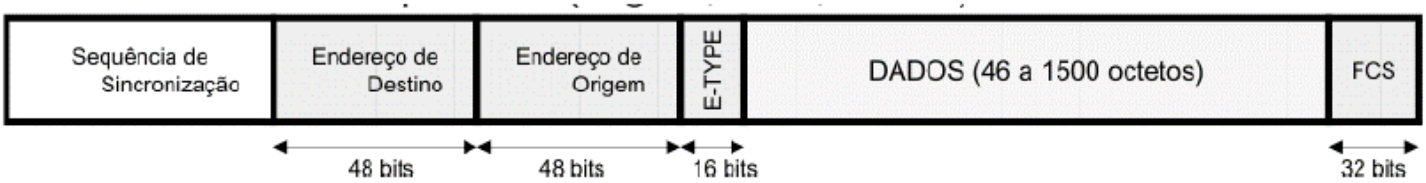
4. TECNOLOGIAS DE REDE LOCAL ETHERNET

1. Redes ethernet CSMS/CD: As redes ETHERNET (IEEE 802.3 / ISO 8802-3) foram originalmente desenvolvidas pela Xerox e atualmente **dominam nas redes locais.** Originalmente o **controlo de acesso ao meio (MAC) era um aspeto fundamental**. A **técnica CSMA/CD usada neste tipo de rede não é ideal,** tratando-se de um **mecanismo que não evita a colisões e com baixa eficiência** sob tráfego elevado. *Exemplos: a primeira versão a rede era constituída por um cabo coaxial ao qual todos os nós eram ligados (topologia BUS). As variantes mais importantes foram o 10base5 (10 Mbps/Digital/500m) e o 10base2 (10 Mbps/Digital/180m).*

2. Domínio de colisão: A técnica CSMA/CD **obriga a que as colisões de dados sejam detetadas por todos os nós antes da transmissão do pacote cessar.** Isto introduz limites na relação entre o tempo de transmissão do pacote e o atraso de propagação. **Para garantir a deteção de colisões por todos os nós fixa-se um tamanho mínimo para os pacotes** (tempo de transmissão) **de 64 octetos**, e um tamanho máximo para a rede (atraso de propagação). Esta **limitação relativa ao tamanho da rede** apenas se aplica ao problema das colisões e designa-se domínio de colisão.
   - O domínio de colisão pode não coincidir com a extensão da rede ETHERNET, os dispositivos "store & forward" isolam os domínios de colisão.
   - **Maiores taxas de transmissão resultam em domínios de colisão cada vez mais pequenos.**

3. Pacotes e endereços: desde a sua origem que as redes ETHERNET **mantiveram o mesmo formato de pacote e endereçamento,** o que **permite compatibilizar totalmente as várias evoluções técnicas ocorridas**. *Exemplo: fibra ótica a 10 Gbps pode ser ligada a segmentos de rede 10base5 e 10base2.* Cada **nó é identificado por um número de 48 bits**, designado de endereço de nó, endereço físico ou endereço MAC.
   - Normalmente estes endereços são representados sob a forma de **6 octetos em notação hexadecimal**, separados por dois pontos. *Exemplo: 00:60:B0:3C:93:DB*
   - Para **garantir que** os endereços são **únicos**, a **cada fabricante** de hardware **é atribuída uma sequência fixa** para os **primeiros 24 bits**.
   - Endereço de "broadcast": FF:FF:FF:FF:FF:FF. Um pacote com este endereço de destino chega a todos os nós da rede.

4. Formato de pacote: **os pacotes na camada de ligação lógica** são habitualmente designados de tramas, frames ou quadros, sendo que formatos de trama diferentes podem coexistir sem problemas na mesma rede ETHERNET. **Formato mais utilizado: "Ethernet II",** também conhecido por DIX (Digital, Intel, Xerox).



   - O **formato de pacote ETHERNET** é tão divulgado que as novas tecnologias como o 802.11 suportam este formato para permitir a interligação direta, mais simples, e implementa tudo o que é necessário, **incluindo identificador para multiplexagem** (E-TYPE) **e código para deteção de erros** (FCS – Frame Check Sequence).

5. Topologias do barramento à estrela: a topologia em barramento de **cabo coaxial das variantes 10base5 e 10base2** proporcionaram redes de custo extremamente reduzido. No início dos anos 90 começaram a **surgir outras implementações baseadas em pares de cobre entrançados (10baseT) e pares de fibras óticas (10baseFL e 10baseFB)**, nestas variantes cada nó possui duas ligações separadas (TX e RX) a um dispositivo concentrador, **a topologia foi então modificada para estrela.** Apesar da nova topologia**, o modo de funcionamento mantém-se e o CSMA/CD impõe restrições quanto ao domínio de colisão, por exemplo para o 10baseT é de 500 metros.** Mas há novas possibilidades: **Comutação, "Full-duplex".**

6. Comutação de tramas: **a topologia em estrela** veio abrir novas possibilidades, pois ao existirem ligações separadas TX e RX para cada nó da rede torna-se possível modificar radicalmente o modo de funcionamento destas redes**. Existindo ligações duplas dedicadas torna-se impossível a ocorrência de colisões, usando um** comutador (switch) **pode-se descativar o CSMA/CD.**
   - Comutador: concentrador/repetidor modificado de forma a receber várias tramas simultaneamente em quaisquer portas; emitir várias tramas simultaneamente em quaisquer portas; armazenar temporariamente tramas quando necessário; fixar os endereços de origem das tramas que vão chegando a cada porta (tabela MAC); e analisar os endereços de destino e retransmitir as tramas apenas nas portas corretas (tabela MAC).

7. Redes ethernet comutadas: funcionamento em "**full-duplex", sem colisões e sem controlo de acesso ao meio**, sendo que as tramas são encaminhadas com base no endereço de nó para que cheguem apenas ao nó de destino (gestão da tabela MAC). Com a eliminação dos domínios de colisão evita-se as restrições nas dimensões máximas da rede e reduz-se ao máximo de congestionamentos por ação dos comutadores.
   - Estes fatores **aumentaram de forma drástica a eficiência geral da rede**, traduzindo-se num **aumento de velocidade** aparente muito mais importante do que qualquer aumento de taxa de transmissão.

100base…	**TX**: Usa dois pares de cobre de um sistema de cablagem tipo 5 ou superior, o comprimento máximo de um segmento é 100 metros. **FX**: Usa duas fibras multimodo, o comprimento máximo de um segmento é 160 metros.
1000base…	**T**: Usa quatro pares de cobre de um sistema de cablagem tipo 5e ou superior, o comprimento máximo de um segmento é 100 metros. Não suporta "full-duplex". **SX**: Usa duas fibras multimodo, o comprimento máximo de um segmento é 220 metros ou 550 metros, respectivamente para fibras de 62,5 ou 50 micons. **LX**: Usa duas fibras monomodo, o comprimento máximo de um segmento pode ir até à dezena de Km de acordo com as especificações do fabricante.
10Gbase…	**SR/LRM/LR/ER/LX4**: são várias normas correspondentes a diversos tipos de fibra óptica que resultam em vários distâncias máximas suportadas que podem ir desde as dezenas de metros até à centena de Km. **CX4/Kx/T**: utilizam vários tipos de cablagem de cobre especial, com características eléctricas especiais ou um número muito elevado de pares de cobre.
40Gbase…	**CR4/SR4/LR4**: a primeira usa 4 pares de um cabo de cobre de características especiais, a segunda utiliza quatro pares de fibras multimodo e a terceira utiliza quatro pares de fibras monomodo.
100Gbase…	**CR10/SR10/LR4/ER4**: As primeiras duas utilizam 10 pares de, respectivamente, cabo de cobre especial e fibras multimodo. As duas últimas utilizam quatro pares de fibras monomodo e diferem no comprimento máximo que podem atingir.

8. Redes locais virtuais (vlan):
   - Norma IEEE802.1Q: uma rede local virtual (VLAN) é uma **rede lógica definida sobre uma rede física,** sendo que sob todos os pontos de vista uma **VLAN deve aparentar ser uma rede totalmente independente e separada.** Existem várias formas de criar redes locais virtuais, sendo que **numa mesma rede física podem etiquetar-se as tramas fazendo com que cada etiqueta** (TAG) **corresponda VLAN diferente.** A norma IEEE 802.1Q **define como se colocam etiquetas de VLAN nas tramas "Ethernet".** Para tal colocam-se as etiquetas nas tramas "Ethernet II" usando o valor 0x8100 no campo E-TYPE e o valor original do campo ETYPE desloca-se 4 octetos, isto é, é acrescentado o campo TCI com 16 bits que contém a identificação da VLAN com 12 bits e é acrescentado um novo campo E-TYPE com 16 bits, totalizando os 4 octetos.
   - Comutadores: podem **definir VLANs** sem recurso a etiquetas ("**untagged**"), por exemplo podem criar-se VLANs correspondentes a subconjuntos do total de portas do comutador. **Transforma-se o comutador em vários comutadores virtuais, mas para suportar mais do que uma VLAN na mesma porta, é necessário usar etiquetes, apenas uma VLAN pode ser "untagged" em cada porta.** Além de VLAN baseadas em portas alguns comutadores também suportam VLAN baseadas em endereços MAC.

9. Redes locais sem fios (WLAN/ IEEE 802.11): a normalização **mais conhecida é a IEEE 802.11** e respetivos aditamentos, sendo que atualmente está disponível a norma 802.11n capaz de funcionar **até 600 Mbps**, usando múltiplos canais simultâneos,

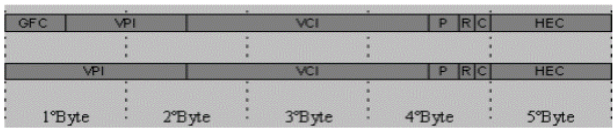
ou seja, vários emissores recetores. Mais penalizador do que a taxa de transmissão é o facto de haver um retorno aos primórdios das redes locais com o meio de transmissão partilhado e um mecanismo de controlo de acesso ao meio do tipo CSMA, o que acarreta ainda mais problemas de segurança do que as redes de meio partilhado de cabo.

- Modulação: usam exclusivamente rádio frequência na banda 2,4 GHz ou 5 GHz, tratando-se de sinais analógicos, sendo que a **transmissão de dados recorre a técnicas de modulação.** Atualmente as técnicas de modulação são complexas, sendo que usam vários sinais em simultâneo com combinações múltiplas **PSK** e **ASK**. A norma **802.11** e respetivos aditamentos definem várias técnicas alternativas de modulação, que conduzem a taxas de transmissão diferentes. A responsabilidade de obter a melhor taxa possível é dos nós, sendo que as opções de menor taxa são mais fiáveis com sinal de baixa intensidade. A gama de frequências usadas e as restrições quanto à **potência de emissão (100 mW)** produzem alcances muito reduzidos, especialmente no interior de edifícios onde é normalmente inferior a 50 metros.
- CSMA/CA: O **problema** das WLAN é o utilizarem **um meio de transmissão partilhado** que **só pode ser usado por um nó em simultâneo**. O mecanismo de controlo de acesso ao meio (MAC) seja 100% eficaz, sob tráfego elevado a taxa nominal é dividida pelo número de nós. **As transmissões são "half-duplex", um nó não pode emitir e receber ao mesmo tempo,** sendo que a receção de sinal em simultâneo com a emissão não é possível devido ao elevado custo, por isso o protocolo CSMA/CD não pode ser usado (não é possível detetar as colisões).
  Em alternativa usa-se o CSMA/CA (Collision Avoidance), que **tenta evitar colisões obrigando os nós a esperar que o meio esteja livre durante um dado período de tempo antes de poderem emitir.** Esta técnica não elimina as colisões, quando elas ocorrem, esse facto tem de ser detetado pelo emissor, para esse efeito sempre que um nó recebe uma trama válida envia ao emissor uma trama **ACK**.
- RTS/CTS: A técnica **CSMA/CA** pode ser combinada com **RTS/CTS**, esta técnica só é usada para o envio de pacotes com tamanho superior ao parâmetro "**RTS Threshold**" definido em cada nó. A técnica RTS/CTS consiste no envio pelo emissor de uma trama "**Request to Send**" ao recetor, eventualmente o recetor responde com "**Clear to Send**" que indica que está pronto a receber. Os outros nós quando recebem um **RTS** ou um **CTS** ficam impossibilitados de emitir durante algum tempo. A técnica **RTS/CTS** é especialmente eficaz no modo infraestrutura em que existe um dispositivo central, o AP ("Access Point") pelo qual todas as comunicações passam. **Numa WLAN sem AP todos os nós podem receber pacotes para retransmitir a outros nós,** sendo este modo de funcionamento designado por "ad-hoc".
- Modo infraestrutura: envolve a existência de um dispositivo central pelo qual todas as comunicações passam, pode-se considerar que se trata de uma topologia em estrela, embora sem fios. Este modo tem grandes vantagens, sendo que uma delas é que a técnica **RTS/CTS** é muito eficaz porque apenas o AP pode responder aos RTS enviando o CTS.
- Células: no modo de infraestrutura a rede é dividida em zonas de cobertura designadas de BBS ("Basic Service Set") e também conhecidas por células. Cada célula é **controlada** por uma "base station", também conhecida por **AP** (**"Access Point"**) e tem um identificador único (BBSID). **Um conjunto de células pode fazer parte de uma mesma infraestrutura** conhecida por ESS ("Extended Service Set"), identificado por **um nome até 32 caracteres** designado de SSID. **Todas as células de um ESS usam o mesmo SSID ("Service Set Identifier").** Nestas condições os nós de rede sem fios podem circular livremente entre os BSS do mesmo ESS sem perderem acesso à rede. **A passagem transparente de célula em célula é conhecida por "roaming".**
- Segmentação: a divisão de uma zona de cobertura em células reduz os efeitos negativos do meio de transmissão partilhado ao reduzir essa partilha a apenas uma célula. **Aumentando o número de células (APs) garante-se que cada célula vai conter um menor número de nós, atenuando os efeitos negativos da partilha do meio de transmissão.** O número de células deve ser o necessário para garantir a cobertura total da área pretendida, mas além disso deve garantir também que o número de nós em cada célula não é muito elevado. **Como referência o número de nós ligados a uma célula deve situar-se sempre abaixo dos 30.** A interligação de **APs** via ligação sem fios ("wireless distribution system") é possível, mas deve ser evitada pois não proporciona o isolamento de meios partilhados como é desejável.
- Tramas: o funcionamento das redes 802.11 é complexo, envolvendo nós de diferentes funções e diversa informação de controlo específica, por isso o formato de trama é também bastante complexo, por exemplo as tramas 802.11 contêm 4 endereços MAC. Apesar destas complexidades internas a ligação direta a redes locais com fios (Ethernet) é simples pois o formato de endereços é igual e os dados e campos de controlo podem ser transportados diretamente entre tramas 802.11 e tramas 802.3. Esta é uma missão do AP. Foi realizado um grande esforço para manter a compatibilidade direta com as tramas 802.3. Por vezes é conveniente usar tramas 802.11 muito pequenas, devido à elevada taxa de erros, mas as tramas 803.3 podem chegar aos 1518 octetos. Para resolver o problema os nós 802.11 são capazes de fragmentar uma trama em segmentos e reagrupar esses segmentos.
- Segurança: tratando-se de uma rede com meio de transmissão partilhado a segurança é desde logo muito precária, uma vez que o meio está livremente acessível (dentro de determinada área) os problemas são ainda maiores. Os **APs** implementam diversas formas de controlo de acesso como parte do processo de entrada de um nó na célula. A autenticação pelo endereço **MAC** do nó não é segura, alternativas mais sólidas são a autenticação de utilizador ou a utilização de uma **chave pré-partilhada (PSK)**.
- Confidencialidade: para garantir a confidencialidade é obrigatório recorrer à criptografia que é suportada pelo **AP**. Para esse efeito é necessário que o AP e o nó possuam uma mesma chave secreta. Esta pode ser pré-partilhada e

nesse caso funciona também como autenticação ou pode ser gerada durante o processo de autenticação do utilizador. Uma outra alternativa consiste na utilização de **criptografia de chave pública**.

5. TECNOLOGIAS WAN: ATM/ISDN E DSL

1. WAN - Redes ATM: A tecnologia ATM (Asynchronous Transfer Mode) **inicialmente** estava a ser **usada em LAN**, no entanto **as redes Ethernet evoluíram para a comutação** e aumentaram de taxa de transmissão, **condenando o ATM em LAN** devido ao seu custo mais elevado. **Atualmente as redes Ethernet começam a invadir o domínio ATM nas redes** WAN (wireless area network) com implementações 10 Gbps e 100 Gbps com alcances na casa das centenas de quilómetros.
   - Canais e caminhos virtuais: **a comutação é feita com circuitos virtuais,** o que trás vantagens significativas se esta técnica for usada entre nós finais e não em simples ligações dedicadas. **Os circuitos virtuais são designados** por canais virtuais e são **identificados por números de 16 bits (VCI),** e definem ligações lógicas entre aplicações nos nós finais. **Internamente a rede ATM faz o encaminhamento entre nós finais, não aplicações, e para simplificar o trabalho da rede definem-se caminhos virtuais que são identificados por números de 12 bits (VPI).** Cada canal virtual (exterior da rede) pertence a um caminho virtual (interior da rede), sendo que todos os canais virtuais com mesmo nó de origem e destino pertencem ao mesmo caminho virtual.
   - Células ATM: outra inovação é a utilização de **PDUs de tamanho fixo** e muito reduzido, que **tomam a designação de** células e têm apenas 53 octetos dos quais 5 são de controlo e os restantes 48 de dados, isso representa um "overhead" de quase 10% (5/53).



Cabeçalho de 2 células (externa/interna):
1. **UNI – User/Network Interface**
2. **NNI – Network/Network Interface**

Nas células UNI o VPI é igual a zero, o seu valor apenas é definido nas células que circulam no interior da rede.

O campo GFC (Generic Flow Control) que **apenas existe nas células UNI serve para controlo de fluxo local e multiplexagem da ligação à rede.** Os campos P e R (Payload Type) indicam o **tipo de dados transportados.** O campo C (Cell Loss Priority) indica **a prioridade da célula em caso de congestionamento**, se tiver este bit com o valor 1 é eliminada em primeiro lugar. O campo HEC (Header Error Correction) contém **um código CRC do cabeçalho que é autocorretor** para erros de 1 bit e deteta erros de mais do que um bit.

   - ISDN: integrado tipo ISDN/RDIS (Integrated Services Digital Network / Rede Digital de Serviços Integrados). A tecnologia a adotar para o B-ISDN (Broadband – ISDN) **seria** precisamente **o ATM**. O facto de ser necessário suportar vídeo, voz e dados conduziu às opções técnicas que foram tomadas, nomeadamente a **utilização de blocos muito pequenos com caminhos virtuais e sem deteção de erros nos dados**. Estas medidas garantem atrasos mínimos nos nós, fundamentais para suportar transmissões em tempo real. Com a expansão da utilização da "internet" a filosofia ISDN deixou de fazer sentido e a tecnologia ATM passou a ser usada para suportar a interligação de encaminhadores da própria "internet". Para suportar as diversas classes de serviços (vídeo, voz e dados) são definidas várias camadas de adaptação conhecidas por AAL (ATM Adaptation Layer), as implementações que servem para suportar dados são o **AAL3/4** e o**AAL5**.
   - (ATM) AAL5: para suportar a **transmissão de pacotes de protocolos de nível superior,** as redes ATM desenvolveram uma **implementação de dupla camada designada por** AAL3/4, sendo que com **a expansão do protocolo IP** foram necessários melhoramentos e eliminação de funcionalidades desnecessárias, deste esforço surgiu o AAL5.

