

APONTAMENTOS DE RCOMP

Índice

T1	8
Transmissão de Dados Digitais	8
MEIOS IDEAIS DE PROPAGAÇÃO:.....	8
FENÓMENOS ASSOCIADOS AOS SINAIS:.....	8
Comunicação em rede.....	9
NÓS DE REDE.....	9
TRANSPORTE PELA REDE – “BROADCAST”	9
TRANSPORTE PELA REDE – COMUTAÇÃO.....	9
PACOTES.....	9
PACOTES – INFORMAÇÃO DE CONTROLO.....	10
T2	10
Comunicação em Rede	10
COMUTAÇÃO DE PACOTES COM CIRCUITOS VIRTUAIS	10
PROCESSO:.....	10
ATRASOS NA REDE:	11
CONTROLO DE FLUXO:.....	11
CONTROLO DE ERROS:.....	11
Arquitecturas e pilhas de protocolos	12
ARQUITECTURAS PROPRIETÁRIAS:	12
MODELO DE REFERÊNCIA OSI:	12
MODELO OSI - CAMADAS:	12
PROTOCOLOS:	13
ARQUITECTURA IEEE 802 (ISO 8802):.....	13
CAMADA MAC:	13
ARQUITECTURA TCP/IP:	14
CONSTITUIÇÃO DA PILHA TCP/IP (PROTOCOLOS MAIS IMPORTANTES):	14
ENCAMINHAMENTO IP:	14
CAMADAS MULTI-PROTÓCOLO:	14
MODELO CLIENTE-SERVIDOR:	14
CONCLUSÕES:	15
T3	15
Tecnologias de rede local ETHERNET	15
REDES ETHERNET – CSMA/CD:	15
REDES ETHERNET – PACOTES E ENDEREÇOS:	15
REDES ETHERNET – FORMATO DE PACOTE:	16
REDES ETHERNET – TOPOLOGIAS (DO BARRAMENTO À ESTRELA):	16

REDES ETHERNET – COMUTAÇÃO DE TRAMAS:	16
REDES ETHERNET COMUTADAS:	16
Redes locais virtuais (VLAN)	17
REDES LOCAIS VIRTUAIS – IEEE802.1Q:	17
REDES LOCAIS VIRTUAIS – COMUTADORES:	17
Redes locais sem fios (WLAN)	17
802.11 – MODULAÇÃO:	18
802.11 – CSMA/CA:	18
IEEE 802.11 RTS/CTS:	18
802.11 – SEGMENTAÇÃO:	19
802.11 – TRAMAS:	19
802.11 – SEGURANÇA:	19
CONFIDENCIALIDADE	19
T4	20
Tecnologias WAN: ATM/ISDN e DSL	20
WAN – REDES ATM:	20
REDES ATM – CANAIS E CAMINHOS VIRTUAIS:	20
REDES ATM – CÉLULAS:	20
REDES ATM – ISDN:	20
REDES ATM – AAL5:	21
WAN – CONECTIVIDADE IP:	21
WAN – INTERLIGAÇÃO DE NÓS IP:	21
WAN – OPERADORES DE TELECOMUNICAÇÕES:	21
WAN – TECNOLOGIAS:	21
WAN – LIGAÇÕES DEDICADAS:	22
WAN – TECNOLOGIAS DE ACESSO:	22
WLL – “WIRELESS LOCAL LOOP”:	22
DSL – DIGITAL SUBSCRIBER LINE/LOOP:	22
ADSL – “ASYMMETRIC DIGITAL SUBSCRIBER LINE”:	23
VDSL – “VERY-HIGH-BIT-RATE DIGITAL SUBSCRIBER LINE”:	23
ACESSO VIA REDE CATV:	23
DOCSIS – DATA OVER CABLE SERVICE INTERFACE SPECIFICATION:	23
ACESSO VIA REDE ELÉCTRICA:	24
Pilha de protocolos TCP/IP	24
TRANSPORTE DE DADOS IP:	24
A CAMADA IP:	24
DATAGRAMAS IP:	24
TAMANHO DOS DATAGRAMAS IP:	25
DATAGRAMAS IP – FRAGMENTAÇÃO:	25

PMTUD (PATH MAXIMUM TRANSMISSION UNIT DISCOVERY):.....	25
T5	26
Pilha de protocolos TCP/IP	26
CAMADA IP - PILHA DE PROTOCOLOS TCP/IP.....	26
DATAGRAMAS IP.....	26
FRAGMENTAÇÃO: (é a solução teórica mais completa).....	26
PMTUD (PATH MAXIMUM TRANSMISSION UNIT DISCOVERY) – É UMA ALTERNATIVA ÀS FALHAS DA FRAGMENTAÇÃO:.....	26
TRANSPORTE DE DATAGRAMAS IP:.....	27
Protocolos ARP; UDP; BOOTP e DHCP.....	27
ARP – ADDRESS RESOLUTION PROTOCOL:.....	27
PROTOCOLO ARP:.....	27
ENDEREÇAMENTO IP:.....	27
PROTOCOLO UDP ("USER DATAGRAM PROTOCOL")	28
PROTOCOLO BOOTP	28
BOOTP DINÂMICO	28
Protocolo ICMP e protocolo TCP	28
PROTOCOLO DHCP ("DYNAMIC Host CONFIGURATION PROTOCOL"):	28
PROTOCOLO ICMP:	28
MENSAGEM ICMP "DESTINO INATINGÍVEL":.....	29
OUTRAS MENSAGENS ICMP IMPORTANTES:.....	29
PROTOCOLO TCP:.....	29
T6	30
Encaminhamento IPv4.....	30
NÓS INTERMÉDIOS:.....	30
ENCAMINHADORES ("ROUTERS"):	30
Encaminhamento estático e encaminhamento dinâmico.....	31
ENCAMINHAMENTO ("ROUTING"):	31
TABELA DE ENCAMINHAMENTO ("ROUTING TABLE"):	31
ENCAMINHAMENTO:	31
REDES LOCAIS E CAMINHO POR OMISSÃO:.....	32
CAMINHOS ALTERNATIVOS:	32
ENCAMINHAMENTO DINÂMICO:	32
ALGORITMOS DISTANCE-VECTOR:.....	32
ALGORITMOS LINK-STATE:	33
Protocolos de encaminhamento: RIP, RIPv2, EIGRP e OSPF	33
SISTEMAS AUTÓNOMOS (AS) – PROTOCOLOS IGP E BGP:.....	33
PROTOCOLOS IGP:.....	33
PROTOCOLO RIPv1:.....	34
PROTOCOLO RIPv2:.....	34

PROTOCOLO OSPF:	34
ÁREAS OSPF:	34
PROTOCOLO EIGRP:	34
Sistemas autónomos e redistribuição de rotas	35
SISTEMAS AUTÓNOMOS EIGRP:	35
ENCAMINHAMENTO ENTRE SISTEMAS AUTÓNOMOS:	35
REDISTRIBUIÇÃO DE ROTAS:	35
T7	36
IPv6 e ICMPv6	36
INTERNET PROTOCOL V6:	36
ENDEREÇAMENTO IPV6 - REPRESENTAÇÃO:	36
IPV6 – TIPOS DE ENDEREÇO:	37
IPV6 – ENDEREÇOS MULTICAST:	37
IPV6 – ENDEREÇOS ANYCAST:	37
PACOTES IPV6:	38
PACOTES IPV6 – CABEÇALHOS DE EXTENSÃO:	38
ICMPV6 (ICMP PARA IPV6):	38
T8	39
Resolução de nomes DNS e WINS/NetBIOS	39
RESOLUÇÃO DE NOMES:	39
RESOLUÇÃO DE NOMES - NETBIOS:	39
NETBIOS – REGISTO DE NOMES:	40
TIPOS DE NOMES NETBIOS (WINDOWS):	40
WINS – SERVIDOR DE NOMES NETBIOS:	40
FUTURO DA RESOLUÇÃO DE NOMES NETBIOS:	41
DNS – DOMAIN NAME SYSTEM:	41
SERVIDORES DNS:	41
DNS – REDE DE NS:	42
RESOLUÇÃO DE NOMES DNS:	42
REGISTOS DNS (RESOURCE RECORDS):	42
TIPOS DE “RESOURCE RECORD”:	43
DNS RR – NOMES E APELIDOS:	43
DNS RR – NS E “GLUE RECORDS”:	44
DNS RR – PTR E DOMÍNIO IN-ADDR.ARPA:	44
CORREIO ELECTRÓNICO:	44
DNS RR – MX:	45
DNS RR – SRV:	45
DNS RR – SPF:	46
DNS RR – LOC:	46

DDNS – DNS DINÂMICO:	46
T9	47
Redes Privadas Virtuais (VPN)	47
VIRTUAL PRIVATE NETWORK (VPN):	47
VPN LAN-LAN (“SITE-TO-SITE VPN”):	47
VPN HOST-LAN (“REMOTE-ACCESS VPN”):	47
INTERLIGAÇÃO DE REDES POR VPN – NÍVEL 2:	48
INTERLIGAÇÃO DE REDES POR VPN – NÍVEL 3:	48
SEGURANÇA DAS VPN:	48
VPN – CHAVES PÚBLICAS:	49
VPN DE UTILIZADOR:	49
VPN DE UTILIZADOR – AUTENTICAÇÃO COM CHAVE PÚBLICA:	49
VPN DE UTILIZADOR – AUTENTICAÇÃO COM CHAVE SECRETA:	50
L2TP - LAYER 2 TUNNELING PROTOCOL:	50
Protocolo PPP	51
PPTP - POINT-TO-POINT TUNNELING PROTOCOL:	51
OUTROS PROTOCOLOS DE VPN:	51
UMA VPN SOBRE TLS - OPENVPN:	51
PPP – POINT TO POINT PROTOCOL:	52
PPP – NCP (NETWORK CONTROL PROTOCOL):	52
T10	53
Gestão de redes	53
GESTÃO DE REDES:	53
MODELO “AGENTE – GESTOR”:	53
PROTOCOLO DE GESTÃO:	53
MIB (“MANAGEMENT INFORMATION BASE”):	53
Protocolo SNMP	54
SNMP (“SIMPLE NETWORK MANAGEMENT PROTOCOL”):	54
SNMPv1 - MENSAGENS:	54
OID – OBJECT IDENTIFIER:	55
EXEMPLO: “INTERFACES”, OID = 1.3.6.1.2.1.2:	55
SEGURANÇA NO SNMP:	55
SNMPv2c, SNMPv3 e RMON:	56
EXEMPLOS SNMP – “TRAPS”:	56
EXEMPLOS SNMP – MIB:	56
EXEMPLOS SNMP – GRÁFICOS DESENHADOS PELO GESTOR:	57
T11	57
Protocolo HTTP	57
TRANSFERÊNCIA DE FICHEIROS EM REDE:	57

HTTP - HYPERTEXT TRANSFER PROTOCOL:.....	57
MENSAGENS HTTP:.....	57
HTTP – PEDIDOS E RESPOSTAS:.....	58
HTTP – LINHAS DE CABEÇALHO (PARÂMETROS DE CABEÇALHO):.....	58
HTTP – GENERAL HEADER FIELDS:.....	58
HTTP – ENTITY HEADER FIELDS:.....	59
HTTP – REQUEST HEADER FIELDS:.....	59
HTTP – RESPONSE HEADER FIELDS:	59
HTTP/1.1 – MÉTODOS OPTIONS E GET:	60
HTTP/1.1 – MÉTODOS HEAD, POST, PUT E DELETE:.....	60
HTTP/1.1 – CÓDIGOS DE RESPOSTA:	60
HTTP/1.1 – CÓDIGOS DE RESPOSTA (4XX E 5XX):.....	61
T12	61
Correio electrónico	61
CORREIO ELECTRÓNICO:	61
CAIXAS DE CORREIO – “MAILBOXES”:.....	61
CORREIO ELECTRÓNICO BASEADO EM SISTEMA DE FICHEIROS:	62
CORREIO ELECTRÓNICO EM REDE:.....	62
MTA – MAIL TRANSPORT AGENT:	62
Protocolos SMTP, POP3 e IMAP	62
SMTP – SIMPLE MAIL TRANSFER PROTOCOL:	62
SMTP – NOME DE DOMÍNIO E REGISTOS MX:.....	63
SMTP – FORMATO DAS MENSAGENS:	63
SMTP – PROTOCOLO:.....	63
ESMTP – EXTENDED SMTP / ENHANCED SMTP:	63
CORREIO ELECTRÓNICO – ACESSO REMOTO:.....	64
POP3 – POST OFFICE PROTOCOL VERSION 3:	64
IMAP4 – INTERNET MESSAGE ACCESS PROTOCOL:.....	64
WEBMAIL:	64
Formato MIME	65
MIME - MULTIPURPOSE INTERNET MAIL EXTENSIONS:.....	65
MIME – “CONTENT-TYPE”:	65
MIME – “CONTENT-TYPE: MULTIPART”:.....	65
MIME – “CONTENT-TRANSFER-ENCODING”:.....	66
MIME – “CONTENT-TRANSFER-ENCODING: QUOTED-PRINTABLE”:.....	66
MIME – “CONTENT-TRANSFER-ENCODING: BASE64”:	66
T13	66
Desenvolvimento de aplicações de rede UDP e TCP	66
PROTOCOLO UDP (“USER DATAGRAM PROTOCOL”):.....	66

"SOCKETS" DE REDE EM UDP:	67
ENVIO E RECEPÇÃO DE DATAGRAMAS UDP:.....	67
PROTÓCOLO TCP ("TRANSMISSION CONTROL PROTOCOL"):	67
ESTABELECIMENTO DA LIGAÇÃO TCP:.....	68
LIGAÇÕES TCP – CANAIS DEDICADOS:.....	68
UDP – ENVIO E RECEPÇÃO DE "DATAGRAMAS":	68
UDP – SERVIÇO NÃO FIÁVEL:.....	69
CLIENTES UDP – TOLERÂNCIA A FALHAS:	69
CLIENTE UDP TOLERANTE A FALHAS – SOCKET NÃO BLOQUEANTE:	69
UDP – ENVIO EM "BROADCAST":.....	70
UDP – ENVIO EM "BROADCAST" – LOCALIZAÇÃO DE APLICAÇÕES:.....	70
UDP – TAMANHO DOS "DATAGRAMAS":	70
UDP – TRANSACÇÕES EM MÚLTIPLOS "DATAGRAMAS":	70
SOCKETS UDP – ASSOCIAÇÃO A ENDEREÇOS REMOTOS:.....	71
LIGAÇÕES TCP - ENVIO E RECEPÇÃO - PROTOCOLO DE APLICAÇÃO:	71
SERVIDORES TCP:.....	72
SERVIDORES TCP MULTI-PROCESSO:.....	72
SERVIDORES UDP:.....	72
SERVIDORES UDP MULTI-PROCESSO:.....	73
PROTOCOLO DE APLICAÇÃO:	73
RECEPÇÃO ASSÍNCRONA:	73

APONTAMENTOS DE RCOMP

T1

Transmissão de Dados Digitais

A solução para a transmissão de dados é a mesma que é utilizada para o armazenamento de dados num dado suporte, é alterar uma propriedade de tal forma que essa propriedade seja uma representação da informação.

Sinais são fenómenos físicos com capacidade de se propagarem (percorrerem uma certa distância)

MEIOS IDEAIS DE PROPAGAÇÃO:

Luz – fibra óptica

Corrente Eléctrica – cobre

Radiação Electromagnética – tal como a luz, propaga-se melhor no espaço vazio, mas ao contrário da luz tem a capacidade de atravessar os materiais sem grandes problemas.

Os sinais electromagnéticos surgem quando existe uma variação de corrente eléctrica.

A frequência (número de ciclos que ocorrem em cada segundo) condiciona fortemente as propriedades do sinal

Baixas frequências: propagação no espaço muito limitada

Rádio frequência: propagação em todas as direcções

Microondas: propagação começa a assemelhar-se à da luz

FENÓMENOS ASSOCIADOS AOS SINAIS:

- Atenuação
- Atenuação x Frequência / Largura de Banda
- Velocidade de Propagação x Frequência
- Ruído

Sinais digitais (produzidos por aplicação de técnicas de codificação)

- Variações bruscas entre patamares bem definidos
- O sinal é produzido directamente dos dados
- Só podem ser usados se for suportada frequência zero

No espaço de tempo que decorre entre essas 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 completamente distorcido.

Os sinais analógicos usam-se quando o meio de transmissão impõe limites bem definidos quanto aos valores da frequência (banda canal) e forma do sinal (sinusoidal)

- Variações contínuas
- Usados quando não é possível recorrer a sinais digitais
- Não é possível produzir directamente dos dados, sendo necessário alterar as propriedades do sinal (modulação) aplicando as seguintes técnicas ou a combinação delas:

FSK – alteração de frequência

Consiste em alterar ligeiramente a frequência do sinal, fazendo corresponder a cada valor da frequência um símbolo da informação a transmitir

ASK – alteração da amplitude

Consiste em alterar ligeiramente a amplitude do sinal, fazendo corresponder a cada valor da amplitude um símbolo da informação a transmitir

PSK – Alteração da fase

Consiste em utilizar diferenças de fase (SALTOS no tempo) para representar símbolos a transmitir

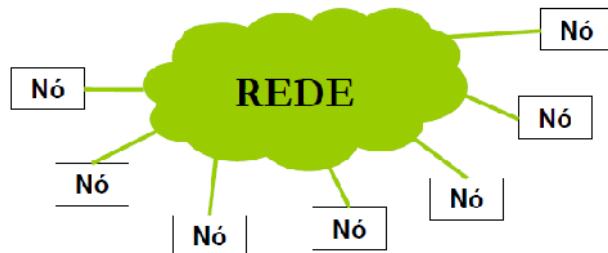
Comunicação em rede



- Numa rede de computadores, qualquer participante é simultaneamente emissor e receptor, estes participantes são conhecidos por nós de rede.
- Numa rede de computadores, pode existir um grande número de nós, qualquer um deles deve ter a possibilidade de transmitir dados a qualquer um dos outros nós da rede.

Nós DE REDE

Os nós são entidades emissoras/receptoras de dados ligados a uma infra-estrutura de transmissão de dados designada por rede.



A rede pode ser um sistema muito simples, pode ser apenas um meio de transmissão partilhado (rede de "broadcast").

Outras redes são muito mais complexas e são capazes de determinar o caminho entre dois nós (rede de comutação).

Cada nó é um potencial emissor e receptor de dados, por isso tem

de ter associado um elemento identificador que seja único em toda a rede.

TRANSPORTE PELA REDE – "BROADCAST"

A rede tem 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.

As redes mais simples usam um meio de transmissão comum para conseguir esse objectivo, neste caso o problema está resolvido desde logo.

Estas redes têm alguns inconvenientes:

- Necessidade de controlo de acesso ao meio (MAC);
- Baixa eficiência sob tráfego elevado; falta de segurança.

TRANSPORTE PELA REDE – COMUTAÇÃO

As redes de comutação são bastante mais complexas, baseiam-se num conjunto de nós intermédios com várias ligações entre si.

Os nós intermédios têm o importante papel de tomar decisões de encaminhamento em função do endereço de destino dos dados.

PACOTES

As redes impõem aos seus nós determinadas regras que visam o bom funcionamento geral. Sendo a rede um sistema partilhado por vários nós estas regras são importantes para evitar desequilíbrios no seu uso.

Uma das regras que quase todos os tipos de rede impõem aos seus nós é um limite quanto ao volume máximo de dados que podem enviar em cada emissão.

Este limite, largamente inferior às necessidades da maioria das aplicações, obriga a dividir a informação a enviar em partes mais pequenas normalmente designadas por pacotes.

PACOTES – INFORMAÇÃO DE CONTROLO

A cada pacote vai ser acrescentada informação de controlo antes dos dados (cabeçalho de controlo) e em muitos casos também após o fim dos dados (cauda).



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 correcto.

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).

T2

Comunicação em Rede

COMUTAÇÃO DE PACOTES COM CIRCUITOS VIRTUAIS

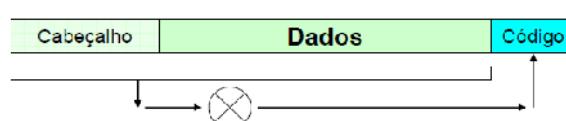
Numa rede de comutação de pacotes, mesmo que os pacotes pertençam todos à mesma transacção, são encaminhados pelos nós intermédios de uma forma independente uns dos outros, podendo até seguir caminhos diferentes e chegar ao destino desordenados.

Um 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.

PROCESSO:

1. O nó de origem pede à rede para criar um circuito virtual com o nó de destino, cujo endereço é indicado.
2. Os nós intermédios da rede definem o caminho e associam-lhe um identificador. A rede devolve o identificador do circuito virtual.
3. Na posse do identificador do circuito virtual o nó de origem pode começar a enviar pacotes. A diferença é que agora não coloca nos cabeçalhos o endereço do nó de destino, mas sim o identificador do circuito virtual.
4. Os nós intermédios encaminham os pacotes segundo o circuito virtual pré-estabelecido, por isso todos os pacotes seguem o mesmo caminho.

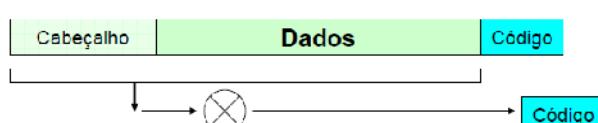
Os erros de transmissão podem ser detectados se o emissor acrescentar ao pacote um código de validação.



Este código é produzido por uma função apropriada que recebe o conteúdo do pacote e produz um código que representa esse conteúdo. O objectivo desta função é que qualquer pequena alteração nos dados de entrada leve à produção de um código diferente.

O código produzido no emissor é enviado juntamente com o pacote, isso dá ao receptor a possibilidade de repetir o processo e confrontar os códigos (se forem diferentes, houve erro de transmissão, se forem iguais, há uma grande probabilidade de não haver erro...)

Existem funções que produzem códigos um pouco mais extensos que são auto-correctores (FEC), apenas se justificam em casos especiais.



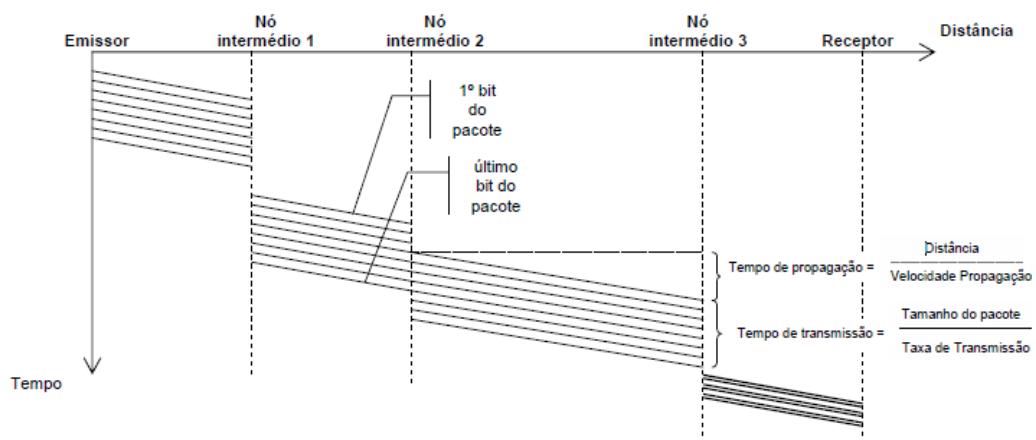
ATRASOS NA REDE:

Pode existir um atraso entre o instante em que um pacote de dados começa a ser emitido e o instante em que ele vai chegar ao destino. Existem 3 razões para isto acontecer:

- Os sinals não se propagam com velocidade infinita, logo existe um atraso de propagação proporcional à distância.
- A emissão/recepção de um pacote de dados não é um processo instantâneo, demora algum tempo, conhecido por tempo de transmissão. 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.
- Nos nós intermédios, os pacotes recebidos seguem uma política de fila de espera (FIFO) antes de serem processados. Em caso de tráfego elevado podem ficar retidos algum tempo num nó intermédio.

Retransmissão após armazenamento integral dos pacotes: store and forward

Retransmissão antes de ter terminado a recepção dos pacotes: cut-through



CONTROLO DE FLUXO:

O controlo de fluxo tem como objectivo regular o fluxo de dados entre emissor e receptor para evitar um overflow no receptor.

Na técnica conhecida por "stop & wait" o emissor tem de aguardar um sinal (ACK) do receptor antes de enviar o pacote seguinte

Inconvenientes: devido aos atrasos de propagação o processo de transmissão torna-se muito lento e de reduzida eficiência.

Para superar este problema criou-se uma variante conhecida por protocolo de janela deslizante.

Em vez de o emissor ter 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.

Depois de enviar W pacotes o emissor tem de aguardar, mas por cada ACK que chega, enviado pelo receptor, torna-se possível emitir mais um pacote.

Em condições ideais, ou seja com um n^o w de pacotes apropriado, não existem paragens de transmissão, o qual só é possível numa ligação full-duplex.

CONTROLO DE ERROS:

O objectivo do controlo de erros é corrigir erros detectados.

Embora se possa recorrer a mecanismos auto-correctores (FEC – Forward Error Correction), na maioria das situações usa-se a retransmissão dos dados (BEC – Backward Error Correction), também conhecida por (ARQ – Automatic Repeat Request), implementada juntamente com o controlo de fluxo.

Para o efeito passam a existir duas respostas possíveis por parte do receptor: ACK e NACK.

O NACK significa que foi detectado 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.

Arquitecturas e pilhas de protocolos

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, adoptou-se uma estratégia de módulos sucessivos, normalmente designados de camadas.