Dados (0 a 65535 bytes)	Alinhamento (0 a 47 bytes)	CTL (2 bytes)	LEN (2 bytes)	CRC (4 bytes)

O campo de "alinhamento" serve para **garantir** que **o comprimento total** é **múltiplo de 48** para que o PDU possa ser dividido em segmentos de 48 bytes que "encaixam" diretamente em células ATM. O campo CTL **não é usado.** O campo LEN indica o **comprimento dos dados (sem alinhamento)** e o campo CRC serve para **detetar erros sobre todo o PDU.** O PDU AAL5 não tem campo de multiplexagem de protocolos, terá de ser usado o valor do VCI para esse efeito.

2. WAN (wireless area network): antes da generalização do protocolo IP, **as redes WAN forneciam um serviço de nível 2 para transferência de dados entre nós finais** que podia ser usado para transportar todo o tipo de dados como dados de protocolos de rede como o IP, o IPX ou outros (filosofia ISDN/RDIS). **Atualmente os clientes das redes WAN apenas pretendem conectividade IP**, uma vez que até para aplicações mais exigentes como transmissão de voz e imagem surgem cada vez mais soluções baseadas na utilização do IP (VoIP, etc.).
   - Interligação de nós IP: os serviços de interligação WAN entre nós finais estão em declínio porque sendo a norma comum o protocolo IP é muito mais lógico e simples usar diretamente esse protocolo do que tentar impor uma tecnologia homogénea de nível 2. As redes **WAN comutadas a funcionar no nível 2 estão a ser substituídas por interligações entre nós IP**, encaminhadores de pacotes IP ("routers IP"), a funcionar no nível 3. A interligação dos encaminhadores IP continua a necessitar de parte da antiga tecnologia WAN, mas faz um uso muito limitado, pois, muitas vezes não são mais do que ligações dedicadas simples, **neste contexto todas as funcionalidades mais avançadas**

dessas tecnologias, como por exemplo o ATM, são totalmente desaproveitadas assistindo-se a um avanço da tecnologia ETHERNET para os domínios WAN.

- Operadores de telecomunicações: normalmente as comunicações de longa distância (WAN) apenas podem ser asseguradas por operadores autorizados. *Exemplo: as emissões privadas via rádio estão sujeitas a várias restrições legais, por exemplo quanto à potência de emissão, que tornam impossível a sua utilização no domínio WAN, sendo ilegal enviar sinais RF 802.11 para o exterior dos edifícios; as ligações privadas por cabo não podem atravessar zonas públicas, tais como atravessar um arruamento sem recorrer a um operador oficial.*
- Tecnologias: uma das vantagens de se utilizar um protocolo de rede global (IP) é que podemos misturar todas as tecnologias de ligação de dados sem qualquer problema. As **interfaces de ligação a essas tecnologias são muitas vezes fornecidas pelos próprios operadores**.
  Exemplos: linhas dedicadas alugadas, analógicas ou digitais, ISDN (RDIS) e rede telefónica analógica, X.25 (TELEPAC), FRAME-RELAY e ATM, etc.).

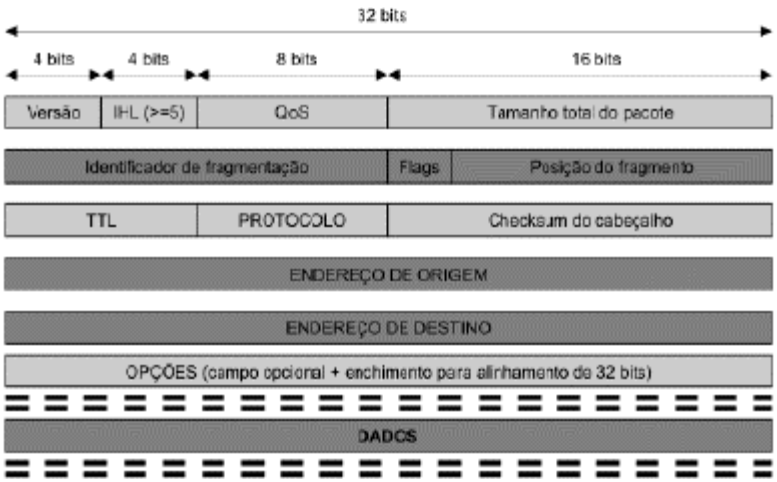
- Ligações dedicadas: em grande parte a utilização atual de infraestruturas WAN limita-se à interligação de encaminhador IP, embora a interligação de encaminhadores possa **recorrer a redes de comutação tais como o ATM ou o FRAME-RELAY**, simples ligações dedicadas são suficientes. **Redes de comutação ou de outros tipos podem ser reduzidas a redes de dois nós,** tornando-se ligações dedicadas, sendo que sobre ligação dedicada usa-se habitualmente o protocolo PPP ("Point to Point Protocol") que foi especialmente concebido para **controlar o transporte de pacotes de rede nestas situações.**

3. Tecnologias de acesso WAN: mesmo tratando-se de um serviço de transporte **IP**, no nível de rede, para que este chegue ao cliente é necessário um **mecanismo de transporte de nível 2 para garantir a ligação entre o cliente e o operador**, estabelecendo uma ligação designada de "Local Loop". Existem alternativas ao uso de fibras óticas que se verificam dispendiosas tais como: acesso sem fio (**WLL**), tipicamente através de um operador **GSM**; ligação à rede telefónica analógica (central telefónica); redes de televisão por cabo (**CATV**); rede alimentação elétrica (Power Line Communication).
   - Wireless local loop: o desenvolvimento da tecnologia de rede sem fios torna viável a sua utilização para **ligação do operador ao subscritor**, para tal as **normas 802.16,** também conhecidas por "Wireless MAN" e WiMAX (WorldWide Interoperability for Microwave Access) são mais apropriadas para este tipo de aplicação do que as 802.11, permitindo taxas de dados até 70 Mbps para distâncias inferiores a 2 Km e com alcances até à centena de Km para taxas de dados mais reduzidas (a 10 Km a taxa máxima é 10 Mbps). *Exemplos: a rede GSM 3G suporta taxas até 16 Mbps e tem a vantagem de já possuir uma cobertura perfeitamente instalada; a rede GSM 4G e novos aditamentos 802.16 deverão atingir 100 Mbps na modalidade móvel e 1 Gbps na modalidade fixa.*
   - DSL (Digital Subscriber line/loop): as técnicas DSL **tiraram partido de uma ligação telefónica já existente entre o cliente e a central telefónica,** constituída por um par de condutores de cobre, sendo capaz de transportar sinais analógicos até quase 2 MHz, contudo a utilização telefónica tradicional ocupa apenas uma pequena faixa até aos 4 KHz.
     Embora a largura de banda disponível seja razoável, a qualidade das linhas é muito precária estando o sinal sujeito a muitas distorções e ruído.
     Por isso as técnicas DSL são obrigadas a dividir o espectro disponível em inúmeros canais com cerca de 4 KHz de largura cada. Durante a fase de iniciação do MODEM cada um dos canais é testado para se determinar quais têm condições de funcionamento aceitáveis, os outros serão descativados. A taxa de transmissão que se pode obter depende do número de canais disponíveis.
   - ADSL (asymmetric digital subscriber line): o ADSL é uma das **técnicas** de acesso com **maior sucesso no presente**. Trata-se de uma variante DSL em que é reservado **um número de canais para circulação de dados no sentido operador para subscritor muito superior ao sentido inverso, criada com a função de cobrir as necessidades cada vez maiores de "downloads".**
   - VDSL (Very High bit digital subscriber line): o objetivo é proporcionar taxas mais elevadas, em modo simétrico ou não, usando mais largura de banda e eventualmente colocando maiores restrições quanto à distância. O **VDSL2** usa bastante mais largura de banda, até aos 30 MHz., sendo que o seu desempenho se degrada com a distância, e a 500 metros já está reduzido a metade. A disponibilidade de qualquer taxa de transmissão DSL está totalmente dependente da qualidade das linhas de transmissão, por isso a maioria dos operadores especifica a taxa máxima associada ao serviço e nunca a taxa mínima.
   - Acesso via rede CATV: as redes de televisão por cabo (CATV) **usam cabos coaxiais para transportarem sinais RF (analógicos) de televisão até aos subscritores.** Para que uma rede CATV possa ser usada como técnica de acesso **tem de ser preparada para o efeito, pois originalmente são redes preparadas para fluxo de sinais apenas no sentido do operador para o subscritor.** A largura de banda disponível numa rede CATV é enorme, começam nos 50 MHz e podem ir até a 1 GHz (950 MHz de largura de banda). Cada canal tem capacidade suficiente para ser partilhado por muitos clientes. **Tratando-se de um meio de transmissão partilhado torna-se necessário um mecanismo de controlo de acesso ao meio (acesso ao canal)**. Por se tratar de um meio partilhado (ao contrário do DSL), para garantir a privacidade, é necessário recorrer a algoritmos de criptografia.

- DOCSIS (data over cable service interface specification): as **normas DOCSIS definem como os canais podem ser usados para transportar dados,** sendo que na camada de ligação física são definidas as larguras dos canais bem como as técnicas de modulação a usar para transportar os dados. Na camada MAC do DOCSIS definem-se os mecanismos de controlo de acesso ao meio partilhado que são o TDMA ("Time division multiple access") e o S-CDMA ("Synchronous Code Division Multiple Access"). Na camada MAC é ainda definido o protocolo BPI ("Baseline Privacy Interface") que garante a confidencialidade dos dados. As taxas máximas de dados dependem da norma. A norma DOCSIS 3.0 permite que um único cliente utilize simultaneamente vários canais. Apesar destes valores, na prática os operadores impõem outros limites inferiores, estes limites são definidos num ficheiro de configuração fornecido pelo operador ao MODEM do cliente via TFTP.
- Acesso via rede elétrica: utilização de cablagens de alimentação elétrica como suporte à transmissão de dados. As **normas X10 de automatização doméstica (domótica) usam esta técnica para troca de informação entre dispositivos.** As normas HomePlug tratam da utilização das linhas de alimentação elétrica para diversos tipos de transmissão de dados (HomePlug 1.0, HomePlug AV, HomePlug BPL, HomePlug Command & Control (HPCC) – aplicações de domótica). A utilização prática do acesso BPL tem-se revelado muito complicada com resultados desanimadores, especialmente se comparados com tecnologias alternativas.

6. PILHA DE PROTOCOLOS TCP/IP

1. Transporte de dados IP: existem várias técnicas de acesso alternativas que não são mais do que ligações de dados entre o operador e o subscritor, mas para que estas ligações possam ser usadas pelo protocolo **IP** são necessários mecanismos apropriados. É necessário definir um formato apropriado para as transferências de dados, sendo os mais comuns são "tramas ETHERNET" ou células ATM (AAL5). Sobre este mecanismo de transporte de nível 2 usa-se o protocolo **PPP** que se encarrega da gestão da ligação dedicada.
2. Camada IP: a **pilha de protocolos** normalmente designada por TCP/IP exerce atualmente um domínio quase total nas comunicações por computador assegurando deste modo a interoperacionalidade direta entre quase todos os tipos de equipamentos ligados por rede. As principais características do protocolo IP são:
   - 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 de dados.
   - Fragmentação e reagrupamento.
   - Deteção de erros apenas no cabeçalho. Parâmetros QoS.
3. 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 64 Kbytes de comprimento, seguem uma organização (alinhamento) de 32 bits.



   - Fragmentação de datagramas IP: 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.

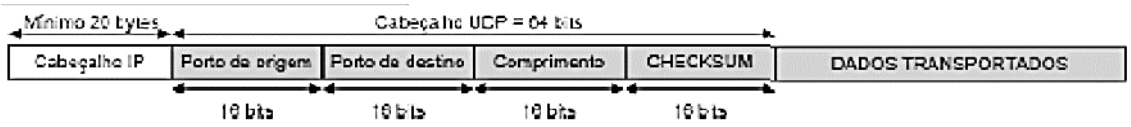
     O primeiro bit do campo "flags" serve para indicar que existem mais fragmentos (valor 1) o valor 0 indica que se trata do último fragmento. **À primeira vista a fragmentação é a solução ideal porque deste modo os valores de MTU são sempre aproveitados ao máximo (menor "overhead").** Na prática, contudo, **a fragmentação sobrecarrega bastante os nós de rede**, em especial porque os nós que recebem os fragmentos 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

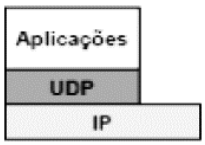
superior ao MTU seguinte ignora-o e devolve uma mensagem de erro ICMP "Destination Unreachable" como o código "fragmentation needed and DF set", indicando ainda o valor do MTU que causou o erro (RFC1191). Usando estas mensagens o nó de origem determina constantemente o PMTU associado a esse caminho.

- 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 dos 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.**

7. PROTOCOLOS ARCP; UDP; BOOTP; DHCP

1. ARP (Address resolution protocol): 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.
   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: 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.
     O protocolo ARP **é necessário para o IPv4, mas deixou de o ser para o IPv6.** No IPv4, os endereços de 32 bits não têm qualquer relação direta com os endereços físicos da rede local, normalmente endereços de 48 bits. Contudo, numa rede local, para comunicar com um dado nó é necessário conhecer o seu endereço físico. Para resolver o problema recorre-se ao protocolo ARP que **usa "broadcast" para criar tabelas de equivalência entre endereços IPv4 e endereços físicos. O IPv6,** como os seus endereços de 128 bits permite que os 48 bits do endereço físico local sejam colocados na parte de nó do endereço IPv6. **Desta forma o protocolo ARP deixa de ser necessário.**
   - Endereçamento IP: como protocolo de rede que é, **define não apenas endereços de nó, mas também endereços de redem,** sendo que um endereço IP (32 bits) é **constituído por uma N bits de rede (mascara de rede) seguidos de (32-N) bits de nó;** 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.**
2. Protocolo UDP ("User datagram protocol"): **o protocolo IP não apresenta as funcionalidades mínimas para ser usado diretamente por aplicações de rede de uso geral,** sendo também um problema a ausência de deteção de erros nos dados. **Mais grave é a ausência de um mecanismo de identificação de aplicações,** sendo que o protocolo UDP implementa estas duas funcionalidades.





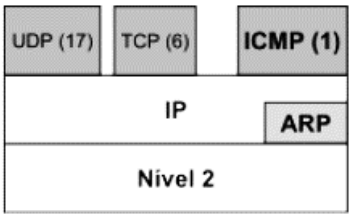
   Num protocolo de aplicação cliente-servidor implementado sobre UDP o servidor associa o seu "socket" a um número de porto local definido no protocolo. Este por sua vez é usado pelo servidor tem de ser um número pré acordado com o cliente, ou seja, tem de estar referido na definição do protocolo de aplicação. Muitas vezes estes **números de porto são também designados** por números de serviço pois permitem o acesso a um determinado serviço. Quando o cliente emite um pedido dirigido ao servidor necessita de saber qual é o número de porto onde o servidor está a receber os pedidos. Isto acontece porque no modelo cliente-servidor o cliente realiza sempre o primeiro contacto. Devido ao tipo de diálogo cliente-servidor o servidor só envia dados ao cliente em resposta a um pedido anterior, por essa razão o servidor não necessita de conhecer antecipadamente o número do porto do cliente, adquire esse conhecimento no momento em que recebe o pedido. Assim o cliente pode usar um número de porto local variável, normalmente solicita um porto livre ao sistema operativo.
3. Protocolo bootp ("BOOT strap Protocol"): é um serviço UDP, que 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

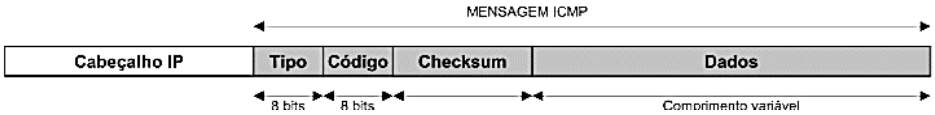
sequer o seu próprio endereço IP. **Como o cliente não sabe a que rede está ligado, não pode usar o endereço de broadcast correspondente, em vez disso usa o endereço "255.255.255.255" que significa broadcast na rede local (seja ela qual for).**

- Bootp dinâmico: apesar do protocolo BOOTP fornecer aos clientes todo o tipo de informações de configuração de que necessitam para funcionar, é, no entanto, necessário que cada endereço MAC seja registado manualmente no ficheiro de configuração. Desta forma **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.**

8. PROTOCOLO ICMP E TCP

1. Protocolo DHCP ("Dynamic host configuration protocol"): é 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.
2. Protocol ICMP (Internet Control Message Protocol): é um protocolo auxiliar de controlo que permite realizar a notificação de vários tipos de anomalias relacionadas com o protocolo IP e ainda desencadear alguns tipos de operações de manutenção.





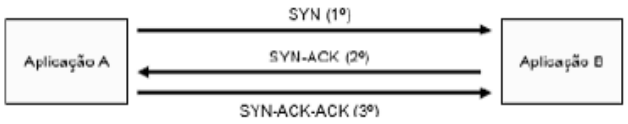
O campo "Tipo" identifica o **tipo de mensagem**, para cada tipo poderão existir vários valores possíveis para o campo "Código". O campo "Checksum" serve para **detetar erros na mensagem ICMP** e o campo "Dados" contém **elementos dependentes do tipo de mensagem ICMP.** *Exemplo: A mensagem ICMP tipo 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 devolvido na mensagem ICMP uma cópia da parte inicial do DATAGRAMA IP que causou o problema.

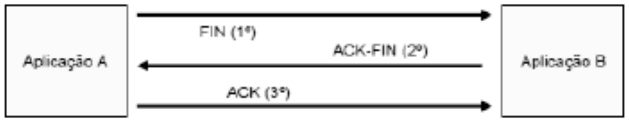
0	NETWORK UNREACHABLE – significa que o DATAGRAMA não chegou à rede de destino.
1	HOST UNREACHABLE – significa que o DATAGRAMA chegou à rede de destino, mas não pode ser entregue no nó de destino.
2	PROTOCOL UNREACHABLE – significa que o DATAGRAMA IP chegou ao nó de destino, mas esse nó não tem o protocolo 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 activa. O campo "MTU seguinte" contém o valor do MTU que causou o problema.
5	Foi usada a opção IP "source route" e falhou

- Mensagem do tipo 4 ("Source Quench"): aviso gerado por um nó saturado, pede que o fluxo de dados seja reduzido.
- Mensagem do tipo 5 ("Redirect"): **indica** ao nó de origem **que está a usar o encaminhador errado para chegar ao destino** pretendido e como tal deve corrigir o encaminhamento, utilizando para esse efeito os primeiros 32 bits da zona de dados da mensagem ICMP que contêm o endereço IP do encaminhador correto. **O nó de origem pode usar esta informação para alterar a sua tabela de encaminhamento.**
- Mensagem do 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 "trace route", gerando erros sucessivos nos vários encaminhadores fica a saber-se o caminho que os dados seguem.

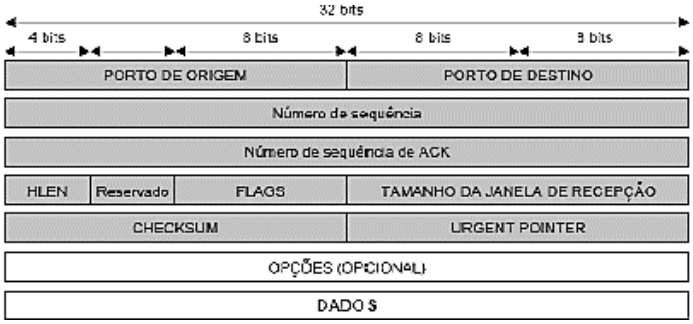
3. Protocolo TCP ("Transmission Control Protocol"): ao contrário do protocolo UDP que é simples, o **TCP é complexo** devido ao conjunto de funcionalidades disponibilizadas. Usando um simples e não fiável serviço de DATAGRAMAS do IPv4 ou IPv6, **o TCP proporciona um serviço de transferência de dados 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 efeito um dos intervenientes envia uma mensagem SYN, que terá como resposta um SYN-ACK, finalmente é enviado o SYNACK-ACK.**



   - Finalização de ligação: qualquer um 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).