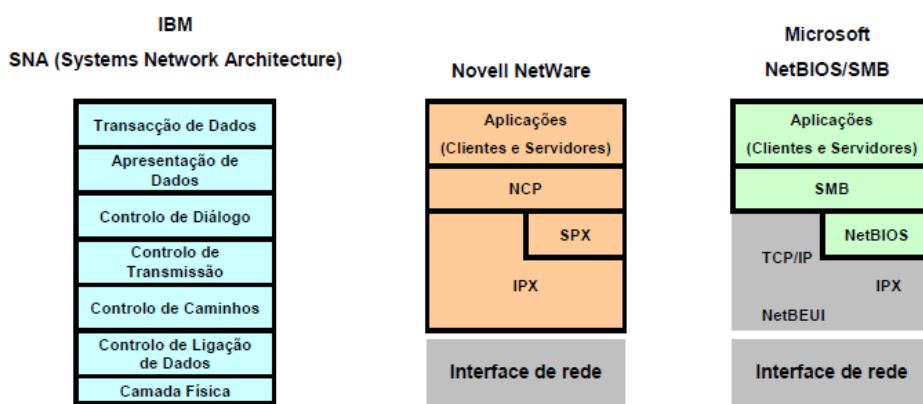
Cada camada resolve uma parte dos problemas.

Modelo ou Arquitectura – forma como as camadas estão organizadas e interagem entre si



ARQUITECTURAS PROPRIETÁRIAS:

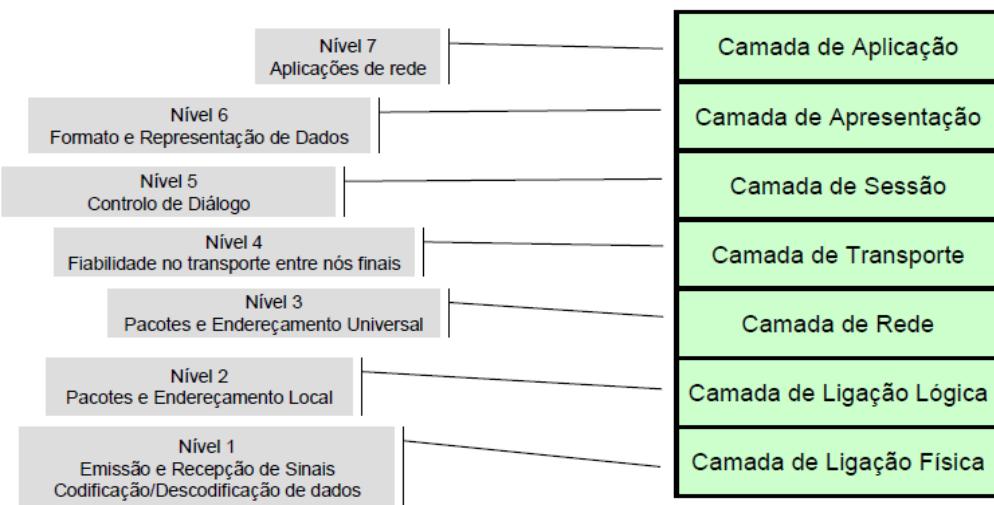
As redes de computadores começaram a surgir espontaneamente no início dos anos 70. Nessa altura cada fabricante desenvolveu o seu próprio sistema fechado, seguindo uma cultura de patentes para evitar que o mesmo fosse copiado por outros. Estas arquitecturas são conhecidas por arquitecturas proprietárias.



MODELO DE REFERÊNCIA OSI:

A ISO (International Organization for Standardization) desenvolveu um sistema de normalização das arquitecturas de rede:

Modelo OSI (Open Systems Interconnection)



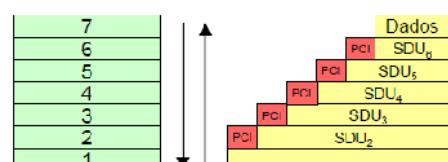
MODELO OSI - CAMADAS:

As camadas sucessivas da pilha interagem entre si segundo um modelo de prestação de serviço no sentido descendente através de pontos de acesso (SAP).

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).

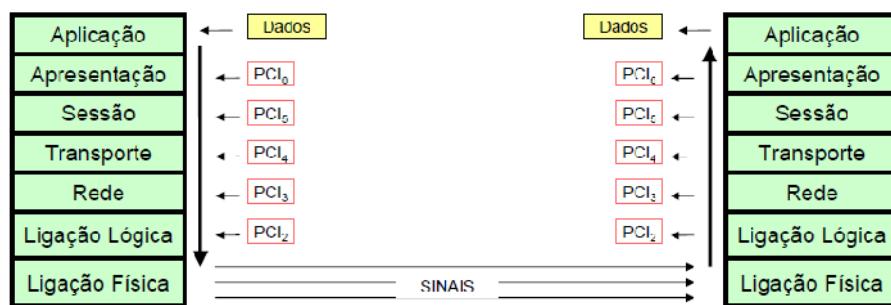
O PCI é adicionado aos dados (SDU – Service Data Unit) que vêm da camada superior.

Em cada camada, o conjunto PCI + SDU é designado por PDU (Protocol Data Unit).

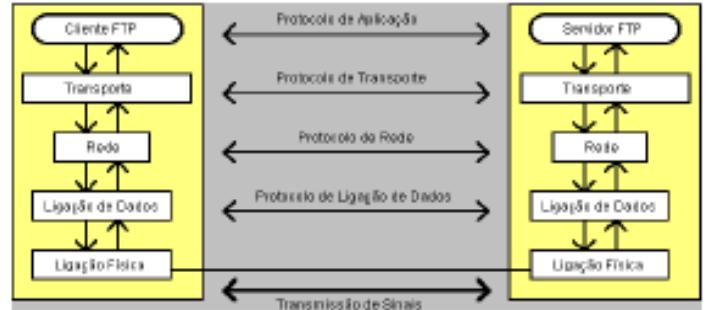


PROTOCOLOS:

As interacções directas 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.



Cada camada tem definido um determinado conjunto de regras, conhecido como protocolo que regulamentam as trocas de informação através do PCI dentro dos objectivos relacionados com as suas funcionalidades. Assim em cada camada está definido um protocolo.



Os grandes objectivos do OSI nunca foram atingidos, devido à enorme complexidade de desenvolver um modelo aberto capaz de contemplar todas as possibilidades.

Embora sob o ponto de vista de interligação de sistemas abertos tenha sido um fracasso, o modelo OSI 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.

Todas as evoluções posteriores dos vários sistemas de rede aproveitaram o modelo de referência OSI (MR-OSI).

ARQUITECTURA IEEE 802 (ISO 8802):

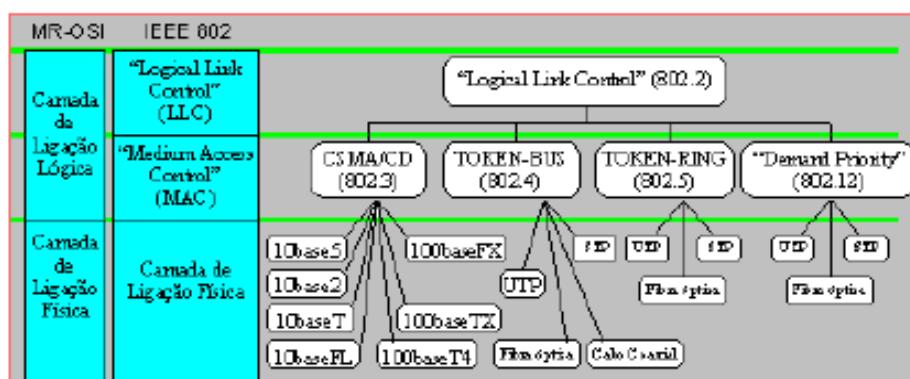
A maioria das tecnologias de rede local estão normalizadas pelo IEEE e pela ISO, estas tecnologias correspondem aos níveis 1 e 2 do MR-OSI.

As redes Ethernet têm, por exemplo o identificador IEEE 802.3 (ISO 8802-3).

Sempre que se produzem evoluções técnicas nestas normas são efectuados aditamentos identificados por letras minúsculas.

Já as redes Ethernet a 100 Mbps são definidas na norma 802.3u e as redes Ethernet a 1 Gbps na norma 802.3z.

A maioria das implementações de rede não usa a camada LLC (camada de ligação lógica) e interagem directamente com a camada MAC (camada de controlo de acesso).



CAMADA MAC:

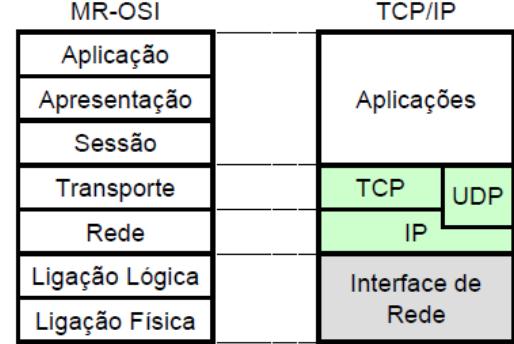
- Assegura a transferência de pacotes de dados a nível local
- Endereçamento de nó
- Detecção de erros

As redes locais evoluem rapidamente para a comutação e os mecanismos de controlo de acesso ao meio deixam de ser usados excepto nas redes sem fios IEEE 802.11.

ARQUITECTURA 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 (PROTOCOLOS 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.

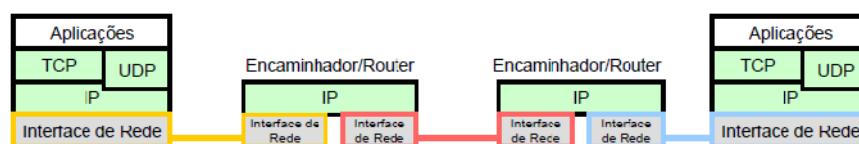
UDP – User Datagram Protocol – Protocolo de pacotes sem fiabilidade, apenas detecção de erros

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.

Isto permite a construção de uma rede global como é o caso da internet.



O endereçamento IP introduz o conceito de endereço de rede.

O objectivo é facilitar o encaminhamento pois passa a ser realizado rede a rede e não só a nível 2.

Por simples 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, sabendo deste modo a que aplicações em particular devem entregar os dados.

Etiquetas – Números de porto ou de serviço.

CAMADAS MULTI-PROTOCOLO:

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).

Este processo repete-se sucessivamente ao longo de uma pilha de protocolos.

MODELO CLIENTE-SERVIDOR:

A quase totalidade das comunicações em rede segue um modelo de diálogo muito simples conhecido por modelo cliente-servidor.

Primeiro:

- O cliente envia um pedido ao servidor, normalmente por acção do utilizador.
- O cliente tem de saber encontrar o servidor, ou seja, 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.
- O número de porto é fixo para cada tipo de servidor.

Segundo:

- Depois de receber o pedido, o servidor executa-o.

- O cliente está à espera de uma resposta.

Terceiro:

- Depois de processar o pedido o servidor responde ao cliente.
- Para saber o endereço do cliente (e número de porto) basta verificar a origem do pedido.

Estas trocas de informação seguem os formatos definidos no respectivo protocolo de aplicação.

Para certos serviços pode não ser necessária uma resposta, para outros este diálogo pode repetir-se sucessivamente.

CONCLUSÕES:

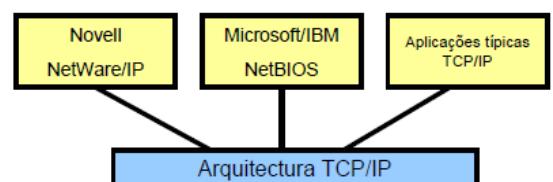
Embora nos tempos iniciais das redes de computadores existisse uma vasta variedade de arquitecturas proprietárias fechadas, a expansão da Internet com a sua arquitectura TCP/IP aberta veio alterar esse panorama.

Nessa altura muitas arquitecturas proprietárias tentaram um processo de abertura que lhe deu novo fôlego, mas a expansão da internet criou um processo irreversível em que o TCP/IP passou a ser obrigatório.

Neste contexto o desaparecimento total das outras arquitecturas é uma mera questão de tempo porque não é eficiente manter muitos protocolos num sistema.

O que se verifica é que as aplicações das arquitecturas proprietárias são modificadas para poderem funcionar sobre TCP/IP.

É necessário não esquecer que o TCP/IP se situa em apenas duas das sete camadas do MR-OSI. Essa é talvez uma das razões do seu sucesso.



T3

Tecnologias de rede local ETHERNET

REDES ETHERNET – CSMA/CD:

As redes ETHERNET (IEEE 802.3 / ISO 8802-3) foram originalmente desenvolvidas pela Xerox nos anos 70.

Actualmente exercem um claro domínio nas redes locais.

Originalmente o controlo de acesso ao meio (MAC) era um aspecto fundamental.

A técnica CSMA/CD usada neste tipo de rede não é propriamente ideal, tratando-se de um mecanismo que não evita as colisões e com baixa eficiência sob tráfego elevado.

Nas primeiras versões 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).

Redes ETHERNET – Domínio de Colisão:

A técnica CSMA/CD obriga a que as colisões de dados sejam detectadas 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 detecção 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.

REDES ETHERNET – PACOTES E ENDEREÇOS:

As redes ETHERNET mantiveram desde a origem o mesmo formato de pacote e endereçamento.