   - Segmentos TCP: 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".



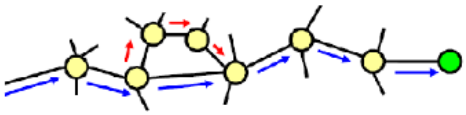
     Em cabeçalhos simples (sem opções):
     IPv4: MSS = MTU – 40
     IPv6: MSS = MTU – 60
     Os portos de origem e destino TCP são totalmente independentes dos portos UDP pois trata-se de duas camadas independentes. O campo HLEN especifica o tamanho do cabeçalho TCP em unidades de 32 bits, pode variar 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.

9. ENCAMINHAMENTO IPV4

1. Nós intermédios: os nós intermédios assumem o papel principal em qualquer tipo de rede de comutação, **uma vez que estas caraterizam-se pela retransmissão da informação entre nós sucessivos (nós intermédios) até chegar ao destino pretendido.** No nível 2 do modelo OSI são designados de comutadores ou switches. Os "datagramas" do protocolo IPv4 também chegam ao destino graças aos nós intermédios, quando operam no nível de rede os nós intermédios são normalmente designados encaminhadores ou "routers" ou por "gateway".
   - Encaminhadores ("Routers"): os encaminhadores são **nós intermédios responsáveis pela retransmissão de pacotes do nível 3.** Operam segundo um protocolo de rede, por exemplo IPv4. Possuem várias interfaces de rede (implementações nível 1 e 2, eventualmente de tipos diferentes), usando essas interfaces recebem e retransmitem "datagramas" IPv4. Quando um encaminhador recebe um "datagrama" para retransmitir analisa o endereço IPv4 de destino contido no cabeçalho. **Com base nesse endereço tem de tomar uma decisão relativamente a "para onde enviar". O encaminhador vizinho para onde o "datagrama" vai ser enviado** é conhecido como próximo nó ("next-hop").

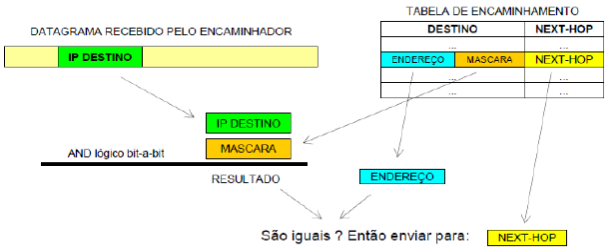
10. ENCAMINHAMENTO ESTÁTICO E ENCAMINHAMENTO DINÂMICO

1. Encaminhamento ("Routing"): a decisão que um **encaminhador tem de tomar consiste em determinar para onde enviar um dado "datagrama", ou seja determinar o "next-hop", ação designada de** encaminhamento ou "routing". Os encaminhadores interagem apenas com os encaminhadores vizinhos, esses são os únicos "**next-hop**" possíveis. Na figura ao lado o encaminhador central tem exatamente 8 encaminhadores vizinhos, ou seja 8 "next-hop" possíveis.



   - Tabela de encaminhamento ("Routing table"): o sucesso do **protocolo IP parte também do** facto de ser assegurado que um **DATAGRAMA IP** emitido em qualquer ponto da **INTERNET** será corretamente encaminhado até ao endereço de destino. As decisões de encaminhamento são feitas com base numa tabela conhecida por tabela de encaminhamento (ou "routing table"). Cada linha da tabela de encaminhamento possui dois elementos fundamentais o destino (um endereço IP associado a uma mascara de rede) e o próximo nó (próximo encaminhador ("**next-hop**") para onde devem ser enviados os dados quando se pretende que estes cheguem ao destino. Trata-se do endereço IP do nó seguinte do percurso ou rota dos dados, é sempre um endereço IP de um nó vizinho.

- Encaminhamento: **denomina-se encaminhamento quando um encaminhador recebe um "datagrama" e usa a coluna "DESTINO" da tabela de encaminhamento para determinar o "next-hop" adequado.** O endereço de destino do "datagrama" é confrontado sequencialmente com cada uma das linhas da tabela de encaminhamento até ser encontrado o DESTINO pretendido.

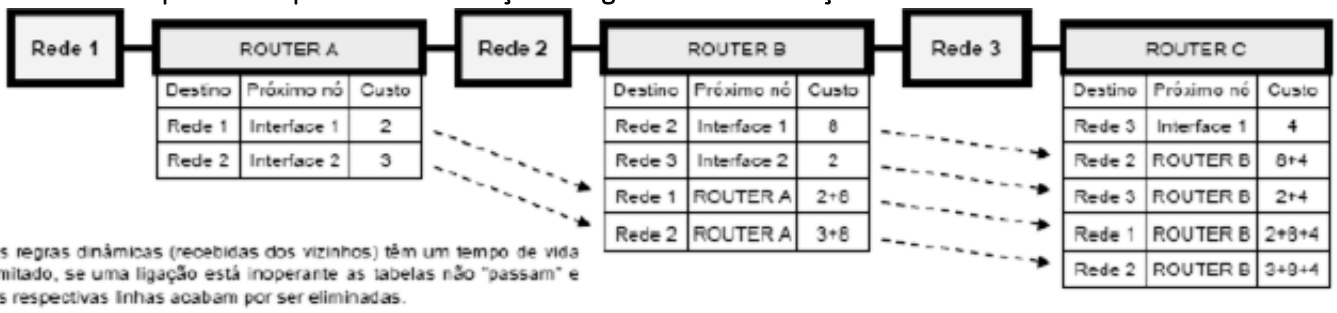


- Redes locais e caminho por omissão: **para cada interface de rede a que um nó está ligado (redes locais) existe uma linha especial na tabela de encaminhamento** em que o "PRÓXIMO NÓ" é a identificação interna dessa interface na rede local e não um endereço IP. **Numa tabela de encaminhamento não é possível ter a identificação de todos os destinos possíveis, usa-se uma linha no fim da tabela para tratar o caminho por omissão ("default route") que serve para identificar todos os destinos não contemplados nas linhas anteriores. O "next-hop" correspondente é muitas vezes conhecido por "default gateway". Todos os nós de rede IP possuem uma tabela de encaminhamento, no caso mais simples apenas com duas entradas: "interface na rede local" e "caminho por omissão".** Todos os endereços na coluna "PRÓXIMO NÓ" pertencem obrigatoriamente a redes locais diretamente ligadas por uma interface.

DESTINO	PRÓXIMO NÓ
192.168.10.0/24	INTERFACE ETHERNET 1
172.14.0.0/16	INTERFACE ETHERNET 2
194.121.12.0/24	172.14.5.100
0.0.0.0/0	192.168.10.200

- Caminhos alternativos: quando apenas existe um caminho possível para chegar ao nó de destino, as tabelas de encaminhamento com duas colunas são suficientes**, no entanto em redes mais complexas é desejável por questões de redundância e distribuição do tráfego, que existam vários caminhos alternativos para chegar a qualquer ponto.**
  Caminhos alternativos significa que **em determinados encaminhadores vão existir linhas da tabela de encaminhamento repetidas com o mesmo destino.** Para escolher o caminho a utilizar por norma deve se utilizar a lógica do menor custo.

2. Encaminhamento dinâmico: a **construção das tabelas de encaminhamento pode ser realizada manualmente** (encaminhamento estático). Em redes complexas esta tarefa tem de ser automatizada, sendo que **a existência de caminhos alternativos** (redundância e distribuição do tráfego) apenas fazem sentido se as tabelas de encaminhamento mantiverem uma representação atualizada do estado da rede, isso não pode ser feito manualmente.
   Para **resolver estes problemas** usam-se protocolos de encaminhamento dinâmico que permite a construção automática das tabelas de encaminhamento. Existem dois tipos de protocolo de encaminhamento dominantes o distance-vector (cada nó divulga junto dos nós vizinhos a sua tabela de encaminhamento) e link-state (cada nó divulga junto dos nós vizinhos a lista de nós aos quais está diretamente ligado (vizinhos)).
   - Algoritmos distance vector: envolvem a **partilha periódica entre nós vizinhos das respetivas tabelas de encaminhamento**, que normalmente se sucede através de uma emissão em broadcast. Inicialmente cada nó possui uma tabela de encaminhamento em que constam apenas as redes a que está diretamente ligado (interfaces de rede, às quais estão associadas um custo com base na distância**. Quando um nó recebe do vizinho uma tabela de encaminhamento, adiciona ao custo de cada linha o custo da interface por onde a informação foi recebida e modifica o PROXIMO NÓ para corresponder ao endereço de origem dessa informação.**



   - Algoritmos link-state: as tabelas de encaminhamento são totalmente construídas em cada encaminhador**, sendo que cada encaminhador tem de conhecer toda a rede. Os vizinhos diretos têm de ser conhecidos de forma a controlar permanentemente o estado da ligação com cada um deles. Esta lista é desta forma partilhada por todos os encaminhadores da rede e sempre que se detetar uma alteração nos estados da ligação reconstrói a lista e divulga-a novamente.** Cada nó de encaminhamento recebe SMS de todos os outros nós contendo a identificação do nó de origem e dos respetivos vizinhos (as ligações entre nós só são aceites com reconhecimento mútuo)**. Com estas informações todos os encaminhadores ficam a conhecer as interligações de toda a rede,** sendo que com conhecimento de todas as interligações da rede, o nó pode agora autonomamente determinar a tabela de encaminhamento. **Trata-se de um algoritmo simples de determinação dos caminhos mais curtos (menos nós intermédios).**

3. Protocolos de encaminhamento IGP e BGP (Sistemas autónomos AS): para efeitos de encaminhamento a internet está dividida em sistemas autónomos (AS), cada sistema **autónomo integra um conjunto de redes IP geridas por uma mesma**

entidade. *Exemplo: um ISP.* Os protocolos de encaminhamento usados no interior de um AS **são independentes da restante internet e dos AS vizinhos** e habitualmente designados de IGP (Interior Gateway Protocols). O **encaminhamento entre sistemas autónomos** é atualmente definido pelo BGP (Border Gateway Protocol). Para **interagir com o BGP,** cada AS tem atribuído um ASN ("Autonomous System Number") **único** que o **identifica**, que são **atribuídos** pelo IANA ("Internet Assigned Numbers Authority"), também responsável pela atribuição deendereçosIPv4/IPv6 e nomes de domínio DNS. **O IANA delega a administração de partes do sistema aos RIR ("Regional Internet Registries").**

- Protocolos IGP distance-vector:
  a. RIPv1 (routing information protocol): cada "router" divulga a sua tabela de encaminhamento por "broadcast" (UDP) nas redes vizinhas de 30 em 30 segundos, ou em períodos variáveis ligeiramente superiores. **Quando uma linha da tabela de encaminhamento não é refrescada durante180 segundos, é marcada como não atingível, ou seja, o número de saltos hops é igual a 16** (métrica). As mascaras de rede não são divulgadas logo apenas suporta endereçamento classful. A métrica é o número de saltos ("hops"), é incrementada por cada "router" que retransmite a tabela. O número de "hops" máximo é 15, o valor 16 significa que o destino não é atingível. Não usa nenhum identificador de área o AS, isso torna impossível a um "router" estar ligado a dois AS distintos que usem RIP. O protocolo é inseguro porque as informações são recebidas sem qualquer procedimento de autenticação.

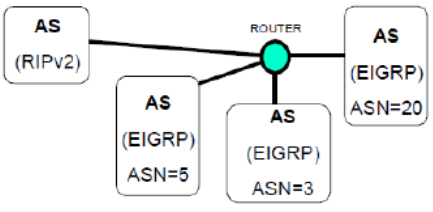
  b. RIPv2: esta versão tenta colmatar algumas limitações da versão 1 tendo sido desenvolvido de forma a ser parcialmente compatível com o RIPv1, mantendo o nº de saltos ("hops") em 15 saltos. Nesta versão, **cada "router" divulga a sua tabela de encaminhamento usando "multicast" em lugar de "broadcast".** As tabelas de encaminhamento divulgadas incluem agora as mascaras de rede, por isso já suporta CIDR ("Classless Inter-Domain Routing"). Suporta a utilização do algoritmo MD5 **permitindo a autenticação baseada numa chave secreta pré-partilhada.**

  c. Protocolos OSPF (open shortest path first): cada "router" tenta **identificar os "routers" vizinhos, recorrendo a "broadcast" e "multicast".** Depois **divulga a lista de vizinhos aos restantes "routers".** Cada "router" monitoriza o estado dos "routers" vizinhos, sempre que se produz alguma alteração repete a divulgação. As transações OSPF usam diretamente o protocolo IP, não recorrem ao UDP. Cada "router" utiliza a informação recebida para construir autonomamente a tabela de encaminhamento. A informação transacionada transporta máscaras de rede, por isso o OSPF suporta CIDR. Suporta também autenticação entre "routers", baseada em MD5 e HMAC-SHA e a métrica pode ser calculada de diversas formas, mas normalmente deriva da taxa se transmissão correspondente à ligação que dá acesso ao caminho.

  d. Áreas OSPF: o OSPF sendo um IGP, não suporta o conceito de AS, sendo que do ponto de vista deste protocolo tudo o que existe é o domínio OSPF (Sob o ponto de vista externo poderá ser um AS, mas não para o OSPF). **Pelas mesmas razões que a Internet está dividida em sistemas autónomos, também um domínio OSPF pode ser dividido em áreas onde o encaminhamento será tratado de forma independente umas das outras, contudo agora o protocolo é o mesmo em todas elas.**

  e. Protocolo EIGRP ("enhanced interior gateway routing protocol "): é um **protocolo hibrido entre distance-vector e link-state**, proprietário do cisco, desenvolvido para resolver os problemas deixados pelo RIP. **Cada "router" mantém uma lista de "routers" vizinhos usando o envio periódico, em "unicast/broadcast/multicast", da mensagem "hello".** Este protocolo trás inúmeras vantagens. *Exemplos: é mantido um controlo mais apertado sobre o estado dos "routers" vizinhos permitindo um reflexo mais rápido do estado da rede nas tabelas de encaminhamento; as tabelas de encaminhamento são divulgadas aos "routers" vizinhos apenas quando ocorre alguma alteração, não há divulgação periódica; o protocolo EIGRP suporta CIDR e usa uma métrica complexa que envolve vários parâmetros, nomeadamente taxa de transmissão da interface, saturação dos nós, atraso na rede, fiabilidade da ligação e MTU; embora não seja usado para efeitos de métrica, também procede à contagem de "hops", normalmente um número de "hops" superior a 100 classifica o destino como não atingível; o número máximo de "hops" pode ser ajustado até 224.*

11. SISTEMAS AUTÓNOMOS E REDISTRIBUIÇÃO DE ROTAS



1. Sistemas autónomos EIGRP: o protocolo EIGRP permite a criação de sistemas autónomos de encaminhamento, associando a cada um deles um número único (Autonomous System Number) que pode **variar de 1 a 65535.** Todas as informações do EIGRP têm um ASN associado, sendo tratadas de forma totalmente independente informações relativas a sistemas autónomos diferentes. A vantagem do EIGRP usar os ASN é que um encaminhador pode estar ligado a vários sistemas autónomos EIGRP diferentes. Os ASN do EIGRP não têm nenhuma relação direta com os ASN do BGP.
2. Encaminhamento entre sistemas autónomos: o objetivo da definição de sistemas autónomos (ou áreas OSPF) **é isolar partes das redes sob o ponto de vista da gestão das tabelas de encaminhamento.** Este processo trás algumas vantagens tais como

administração mais simples uma vez que existem menos redes; tabelas de encaminhamento mais pequenas; menor tráfego de rede (propagação do protocolo limitada). **Os sistemas autónomos não são criados arbitrariamente, devem corresponder a partes isoladas das redes, muitas vezes com apenas com uma ligação aos restantes sistemas autónomos.** Para facilitar o encaminhamento no exterior do sistema autónomo é desejável que as redes IP de um sistema autónomo constituam um bloco CIDR único. **O encaminhamento entre sistemas autónomos pode ser conseguido pela inserção manual de regras de encaminhamento nas tabelas no interior de cada AS.** Estas regras estáticas podem depois ser propagadas a todos os "routers" do AS pelo protocolo de encaminhamento usado no seu interior.

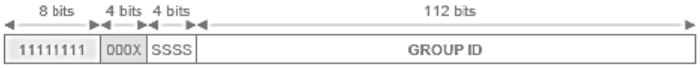
3. Redistribuição de rotas: para **garantir o encaminhamento entre sistemas autónomos podem ser configuradas regras estáticas, manualmente, nos "routers"** de fronteira, que indiquem como chegar às redes de cada um dos sistemas autónomos, de preferência sob a forma de blocos CIDR únicos. Outra alternativa é a "redistribuição de rotas" ou "route map" que consiste em definir formas automáticas de copiar regras de encaminhamento entre dois AS vizinhos, operação realizada no "router" de fronteira que faz parte de ambos os AS. Uma das dificuldades na "redistribuição de rotas" coloca-se quando os AS vizinhos utilizam protocolos de encaminhamento diferentes. Devido à variedade dos tipos de métrica usado por cada um torna-se necessário muitas vezes arbitrar alguns valores.

12. IIPv6 E ICMPv6

1. Internet protocol v6: a expansão da internet levou a um esgotamento dos endereços IPv4 disponíveis, desta forma foram tomadas medidas para permitir um aproveitamento melhor do espaço de endereçamento de 32 bits: definição de 3 classes de rede com 8, 16 ou 24 bits (na versão inicial as redes IPv4 usavam sempre 8 bits para identificar a rede); definição livre de outras máscaras de rede mais ajustadas às realidades de cada situação concreta (por exemplo mascaras de 30 bits para ligações dedicadas); utilização de redes privadas associado a dispositivos capazes de traduzir endereços (NAT – Network Address Translation). Cada conjunto de redes privadas necessita apenas de um endereço oficial.
   Tendo como principal objetivo resolver as limitações do espaço de endereçamento de 32 bits foi desenvolvido um sucessor do IP versão 4, inicialmente conhecido por IP-NG ("Next Generation") foi-lhe atribuído o número de versão 6, sendo atualmente conhecido por IPv6.
   - Endereços de 128 bits: os endereços IPv6 são **constituídos por 4 vezes mais bits do que os IPv4**, 2^{128} no IPv6 contra 2^{32} no IPv4, desta forma o espaço de endereçamento do IPv6 é "imenso", permitindo quase um espaço de endereçamento IPv4 para cada habitante da Terra. Alguns aspetos do protocolo IP foram melhorados na versão 6 (IPv6), tais como: **maior espaço de endereçamento**; **abandono da fragmentação**, apenas "Path MTU discovery" (PMTUD); **integração do endereço físico (MAC) no endereço IPv6 (ARP desnecessário)**; suporte **de MULTICAST** (no IPv4 é uma opção); configuração automática de nós, evitando a necessidade **do BOOTP/DHCP**; suporte JUMBO GRAMS – pacotes IP até 4 Gb (o IPv4 apenas suporta 64Kb); IPSEC – protocolo de autenticação e confidencialidade integrado.
   - Representação de endereços IPv6: são representados sob a forma de texto através de uma **sequência de 8 conjuntos de 16 bits em notação hexadecimal, separados por ":".** *Exemplo: XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX*
     A grande extensão dos endereços torna a sua representação pouco cómoda e pouco legível, sendo que no sentido de facilitar a sua representação e leitura, em cada conjunto de 16 bits eliminam-se os zeros à esquerda, além disso os conjuntos de 16 bits com valor nulo são omitidos. *Exemplo: "0000:3278:0A04:0005:0000:0000:0000:0034" = ":3278:A04:5::34".* **Nunca pode existir mais do que uma sequência "::" pois representa um número indeterminado de conjuntos nulos.**
     Os endereços IPv4 podem ainda ser representados na forma IPv6, isso é útil quando há necessidade de encaminhar pacotes IPv6 através de uma rede IPv4 ("IPv4 compatible"), ou quando pretendemos representar um nó que não possui IPv6 ("IPv4 mapped"). Nestes casos o endereço IPv4 ocupa os dois últimos conjuntos e pode ser representado na notação IPv4. *Exemplo: ::193.136.62.9 e ::FFFF:193.136.62.9*
   - Tipos de endereço: **o espaço de endereçamento é estruturado em redes**, sendo a parte inicial do endereço usada para identificar a rede (prefixo de rede) e a parte restante identifica um nó nessa rede.
     Exemplo: 2001:0DB8:2B00::/40 (representa uma rede com mascara de 40 bits)
     1º nó da rede: 2001:0DB8:2B00::1 (2001:0DB8:2B00:0000:0000:0000:0000:0001)
     Último nó da rede: 2001:0DB8:2BFF:FFFF: FFFF: FFFF: FFFF: FFFF
     (o IPv6 não usa endereços de BROADCAST, para o mesmo efeito existem endereços MULTICAST)
     O protocolo IPv6 suporta 3 tipos de endereço:
     a. UNICAST (identifica um nó único numa dada rede);

     b. MULTICAST (identificam conjuntos de nós, os dados têm de ser entregues em todos eles):

Os endereços MULTICAST identificam-se por terem os primeiros 8 bits com o valor um.



O bit X tem o valor zero para endereços MULTICAST normalizado ("well-known") e o valor um para outros grupos de nós.

Os 4 bits seguintes (SSSS) definem a zona limite ("SCOPE") até onde o MULTICAST pode ser aplicado:

1 – "Node-Local" – Apenas no nó emissor
2 – "Link-Local" – Na mesma rede física (nível 2)
5 – "Site-Local"
8 – "Organization-Local"
E – "Global" – Toda a INTERNET

Os endereços MULTICAST "well-known" servem por exemplo para identificar serviços, por exemplo:

"FF02:0:0:0:0:0:0:C" identifica todos os servidores DHCPv6 da rede local (SCOPE=2).
"FF02::1" identifica todos os nós da rede local (equivalente ao BROADCAST do IPv4).
"FF02::2" identifica todos os ROUTERS da rede local.
"FF01::43" identifica os servidores NTP existentes no mesmo nó (SCOPE=1).
"FF0E::43" identifica todos os servidores NTP da INTERNET (SCOPE=E).

c. ANYCAST (identificam conjuntos de nós, os dados são entregues em apenas um deles): os endereços ANYCAST são endereços UNICAST normais, a única diferença é que **são atribuídos a vários nós da rede**. Pode ser útil de diversas formas, quando um encaminhador recebe um pacote destinado a um endereço ANYCAST determina qual é o nó que está mais próximo dentro do conjunto de nós que possui esse endereço ANYCAST. O endereço "subnet-router any cast address" é um exemplo deste tipo de endereço, é constituído pelo prefixo de uma dada rede IPv6, seguido de zeros, serve para identificar um dos encaminhadores dessa rede.

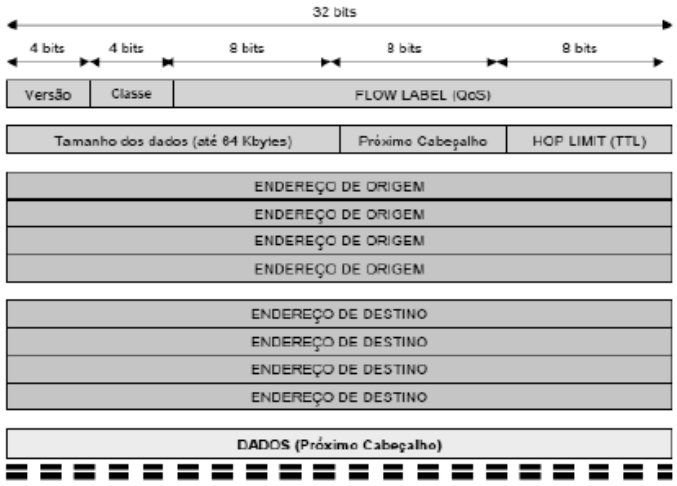
Exemplos: Os endereços UNICAST estão divididos em várias gamas de acordo com os valores dos primeiros bits:

010… – Endereços UNICAST associados a fornecedores de serviço.
100… - Endereços UNICAST associados a zonas geográficas
11111110 10… - Endereços privados UNICAST para uso local (LINK), sem encaminhamento.
11111110 11… - Endereços privados UNICAST para uso local (SITE) dentro da organização.

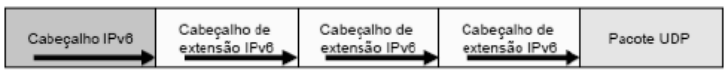
Alguns endereços especiais são:

:: (0:0:0:0:0:0:0:0), tal como no IPv4, zero representa um endereço desconhecido.
::1 (0:0:0:0:0:0:0:1) endereço de LOOPBACK (equivalente ao 127.0.0.1 do IPv4).

- Pacotes IPv6: o cabeçalho IPv6 denota várias simplificações, **incluindo a eliminação do CHECKSUM**, ficando com um comprimento fixo de 40 bytes, dos quais 32 são ocupados com os endereços de origem e destino. Alguns campos mudam de nome, mas mantêm em grande medida a funcionalidade anterior, por exemplo o identificador de protocolo tem agora a designação "Próximo Cabeçalho". O 2º campo ("Classe de Tráfego") serve **para definir a prioridade e tipo de tratamento que o pacote pode ter por parte da rede (ROUTERS),** por exemplo se, se trata de uma aplicação interativa ou se a recuperação de dados é irrelevante. Está associado ao 3º campo de 24 bits que é usado como identificador de um dado fluxo de dados com determinadas características QoS negociadas.



- Cabeçalhos de extensão: apesar de um tamanho de cabeçalho fixo (40 bytes) o IPv6 também suporta opções que envolvem a existência de mais informação de controlo. O campo "NEXT HEADER" do cabeçalho IPv6 é usado **para identificar o protocolo a que pertencem os dados transportados (multiplexagem, com os mesmos identificadores que eram usados no campo PROTOCOL do IPv4).** O IPv6 permite, contudo, que o "NEXT HEADER" **seja um bloco de opções IPv6** designado de "Cabeçalho de extensão". Os cabeçalhos de extensão começam pelos campos "NEXT HEADER" e "LENGTH", tornando-se possível a existência de uma sucessão de cabeçalhos.



Valores especiais do campo "NEXT HEADER":

0	HOP-BY-HOP OPTIONS – opções que necessitam de ser processadas nos nós intermédios, EX.: JUMBOPAYLOAD
43	ROUTING – opções de encaminhamento (Ex.: SOURCE-ROUTING)
44	FRAGMENTAÇÃO – no IPv6 a fragmentação em nós intermédios não é suportada, apenas entre nós finais
50	ENCAPSULAMENTO – TUNNELING – CONFIDENCIALIDADE - INTEGRIDADE
51	AUTENTICAÇÃO - CONFIDENCIALIDADE- INTEGRIDADE
60	DESTINATION OPTIONS – opções que apenas necessitam de ser verificadas no nó final de destino.

- ICMPv6 (ICMP para IPv6): embora muito semelhante ao protocolo ICMP usado com o IPv4 (ICMPv4) foi necessário realizar algumas adaptações e surgiu assim o **ICMPv6 com identificador de protocolo 58.** O formato das mensagens ICMPv6 é igual ao das mensagens ICMPv4, ou seja: **TIPO (8 bits) + CÓDIGO (8 bits) + CHECKSUM (16 bits) + DADOS (comprimento variável) As diferenças estão nos tipos de mensagens:**

TIPO ICMPv6	
1	Destino inatingível – o pacote não chegou ao destino, o código indica a razão (existem códigos diferentes do ICMPv4).
2	Pacote demasiado grande – pacote não cabe no MTU seguinte, usado para PATH MTU DISCOVERY
3	Tempo excedido (TTL) – idêntico ao IPv4 (Código 0 = TTL esgotado; Código 1 = Reagrupamento falhou)
4	Erro no cabeçalho IP, é indicada a posição do erro no cabeçalho do pacote IP original.
128/129	Respectivamente pedido e ECHO e resposta de ECHO. Implementação igual à do ICMPv4.
130	"GROUP MEMBERSHIP QUERY" – enviada aos ROUTERS para obter informação sobre grupos locais MULTICAST.
131	"GROUP MEMBERSHIP REPORT" – enviada pelos ROUTERS em resposta aos "GROUP MEMBERSHIP QUERY".
132	"GROUP MEMBERSHIP REDUCTION" – enviada aos ROUTERS quando um nó pretende sair de um grupo MULTICAST.
133	Pedido de ROUTER - enviada para o endereço MULTICAST ALL-ROUTERS para obter uma lista de ROUTERS.
134	Anúncio de ROUTER - enviada pelos ROUTERS em resposta ao pedido anterior.
135	Pedido de vizinho – usado para obter um endereço físico de um nó vizinho (equivalente a um pedido ARP). Normalmente desnecessário.
136	Anúncio de vizinho – resposta ao pedido anterior (equivalente a uma resposta ARP).
137	REDIRECT – enviada pelo 1º ROUTER quando existe um caminho mais directo ou o nó de destino é vizinho.

13. RESOLUÇÃO DE NOMES DNS E WINS/ NetBios

1. Resolução de nomes: a manipulação de endereços de nó pelos utilizadores e administradores não é cómoda, mas sob o ponto de vista da rede é o único elemento aceitável para identificar um nó sem ambiguidades, **desta forma o objetivo é estabelecer uma ligação entre os utilizadores e os endereços de rede de tal forma que os primeiros não tenham de interagir diretamente com os segundos.** Apoiados neste serviço os utilizadores podem usar nomes de máquinas bastante mais representativos para o ser humano.
2. NetBIOS:
   - Resolução de nomes: um sistema de resolução de nomes bem-sucedido é o NetBios, este integrado em redes PEER-TO-PEER, em que não existem servidores, envolve o envio em BROADCAST do nome a resolver ("NAME QUERY"), desta forma o detentor desse nome vai ter oportunidade de responder e revelar o seu endereço de nó.
   - Registo de nomes: concebido para redes sem servidores, no NetBIOS cada nó tem a missão de responder aos pedidos de rede relativos ao seu nome, havendo na mesma a necessidade ter algumas garantias de que os nomes são únicos**, para esse efeito quando os nós arrancam enviam para a rede em BRODCAST vários pedidos de registo com o nome pretendido, durante este processo, qualquer nó que já esteja a usar o mesmo nome deve responder com uma mensagem de erro.** A ausência de qualquer resposta aos sucessivos pedidos de registo ("NAME REGISTRATION"), significa que o registo foi bem-sucedido. Um nó NetBIOS, antes de ser desligado anuncia à rede a libertação do nome que estava a usar ("NAME RELEASE").
     - Tipos de nomes (WINDOWS):os **nomes NetBIOS podem ser únicos ou de grupo,** sendo que os nomes de grupo são registados da mesma forma que os nomes únicos, mas podem estar registados em vários nós em simultâneo. **Trata-se de um mecanismo simples de MULTICAST que pode ser usado para definir estruturas lógicas de grupos de nós na rede, por exemplo as redes Microsoft usam nomes de grupo para implementar os conceitos de WORKGROUP e DOMAIN.** Na norma original um nome NetBIOS pode ter até 16 caracteres, na implementação mais divulgada (Redes Windows) o 16º serve para identificar o tipo de nome. Habitualmente o tipo de nome é representado em notação hexadecimal separado por um "#" do nome. Por exemplo "SERVIDOR1#20" representa o nome "SERVIDOR1" do tipo 20 (hexadecimal). Alguns dos tipos de nomes importantes para as redes Windows são:

00	Nome único de uso geral
01	Associado ao nome especial "__MSBROWSE__" que identifica o colector local de listas de nomes.
03	Nome de utilizador (clientes Windows antigos) e nome de nó/servidor.
1B	Associado ao nome do domínio identifica o PDC.
1C	Associado ao nome do domínio identifica um servidor de LOGIN no domínio.
1D	Associado ao nome do domínio identifica o representante local do domínio.
1E	Nome de domínio ou grupo de trabalho (WORKGROUP), todos os membros possuem este nome.
20	Nome de servidor (File Server)

Cada nó de rede NetBIOS contém vários nomes, por exemplo:

MYSERVER#00
MYSERVER#03
MYSERVER#20
__MSBROWSE__#01
MYDOMAIN#1B
MYDOMAIN#1C
MYDOMAIN#1D
MYDOMAIN#1E
HST222#00
HST222#03
HST222#20

3. Wins ("Windows Internet Name Service"): baseado em BROADCAST apresenta vários **problemas** tais como a insegurança e falta de fiabilidade (registo por omissão; ambiente de confiança geral entre nós), a limitação de propagação (sistema limitado a uma única rede) e o elevado tráfego (penalizador para o desempenho das redes pois anula a segmentação de nível 2). Todos os pedidos são enviados em UNICAST para o servidor WINS, o que significa que este pode encontrar-se

numa rede remota e servir clientes de várias redes em simultâneo. Além desta vantagem a transição para um modelo cliente/servidor leva a um aumento geral da segurança e fiabilidade, sendo que agora todos os pedidos têm resposta e os registos de nomes estão associados a um tempo de vida (esgotado esse tempo são eliminados).

- Futuro da resolução de nomes NetBIOS: a utilização de servidores WINS em substituição do BROADCAST veio resolver quase todos os problemas e limitações que o sistema tinha anteriormente. Os servidores WINS podem ser interligados de várias formas (usando protocolos proprietários), por exemplo numa perspetiva de replicação ou de consulta/registo remoto. Apesar de eficiente num ambiente limitado, **a sua aplicação em larga escala é problemática por se tratar de uma estrutura de nome totalmente rasa,** para suportar um milhão de máquinas será necessário gerir uma base de dados com um milhão de registos distribuídas por uma grande área (sem nunca haver registos repetidos). Tal como a pilha de protocolos TCP/IP fez desaparecer outras implementações, também é previsível que o sistema **DNS** acabe por substituir totalmente o **NetBIOS**. Isso já é possível nos sistemas que usam "Active Directory". De momento verifica-se a coexistência **WINS/DNS**, alguns servidores WINS podem ser até configurados para recorrer a DNS quando não conseguem resolver um nome. Os clientes Windows também combinam o recurso a **NetBIOS** em **BROADCAST**, **WINS** e **DNS**, criando por vezes alguma confusão.

4. DNS ("Domain Name System"): é estruturado em árvore de tal forma que cada ramo é administrativamente independente dos outros ramos. A independência entre ramos existe porque **cada nome apenas tem significado no ramo em que é definido, para identificar globalmente um nome é necessário especificar não apenas o nome, mas também o ramo.** Os ramos desta estrutura são conhecidos por nomes de domínios.
   - Servidores DNS: a estrutura lógica do DNS usa como plataforma uma rede de servidores de nomes, sendo que os vários servidores de nomes (NS) comunicam entre si de tal forma que **cada um deles é capaz de resolver qualquer nome qualificado de qualquer domínio.** Para um cliente poder resolver qualquer nome da INTERNET basta-lhe conhecer o endereço de um servidor de nomes. Cada NS contém uma base de dados com todos os registos, mas apenas dos domínios que serve (normalmente apenas um domínio). Estes servidores têm autoridade sobre o domínio ("authoritative DNS servers"). Isso permite-lhe responder diretamente a pedidos referentes a esse domínio. **Para responder relativamente a toda a INTERNET o servidor de nomes tem de recorrer a outros servidores de nomes.**
   - Redes de NS: para que o conjunto de NS de todos os domínios permita o funcionamento em conjunto **é necessário que cada servidor de nomes contenha registos (nomes) do domínio que serve (ou cópia obtida do servidor principal), o endereço dos servidores de nomes da raiz (acima dos domínios de topo) e o endereço dos servidores de nomes de cada subdomínio.**
     Todos os outros domínios e servidores de nomes são desconhecidos, mesmo assim é possível resolver qualquer nome. **A resolução é um processo descendente que começa num servidor de nomes da raiz.** Uma vez que cada servidor de nomes é obrigado a conhecer os servidores de nomes do domínio imediatamente abaixo, este processo conduz sempre ao servidor de nomes correto.

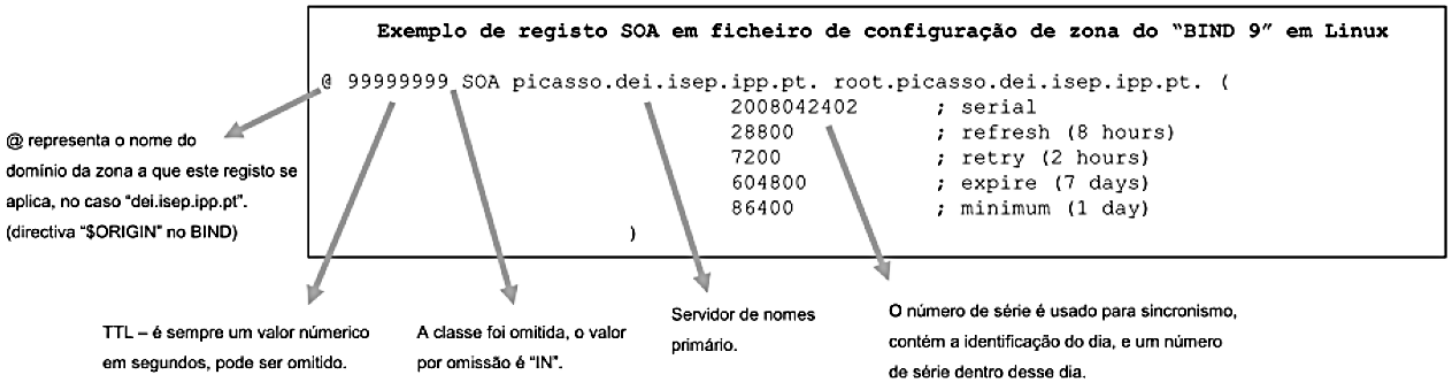
   - Resolução de nomes DNS: a resolução de nomes funciona em sentido descendente desde os domínios de topo. **O NS local contacta um NS de raiz e pede-lhe um servidor de nomes de "pt", o "ROOT NS" devolve-lhe o nome de um servidor de nomes do domínio "pt**". O processo repete-se até chegar ao último elemento do nome. **Quando se pede um NS de um domínio é fornecido um nome e não um endereço.** Se o nome do NS pertence ao domínio, então ocorre um bloqueio do sistema devido a uma dependência circular. Para resolver este problema adiciona-se ao domínio acima um registo com o endereço IP do NS. Este registo é conhecido por "glue record".
     Os servidores de nomes têm de manter em cache as respostas que vão obtendo, sendo que para o efeito cada resposta tem associado um tempo de vida (TTL). **O tempo de vida é definido pelo administrador de cada domínio e pode ter valores mais ou menos elevados. O caching das respostas reduz de forma muito significativa o recurso direto aos servidores de nomes de topo, nomeadamente os de raiz.** Quanto maior for o TTL estipulado pelo administrador menor será a quantidade de pedidos que s respetivos servidores vão receber.