Isto permite compatibilizar totalmente as várias evoluções técnicas ocorridas.

Fibra óptica 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.

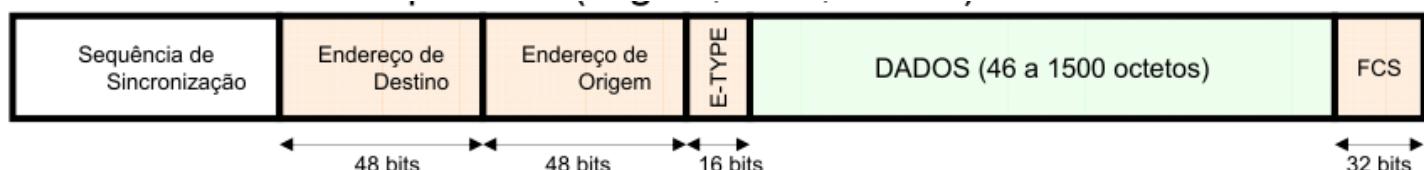
- Normalmente estes endereços são representados sob a forma de 6 octetos em notação hexadecimal, separados por dois pontos, por exemplo: 00:60:B0:3C:93:DB.
- Para garantir que os endereços são únicos, a cada fabricante de hardware é atribuída uma sequência fixa para os primeiros 24 bits.
- O endereço FF:FF:FF:FF:FF:FF é o endereço de "broadcast", um pacote com este endereço de destino chega a todos os nós da rede.

REDES ETHERNET – FORMATO DE PACOTE:

Os pacotes na camada de ligação lógica são habitualmente designados de tramas, frames ou quadros.

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 directa, mais simples, e implementa tudo o que é necessário, incluindo identificador para multiplexagem (E-TYPE) e código para detecção de erros (FCS – Frame Check Sequence).

REDES ETHERNET – 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 ópticas (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"

REDES ETHERNET – 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 desactivar 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).
- Analisar os endereços de destino e retransmitir as tramas apenas nas portas correctas (tabela MAC).

REDES ETHERNET COMUTADAS:

- Funcionamento em "full-duplex", sem colisões e sem controlo de acesso ao meio.
- Encaminhamentos com base no endereço de nó para que as tramas cheguem apenas ao nó de destino (gestão da tabela MAC).
- Eliminação dos domínios de colisão, evitando as restrições nas dimensões máximas da rede.
- Redução ao máximo de congestionamentos por acção dos comutadores.

Estes factores 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 microns. 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 várias 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.

Redes locais virtuais (VLAN)

REDES LOCAIS VIRTUAIS – IEEE802.1Q:

Uma rede local virtual (VLAN) é uma rede lógica definida sobre uma rede física.

Sob todos os pontos de vista uma VLAN deve apresentar ser uma rede totalmente independente e separada.

Há várias formas de criar redes locais virtuais.

Numa mesma rede física podem etiquetar-se as tramas fazendo com que cada etiqueta (TAG) corresponda a uma rede virtual diferente.

A norma IEEE 802.1Q define como se colocam etiquetas de VLAN nas tramas "Ethernet"

Colocam-se as etiquetas nas tramas "Ethernet II" usando o valor 0x8100 no campo E-TYPE e o valor original do campo E-TYPE 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.

REDES LOCAIS VIRTUAIS – COMUTADORES:

Os 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 etiquetas, apenas uma VLAN pode ser "untagged" em cada porta.

Além de VLAN baseadas em portas alguns comutadores também suportam VLAN baseadas em endereços MAC.

Redes locais sem fios (WLAN)

Representam uma evolução muito importante no sentido da acessibilidade e mobilidade por um lado e simplificação das instalações por outro lado.

Normalização mais conhecida: IEEE 802.11 e respectivos aditamentos

Actualmente já está disponível a norma 802.11n capaz de funcionar até 600 Mbps, usando múltiplos canais simultâneos (vários emissores/receptores).

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.

802.11 – MODULAÇÃO:

Embora a norma 802.11 original tivesse prevista uma implementação alternativa baseada em luz infravermelha, todas as evoluções posteriores usam exclusivamente rádio frequência na banda 2,4 GHz ou 5 GHz.

Tratando-se de sinais analógicos por natureza, a transmissão de dados recorre a técnicas de modulação.

As técnicas de modulação usadas actualmente são bastante complexas, usando vários sinais em simultâneo com combinações múltiplas PSK e ASK, desenvolvidas na fase final da evolução dos modems de linha telefónica, para as linhas DSL e para as redes de telemóvel.

A norma 802.11 e respectivos aditamentos definem várias técnicas alternativas de modulação, que conduzem a várias taxas de transmissão.

Cabe aos nós tentar as várias técnicas para obter a melhor taxa possível.

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.

802.11 – CSMA/CA:

O maior problema das redes locais sem fios é o facto de usarem um meio de transmissão partilhado que só pode ser usado por um nó de cada vez.

- Mesmo que 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

A recepção 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 detectar 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 detectado pelo emissor, para esse efeito sempre que um nó recebe uma trama válida envia ao emissor uma trama ACK.

IEEE 802.11 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 “RTSThreshold” definido em cada nó.

A técnica RTS/CTS consiste no envio pelo emissor de uma trama “Request to Send” ao receptor, eventualmente o receptor 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 infra-estrutura 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.

Este modo de funcionamento, sem AP, é conhecido por “ad-hoc”.

802.11 – MODO INFRA-ESTRUTURA:

O modo de infra-estrutura 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.

O modo de infra-estrutura tem grandes vantagens, uma delas é que a técnica RTS/CTS passa a ser muito eficaz porque apenas o AP pode responder aos RTS enviando o CTS.

802.11 – Células:

No modo de infra-estrutura 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 (BSSID).

Um conjunto de células pode fazer parte de uma mesma infra-estrutura 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".

802.11 – 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.

802.11 – TRAMAS:

O funcionamento das redes 802.11 é bastante 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 directa 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 directamente entre tramas 802.11 e tramas 802.3. Esta é uma missão do AP.

Foi realizado um grande esforço para manter a compatibilidade directa 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.

802.11 – 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.

Controlo de acesso:

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 nessa 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.

Tecnologias WAN: ATM/ISDN e DSL

WAN – REDES ATM:

A tecnologia ATM (Asynchronous Transfer Mode) está disponível há bastante tempo, mas a expansão da sua utilização não foi a esperada, uma vez que o seu objectivo inicial que era assegurar a interligação de nós finais nunca foi atingido.

Na altura em que a tecnologia ATM começava a ser usada em LAN, as redes Ethernet evoluíram para a comutação e aumentaram de taxa de transmissão, condenando o ATM em LAN devido ao seu custo mais elevado.

Actualmente assiste-se a uma continuação deste processo e as redes Ethernet começam a invadir o domínio ATM nas redes WAN com implementações 10 Gbps e 100 Gbps com alcances na casa das centenas de quilómetros.

REDES ATM – CANAIS E CAMINHOS VIRTUAIS:

- Comutação com circuitos virtuais
- Vantagens mais 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).
- Todos os canais virtuais com mesmo nó de origem e destino pertencem ao mesmo caminho virtual

REDES ATM – CÉLULAS:

Outra inovação é a utilização de PDUs de tamanho fixo e muito reduzido.

Estes PDUs 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).

GFC	VPI	VCI	P R C	HEC
VPI	VCI	P R C	HEC	
1°Byte	2°Byte	3°Byte	4°Byte	5°Byte

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 auto-corrector para erros de 1 bit e detecta erros de mais do que um bit.

REDES ATM – ISDN:

As principais características das redes ATM derivam de terem sido desenvolvidas com o objectivo de fornecerem serviços integrados tipo ISDN/RDIS (Integrated Services Digital Network / Rede Digital de Serviços Integrados).

A tecnologia a adoptar 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 detecção 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.

REDES 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.

Com a enorme expansão do protocolo IP este tipo de utilização das redes ATM tornou-se cada vez mais importante e foram necessários melhoramentos e eliminação de funcionalidades desnecessárias, deste esforço surgiu o AAL5.

PDU	Dados (0 a 65535 bytes)	Alinhamento (0 a 47 bytes)	CTL (2 bytes)	LEN (2 bytes)	CRC (4 bytes)
Camada AAL5:					

- 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” directamente 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 detectar 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.

O futuro das redes ATM não parece muito promissor e a expansão das redes Ethernet para o domínio WAN pode ser assinalar o início do seu fim.

WAN – CONECTIVIDADE IP:

A expansão da “INTERNET” provocou uma alteração radical no modo como as redes WAN são encaradas.

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).

Actualmente 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.).

WAN – 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 directamente 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.

WAN – OPERADORES DE TELECOMUNICAÇÕES:

Normalmente as comunicações de longa distância (WAN) apenas podem ser asseguradas por operadores autorizados.

- 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. Por exemplo, é ilegal enviar sinais RF 802.11 para o exterior dos edifícios.
- As ligações privadas por cabo não podem atravessar zonas públicas. Por exemplo, não é possível atravessar um arruamento sem recorrer a um operador oficial.
- A única alternativa privada possível é a utilização de feixes de luz laser quando existe linha de visão entre os pontos a ligar. Esta opção está livre de limitações legais, mas tem limitações técnicas, em especial relativamente às condições de propagação, estando normalmente limitada a distâncias inferiores a 3 Km.

WAN – 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.



Existem actualmente uma grande variedade de tecnologias propostas pelos operadores de telecomunicações autorizados.

As interfaces de ligação a essas tecnologias são muitas vezes fornecidas pelos próprios operadores (ex.: linhas dedicadas alugadas, analógicas ou digitais, ISDN (RDIS) e rede telefónica analógica, X.25 (TELEPAC), FRAME-RELAY e ATM, etc.)

WAN – LIGAÇÕES DEDICADAS:

Em grande parte a utilização actual de infra-estruturas WAN limita-se à interligação de encaminhador IP.

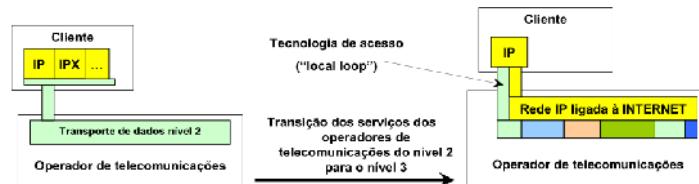
Embora a interligação de encaminhadores possa recorrer a redes de comutação tais como o ATM ou o FRAME-RELAY, simples ligações dedicadas são suficientes.

Mesmo as redes de comutação ou de outros tipos podem ser reduzidas a redes de dois nós, tornando-se ligações dedicadas.

Sobre a 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.

Estas tecnologias são actualmente usadas pelos operadores de telecomunicações para proporcionar aos clientes serviços de transporte IP em complemento à abordagem tradicional em que era proporcionado transporte de dados de nível 2, sobre o qual o cliente usava o protocolo IP ou outros.

WAN – TECNOLOGIAS DE ACESSO:



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.

Esta ligação é conhecida por "Local Loop".

Soluções além da fibra óptica (demasiado dispendiosa):

- 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 eléctrica (Power Line Communication).

WLL – "WIRELESS LOCAL LOOP":

O crescente desenvolvimento da tecnologia de rede sem fios começa a tornar viável a sua utilização para ligação do operador ao subscritor.

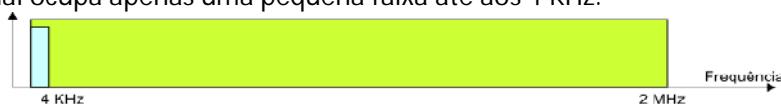
As normas 802.16, também conhecidas por "Wireless MAN" e WiMAX (World Wide Interoperability for Microwave Access) são mais apropriadas para este tipo de aplicação do que as 802.11, permitindo taxas de dados até 70 Mbps para distâncias inferiores a 2 Km e com alcances até à centena de Km para taxas de dados mais reduzidas (a 10 Km a taxa máxima é 10 Mbps).

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 procuram tirar partido da uma ligação telefónica já existente entre o cliente e a central telefónica. Essa ligação, constituída por um par de condutores de cobre, é capaz de transportar sinais analógicos até quase 2 MHz, contudo a utilização telefónica tradicional ocupa apenas uma pequena faixa até aos 4 KHz.



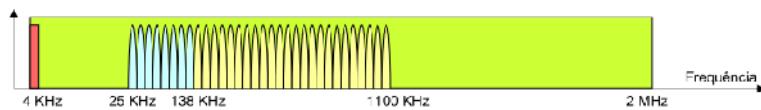
Embora a largura de banda disponível seja razoável, a qualidade das linhas é muito precária estando o sinal sujeito a muitas distorções e ruído. Por isso as técnicas DSL são obrigadas a dividir o espectro disponível em inúmeros canais com cerca de 4 KHz de largura cada.

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 desactivados. 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-RATE DIGITAL SUBSCRIBER LINE”:

O objectivo do VDSL 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.

O desempenho do VDSL2 degrada-se bastante 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 (análogicos) 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 (Data Over Cable Service Interface Specification) definem como os canais podem ser usados para transportar dados.