5. Registos DNS (Resource Records): a base de dados de um servidor DNS é constituída por registos de diferentes tipos com diversas finalidades (Resource Records). **Cada registo (RR) tem os seguintes elementos:**

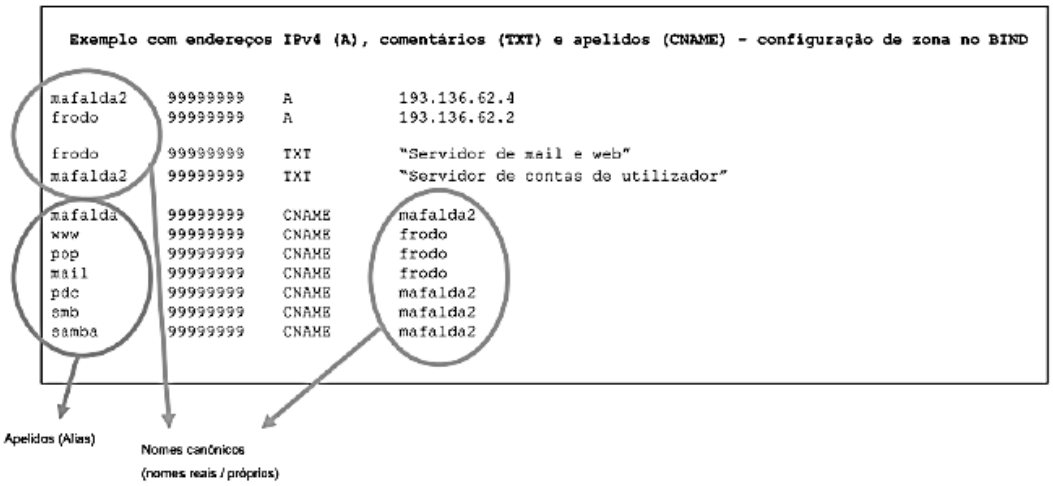
NOME (até 255 caracteres)	- Nome da entidade a quem se aplica o registo (proprietário do registo), terminado com ZERO.
TIPO	- Número de 16 bits que identifica o tipo de registo (RR TYPE)
CLASSE	- Número de 16 bits que identifica a classe (RR CLASS), para no IP apenas se usa o valor 1 (classe IN)
TTL	- Número de 32 bits que o tempo de vida em segundos do registo
RDLENGTH	- Número de 16 bits que define o tamanho do campo de dados em octetos
DADOS (comprimento variável)	- Dados que constituem o valor do registo (RDATA)

Os registos DNS (RR) são armazenados pelos servidores de nomes e são fornecidos aos clientes e outros servidores quando solicitados através de pedidos através da rede (DNS QUERY). Os servidores de nomes DNS atendem os pedidos devidamente formatados, no porto 53, normalmente as mensagens de pedido são encapsuladas em DATAGRAMAS UDP. Cada pedido contém uma ou mais perguntas segundo o formato NOME (até 255 carateres) – TIPO – CLASSE. No campo "TIPO", além dos valores de tipo de registo (RR TYPE), podem ser usados alguns valores especiais: o valor 255 ("*") representa "qualquer tipo" e o valor 252 ("AXFR") representa todos os RR do domínio, os pedidos AXFR são usados para sincronizar os vários servidores de nomes de um domínio, devido ao volume de registos envolvidos, neste caso recorresse a uma ligação TCP para o mesmo número de porto usado em UDP.

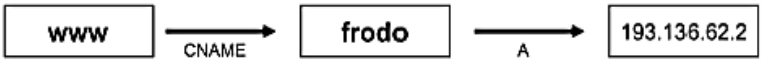
TIPO (RR TYPE)	Nome do Tipo - Objectivo	NOME (Proprietário)	DADOS
1	A - Endereço IPv4 correspondente um nome de nó	Nome de nó	Endereço IPv4
2	NS - Servidor de nomes	Nome de domínio (sub-domínio)	Nome do NS (qualificado)
5	CNAME - Nome alternativo (alias)	Nome alternativo ou apelido (alias)	Nome oficial (Nome de nó)
6	SOA (Start Of Authority) define parâmetros do dom.	Nome do domínio	Diversos, incluindo nº de série da base de dados, ...
12	PTR – nome correspondente a um endereço	Endereço (nome em IN-ADDR.ARPA.)	Nome de nó (qualificado)
15	MX – define um "mailhub" do domínio	Nome do domínio	Prioridade+Nome do servidor
16	TXT – define um comentário	Nome de domínio	Texto livre (comentário)
28	AAAA - Endereço IPv6 correspondente um nome de nó	Nome de nó	Endereço IPv6
29	LOC – define a localização geográfica do domínio	Nome de domínio	Coordenadas geográficas
33	SRV – define um serviço de rede	_serviço._protocolo.Nome de domínio	Prioridade+Peso+Porto+Nome do servidor
99	SPF – "Sender Policy Framework"	Nome de domínio	Restrições ao envio de mail em nome do domínio



- Nomes e apelidos:



Uma vez que este exemplo se encontra associado à definição da zona "dei.isep.ipp.pt", entre outros, o nome "www.dei.isep.ipp.pt" vai ser resolvido para o endereço "193.136.62.2".



- NS e Glue records: os registos NS são fundamentais no sistema DNS, são eles que garantem a ligação descendente entre os domínios, sendo que para manter a ligação, cada domínio tem de conhecer os servidores de nomes dos seus subdomínios. Os glue records (destacados a verde na imagem seguinte) são registos de endereço (A) que não pertencem ao domínio, mas são necessários para obter os endereços dos servidores de nomes. Sem o "glue record" a última resolução não seria possível, pois quem a deveria realizar seriam os servidores de nomes do domínio "dei.isep.ipp.pt".

Exemplo com sub-domínio (dei.isep.ipp.pt) - configuração de zona (isep.ipp.pt) no BIND

```
$ORIGIN isep.ipp.pt
@    IN SOA nsrv1.isep.ipp.pt admin.isep.ipp.pt 2007120500 2h 15M 3W12h 2h20M

@           IN          NS          nsrv1.isep.ipp.pt
@           IN          NS          nsrv2.isep.ipp.pt

nsrv1       IN          A           193.136.6.40
nsrv2       IN          A           193.136.6.47


dei.isep.ipp.pt         IN          NS          picasso.dei.isep.ipp.pt
dei.isep.ipp.pt         IN          NS          slave.dei.isep.ipp.pt

picasso.dei.isep.ipp.pt IN          A           193.136.62.3
slave.dei.isep.ipp.pt   IN          A           193.136.62.110
```

Os registos NS definem os nomes qualificados dos servidores de nomes

- PTR E domínio IN-ADDR.ARPA: o domínio especial "IN-ADDR.ARPA" contém um registo da estrutura de endereços IPv4, sendo que cada nível de subdomínio corresponde a um octeto do endereço IPv4 e o octeto da esquerda corresponde ao subdomínio superior. Os nomes "IN-ADDR.ARPA" servem para resolução inversa, ou seja, obter o nome DNS usando o endereço como ponto de partida. Os registos PTR permitem definir estes nomes correspondentes a endereços.
- Correio eletrónico: os domínios DNS são usados pelo correio eletrónico da internet para identificar destinatários no domínio (DESTINATÁRIO@NOME-DO-DOMÍNIO). Para entregar as mensagens de correio ao domínio é necessário identificar e contactar um dos seus servidores de correio SMTP ("Simple Mail Transfer Protocol").
- MX (MaileXchanger): os registos MX servem para identificar de forma mais eficiente os servidores de correio de um domínio, relativamente à alternativa anterior têm a vantagem de permitir definir vários servidores com diferentes níveis de preferência e além disso podem ser implementados sem recurso ao domínio superior. Um domínio pode ter vários registos MX, cada um definindo o nome do servidor (tem de ser um nome canónico, não pode ser um apelido) e o nível de preferência de 16 bits (números inferiores significam preferência mais elevada).
- Registos SRV: os registos SRV (RFC 7282) têm como objetivo permitir aos clientes identificar num domínio servidores de determinado tipo (serviços). Como consequência estes registos servem também para os servidores divulgarem os seus serviços (Ex.: Active Directory). Os registos SRV são associados a nomes simbólicos que representam tipos de serviço no contexto de um domínio, na forma:

  _{Serviço}._{Protocolo}.{Nome-do-domínio}

  O caractere sublinhado serve para evitar conflitos com nomes de domínio "normais".
  {Serviço} é um identificador de serviço normalizado.
  {Protocolo} identifica a plataforma de transporte a usar ("udp" ou "tcp").

  O registo SRV propriamente dito contém um nível prioridade semelhante ao dos registos MX, e um nível de peso que se aplica entre registos com o mesmo nome e mesma prioridade. Segue-se o número de porto usado pelo serviço e o nome canónico do servidor.
- Sender Policy Framework (SPF): o documento RFC 4408 (Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1) define formas de os domínios divulgarem as suas políticas relativamente a quem pode emitir mensagens de correio eletrónico em nome de utilizadores desse domínio. Quando um servidor SMTP recebe uma mensagem, deve verificar no domínio do remetente (campo From) se quem está a tentar enviar tem autorização para tal. O registo DNS SPF é um registo de texto associado ao nome do domínio, uma vez que este tipo de registo é bastante recente, também pode ser implementado através de um registo TXT, como nem todos os clientes e servidores DNS suportam os registos SPF, é aconselhável definir os dois.
- Registo LOC:

  O registo **LOC** serve para definir a localização geográfica do domínio, como tal é normalmente associado ao nome de domínio.

  O registo **LOC** contém:
  - Latitude em graus, minutos e segundos
  - Longitude em graus, minutos e segundos
  - Altitude em metros.
  - Tamanho em metros (diâmetro da esfera que contém o local)
  - Precisão horizontal e precisão vertical

Exemplo de registo LOC - configuração de zona no BIND

```
dei.isep.ipp.pt         IN  LOC   41 10 39.782 N 8 36 28.578 W 50.00m 100m 10m 10m
```

O registo define as características geográficas do domínio "dei.isep.ipp.pt" como sendo:
latitude = 41º 10' 39,782" Norte; longitude = 8º 36' 28,578" Oeste; altitude = 50 metros; dimensão = 100 metros; precisão horizontal = 10 metros e precisão vertical = 10 metros;

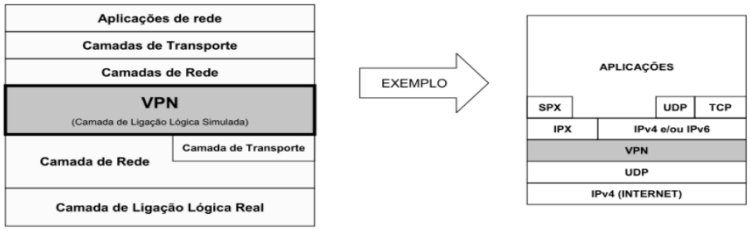
Exemplos de interrogação de registos LOC

```
-bash-3.00$ host -t LOC yahoo.com
yahoo.com location 37 23 30.900 N 121 59 19.000 W 7.00m 100m 100m 2m
-bash-3.00$ host -t LOC ckdhr.com
ckdhr.com location 42 21 43.528 N 71 5 6.284 W -25.00m 1m 3000m 10m
-bash-3.00$ host -t LOC dei.isep.ipp.pt
dei.isep.ipp.pt location 41 10 39.782 N 8 36 28.578 W 50.00m 100m 100m 10m
```

6. DDNS (DNS dinâmico): as bases de dados DNS foram concebidas para serem raramente alteradas, as alterações são feitas manualmente, os valores TTL normalmente usados atestam isso mesmo, existem, contudo, **situações em que o registo automático do nome pelo próprio cliente seria desejável, nomeadamente quando o cliente não possui um endereço fixo.** O carácter dinâmico dos registos torna a administração muito mais simples. *Exemplo: com os registos SRV, os atuais servidores Windows, nomeadamente os controladores de domínio anunciam-se no servidor DNS através de registos SRV.* O documento RFC 2136 adiciona ao sistema de mensagens DNS (RFC 1035) as funcionalidades necessárias para atualizações dinâmicas de registos DNS. É usado pelo comando "nsupdate". Os processos de alteração da base de dados estão protegidos por mecanismos de segurança tais como o "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)". Os servidores DNS também podem ser utilizados por outras vias paralelas, alteram os ficheiros de configuração DNS e obrigam o servidor DNS a reler os mesmos, por exemplo o comando "ddclient" funciona deste modo usando pedidos HTTP.

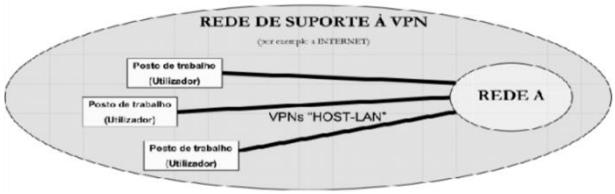
14. REDES PRIVADAS VIRTUAIS VPN

1. Virtual Private Network (VPN): uma VPN é uma **infraestrutura de comunicação de nível 2 (camada de ligação lógica) que é simulada sobre uma outra rede, tipicamente uma infraestrutura de nível 3 (camada de rede).**
   A designação VIRTUAL tem origem no facto de se tratar de uma infraestrutura simulada (não real), normalmente uma ligação "ponto-a-ponto"; PRIVADA advém do facto de serem usados mecanismo de seguranças que garantem a confidencialidade dos dados que circulam na VPN.



2. VPN LAN-LAN (site- to-site VPN): destinam-se a interligar redes, sendo implementadas pelos administradores das redes como outro qualquer tipo de ligação entre duas redes. **Os utilizadores das redes usufruem destas ligações VPN sem necessidade de conhecerem a sua existência.**
   Dadas as suas características, devem ter um carácter permanente, podendo ser estabelecidas e recuperadas automaticamente sem intervenção manual.

3. VPN HOST-LAN ("Remote access VPN"): servem para ligar nós individuais a uma rede remota, sendo que tipicamente é **criada por iniciativa do utilizador de um posto de trabalho, recorrendo a uma aplicação cliente instalado no posto local que comunica com um servidor na rede remota.**
   Neste tipo de rede, o controlo de acesso baseia-se na autenticação do utilizador, após esse processo o posto de trabalho recebe um endereço de rede pertencente à rede remota que lhe permite operar como se estivesse diretamente ligado a essa rede.



4. Interligação de redes VPN nível 2: a interligação por VPN no nível 2 **consiste na retransmissão de tramas de nível 2 através da VPN que se comporta então como uma ponte ("bridge").** Todos os dados que circulam em tramas numa das redes propagam-se até à rede remota, independentemente dos protocolos em causa. As duas redes interligadas são obrigatoriamente do mesmo tipo, caso contrário os formatos de trama seriam diferentes. Sob o ponto de vista das camadas superiores, as duas redes passam a ser apenas uma única. **O tráfego de "broadcast" de nível 2 propaga-se através da VPN, permitindo o funcionamento de protocolos tais como o ARP.**
5. Interligação de redes VPN nível 3: a interligação por VPN no nível **3 consiste na retransmissão de pacotes de rede através da VPN que se comporta então como um encaminhador ("router").** Se houver necessidade de suportar vários protocolos de rede serão necessários vários encaminhadores em paralelos ("router" multiprotocolo). **Como as redes não estão ligadas no nível 2, o tráfego em "broadcast" não passa através da VPN.** As redes interligadas mantêm-se separadas no nível 2 e irão por isso constituir redes distintas sob o ponto de vista dos protocolos de nível 3.
6. Segurança das VPN: tratando-se transferências de dados que usam infraestruturas potencialmente inseguras e nas quais é possível todo tipo de intervenções de terceiros, a introdução de mecanismos de segurança é fundamental. Tanto a nível de autentificação (**garantir a autenticidade dos intervenientes (nós da VPN), ou seja, máquinas/servidores ou utilizadores**), como de privacidade (**garantir que os dados que são transferidos pela VPN não serão acessíveis e terceiros. Dado que se**

tratam de redes públicas não é possível controlar o acesso, logo é necessário recorrer à cifragem). A técnica de cifragem convencional é conhecida por criptografia simétrica e implica a partilha entre os dois envolvidos de uma chave secreta (PSK – PreSharedKey). Sendo um segredo pré-partilhado, este é um bom mecanismo de autenticação. Se uma das partes não possui a chave correta não vai poder comunicar. A operação de distribuição da chave pode ser complicada, na sua versão mais simples é realizada pelo administrador das duas extremidades da VPN.

   - Chaves públicas: a técnica de cifragem conhecida por criptografia de chave pública veio resolver de forma radical as dificuldades na distribuição das chaves. A chave usada para cifrar é pública, mas não serve para decifrar, isso é conseguido com uma outra chave designada de privada. A vantagem é que a chave privada nunca tem de ser transferida. Os certificados são um elemento importante porque garantem a autenticidade dos intervenientes. Se os nós forem controlados pelo mesmo administrador (VPN LAN-LAN) os certificados podem ser instalados manualmente para maior segurança.

7. VPN de utilizador: é tipicamente uma VPN HOST-LAN criada por iniciativa do utilizador, sendo que os dois nós da VPN assumem mais claramente uma relação cliente-servidor. O cliente contacta o servidor no endereço de rede fornecido pelo utilizador. Será então exigido ao utilizador elementos de autenticação, habitualmente constituídos pelo par "NOME-DE-UTILIZADOR mais PASSWORD". Posteriormente o servidor fornece ao cliente os parâmetros de configuração de rede para o cliente poder funcionar, sendo eles o "NOME-DE-UTILIZADOR e PASSWORD" que serve como autenticação do utilizador perante o servidor VPN que faz assim o controlo de acesso ao serviço. Se para o servidor é importante verificar a autenticidade do cliente (UTILIZADOR), também para o utilizador do serviço é importante ter algumas garantias, não convém que a PASSWORD seja entregue ao "primeiro servidor que aparecer".
   - Autentificação com chave pública: nesta abordagem começa-se por criar uma ligação segura e autenticada com base em certificados de chave pública que cada um envia ao parceiro. É particularmente importante a validação do certificado de chave pública do servidor por parte do cliente. Através da ligação segura criada é possível então enviar a PASSWORD para autenticação do utilizador.
   - Autentificação com chave privada: para se conseguir distribuir uma chave secreta por cliente e servidor e simultaneamente autenticar ambos pode-se lançar mão de um segredo que mais ninguém conhece: "a PASSWORD do utilizador". Um algoritmo produz uma chave secreta usando a PASSWORD, então reproduzindo o processo nos dois pontos temos uma chave secreta: CHAVE SECRETA = HASH (SALT + PASSWORD). Como a chave é secreta, tal como acontecia no PSK, o simples facto de a ligação segura funcionar autentica os intervenientes. Note-se que a PASSWORD nunca é transmitida. O sistema é de uma forma geral seguro, contudo baseia-se na PASSWORD do utilizador. Este é o seu ponto fraco, se a PASSWORD do utilizador for fraca, pode ficar a comprometer toda a segurança para o utilizador. O protocolo CHAP (Challenge Handshake Authentication Protocol) baseia-se nestes princípios de funcionamento. A necessidade de o servidor conhecer a PASSWORD do utilizador pode ser um obstáculo à sua implementação em alguns tipos de sistema, como por exemplo os da família Unix.

8. L2TP - Layer 2 tunneling protocol: é um protocolo de túnel simples, não implementa mecanismos de autenticação nem de privacidade. Normalmente a autenticação é assegurada pelo protocolo PPP e a confidencialidade é assegurada pelo IPsec. Este é uma parte integrante do IPv6 e um protocolo extra no IPv4, permite criar ligações seguras e autenticadas, baseadas quer em chaves secretas pré-partilhadas (PSK), quer em certificados de chave pública. A missão do L2TP é criar os túneis e transferir o respetivo tráfego, usando o modelo cliente/servidor, o LNS ("L2TP Network Server") é contactado no porto UDP 1701 pelo LAC ("L2TP Access Concentrator") para se estabelecer o túnel. Cada túnel é ainda dividido em sessões, para cada protocolo acima do L2TP será usada uma sessão diferente. Para implementar autenticação de utilizador é necessário recorrer ao protocolo PPP.
   - PPTP ("point-to-point tunneling protocol"): predecessor do L2TP. Não usa IPsec, o PPTP acaba por ser mais simples de configurar porque não exige chaves pré partilhadas (PSK) ou certificados de chave pública. O PPTP usa uma ligação TCP (porto 1723) para controlar a o túnel de dados que funciona sobre o protocolo GRE ("Generic Routing Encapsulation"). O protocolo GRE foi desenvolvido pela Cisco para criar túneis sobre o protocolo IP e tem o identificador de protocolo número 47. O protocolo GRE não foi desenvolvido para uso direto pelas aplicações (não define números de porto), o que provoca grandes problemas na tradução de endereços (NAT) nas redes privadas. Nem o PPTP, nem o GRE implementam mecanismos de segurança (privacidade/ autenticação), uma vez que são asseguradas pelo protocolo PPP. O protocolo PPP pode suportar diversos mecanismos de autenticação e privacidade, no contexto atual a Microsoft usa o protocolo de autenticação MSCHAPv2 (CHAP = Challenge Handshake Authentication Protocol) que além da autenticação o utilizador via PASSWORD permite gerar uma chave secreta para o protocolo MPPE ("Microsoft Point-to-Point Encryption") baseado no RC4 ("Rivest Cipher 4"). Juntamente com o MPPE a Microsoft usa ainda o MPPC ("Microsoft Point-to-Point Compression").

9. OpenVPN: implementação open Source de VPN sobre TLS, com privacidade através de criptografia de chave pública para distribuir a chave secreta, seguida de criptografia convencional com a chave que foi distribuída, mas também são suportadas chaves pré partilhadas (PSK). Quando há utilizadores envolvidos (VPN HOST-LAN) é possível a autenticação por PASSWORD, a qual é enviada diretamente ao servidor através de uma ligação segura já estabelecida. Nesta fase é fundamental que o servidor já se tenha autenticado perante o cliente, caso contrário corremos o risco de estar a entregar a PASSWORD a um desconhecido. A autenticação

do servidor perante o cliente deve ser feita através do certificado de chave pública do servidor, instalado manualmente no cliente. O OpenVPN pode funcionar (porto 1194) tanto sobre UDP como TCP

10. PPP ("Point to point protocol"): deriva diretamente do protocolo HDLC e assegura de forma bastante eficiente e completa o transporte de dados de nível 2 através de uma ligação dedicada ponto a ponto. Possui diversos mecanismos apropriados ao estabelecimento da ligação lógica entre os dois pontos, incluindo por exemplo mecanismos de autenticação.

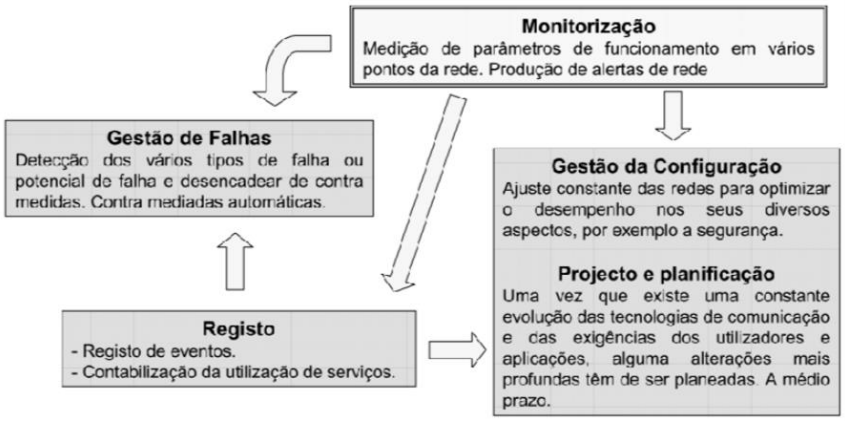
Marcador (7E)	Endereço (FF)	Controlo (03)	Protocolo	DADOS	FCS	Marcador (7E)

O campo "protocolo" serve para identificar os dados que são transportados; alguns protocolos são usados pelo próprio PPP. O identificador de protocolo 0xC021 é usado pelo protocolo LCP (Link Control Protocol). O protocolo LCP é o responsável pelo estabelecimento e manutenção da ligação nível 2. O LCP lida com a autenticação, por exemplo usando o "Challenge Handshake Authentication Protocol" (CHAP), e com a configuração da ligação de dados, negociando por exemplo o MTU.

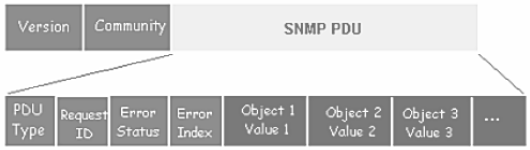
- NCP (network control protocol): usa os identificadores de protocolo "0x8?". É responsável pela interação com os protocolos de nível 3, por exemplo trata de definição dos parâmetros necessários a cada protocolo em particular. Para cada protocolo de rede existe um NCP específico.

15. GESTÃO DE REDES

1. Modelo "Agente-gestor": a maioria dos sistemas de gestão de rede adota um modelo do tipo cliente servidor no qual estão envolvidas duas entidades: O sistema gerido (repetidor, comutador, router, servidor, etc.) utiliza um repositório de informação conhecido por MIB ("Management Information Base"). O agente é um serviço de rede residente no sistema gerido e que permite o acesso à MIB usando um protocolo de gestão através da rede. O sistema gestor dialoga com os agentes residentes nos vários dispositivos da rede e constrói uma visão global. A gestão da rede centraliza-se no sistema gestor, que pode ser mais ou menos automatizado.



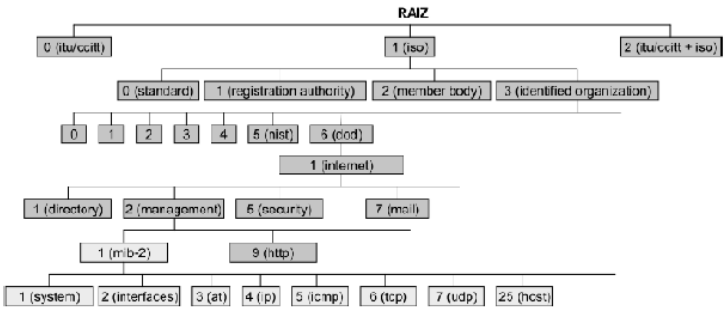
2. Protocolo de gestão: envolve dois tipos de transações: os pedidos enviados pelo sistema gestor aos agentes, neste caso os agentes assumem o papel de servidores (modelo cliente-servidor). Os pedidos são relativos a operações de gestão, quer de consulta, quer de alteração da configuração. Os alarmes enviados ao sistema gestor, por iniciativa dos agentes. Normalmente trata-se de alertas relativos a eventos ocorridos na rede ou até simples registos de atividades.
3. MIB ("management information base"): é o conjunto de dados associados a cada dispositivo de rede com capacidade de gestão, sendo na verdade uma visão externa (segundo o protocolo de gestão) do conjunto de dados internos do sistema. O "Agente" é responsável por criar essa visão conceptual (base de dados virtual) e interagir externamente segundo ela. Por ser especialmente adequado para este tipo de aplicação, a definição da MIB é muitas vezes orientada a objetos, sendo o paradigma dos objetos levado mais ou menos longe conforme a implementação em causa. Numa MIB de objetos pura, cada objeto de gestão é uma instanciação de uma classe (definida numa estrutura de classes e subclasses com herança). Cada classe define os métodos apropriados para interagir com o objeto.
4. SNMP ("Simple network management protocol"): estando as redes cada vez mais centradas na pilha TCP/IP e atendendo às várias evoluções que o protocolo SNMP tem sofrido no sentido do seu enriquecimento, não é de esperar a sua substituição no futuro. A versão 1 (SNMPv1) ainda é muito usada devido à quantidade de dispositivos que não suporta outras versões. Neste protocolo existem 5 mensagens possíveis, que são enviadas sob a forma de datagramas UDP. O Agente recebe pedidos no porto 161, segundo o modelo cliente-servidor e o Gestor recebe alarmes ("Traps") no porto 162. No SNMPv1 os objetos da MIB são simples variáveis. As mensagens "get request" permitem obter os valores de objetos da MIB através da mensagem "get response". A mensagem "set request" permite alterar o valor dos objetos, sendo confirmada por uma "get response". Na MIB os objetos são armazenados em sequência, a mensagem "getnext" permite uma consulta sequencial.



- Mensagens SNMPv1: seguem um formato geral, o campo "Version" identifica a versão do protocolo, o valor zero identifica o SNMPv1. O campo "Community" contém uma cadeia de caracteres que pode ser usada para controlo de acesso. "PDU Type" identifica o tipo de mensagem: O campo "Request ID" identifica um pedido, o valor é repetido na resposta o que permite relacionar a mesma com um pedido formulado anteriormente. Dado que o SNMP usa uma plataforma de transporte não fiável (UDP) esta característica torna-se importante. "Error Status" identifica o resultado da operação: Em caso de erro,

"Error Index" pode servir para indicar relativamente a qual dos objetos ocorreu esse erro. As mensagens tipo 0, 1, e 3 usam sempre o valor zero nos campos de erro. A mensagem "Trap" possui um formato diferente.

- OID – object identifier: os objetos (variáveis) residentes na MIB SNMP não são identificados por nomes. A norma ASN.1 (Abstract Syntax Notation One) **define um sistema universal de nomeação baseado numa árvore numérica.** Cada tipo de objeto (classe) é identificado pela sequência de números até à raiz, por exemplo o objeto "internet" é identificado por "1.3.6.1", todos os objetos abaixo começam por esta sequência, os objetos relativos à gestão de rede começam por "1.3.6.1.2.1". No ramo das interfaces (1.3.6.1.2.1.2) está definido o tipo de objeto "ifNumber" que contém o número de interfaces existentes. O tipo ifTable é uma tabela de objetos do tipo ifEntry (1.3.6.1.2.1.2.2.1), cada objeto do tipo ifEntry contém o objeto ifIndex que contém o número da interface, o seu valor pode ir de 1 a ifNumber e não pode haver números repetidos. **O protocolo SNMP usa os OID, contudo os OID referem-se a classes de objetos e não instâncias de objetos como os que existem na MIB do agente. Uma vez que podem existir várias instâncias da mesma classe, o SNMP acrescenta mais um número para identificar a instância, começando por zero para a primeira instância.** Por exemplo para saber quantas interfaces de rede um dispositivo tem envia-se o pedido "GET 1.3.6.1.2.1.2.1.0". Em tabelas, o SNMP usa determinados valores dos elementos da tabela para os identificar. No caso do "ifTable", usa o "ifIndex", portanto ifDesc.8 ou 1.3.6.1.2.1.2.2.1.2.8 representa a descrição da oitava interface que o sistema possui. Outras classes de objetos tipo tabela são: "atTable", "ipAddrTable"; "ipRoutingTable"; "tcpConnTable" e "egpNeighTable". Para todas elas o SNMP define regras específicas para identificar cada elemento da tabela.



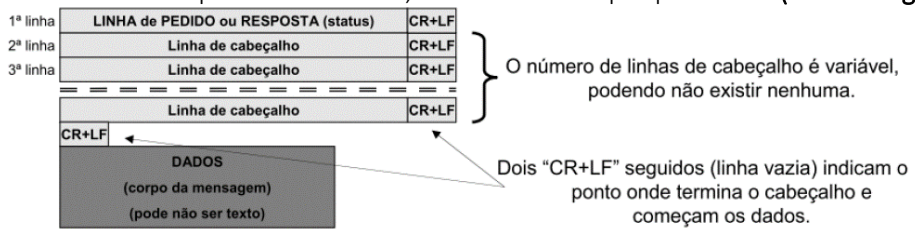
- Segurança: no SNMPv1 e SNMPv2c os mecanismos de segurança são elementares, normalmente é possível definir dois tipos de acesso: "leitura apenas" e "leitura e escrita", sendo estes dois tipos de acesso são associados a "comunidades" distintas. O identificador (nome) da comunidade serve ele próprio como "palavra-chave", além disso não existe qualquer tipo de cifragem, ou seja, **o nome da comunidade ("password") circula pela rede sem qualquer proteção.** O SNMPv2p e SNMPv2u diferem do SNMPv2c pela existência de mecanismos de segurança mais sofisticados, mas não tiveram grande sucesso. No SNMPv3 os mecanismos de segurança foram finalmente revistos. O SNMPv3 usa criptografia simétrica para garantir quer a autenticação (e controlo/diferenciação de acesso) quer a privacidade. **Na implementação destes mecanismos de segurança está implícita a distribuição prévia manual de um segredo (chave secreta) entre os dois extremos da ligação, eventualmente a através da "palavra-chave" do utilizador.** Apesar da evolução ainda há algumas limitações importantes, a autenticação baseia-se em MD5 ou SHA e a cifragem usa DES. Tanto a autenticação como a cifragem são opcionais, mas para ter cifragem é obrigatória a autenticação pois a chave secreta para a cifragem é gerada durante a autenticação.
- SNMPV2C, SNMPV3 e RMON: além de alterações e atualizações na MIB, o SNMPv2 introduz novas mensagens: "Getbulk Request" (serve para obter volumes de informação da MIB superiores aos permitidos pelo "Get Request") e "Info Request" (equivalente à mensagem "Trap", mas com confirmação da receção).
  As novas versões vieram dar suporte a uma maior variedade de aplicações, nomeadamente em redes de muito grande dimensão onde a recolha de informação tem de ser hierarquizada. Nesse domínio o suporte de ligações agente-agente traz novas possibilidades.
  Os dispositivos com capacidade RMON ("Remote monitoring") funcionam como gestores locais recolhendo informação dos agentes próximos via SNMP. **Além disso têm capacidade de monitorizar diretamente a rede (funcionamento como sonda) guardando todas estas informações numa MIB apropriada (MIB RMON). As MIB RMON podem depois ser consultadas pelo gestor SNMP central.**

16. PROTOCOLO HTTP

1. Transferência de ficheiros em rede: com o surgimento dos documentos de híper texto em rede, o protocolo mais usado nessa altura para transferir ficheiros, o FTP ("File Transfer Protocol"), revelou-se inapropriado para esse tipo de aplicação. **No entanto, o FTP é complexo e torna-se lento quando se pretende realizar muitas transferências de ficheiros de dimensão reduzida, envolvendo diversos servidores**, situação típica em documentos de híper texto como HTML ("Hypertext Markup Language"). Para "carregar" completamente um ficheiro HTML é necessário obter o ficheiro e também um conjunto mais ou menos vasto de outros ficheiros correspondentes a referências existentes. **O FTP, com a necessidade de uma ligação de controlo de sessão e autenticação de utilizador não é de todo adequada para este tipo de aplicação. Muitas vezes demora mais tempo a estabelecer a sessão FTP do que a transferir o ficheiro.**
2. HTTP ("hypertext transfer protocol"): o objetivo do http é proporcionar uma forma expedita de transferir ficheiros, segundo o modelo cliente servidor, com especial predomínio para as transferências no sentido servidor para cliente. **O cliente começa por estabelecer uma ligação TCP com o servidor que está à espera no porto número 80, depois de estabelecida a**

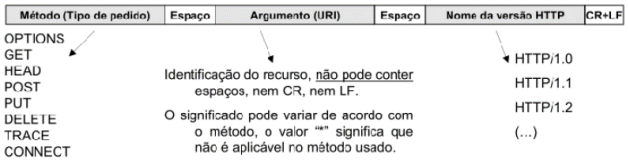
ligação o protocolo HTTP usa-a para comunicação entre as duas entidades. Seguindo o modelo cliente / servidor, o cliente envia um "pedido HTTP" e servidor devolve uma "resposta HTTP". O "HTTP 1.1" define vários tipos de pedido: OPTIONS; GET; HEAD; POST; PUT; DELETE; TRACE e CONNECT. Nem todos são suportados pelo "HTTP 1.0".

- Mensagens: as mensagens HTTP (pedidos e respostas) obedecem a um formato bem definido, organizado em linhas de texto de comprimento variável, terminadas sempre por CR+LF (CR=Carriage Return; LF=Line Feed):

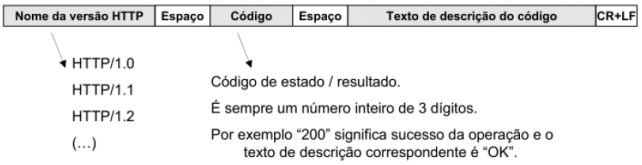


- Pedidos e respostas:

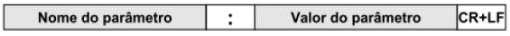
A linha de pedido (1ª linha do pedido) tem o seguinte formato:



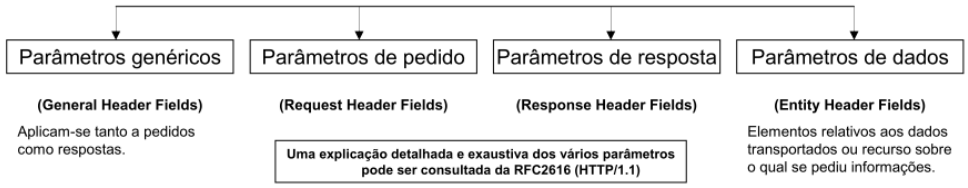
A linha de resposta / estado (1ª linha da resposta) tem o seguinte formato:



- Linhas/ parâmetros de cabeçalho: as linhas de cabeçalho servem para implementar diversas funcionalidades do protocolo HTTP, a sua forma geral é:



O "nome do parâmetro" é um identificador com significado especial para o protocolo HTTP, a interpretação deste identificador não é sensível a maiúsculas e minúsculas. Segue-se imediatamente o sinal de dois pontos. O "valor do parâmetro" pode ser precedido de caracteres brancos (ESPAÇO, TAB, etc.) que deverão ser ignorados. O valor de um parâmetro pode ocupar mais do que uma linha ("holding"), sempre que após o "CR+LF" surge um caractere branco, então trata-se de uma continuação da linha anterior e não uma nova linha.



- General header fields:

Cache-Control	Este parâmetro permite controlar os vários atributos do armazenamento da informação ao longo do percurso entre cliente e servidor, alguns valores possíveis são "no-cache"; "no-store"; "max-age"; "public"; "private".
Connection	O valor "close" indica que a ligação TCP deve ser fechada após a transferência dos dados. No HTTP/1.1 as ligações TCP entre cliente e servidor podem ser mantidas para além da satisfação do primeiro pedido. Este é o comportamento por omissão no HTTP/1.1 e seguintes. Se um pedido ou uma resposta contém a linha de cabeçalho "Connection: close" a ligação será quebrada.

Date	Contém a data/hora (em formato HTTP) a que a mensagem foi produzida.
Pragma	O valor "no-cache" indica que não devem ser usadas cópias em "cache", é equivalente ao valor "no-cache" no parâmetro "Cache-Control". O HTTP/1.0 não suporta o parâmetro "Cache-Control".
Warning	Permite adicionar à mensagem um aviso, entre outros elementos o aviso contém um código numérico de 3 dígitos, a identificação da entidade que o adicionou, uma descrição em texto e a data.

- Entity header fields: estes parâmetros aplicam-se ao documento transportado ou referido, os mais usados são:

Allow	Contém um conjunto de identificadores de métodos (GET; POST; etc.) que são aceites para o documento em causa.
Content-Encoding	Contém o identificador do método de codificação aplicado ao documento, por exemplo "gzip".
Content-Language	Contém o identificador da linguagem asscciada ao documento, por exemplo "pt-PT" ou "pt-BR".
Content-Length	Contém o tamanho do documento, em octetos (bytes).