Na camada de ligação física o DOCSIS define 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 ELÉCTRICA:

Utilização de cablagens de alimentação eléctrica 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 eléctrica 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.

Em Portugal, recentemente a EDP abandonou a ideia de disponibilizar este tipo de tecnologia aos seus clientes.

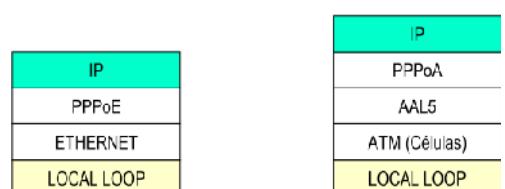
Pilha de protocolos TCP/IP

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.



A CAMADA IP:

A pilha de protocolos normalmente designada por TCP/IP exerce actualmente um domínio quase total nas comunicações por computador assegurando deste modo a inter-operacionalidade directa entre quase todos os tipos de equipamentos ligados por rede.

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

Principais características do protocolo IP:

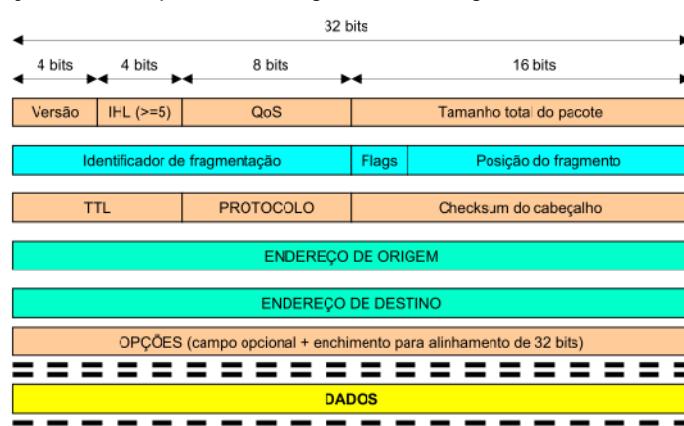
- Apresenta apenas as funcionalidades estritamente necessárias.
- Definição de um formato de dados (Pacote IP).
- Definição de endereços de rede, e dentro de cada rede, endereços de nó.
- Tempo de vida dos pacotes. Identificador para multiplexagem de dados.
- Fragmentação e reagrupamento.
- Detecção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

DATAGRAMAS IP:

Um dos aspectos importantes de um protocolo com o objectivo 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:

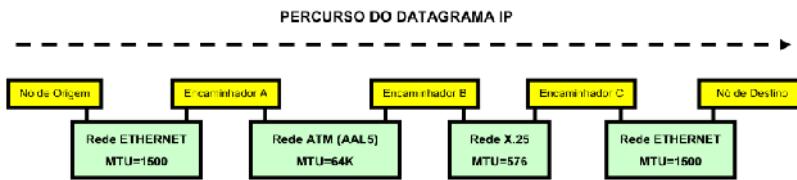


TAMANHO DOS DATAGRAMAS IP:

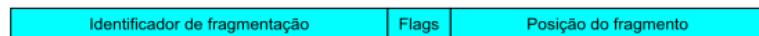
Embora os “datagramas” IP possam ter até 64 Kbytes de comprimento, eles têm de ser transportados com recurso a tecnologias de nível 2 que podem não suportar esse volume de dados em cada PDU.

O volume de dados que cada PDU de nível 2 pode transportar é conhecido por MTU (“Maximum Transmission Unit”), por exemplo numa trama ETHERNET-II o MTU é 1500 octetos.

O problema torna-se complicado porque o percurso de um determinado datagrama pode envolver muitos tipos diferentes de tecnologias de nível 2 com diferentes valores de MTU.



DATAGRAMAS IP – FRAGMENTAÇÃO:



A solução teórica mais completa é a fragmentação que é directamente 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.

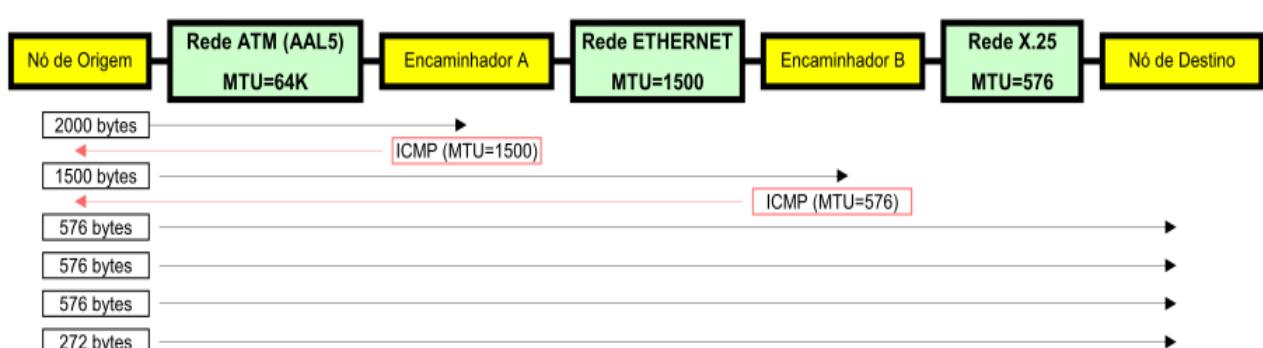
PMTUD (PATH MAXIMUM TRANSMISSION UNIT DISCOVERY):

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.



Pilha de protocolos TCP/IP

CAMADA IP - PILHA DE PROTOCOLOS TCP/IP

Principais características do protocolo IP:

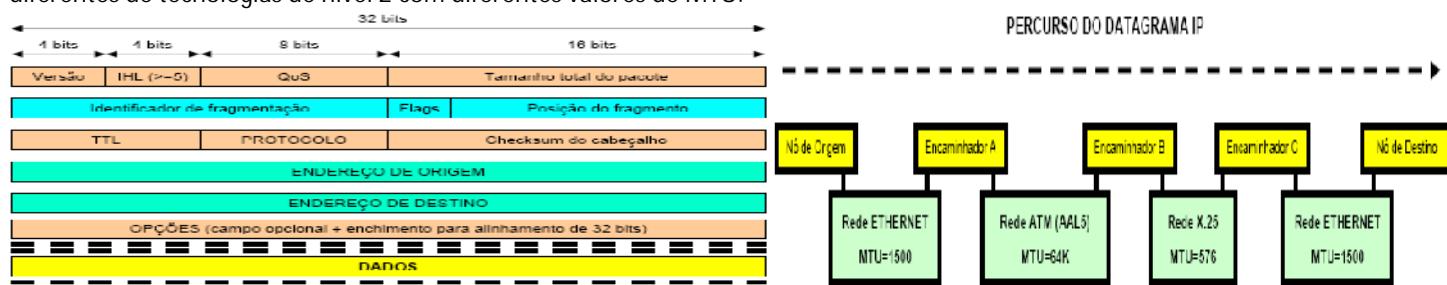
- Apresenta apenas as funcionalidades estritamente necessárias;
- Definição de um formato de dados (Pacote IP);
- Definição de endereços de rede e dentro de cada rede, endereços de nó;
- Tempo de vida dos pacotes. O "tempo de vida" (TTL) dos pacotes IP permite evitar consequências graves se ocorrerem circuitos fechados no encaminhamento IP:

O tempo de vida dos pacotes IP é um número inteiro de 8 bits que funciona como contador decrescente de saltos. No nó de origem é atribuído um dado valor não nulo, sempre que um encaminhador faz a transposição do pacote de uma rede para outra, decrementa uma unidade no tempo de vida. Se num encaminhador o tempo de vida de um pacote chega a zero, então o pacote é destruído e é gerada uma mensagem de erro ICMP. O tempo de vida dos pacotes IP tem várias utilidades, uma delas é que garante que se existir um erro grave nas tabelas de encaminhamento e estas definam um percurso circular fechado, os pacotes que entram nesse percurso não vão circular indefinidamente, assim que o tempo de vida chega a zero são eliminados. Evita-se assim o colapso da rede nestas situações de erro. Identificador para multiplexagem de dados;

- Fragmentação e reagrupamento;
- Detecção de erros apenas no cabeçalho. Parâmetros QoS.

DATAGRAMAS IP

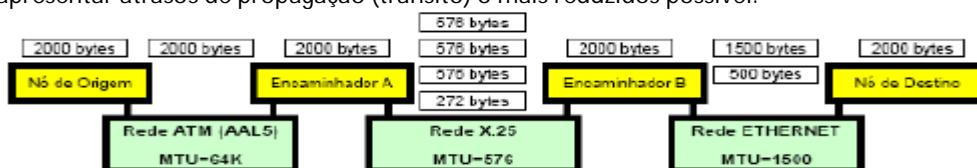
Um dos aspectos importantes de um protocolo é a definição de um formato para os pacotes que transportam os dados ("datagrama"). Uma vez definido o IP dificilmente poderá ser alterado. Os datagramas IP podem ter 64 Kbytes de comprimento, seguem uma organização (alinhamento) de 32 bits, mas embora os "datagramas" IP possam ter até 64 Kbytes de comprimento, eles têm de ser transportados com recurso a tecnologias de nível 2 que podem não suportar esse volume de dados em cada PDU. O volume de dados que cada PDU de nível 2 pode transportar é conhecido por MTU ("Maximum Transmission Unit"), por exemplo numa trama ETHERNET-II o MTU é 1500 octetos. O problema torna-se complicado porque o percurso de um determinado datagrama pode envolver muitos tipos diferentes de tecnologias de nível 2 com diferentes valores de MTU.



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

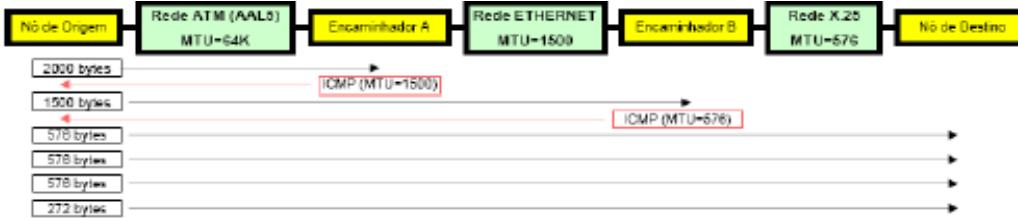
FRAGMENTAÇÃO: (é a solução teórica mais completa)

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 neste modo os valores de MTU são sempre aproveitados ao máximo (menor "overhead"). Na prática contudo, a fragmentação sobrecarrega bastante os nós de rede, em especial os nós que recebem os fragmentos e têm de fazer o reagrupamento. Este problema é especialmente desfavorável nos encaminhadores que devem apresentar atrasos de propagação (trânsito) o mais reduzidos possível.



PMTUD (PATH MAXIMUM TRANSMISSION UNIT DISCOVERY) – É UMA ALTERNATIVA ÀS FALHAS DA FRAGMENTAÇÃO:

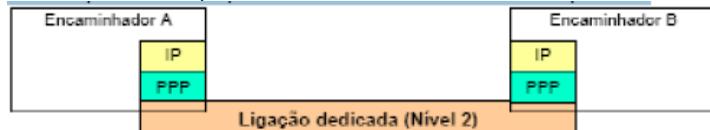
A solução é simples: o segundo bit do campo "flags" do cabeçalho IP é colocado com o valor um, serve para indicar que o datagrama não pode ser fragmentado (DF). A técnica PMTUD tem como objectivo determinar o tamanho máximo que um pacote pode ter para atingir um determinado destino (percorrer um dado caminho) sem necessidade de ser fragmentado. 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. Quando o nó emissor recebe esta mensagem percebe que o pacote se perdeu por ser demasiado grande, vai então voltar a enviar os dados, mas em pacotes mais pequenos. A técnica PMTUD começa por usar pacotes grandes, sendo progressivamente reduzido até que deixe de ocorrer o erro referido.



TRANSPORTE DE DATAGRAMAS IP:

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

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



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



Protocolos ARP; UDP; BOOTP e DHCP

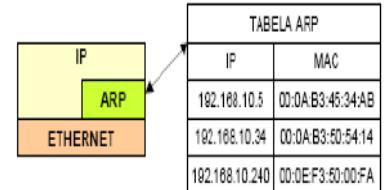
ARP – ADDRESS RESOLUTION PROTOCOL:

O protocolo ARP tem como objectivo 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.

ARP – Address Resolution Protocol A camada ARP gera uma tabela de equivalências entre endereços IP dos nós IP da rede local e os endereços MAC respectivos.

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 respectivo endereço MAC de destino para colocar na trama (encapsulamento no nível 2). Quando é necessário um endereço que não se encontra na tabela, então é usado o protocolo ARP propriamente dito para determinar esse endereço que será depois adicionado à tabela.



PROTOCOLO ARP:

O protocolo ARP é usado quando é necessário um endereço de nível 2 (MAC) correspondente a um dado endereço de nível 3 (IP), e este não se encontra na tabela ARP. A camada ARP começa por enviar um PEDIDO ARP que contém o endereço IP pretendido, este pedido é enviado em BROADCAST, por isso todos os nós da rede recebem. Todas as camadas ARP estão à escuta destes pedidos, quando os recebem comparam o seu próprio endereço IP com o que consta no pedido, se forem iguais enviam a RESPOSTA ARP que contém o endereço MAC pretendido. Ao receber a resposta, o nó que desencadeou o processo, adiciona os novos dados à sua tabela ARP, estes novos dados serão válidos apenas durante algum tempo.