Content-MD5	Contém o resultado da aplicação do algoritmo MD5 ao documento, serve para controlo de integridade, mas não garante qualquer tipo de segurança.
Content-Type	Identifica o conteúdo do documento, o identificador é constituído da seguinte forma: "TIPO/SUB-TIPO ; parâmetros". Exemplo: "Content-Type: text/html; charset=ISO-8859-4"
Expires	Contém a data/hora (formato HTML) em que o documento em "cache" perde a validade.
Last-Modified	Contém a data/hora (formato HTML) em que o documento foi alterado pela última vez. Normalmente o servidor obtém este valor do sistema de ficheiros.

- Request header fields:

Accept	Contém um conjunto de identificadores de tipo ("Content-Type") que serão aceitas como resposta ao pedido.
Accept-Charset	Idêntico ao anterior, mas para identificadores de conjuntos de caracteres.
Accept-Encoding	Idêntico ao anterior, mas para tipo de codificação ("Content-Encoding").
Accept-Language	Idêntico ao anterior, mas para tipo de linguagem ("Content-Language").
Authorization	Serve para autenticação do utilizador. Contém um "string" de validação, por exemplo contendo o nome de utilizador e respectiva "password". Torna-se necessário após uma resposta "401 Unauthorized" do servidor.

From	Contém o endereço de correio electrónico do utilizador.
If-Match	Permite definir uma condição baseada numa propriedade do documento "Entity Header" para que o pedido seja atendido.
Referer	Contém o URI do documento de onde partiu a referencia ao URI pedido.
User-Agent	Contém a identificação da aplicação que emitiu o pedido, normalmente um "BROWSER".
Cookie	Contém um par "nome=valor" que foi fornecido pelo servidor e serve para este identificar a sessão do cliente.

- Response header fields:

Location	Contém o URI absoluto do documento pedido.
Retry-After	Associado a uma resposta "503 Service Unavailable" ou a uma resposta "3xx", indica que o cliente deve voltar a enviar o pedido mais tarde.
Server	Contém um texto de identificação da aplicação servidora. Exemplo: "Apache/1.3.27 (Unix) (Red-Hat/Linux)"

WWW-Authenticate	Acompanha a resposta "401 Unauthorized". Contém uma identificação do método de autenticação que o cliente deve usar e parâmetros associados ao processo. Estão previstos dois métodos de autenticação: "Basic" – neste caso o conjunto USERNAME/PASSWORD são enviados em forma legível no pedido através do parâmetro "Authorization". Apenas é aceitável se usado sobre TLS (HTTPS). "Digest" – forma segura de autenticação em que é enviado o resultado da aplicação do algoritmo MD5 e não a PASSWORD.
Set-Cookie Set-Cookie2	Contém um par "nome=valor" que o cliente deve guardar e fornecer em todos os pedidos subsequentes com este servidor.

- HTTP1.1 - Métodos options e get:

OPTIONS	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

GET	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

Options: serve para obter uma lista de métodos aceites para acesso a um dado URI, ou genericamente pelo servidor (nesse caso o URI deve ser um asterisco). Na resposta "200 OK" será incluído o parâmetro "Allow" com a lista de métodos suportados e eventualmente outros parâmetros que sirvam para definir as capacidades do servidor.
Get: serve para **obter o documento identificado por "URI"**, sendo que se este identificar uma unidade de processamento (Ex.: ficheiro executável), então o servidor executa essa unidade e devolve o seu resultado ("output"). Esta técnica é conhecida por CGI (Common Gateway Interface). Neste contexto dos CGI podem ser fornecidos dados pelo cliente ao servidor (normalmente recolhidos por um formulário) esses dados têm de ser acrescentados ao URI, sendo separados do nome do objeto por um ponto de interrogação. O que se segue ao ponto de interrogação é conhecido por "query string" e pode **ser composto por vários campos**, separados por "&".
Exemplo: http://www.server1.net/login?username=teste&password=nenhuma&departamento=5
O **método GET não é a forma ideal para fornecer dados a um CGI no servidor**. Sendo que os **valores dos campos aparecem visíveis no URI** o que nem sempre será o mais adequado sob o ponto de vista de **privacidade**, além disso apenas são suportados valores em formato de texto. Por outro lado, alguns clientes servidores impõem limites ao tamanho do URI. O método POST é **mais adequado para este tipo de aplicação.**

- HTTP1.1 – Métodos HEAS, POST, PUT e DELTE:

HEAD	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

Serve para obter uma resposta exactamente igual à que seria obtida com o método **GET**, mas o documento não é enviado. Todos os parâmetros de cabeçalho devem ser iguais aos que seriam obtidos usando o método **GET** com o mesmo **URI**.

POST	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

O objectivo geral do **POST** é enviar dados a um **URI**. A forma como o método é processado é da responsabilidade do servidor, tipicamente o **URI** corresponde a um **CGI**, a diferença relativamente ao método **GET** é que os dados são enviados no corpo da mensagem, dessa forma não há restrições quanto ao volume de dados ou tipo de dados enviados.

A técnica de **CGI** assume uma importância bastante grande na utilização actual do **HTTP**, foram desenvolvidas ou adaptadas diversas linguagens de programação especialmente para este efeito.
Relativamente a linguagens interpretadas (scripts) destacam-se o **PHP**, **Perl**, **Python** e **ASP**.

PUT	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

Trata-se do método inverso do **GET**, ou seja serve para colocar um documento no servidor. O nome a dar ao documento é fornecido no **URI**, o conteúdo do documento é transportado no corpo da mensagem.

DELETE	Espaço	Argumento (URI)	Espaço	HTTP/1.1	CR+LF

Serve para eliminar um documento (**URI**) do servidor.

- HTTP1.1 – Códigos de resposta:

As respostas HTTP podem agrupar-se em 5 categorias:

HTTP/1.1	Espaço	1XX	Espaço	Texto de descrição do código	CR+LF

Os códigos **1XX** são de informação, não existiam no "**HTTP/1.0**". Exemplos:
"**100 Continue**" – indica que a primeira parte do pedido foi recebida e que o servidor aguarda algo mais

HTTP/1.1	Espaço	2XX	Espaço	Texto de descrição do código	CR+LF

Os códigos **2XX** indicam sucesso na operação realizada. Exemplos:
"**200 OK**" – indica sucesso num **GET**, **HEAD** ou **POST**

HTTP/1.1	Espaço	4XX	Espaço	Texto de descrição do código	CR+LF

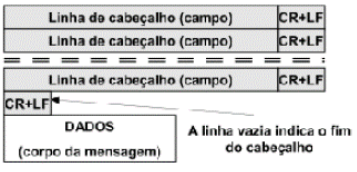
Os códigos **4XX** indicam um erro da responsabilidade do cliente. Exemplos:
"**400 Bad Request**" – o pedido foi mal formulado e não foi compreendido pelo servidor.
"**401 Unauthorized**" – o pedido só pode ser satisfeito após autenticação do utilizador.
"**402 Payment Required**"
"**403 Forbidden**" – o acesso ao recurso não é permitido.
"**404 Not Found**" – o pedido foi compreendido, mas o recurso não existe.
"**405 Method Not Allowed**" – o método usado no pedido não é aceitável para o **URI**.

HTTP/1.1	Espaço	5XX	Espaço	Texto de descrição do código	CR+LF

Os códigos **5XX** indicam um erro da responsabilidade do servidor. Exemplos:
"**500 Internal Server Error**" – erro grave no servidor que impede o seu funcionamento normal.
"**501 Not Implemented**" – o pedido necessita de uma funcionalidade não disponível.
"**503 Service Unavailable**" – o pedido não pode ser satisfeito devido a uma anomalia temporária.
"**505 HTTP Version Not Supported**" – o servidor não suporta a versão **HTTP** indicada no pedido.

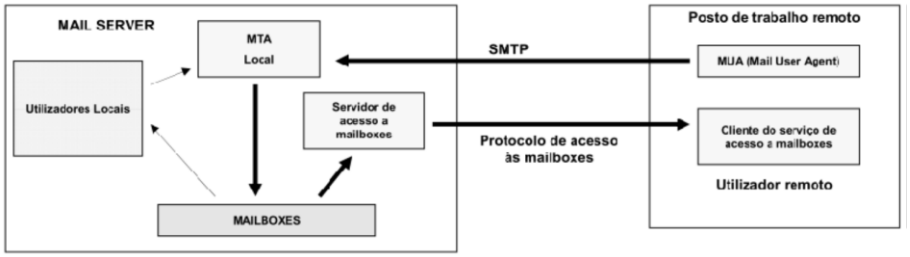
17. CORREIO ELETRÓNICO

1. Correio eletrónico: o objetivo do correio eletrónico é o envio de mensagens "off-line" (não interativo) entre utilizadores. Sendo que, o destinatário pode não estar presente no momento em que a mensagem chega ("off-line"). **Cada destinatário possui um local de armazenamento de mensagens onde o sistema de correio deposita as mensagens que lhe são destinadas.** Esse local de armazenamento é conhecido por caixa-do-correio ("mailbox") – **o sistema é implementado usando um sistema de ficheiros com permissões de utilizador.**
   - Correio eletrónico baseado em sistema de ficheiros: os sistemas de correio eletrónico desenvolveram-se usando simples sistemas de ficheiros partilhados. **Tanto as mail boxes dos utilizadores como a fila de entrada de correio são objetos do sistema de ficheiros, ficheiros e/ou diretórios.** Um sistema deste tipo está totalmente contido num único servidor e não utiliza a rede diretamente, sendo que a identificação dos utilizadores faz-se recorrendo apenas ao nome do utilizador.
   - Correio eletrónico em rede: com o progressivo desenvolvimento das redes de computadores, surgiu a necessidade de alargar o funcionamento dos sistemas de correio eletrónico existentes de tal modo que, **utilizadores de sistemas centrais diferentes possam também comunicar uns com os outros.** A comunicação entre sistemas centrais de correio recorre a uma infraestrutura de rede e será realizada segundo um protocolo de aplicação reconhecido pelos dois intervenientes. A identificação dos utilizadores (remetente e destinatário) necessita agora de mais um elemento, a **identificação do sistema de correio a que esse utilizador pertence, por exemplo: utilizador@sistema.**
2. MTA ("mail transport agent"): o sistema de processamento de correio residente em cada sistema passa a ter a **capacidade de dialogar através da rede com outros sistemas** e é designado de MTA ("Mail Transport Agent" ou "Message Transfer Agent"). Os utilizadores locais continuam a usar diretamente o sistema de ficheiros para enviarem correio e lerem o correio das respetivas mail boxes. O mesmo protocolo que é usado para envio de correio entre os **MTA** pode também ser **usado para utilizadores remotos enviarem correio**, através de software adequando designado de MUA ("Mail User Agent").
3. SMTP ("simple mail transfer protocol"): é **o protocolo de aplicação mais usado para transferir correio** entre sistemas. A identificação dos utilizadores (mailboxes) usa a forma: UTILIZADOR@NOME-DNS, "NOME-DNS" é o nome DNS qualificado do servidor de correio onde a mailbox desse utilizador se encontra. **Quando o MTA processa uma mensagem verifica se o "NOME-DNS" corresponde ao seu próprio nome,** nesse caso procura a mailbox local correspondente ao "UTILIZADOR" e deposita lá a mensagem. Se o "NOME-DNS" pertence a outro servidor, contacta esse servidor (resolvendo o nome DNS) e envia-lhe a mensagem usando o protocolo SMTP.
   - Nome de domínio e registos mx: a identificação de mailboxes usa a forma **UTILIZADOR@NOME-DNS** "**NOME-DNS**" **identifica o servidor de correio**, o endereço correspondente será contactado para efeitos de envio de correio. Na prática torna-se **mais cómodo identificar utilizadores em domínios DNS e não em servidores**. Para conseguir isso pode-se recorrer ao domínio acima e criar um registo **A** que provoque a resolução do nome de domínio para o endereço do servidor de correio. Atualmente o sistema **DNS** implementa registos apropriados para resolver este problema de uma forma mais eficiente, os registos MX ("Mail Exchanger"). Os registos **MX** associam diretamente o nome do domínio a um ou vários endereços dos servidores de correio desse domínio. Os **MTA** atuais resolvem o nome do domínio pedindo o respetivo registo **MX** e não o registo **A**.
   - Formato das mensagens: as mensagens de correio eletrónico são **constituídas por um cabeçalho seguido do corpo da mensagem**, sendo que cada linha de cabeçalho contém um identificador de campo e o respetivo valor, separado por ":". Os valores dos campos, por sua vez, podem ocupar mais do que uma linha, nesse caso as linhas de continuação devem começar com um espaço em branco.



   - Protocolo: o SMTP usa uma **ligação TCP para transferir a mensagem de correio eletrónico**, para esse efeito os MTA aceitam ligações TCP no número de porto 25. Depois de estabelecida a ligação inicia-se um diálogo baseado em linhas de texto terminadas por CR mais LF, seguindo um conjunto de comandos suportado (RFC 821).
   - ESMTP ("extended smtp / enhanced smtp"): o **ESMTP (RFC 1869)** possui um conjunto mais vasto de comandos do que o SMTP **normal.** O cliente que deseja usar ESMTP em lugar de **SMTP** usa o comando "**EHLO**" em lugar do habitual comando "**HELO**". Se o servidor suporta **ESMTP** responde com um código de sucesso (**250**), caso contrário responde com um código de erro (**5xx**), nesse caso o cliente terá de enviar um "**HELO**" e limitar-se ao **SMTP** normal. A identificação das extensões suportadas pelo servidor é fornecida ao cliente juntamente com a resposta ao "**EHLO**".

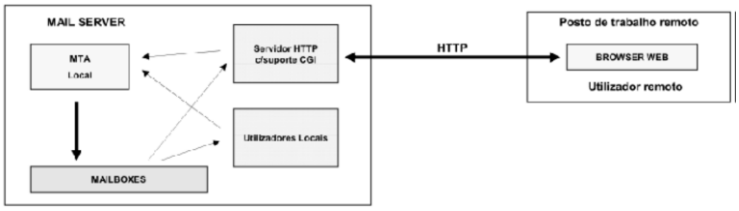
4. Acesso remoto: os utilizadores remotos de um sistema de correio podem recorrer ao protocolo **SMTP** para emitir mensagens, mas para poderem aceder às respetivas mailboxes torna-se necessário um protocolo adicional. **Atualmente os dois protocolos mais usados para acesso a mailboxes remotas são o IMAP4 e o POP3.**



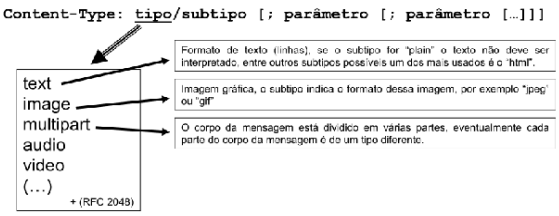
   - POP3 ("post office protocol version 3"): usa uma ligação TCP dirigida ao porto 110 do servidor, as mensagens trocadas entre o cliente comandos sob a forma de linhas de texto terminadas por CR+LF. Depois de o cliente POP3 receber a frase de identificação do servidor deve autenticar-se, o exemplo seguinte apresenta a "negrito" enviadas pelo cliente:

```
+OK POP3 frodo.dei.isep.ipp.pt 2004.89rdk server ready
USER andre
+OK User name accepted, password please
PASS xxxxxx
+OK Mailbox open, 0 messages
STAT
+OK 0 0
LIST
+OK Mailbox scan listing follows
.
QUIT
+OK Sayonara
```

   Além da autenticação baseada nos comandos USER/PASS, que só deve ser usada sobre ligações seguras (POP3S), também é suportada a autenticação tipo CHAP com o comando APOP.
   - IMAP4 ("internet message access protocol"): o protocolo IMAP4 (IMAP4rev1 - RFC 3501) é bastante mais interativo, também **usa uma conexão TCP,** nesta caso para o porto 143, mas normalmente a ligação do cliente com o servidor mantém-se ativa constituindo uma sessão interativa. **Entre outras funcionalidades, o IMAP4 permite: consulta da lista de mensagens disponíveis na mailbox; leitura de uma mensagem específica, ou até uma parte de uma mensagem; marcação de mensagens com estados (no servidor); organização da mailbox em pastas (no servidor); pesquisa de mensagens (no servidor).**
   - Webmail: trata-se de aplicações **CGI** que são executadas por um servidor **HTTP** residente na mesma máquina onde o sistema de correio está a funcionar. Os **CGI** que constituem o **WebMail** interagem com o sistema de correio eletrónico do mesmo modo que os utilizadores locais. **Deste modo eliminam todos os inconvenientes do acesso remoto sem necessidade de clientes especiais.**



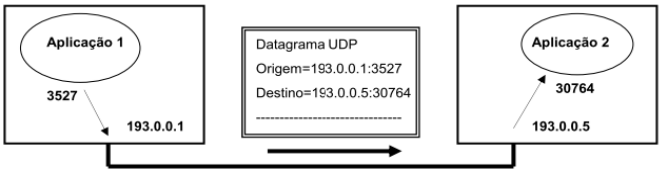
5. MIME ("multipurpose internet mail extensions"): é um formato de mensagens que permite ultrapassar as limitações das mensagens de texto simples. Embora tenha sido desenvolvido para o correio eletrónico é atualmente usado em muitos outros protocolos como por exemplo o **HTTP**. **Uma mensagem de correio eletrónico é identificada como estando em formato MIME através da presença do campo "MIME-Version" no cabeçalho, a versão atualmente em uso é a "1.0".** A mensagem em formato MIME possuem outras linhas de cabeçalho fundamentais para identificar quer o tipo de dados transportados no corpo ("Content-Type:") quer a forma como esses dados estão representados ("Content-Transfer- Encoding:"). No contexto do **SMTP** a mensagem é obrigatoriamente de texto, contudo esse texto pode ser usado para representar qualquer tipo de dados.
   - "Content-type": fornece informação sobre o tipo de dados transportados no corpo da mensagem. Os parâmetros são opcionais, são constituídos por pares "atributo=valor" e permitem definir mais detalhes sobre os dados. *Exemplo: Content-Type: text/html; charset=us-ascii.* **O campo "Content-Type" é importante porque permite à aplicação de destino interpretar os dados para os apresentar corretamente ao utilizador.**



   - "Content-type: multipart": o conteúdo "**multipart**" é bastante usado porque permite transportar numa única mensagem vários tipos de conteúdos diferentes em partes separadas.
   - "Content-transfer-encoding": o **SMTP** apenas suporta texto simples com caracteres de 7 bits, todos os outros tipos de conteúdos tem de ser representados recorrendo apenas a este formato de texto simples.
   - "Content-transfer-encoding: quoted-printable": o objetivo é permitir a representação de qualquer tipo de texto em formato de 7 bits. **A codificação baseia-se, entre outros, nos seguintes princípios: qualquer caractere de 8 bits (octeto), com a exceção de CR/LF pode ser representado pela sequência "=XX", onde XX representa o valor do octeto em notação hexadecimal; os caracteres com códigos ASCII 33 a 60 e 62 a 126 não necessitam de ser convertidos; os caracteres brancos (códigos ASCII 7 e 32) também não necessitam de ser convertidos, mas se ocorrerem no fim da linha, o final da linha terá de ser assinalado com o sinal "="; as linhas codificadas não podem ter mais do que 72 caracteres, o sinal "=" permite criar uma quebra de linha "soft", apenas para efeitos de texto codificado.**
   - "Content-transfer-encoding: base64": pode ser usada para qualquer tipo de dados, no entanto os dados codificados ocupam cerca de 33% mais espaço do que os dados originais.