O protocolo ARP é necessário para o IPv4, mas deixou de ser para o IPv6. No IPv4, os endereços de 32 bits não têm qualquer relação directa 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:

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

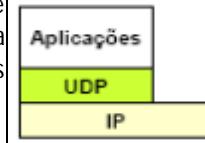
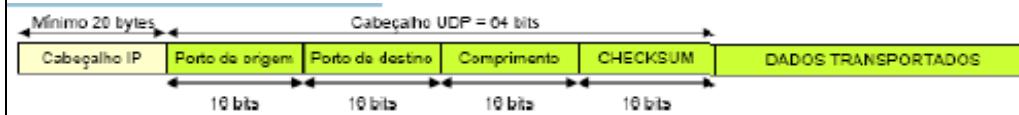
-Um endereço IP (32 bits) é constituído por uma N bits de rede (máscara de rede) seguidos de (32-N) bits de nó.

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

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

PROTOCOLO UDP ("USER DATAGRAM PROTOCOL")

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

O número do porto local 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.

PROTOCOLO BOOTP

O protocolo BOOTP ("BOOT strap Protocol") é um exemplo de um serviço UDP, no caso, o servidor recebe datagramas UDP no porto 67. Como tal os clientes BOOTP sabem que devem enviar os seus pedidos para o número de porto 67. O objectivo deste protocolo é obter informação para configuração IP do nó. Assim sendo, o cliente nada conhece sobre a rede a que está ligado, nem sequer o seu próprio endereço IP.

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

BOOTP DINÂMICO

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

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

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

Protocolo ICMP e protocolo TCP

PROTOCOLO DHCP ("DYNAMIC Host CONFIGURATION PROTOCOL"):

O protocolo DHCP não é mais do que uma extensão do protocolo BOOTP, ao qual é adicionado o conceito de aluguer do endereço (lease) por determinado tempo.

Quando um cliente recebe um endereço via DHCP tem de controlar o tempo de aluguer e, se pretender continuar a usar o endereço, tem de renovar o pedido antes que o aluguer se esgote.

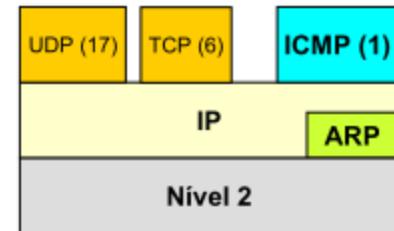
Uma vez esgotado o tempo de aluguer, o servidor DHCP tem a liberdade de fornecer esse endereço IP a outro cliente.

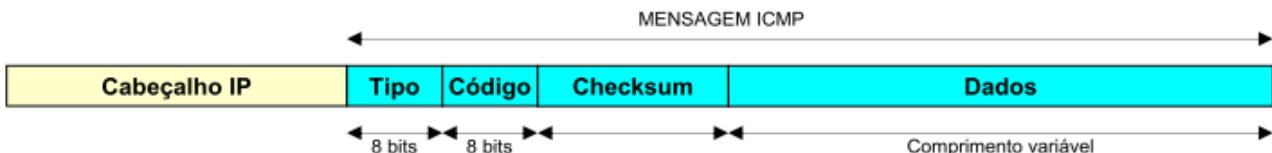
Na prática os servidores DHCP tentam manter sempre o IP de cada cliente, apenas quando se esgota a gama disponível é que os endereços são reutilizados para clientes diferentes.

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

PROTOCOLO ICMP:

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





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

Por exemplo a mensagem ICMP tipo 0 significa "echo reply", é usada em conjunto com o tipo 8 ("echo request") e servem para teste de conectividade sendo usadas pelo bem conhecido comando "ping".

Em ambos os casos o código deverá ser zero, os dados são constituídos por um identificador e um número de sequência, ambos com 16 bits, seguidos de um padrão de bits que o requisitante pode colocar e no pedido e será devolvido na resposta.

MENSAGEM ICMP "DESTINO INATINGÍVEL":

A mensagem ICMP de tipo 3 ("Destination Unreachable") é enviada ao nó de origem quando um Datagrama IP não pode ser entregue no endereço de destino pretendido.



O campo "Código" é usado para indicar a razão dessa falha. O campo "MTU seguinte" é usado apenas para o código 4. Em qualquer caso é devolvida na mensagem ICMP uma cópia da parte inicial do DATAGRAMA IP que causou o problema.

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

OUTRAS MENSAGENS ICMP IMPORTANTES:

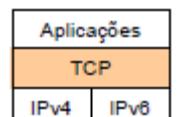
Mensagem tipo 4 ("Source Quench") – aviso gerado por um nó saturado, pede que o fluxo de dados seja reduzido.

Mensagem tipo 5 ("Redirect") – indica ao nó de origem que está a 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 correcto. O nó de origem pode usar esta informação para alterar a sua tabela de encaminhamento.

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

PROTOCOLO TCP:

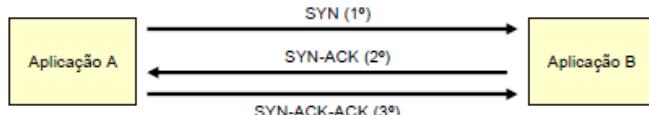
Ao contrário do protocolo UDP que é muito simples, o "Transmission Control Protocol" (TCP) é comparativamente bastante complexo devido ao conjunto de funcionalidades disponibilizadas.



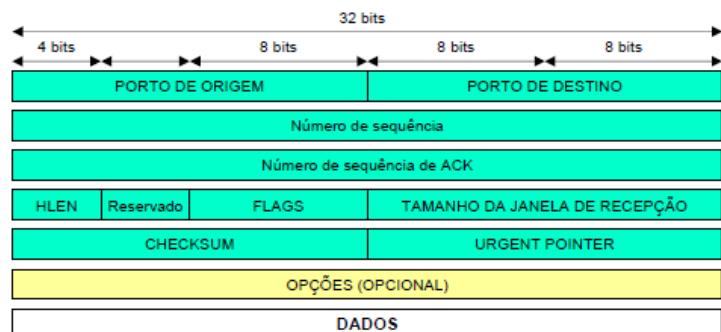
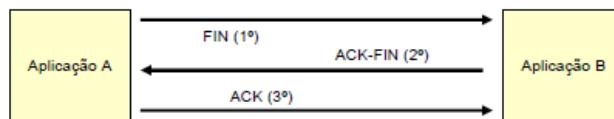
Usando um simples e não fiável serviço de DATAGRAMAS do IPv4 ou IPv6, o TCP proporciona um serviço de transferência de dados em fluxo através de uma ligação lógica.

A operação do TCP utiliza controlo de erros e de fluxo baseado a ARO continuo que garante a total ausência de erros.

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



Finalização de ligação – qualquer 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 detectar 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.

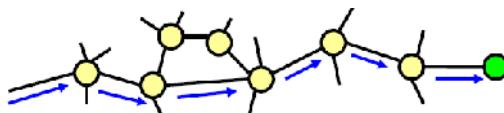
T6

Encaminhamento IPv4

NÓS INTERMÉDIOS:

Os nós intermédios assumem o papel principal em qualquer tipo de rede de comutação.

As redes de comutação caracterizam-se pela retransmissão da informação entre nós sucessivos (nós intermédios) até chegar ao destino pretendido.



No nível 2 do modelo OSI são habitualmente designados de comutadores ou “switches”.

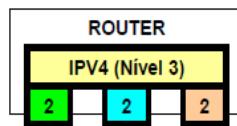
Os “datagramas” do protocolo IPv4 também chegam ao destino graças aos nós intermédios, quando operam no nível de rede os nós intermédios são normalmente designados encaminhadores ou “routers” ou por “gateway”.

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".

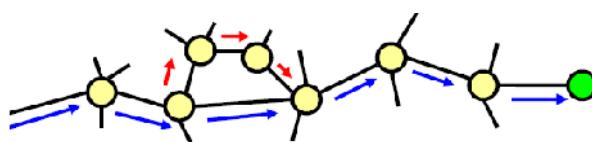
O encaminhador vizinho para onde o "datagrama" vai ser enviado é conhecido como próximo-nó ("next-hop").

Encaminhamento estático e encaminhamento dinâmico.

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".

Esta decisão e a aplicação da mesma são designadas 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 exactamente 8 encaminhadores vizinhos, ou seja 8 "next-hop" possíveis.

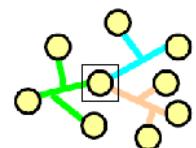
TABELA DE ENCAMINHAMENTO ("ROUTING TABLE"):

Uma das principais razões do actual sucesso do protocolo IP é o facto de ser assegurado que um DATAGRAMA IP emitido em qualquer ponto da INTERNET será correctamente encaminhado até ao endereço de destino.

As decisões de encaminhamento, ou seja a escolha do próximo nó ("next-hop") apropriado, são realizadas usando uma tabela conhecida por tabela de encaminhamento (ou "routing table").

Cada linha da tabela de encaminhamento possui dois elementos fundamentais:

DESTINO – identificação do destino, ou seja um endereço IP associado a uma máscara de rede.
Normalmente representa uma rede remota

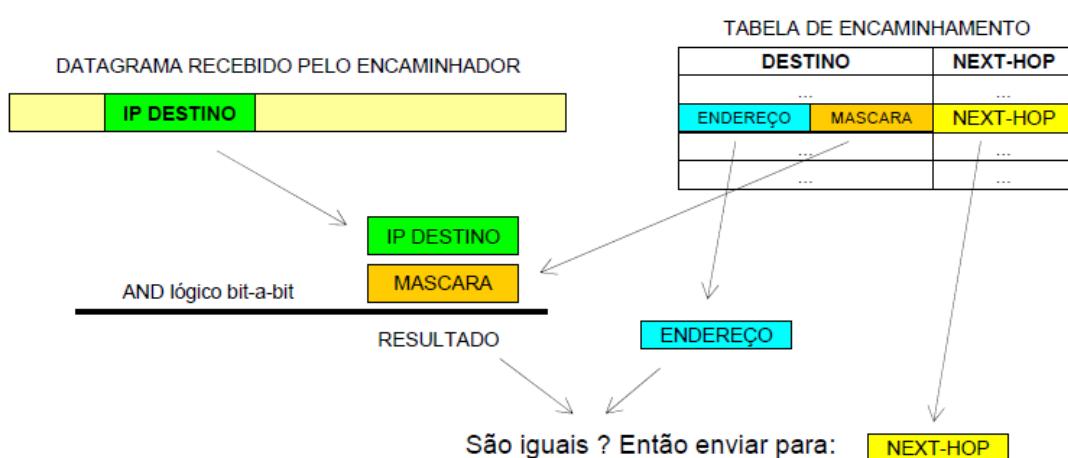


PRÓXIMO NÓ – próximo encaminhador ("next-hop") para onde devem ser enviados os dados quando se pretende que estes cheguem ao DESTINO. Trata-se do endereço IP do nó seguinte do percurso ou rota dos dados, é sempre um endereço IP de um nó vizinho.

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".

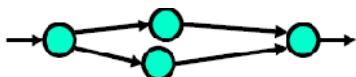
Todos os endereços na coluna "PRÓXIMO NÓ" pertencem obrigatoriamente a redes locais directamente ligadas por um interface.

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.

Em redes mais complexas, como por exemplo as redes de trânsito que interligam zonas da INTERNET, é vulgar e 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.

É necessário definir novos critérios para tomar uma decisão de escolha do melhor caminho.

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
194.121.12.0/24	192.168.10.2
0.0.0.0/0	192.168.10.200

Para permitir uma decisão no sentido de que seja usado o melhor caminho, a cada linha da tabela de encaminhamento são adicionados outros parâmetros que deverão quantificar o custo da sua utilização (caminho até ao destino).

Quando existem duas linhas com o mesmo DESTINO será usada a de menor custo.

ENCAMINHAMENTO DINÂMICO:

A construção das tabelas de encaminhamento pode ser realizada manualmente (encaminhamento estático), contudo para redes extensas e complexas essa tarefa torna-se quase impossível e tem de ser automatizada.

As vantagens da existência de caminhos alternativos (redundância e distribuição do tráfego) apenas fazem sentido se as tabelas de encaminhamento mantiverem uma representação actualizada 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. Estes protocolos envolvem a troca de informação de controlo entre os nós encaminhadores.

Existem dois tipos de protocolo de encaminhamento dominantes:

DISTANCE-VECTOR - cada nó divulga junto dos nós vizinhos a sua tabela de encaminhamento.

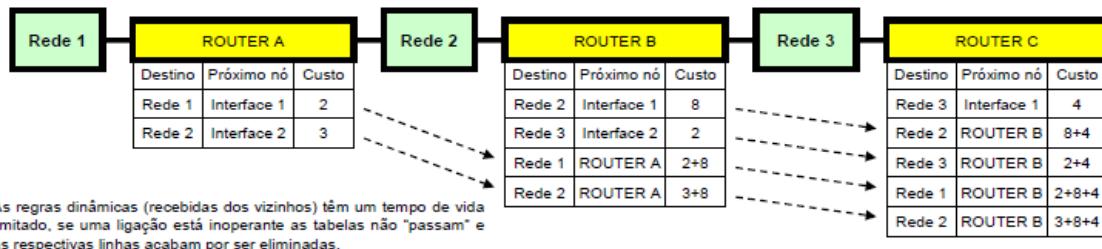
LINK-STATE - cada nó divulga junto dos nós vizinhos a lista de nós aos quais está directamente ligado (vizinhos).

ALGORITMOS DISTANCE-VECTOR:

Este algoritmo envolve a divulgação periódica entre nós vizinhos das respectivas tabelas de encaminhamento. Esta divulgação pode recorrer a uma simples emissão em BROADCAST.

Inicialmente cada nó possui uma tabela de encaminhamento em que constam apenas as redes a que está directamente ligado (interfaces de rede). A cada uma das interfaces está associado um custo (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:

Nos algoritmos LINK-STATE as tabelas de encaminhamento são totalmente construídas em cada encaminhador. Para isso cada encaminhador tem de conhecer toda a rede.

Cada nó de encaminhamento tem de conhecer os seus vizinhos directos e controlar permanentemente o estado da ligação com cada um deles. Na posse dessa lista divulga-a por todos os encaminhadores da rede. Sempre que detectar uma alteração nos estados das ligação reconstrói a lista e divulga-a novamente.

Cada nó de encaminhamento recebe portanto mensagens de todos os outros nós contendo a identificação do nó de origem e dos respectivos 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.

Na posse do conhecimento de todas as interligações da rede, o nó pode agora autonomamente determinar a tabela de encaminhamento.

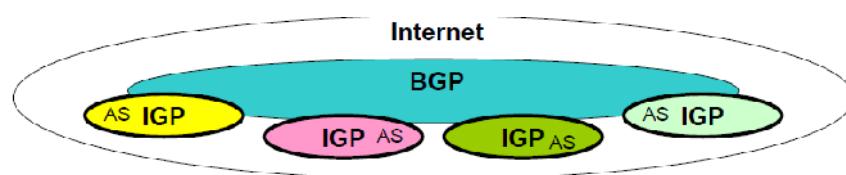
Trata-se de um algoritmo simples de determinação dos caminhos mais curtos (menos nós intermédios).

Protocolos de encaminhamento: RIP, RIPv2, EIGRP e OSPF

SISTEMAS AUTÓNOMOS (AS) – PROTOCOLOS IGP E BGP:

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, por 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 actualmente definido pelo BGP (Border Gateway Protocol).

Para interagir com o BGP, cada AS tem atribuído um ASN ("Autonomous System Number") único que o identifica.

Os ASN são atribuídos pelo IANA ("Internet Assigned Numbers Authority"), também responsável pela atribuição de endereços IPv4/IPv6 e nomes de domínio DNS.

O IANA delega a administração de partes do sistema aos RIR ("Regional Internet Registries").

PROTOCOLOS IGP:

Os protocolos de encaminhamento usados no interior dos AS são designados "Interior Gateway Protocols".

A utilização de um ASN único, fornecido pelo IANA apenas é importante se o AS estiver em contacto com o BGP, caso contrário podem usar-se por exemplo os valores 64512 a 65534 que foram reservados para uso privado.

Principais Protocolos IGP:

- DISTANCE-VECTOR: RIP (Routing Information Protocol) e o IGRP (Interior Gateway Routing Protocol).
- LINK-STATE: OSPF (Open Shortest Path First) e o IS-IS (Intermediate System to Intermediate System).
- Protocolos mistos ou híbridos: o protocolo proprietário da CISCO "Enhanced Interior Gateway Routing Protocol" (EIGRP) possui características de ambas as categorias.

PROTOCOLO RIPv1:

Cada "router" divulga a sua tabela de encaminhamento por "broadcast" (UDP) nas redes vizinhas.

A divulgação ocorre de 30 em 30 segundos, ou em períodos variáveis ligeiramente superiores.

Quando uma linha da tabela de encaminhamento não é refrescada durante 180 segundos, é marcada como não atingível (hops=16).

- Não divulga máscaras de rede, 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 ou 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.

PROTOCOLO RIPv2:

A versão 2 do protocolo RIP 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.

- Cada "router" divulga a sua tabela de encaminhamento usando "multicast" em lugar de "broadcast".
- As tabelas de encaminhamento divulgadas incluem agora as máscaras 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.

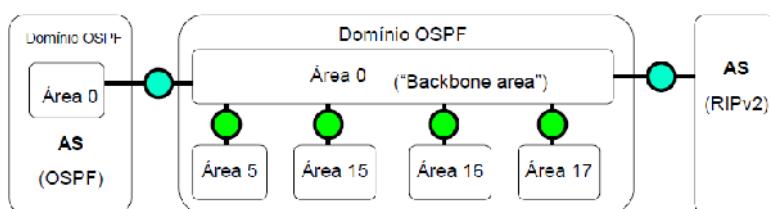
PROTOCOLO OSPF:

Trata-se de um protocolo "link-state", 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 transacções OSPF usam directamente o protocolo IP, não recorrem ao UDP.
- Cada "router" utiliza a informação recebida para construir autonomamente a tabela de encaminhamento. A informação transaccionada transporta máscaras de rede, por isso o OSPF suporta CIDR.
- Suporta autenticação entre "routers", baseada em MD5 e HMAC-SHA.
- A métrica pode ser calculada de diversas formas, mas normalmente deriva da taxa de transmissão correspondente à ligação que dá acesso ao caminho.

ÁREAS OSPF:

O OSPF é um IGP, por isso não suporta o conceito de AS, sob o ponto de vista do OSPF 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 independenteumas das outras, contudo agora o

protocolo é o mesmo em todas elas.



TIPOS DE ROUTERS OSPF

- | |
|--|
| ASBR ("Autonomous System Boundary Router") |
| ABR ("Area Border Router") |

PROTOCOLO EIGRP:

O protocolo EIGRP é um melhoramento do protocolo IGRP desenvolvido pela Cisco para fazer face aos problemas do RIP. Na sua essência é um protocolo "distance-vector", mas tem algumas características dos protocolos "link-state".

Cada "router" mantém uma lista de "routers" vizinhos usando o envio periódico, em "unicast/broadcast/multicast", da mensagem "hello".

Vantagens:

- 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.

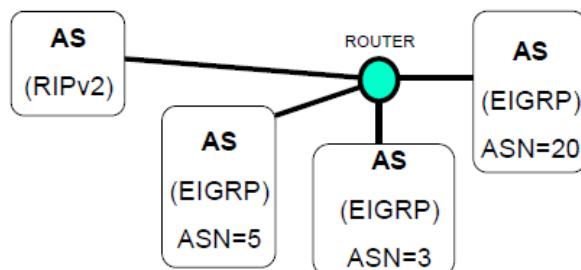
Sistemas autónomos e redistribuição de rotas



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 directa com os ASN do BGP.

ENCAMINHAMENTO ENTRE SISTEMAS AUTÓNOMOS:

O objectivo da definição de sistemas autónomos (ou áreas OSPF) é isolar partes das redes sob o ponto de vista da gestão das tabelas de encaminhamento.

Vantagens:

- 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.

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.

Uma 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.

T7

IPv6 e ICMPv6

INTERNET PROTOCOL V6:

Nos anos 90 a expansão da INTERNET atingiu valores inicialmente impensáveis que conduziram a uma situação de esgotamento dos endereços IPv4 disponíveis. Foram tomadas várias medidas para permitir um melhor aproveitamento do espaço de endereçamento de 32 bits do IPv4:

- 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 máscaras 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 objectivo 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 actualmente conhecido por IPv6.

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.

Aspectos melhorados do protocolo IP na versão 6 (IPv6):

- 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.

ENDEREÇAMENTO IPv6 - REPRESENTAÇÃO:

Os endereços IPv6 são representados sob a forma de texto através de uma sequência de 8 conjuntos de 16 bits em notação hexadecimal, separados por “::”.

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

A grande extensão dos endereços torna a sua representação pouco cómoda e pouco legível.

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. Por exemplo:

“0000:3278:0A04:0005:0000:0000:0034” = “: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, por exemplo (respectivamente):

::193.136.62.9 e ::FFFF:193.136.62.9

IPv6 – TIPOS DE ENDEREÇO:

Tal como acontecia no IPv4 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. Por exemplo

2001:0DB8:2B00::/40 (representa uma rede com máscara de 40 bits)

1º nó da rede: 2001:0DB8:2B00::1 (2001:0DB8:2B00:0000:0000:0000:0001)

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:

UNICAST – identifica um nó único numa dada rede.

MULTICAST – identificam conjuntos de nós, os dados têm de ser entregues em todos eles.

ANYCAST – identificam conjuntos de nós, os dados são entregues em apenas um deles.

IPv6 – ENDEREÇOS MULTICAST:

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: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).

IPv6 – ENDEREÇOS ANYCAST:

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.

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.

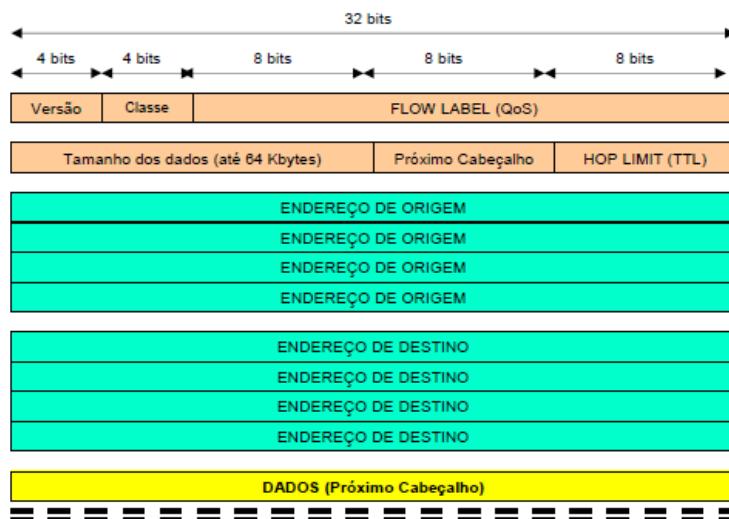
Alguns endereços especiais são:

- :: (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 interactiva 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.



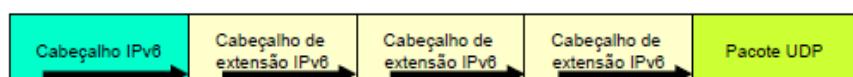
PACOTES IPv6 – CABEÇALHOS DE EXTENSÃO:

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.



Além dos valores normalizados para os protocolos de transporte, tais como 6 para TCP e 17 para UDP, o campo "NEXT HEADER" suporta valores especiais que identificam cabeçalhos de extensão:

0	HOP-BY-HOP OPTIONS – opções que necessitam de ser processadas nos nós intermédios, EX.: JUMBO PAYLOAD
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)

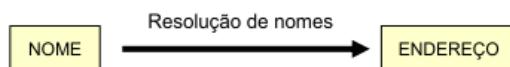
As diferenças estão nos tipos de mensagens:

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.

T8

Resolução de nomes DNS e WINS/NetBIOS

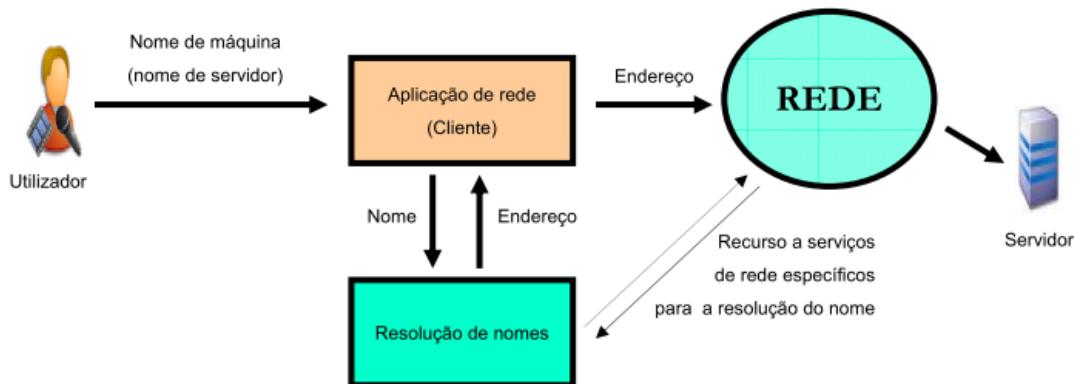
RESOLUÇÃO DE NOMES:



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.