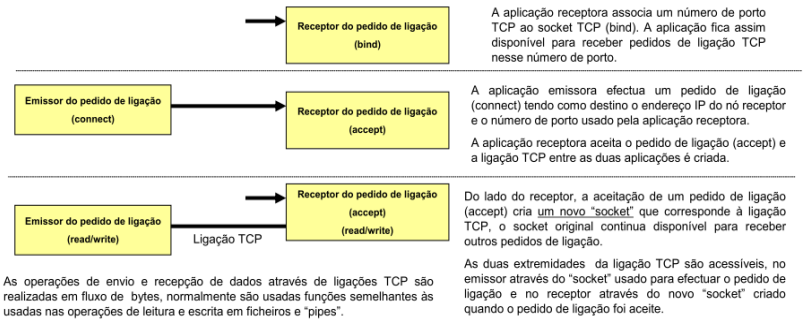
18. DESENVOLVIMENTO DE APLICAÇÕES DE REDE UDP TCP

1. Protocolo UDP ("User datagram protocol"): permite o envio de blocos de dados de tamanho variável, **destinado a ser usado por aplicações de rede, proporcionando uma forma de identificação de aplicações individuais.** A origem e destino de cada "datagrama" são identificados por números de 16 bits, conhecidos por números de porto. Os números de porto associados aos endereços **IP** dos nós identificam universalmente a origem e destino de cada "**datagrama**". Os números de porto **UDP** são ainda associados às aplicações através da operação "bind". Não podem existir duas aplicações no mesmo nó a usar o mesmo número de porto.



   - "Sockets" de rede: sob o ponto de vista de desenvolvimento de aplicações, as interações com a rede baseiam-se no **conceito de "socket", no caso do UDP um "socket" é associado a um número de porto local após o que fica em condições de receber e enviar "datagramas" UDP.** A associação de um "socket" UDP a um número de porto ("bind") é condicionada pela obrigatoriedade de nunca existirem nesse nó dois "Sockets" associados ao mesmo número de porto. Algumas aplicações não necessitam de usar nenhum número de porto em particular, para estes casos, é normalmente possível solicitar "um qualquer" número de porto que esteja livre. Geralmente isto é conseguido **efetuando o "bind" ao número de porto zero.** A maioria das comunicações de rede usa o modelo cliente-servidor, nesse contexto o servidor deve usar um número de porto pré acordado com o cliente. Por outro lado, para o cliente, qualquer número de porto serve.
   - Envio e receção de datagramas: **o serviço de "datagramas" UDP é não orientado à conexão e não oferece qualquer tipo de garantias.** Cada "datagrama" é tratado individualmente, por isso em cada operação de envio de um "datagrama" é necessário fornecer o endereço IP de destino e o número de porto de destino. **Para cada operação de envio de uma "datagrama" deve existir no endereço IP de destino especificado uma aplicação UDP à escuta no número de porto de destino especificado.**
     Sob o ponto de vista da aplicação, a receção é normalmente síncrona, ou seja, **a operação de receção bloqueia a execução da aplicação (processo ou thread) até que seja recebido um datagrama.** Após a receção de um "datagrama" UDP **a aplicação recetora tem acesso ao endereço IP de origem e número de porto de origem.** No caso de se tratar de um servidor pode usar estes elementos para enviar a resposta ao cliente. O envio de "datagramas" UDP não oferece qualquer tipo de garantias, nem qualquer "feedback", ou seja, o emissor não tem forma de saber se o "datagrama" chegou ou não ao destino. Existe uma única exceção a esta falta de "feedback": quando o nó de destino está operacional, e o número de porto de destino não está em uso por nenhuma aplicação, o nó de destino emite uma mensagem ICMP "Destination port unreachable", no nó de origem a API associa então o erro ao "socket" emissor.
     Todas as operações de envio e receção de rede são geridas pelo sistema operativo através de filas **FIFO**, ou seja, **mesmo que a aplicação não solicite explicitamente a receção de um "datagrama", eles podem estar a ser recebidos e serão disponibilizados à aplicação pela ordem em que chegaram, quando a aplicação o solicitar a sua receção.**
     O envio de "**datagramas**" é realizado através da invocação de uma "**system-call**" que recebe como argumentos um bloco de dados com determinada dimensão e um endereço de destino (endereço IP + número de porto UDP). A "system-call" solicita a receção de um "**datagrama**" devolve um bloco de dados com determinada dimensão e o endereço de origem do "**datagrama**" (**endereço IP + número de porto UDP**). Normalmente as receções de dados são bloqueantes, ou seja, se não existir nenhum "**datagrama**" na fila, a operação de receção bloqueia o **processo/thread** até que chegue um "**datagrama**".
   - Envio em "Broadcast": em **IPv4 o endereço de "broadcast" de uma rede é o último endereço dessa rede**, ou seja, o endereço de rede em que todos os bits à direita da mascara de rede têm o valor 1. A colocação no código de uma aplicação de um endereço de "**broadcast**" nesta forma não é, contudo, aceitável pois nesse caso a aplicação apenas funcionaria nessa rede. Para esse efeito o referido endereço de "**broadcast**" deverá ser colocado num ficheiro de configuração que é lido pela aplicação. Em alternativa, se apenas for pretendido o "**broadcast**" na rede local, pode ser usado o endereço "**255.255.255.255**" que permite o envio em "**broadcast**" na rede local independentemente de qual seja essa rede. O que permitir a um cliente contactar um servidor que se encontra na rede local, mas cujo endereço desconhece.
   - Envio em "Broadcast - localização de aplicações: numa arquitetura cliente-servidor pode adotar-se uma das técnicas seguintes:
     - Anúncio de servidores: um servidor pode enviar periodicamente em "**broadcast**", um **"datagrama" UDP** onde anuncia à rede a sua presença. Ao fazer este anúncio dá a conhecer a sua localização (**endereço IPv4**). Os clientes deste tipo de aplicação começam por escutar a rede recebendo **"datagramas" UDP** no porto pré combinado, constroem assim uma lista de servidores disponíveis guardando os respetivos **endereços IPv4**.
     - Pedido de servidores: O cliente envia em "**broadcast**", um **"datagrama" UDP** para um número de porto pré combinado, onde solicita servidores. Estes respondem, permitindo igualmente ao cliente a construção de uma lista de servidores disponíveis e respetivos **endereços IPv4**.

2. Clientes UDP – Tolerância de falhas: no contexto do modelo cliente-servidor, as limitações do protocolo **UDP** levam a que cada pedido seja tratado independentemente de pedidos anteriores (servidor idem potente). **Sob o ponto de vista de falhas na entrega de "datagramas" o servidor fica numa posição cómoda pois o seu funcionamento nunca será afetado por esse tipo de falhas.** O cliente pelo contrário, depois de enviar o pedido, **fica dependente da chegada de uma resposta do servidor**. A chave para resolver o problema passa por evitar o bloqueio da aplicação à espera da receção da resposta definindo um tempo máximo ara essa operação ("**timeout**").
   - Sockets não bloqueantes: é possível alterar o comportamento de um "**socket**" de forma a tornar-se não bloqueante, depois disso as operações realizadas sobre ele que levariam a um bloqueio retornam de imediato com erro. Torna-se então simples criar um ciclo que executa um número determinado de tentativas de receção em sucessivos intervalos de tempo, perfazendo no total o "**timeout**".
   - Threads: uma outra possibilidade é criar um "**thread**" para efetuar a receção e cancelar o "**thread**" passado um período de tempo pré determinado (" ") caso não tenha chegado nenhuma resposta.
   - Monitorização de "sockets": alguma **API** de "**sockets**" dispõem de funções de monitorização de "**sockets**", é o caso da "**system-call**" "**select**", permite por exemplo monitorizar um "**socket**" **UDP** para determinar se existe algum "**datagrama**" disponível para ser recebido. A "**system-call**" "**select**" permite associar um tempo máximo à operação.
   - Sinais: nalguns sistemas operativos, os processos são avisados da chegada de dados a um "**socket**" através de sinais, é o caso do sinal **SIGIO** no sistema **Unix**. Esta característica pode ser aproveitada.
3. Protocolo TCP ("transmission control protocol"): proporciona um serviço de qualidade significativamente superior ao do protocolo UDP, **permitindo criar ligações lógicas bidirecionais entre aplicações residentes em nós de rede distintos,** vulgarmente designadas "conexões TCP" fornecem garantias de entrega dos dados na ordem em que são emitidos. As ligações **TCP** são exclusivas dos dois nós entre os quais são criadas, são canais de comunicação dedicados nos quais não é possível a intervenção de terceiros. Sob o ponto de vista das aplicações**, as transações de dados via TCP são realizadas em fluxo, não existe o conceito de bloco de dados**, todos os dados são enviados e recebidos byte a byte num fluxo contínuo. Estas transações são apenas possíveis após uma fase prévia de estabelecimento da ligação TCP ("conexão TCP").
   - Estabelecimento da ligação TCP: para ser possível estabelecer a ligação TCP entre duas aplicações, **uma delas assume o papel de recetor do pedido de ligação**, para isso associa o "socket" TCP a um número de porto TCP previamente acordado com o emissor do pedido de ligação. De seguida fica à espera. **A outra aplicação pode então emitir um pedido de estabelecimento de ligação TCP especificando como destino o endereço IP do nó onde se encontra a primeira aplicação e o número de porto que essa aplicação está a usar.**



   - Canais dedicados- ligações TCP: antes de a ligação TCP ser estabelecida**, cada um dos "sockets" tem associado a si apenas o número de porto local.** Após o estabelecimento da ligação TCP, os "sockets" correspondentes à ligação têm associados também o endereço IP remoto e o número de porto remoto. **Pode observar-se que no recetor existem "sockets" diferentes associados ao mesmo número de porto local, algo que não é possível em UDP.** Os dados TCP que chegam da rede são disponibilizados no "socket" apenas e só se o número de porto de destino dos dados corresponde ao porto local, o endereço IP remoto corresponde ao endereço de origem dos dados e o número de porto remoto corresponde ao número de porto de origem dos dados.
4. Ligação UDP: apesar do UDP ser um protocolo sem ligação, contudo é possível associar a um "**socket**" local um "**socket**" remoto através do endereço **IP** e número de porto deste último. Na **API** de linguagem C, a "**system-call**" usada para este efeito é a mesma que é usada para estabelecer uma ligação **TCP**, contudo neste caso não se trata de nenhum tipo de ligação.

   Por exemplo, ao contrário do que acontece com uma ligação **TCP**, sempre que a aplicação entender pode associar "**socket**" a um outro endereço remoto, as vezes que quiser. A associação de um **"socket" UDP** a um endereço remoto tem efeito quer sob o ponto de vista de emissão de "**datagramas**" quer na receção:
   - Emissão – em cada emissão de um "datagrama" deixa de ser necessário especificar o destino, sendo usado sempre o endereço remoto que foi associado ao "socket".
   - Receção – o "socket" passa a receber apenas "datagramas" cujo endereço de origem corresponde ao endereço remoto que lhe foi associado.

   Não existe mais nenhum efeito destas associações, todas as características de falta de fiabilidade do **UDP** persistem. Este tipo de associação pode, contudo, ser bastante útil em algumas aplicações, em particular a filtragem de "**datagramas**" que fica associada ao processo de receção. Os servidores **UDP** multiprocesso aproveitam esta propriedade.

5. Ligação TCP: o protocolo TCP tem **a vantagem de garantir a entrega fiável dos dados na ordem exata em que são emitidos, sem qualquer limitação relativamente ao volume de dados enviados ou recebidos**. O envio e receção de dados através de uma ligação TCP são realizados byte a byte. Tem de haver uma correspondência exata entre o número de bytes cuja leitura é solicitada numa extremidade e os número de bytes que foram escritos na outra extremidade.
   - Se for solicitada a leitura de mais bytes do que os que foram escritos, a leitura vai ficar bloqueada à espera que surjam os bytes em falta. As consequências são óbvias.
   - Se for solicitada a leitura de menos bytes do que os que foram escritos, a leitura conclui-se sem problemas, mas os bytes que não foram lidos vão surgir na leitura seguinte.

   A garantia de que esta correspondência existe (sincronização entre leituras e escritas) é da responsabilidade do protocolo de aplicação onde estão definidas todas as trocas de informação entre as entidades envolvidas, tipicamente clientes e servidores.
   - Envio e receção (protocolo de aplicação): quando o protocolo de aplicação usa uma ligação TCP é da sua responsabilidade garantir que vai existir uma correspondência exata entre as operações de escrita e operações de leitura, realizadas nas extremidades opostas da ligação. Existem três abordagens ao problema que podem ser usadas isoladamente ou combinadas:
     - Blocos de tamanho fixo: se o protocolo de aplicação estabelecer mensagens de comprimento fixo em cada situação, a leitura dessas mensagens não apresenta qualquer problema. Esta solução pode conduzir a algum desaproveitamento da rede se o volume de informação útil transmitida for inferior ao tamanho fixo da mensagem.
     - Indicador do tamanho do bloco: o emissor começa por informar o recetor do tamanho da mensagem que se segue. Com esse conhecimento o recetor pode solicitar a leitura do número exato de bytes necessário. Exemplo: campo "Content-Length" do protocolo HTTP.
     - Marcador de fim de bloco: o protocolo estabelece um marcador para indicar o fim do bloco. O recetor efetua a leitura byte a byte até surgir o marcador. É simples de implementar quando se pode garantir que os dados nunca contêm o marcador escolhido. No cabeçalho das mensagens HTTP a sequência CR/LF é usada para identificar o fim das linhas e a sequência CR/LF/CR/LF é usada para identificar o fim do cabeçalho.
6. Servidores UDP: devido às limitações do **protocolo UDP**, um **servidor UDP** tem normalmente um funcionamento muito simples, limita-se a receber um pedido número de porto definido no protocolo de aplicação, sob a forma de um "**datagrama**", processar o pedido e enviar a resposta ao cliente. **Dada a ausência de qualquer canal de comunicação dedicado entre cliente e servidor, cada pedido é tratado individualmente.** Quando o servidor recebe um pedido guarda o endereço de origem para após o processamento enviar a resposta ao cliente. **Não existe qualquer tipo de sessão, o servidor atende os pedidos pela ordem de chegada.** Implementar protocolos de aplicação com sessão sobre UDP não é impossível, mas torna o servidor mais complexo. **O servidor terá de armazenar vários contextos de comunicação correspondentes a cada um dos clientes com quem está a comunicar e mediante a chegada de um pedido selecionar o contexto correto tendo como critério o endereço do cliente (endereço + número de porto).**
   - Servidores UDP multiprocesso: os servidores UDP normalmente são implementados num único processo, cada **porto UDP** é um ponto de receção único que não pode ser usado simultaneamente por vários processos sem conflitos. **Com vários processos a ler do mesmo "socket" nunca se saberia qual dos processos receberia o "datagrama".** Contudo através da associação do **"socket" UDP** a um endereço remoto obtém-se um ponto distinto de receção pois esse "**socket**" deixa de receber "**datagramas**" de outros endereços (endereço + porto) que não o que foi associado.

     Quando o servidor recebe um pedido cria uma cópia do "**socket**" (**DUP**) e associa a essa cópia o endereço de origem do "**datagrama**" (**CONNECT**), de seguida cria um processo filho (**FORK**), o processo filho usa a cópia que está associada ao endereço remoto. O processo pai continua a usar o "**socket**" original que não está associado a nenhum endereço remoto. Sempre que chega um "datagrama" ao **porto UDP**, é verificado se o endereço de origem corresponde a um endereço associado a um "**socket**", nesse caso disponibiliza os dados apenas nesse "**socket**".
   - Tamanho dos datagramas: teoricamente um **"datagrama" IPv4** pode ter 65535 bytes, por isso o **"datagrama" UDP** poderia ter esse comprimento descontando o tamanho do cabeçalho *IPv4* (20 a 60 bytes) e o tamanho do cabeçalho **UDP** (8 bytes). **No entanto, atualmente não se usa fragmentação, por outro lado a RFC 791 (IPv4) diz que todos os nós são obrigados a suportar "datagramas" pelo menos até aos 576 bytes.**

     A forma de garantir que o volume excessivo de dados de um **"datagrama" UDP** não vai impedir a sua chegada ao destino é evitar ultrapassar 512 bytes (RFC 791), sendo que pode ser necessário dividir a informação a enviar em vários datagramas ou esquecer os **"datagramas" UDP** e optar antes por **conexões TCP**.

     A divisão de um bloco de dados por vários **"datagramas" UDP** vai obrigar as aplicações a realizarem uma série de procedimentos de verificação. No mínimo, o recetor terá de saber o número total de "**datagramas**" e estes terão de ser numerados.
7. Servidores TCP: quando o **servidor TCP** aceita um pedido de **ligação TCP** de um cliente, fica estabelecida uma **ligação TCP** e do lado do servidor é criado um novo "**socket**" associado a essa ligação. O formato das mensagens trocadas entre cliente e servidor através da **ligação TCP** é definido pelo **protocolo de aplicação**, mas graças ao carácter persistente da **ligação TCP** constitui uma sessão no sentido em que estas interações podem não se limitar apenas a um pedido e respetiva resposta.

- Servidores TCP multiprocesso: depois de um servidor TCP aceitar uma ligação de um cliente tem de se manter disponível para aceitar outros clientes, ou seja, tem de dialogar com o cliente segundo o protocolo de aplicação através do "**socket**" correspondente à **ligação TCP**, mas também tem de aceitar novos pedidos de ligação.
  O servidor tem de lidar com dois "**sockets**" aceitando novos clientes no "**socket**" original (**accept**) e dialogando com o cliente no novo "**socket**" correspondente à **ligação TCP**. Existem muitas formas de resolver o problema, em **Unix** uma das mais populares é criar um processo filho exclusivo para o novo cliente. Esta forma de implementar o **servidor TCP** tem a vantagem de criar um processo independente para atender cada cliente. Cada um destes processos filho fica inteiramente dedicado a servir um cliente em particular com todas as vantagens que daí advêm.

8. Protocolos de aplicação: é um conjunto de regras que visam **permitir a troca de informação sem ambiguidades** entre duas entidades que comunicam através de uma infraestrutura de rede. Quando as entidades envolvidas são aplicações finais (situadas no nível 7 – camada de aplicação do MR-OSI) estes protocolos são designados "protocolos de aplicação".
   O protocolo é uma especificação o mais formal e exata, possível de todos os formatos de mensagens usados nas diferentes fases dos diálogos; todas os diálogos e ações possíveis, os respetivos objetivos e resultados possíveis; procedimentos de deteção e recuperação de erros.
   Os protocolos de aplicação devem ser flexíveis permitindo a implementação de novas funcionalidades mantendo a compatibilidade com versões anteriores.
9. Receção assíncrona: a disponibilidade de dados para serem recebidos da rede constitui eventos assíncronos no sentido em que a aplicação não tem um controlo preciso sobre o instante em que vão ocorrer. **Muitas aplicações limitam-se a solicitar a leitura e aguardar (bloquear) até que os dados estejam disponíveis, este tipo de procedimento pode designar-se receção síncrona.** Para muitas aplicações esse procedimento não é aceitável porque estão a usar vários "**sockets**" e não sabem em qual deles vão surgir os primeiros dados, necessitando de executar outras tarefas enquanto não chegam dados. Existem várias soluções para receção assíncrona:
   - "sockets" não bloqueantes: obriga a aplicação a periodicamente realizar uma sequência de tentativas de leitura nos vários "sockets" para verificar se existem dados. Este método pode ser mais eficiente se for desencadeado apenas quando o sistema operativo alerta o processo para o facto de terem chegado dados (Ex.: sinal SIGIO nos sistemas Unix).
   - "threads" ou processos: nesta solução é criado um "thread" ou um processo para ler de cada "socket", apenas esse processo ou "thread" fica bloqueado à espera de dados.
   - Função específica para monitorizar um conjunto de "sockets": esta função recebe como argumento um conjunto de "**sockets**" e desbloqueia quando chegam dados a qualquer um deles, ou se esgota o "**timeout**" também fornecido à função.