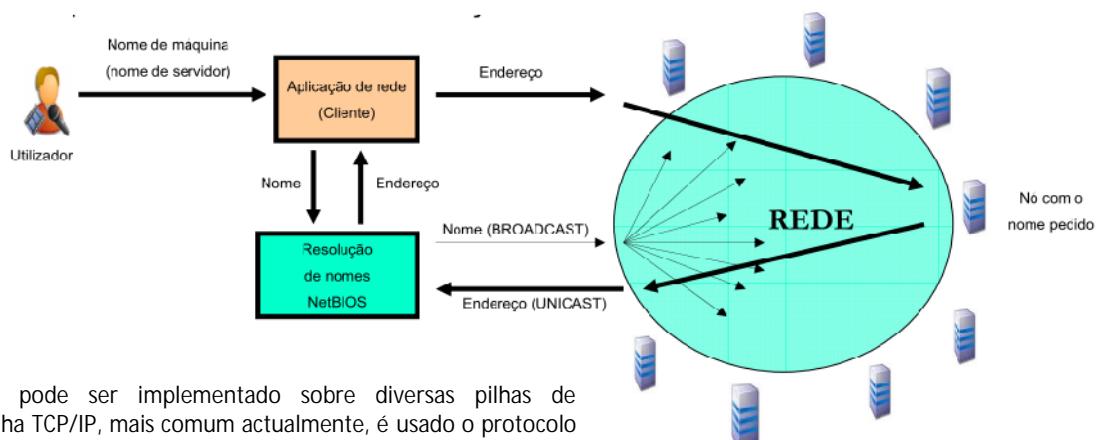
A resolução de nomes tem como objectivo estabelecer uma ligação entre os utilizadores e os endereços de rede de tal forma que os primeiros não tenham de interagir directamente com os segundos.

Apoiados neste serviço os utilizadores podem usar nomes de máquinas bastante mais representativos para o ser humano.



RESOLUÇÃO DE NOMES - NETBIOS:

De entre vários, um dos sistemas de resolução de nomes que teve bastante sucesso foi o sistema NetBIOS. Integrado em redes PEER-TO-PEER, em que não existem servidores, o NetBIOS 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.



Este mecanismo pode ser implementado sobre diversas pilhas de protocolos. Na pilha TCP/IP, mais comum actualmente, é usado o protocolo UDP e o serviço é acessível no porto 137.

NETBIOS – REGISTRO DE NOMES:

O sistema de nomes NetBIOS foi concebido para redes sem servidores, cada nó tem a missão de responder aos pedidos de rede relativos ao seu nome.

Mesmo sem servidores há necessidade de ter algumas garantias de que os nomes são únicos, para esse efeito quando os nós arrancam enviam para a rede em BROADCAST 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 NETBIOS (WINDOWS):

Os nomes NetBIOS podem ser únicos ou de grupo, os nomes de grupo são registados pela mesma forma que os nomes únicos, mas podem estar registados por 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

WINS – SERVIDOR DE NOMES NETBIOS:

O protocolo de resolução de nomes NetBIOS baseado em BROADCAST apresenta vários problemas:

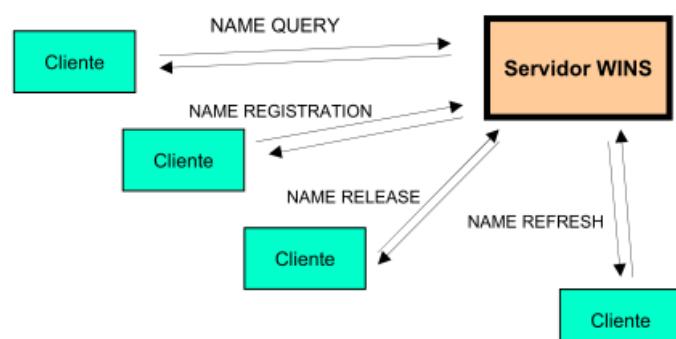
- Insegurança e falta de fiabilidade (registo por omissão; ambiente de confiança geral entre nós)
- Limitação de propagação do BROADCAST (sistema limitado a uma única rede)
- O tráfego em BROADCAST é penalizador para o desempenho das redes pois anula a segmentação de nível 2.

Para resolver estes problemas a Microsoft desenvolveu o serviço WINS ("Windows Internet Name Service"), contendo muito poucas alterações ao protocolo original de resolução de nomes, a principal diferença é que deixa de ser usado o BROADCAST.

Todos os pedidos são enviados em UNICAST para o servidor WINS, isso significa o servidor WINS pode encontrar-se numa rede remota e servir clientes de várias redes.

Além desta vantagem a transição para um modelo cliente/servidor leva a um aumento geral da segurança e fiabilidade. Agora todos os pedidos têm resposta.

Os registos de nomes estão associados a um tempo de vida, esgotado esse tempo são eliminados.



FUTURO DA RESOLUÇÃO DE NOMES NETBIOS:

O responsável por trazer até aos dias de hoje o NetBIOS é a Microsoft e os seus sistemas operativos. 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 perspectiva 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 registo distribuídas por uma grande área (sem nunca haver registo repetidos).

Outra lição que a história das redes e protocolos ensinou é que a coexistência de duas implementações paralelas que fazem o mesmo não dura muito tempo. 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".

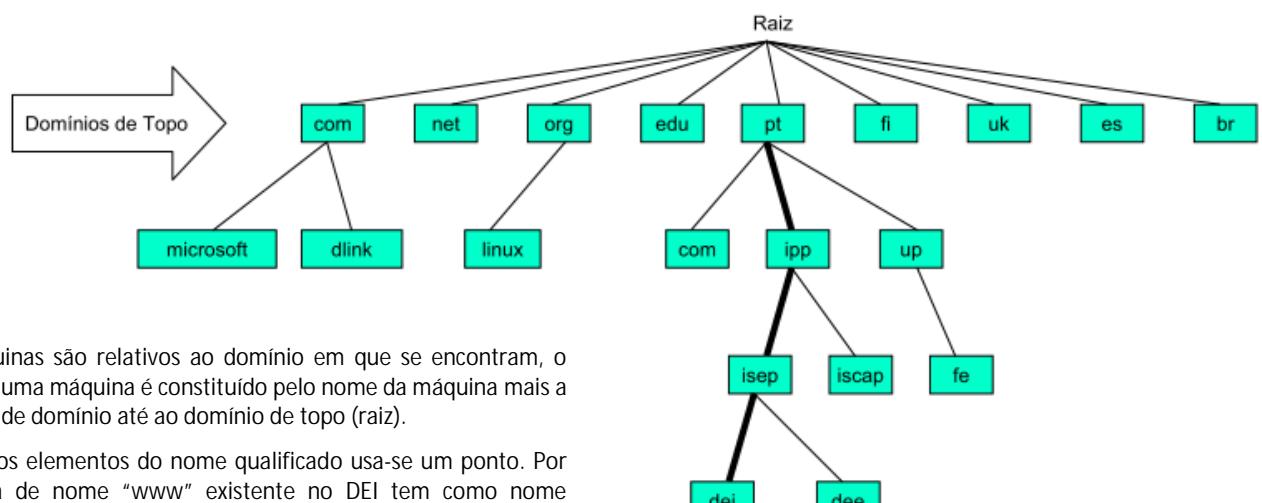
De momento verifica-se a coexistência WINS/DNS, alguns servidores WINS podem ser até configurados para recorrer a DNS quando não conseguem resolver um nome. Os clientes Windows também combinam o recurso a NetBIOS em BROADCAST, WINS e DNS, criando por vezes alguma confusão.

DNS – DOMAIN NAME SYSTEM:

O "Domain Name System" tem a grande vantagem de ser 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.

A figura apresenta alguns exemplos:

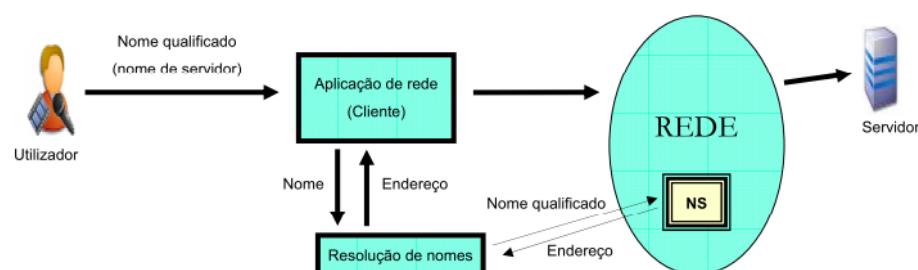


Os nomes das máquinas são relativos ao domínio em que se encontram, o nome qualificado de uma máquina é constituído pelo nome da máquina mais a sequência de nomes de domínio até ao domínio de topo (raiz).

Para separar os vários elementos do nome qualificado usa-se um ponto. Por exemplo a máquina de nome "www" existente no DEI tem como nome qualificado "www.dei.isep.ipp.pt"

SERVIDORES DNS:

A estrutura lógica do DNS usa como plataforma uma rede de servidores de nomes. 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 servidor de nomes (NS) contém uma base de dados com todos os registo, mas apenas dos domínios que serve (normalmente apenas um domínio). Estes servidores têm autoridade sobre o domínio ("authoritative DNS servers"). Isso

permite-lhe responder directamente 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.

DNS – REDE DE NS:

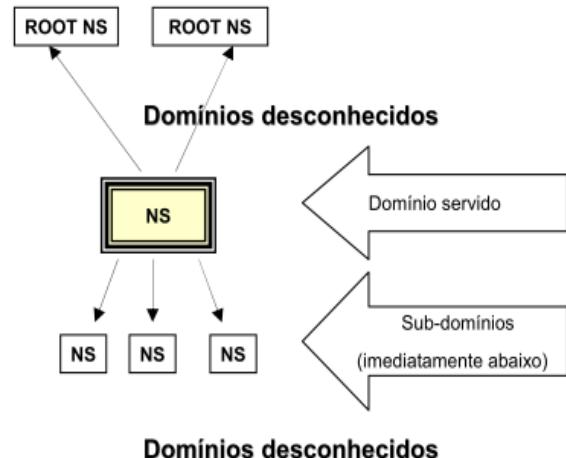
Para que o conjunto de servidores de nomes (NS) de todos os domínios permita o funcionamento em conjunto é necessário que cada servidor de nomes contenha a seguinte informação:

- Registos (nomes) do domínio que serve (ou cópia obtida do servidor principal)
- Endereço dos servidores de nomes da raiz (acima dos domínios de topo)
- Endereço dos servidores de nomes de cada sub-domínio

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.

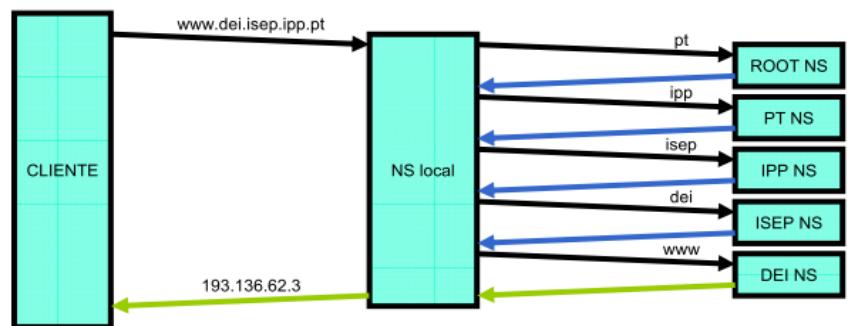
Uma vez que cada servidor de nomes é obrigado a conhecer os servidores de nomes do domínio imediatamente abaixo, este processo conduz sempre ao servidor de nomes correcto.



RESOLUÇÃO DE NOMES DNS:

A resolução de nomes funciona em sentido descendente desde os domínios de topo. Se algures na INTERNET uma aplicação necessita do endereço de www.dei.isep.ipp.pt começa por contactar o servidor de nomes local cujo endereço conhece. 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. 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 directo 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 os respectivos servidores vão receber.

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 caracteres)	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 recorre-se a uma ligação TCP para o mesmo número de porto usado em UDP.

TIPOS DE "RESOURCE RECORD":

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

Exemplo de registo SOA em ficheiro de configuração de zona do "BIND 9" em Linux

```
@ 99999999 SOA picasso.dei.isep.ipp.pt. root.picasso.dei.isep.ipp.pt. (
    2008042402 ; serial
    28800        ; refresh (8 hours)
    7200         ; retry (2 hours)
    604800       ; expire (7 days)
    86400        ; minimum (1 day)
```

@ representa o nome do domínio da zona a que este registo se aplica, no caso "dei.isep.ipp.pt". (directiva "\$ORIGIN" no BIND)

TTL – é sempre um valor numérico em segundos, pode ser omitido.

A classe foi omitida, o valor por omissão é "IN".

Servidor de nomes primário.

O número de série é usado para sincronismo, contém a identificação do dia, e um número de série dentro desse dia.

DNS RR – NOMES E APELIDOS:

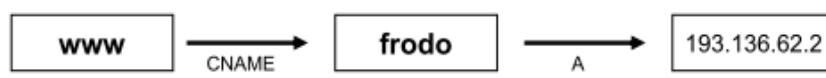
Exemplo com endereços IPv4 (A), comentários (TXT) e apelidos (CNAME) - configuração de zona no BIND

mafalda2	99999999	A	193.136.62.4
frodo	99999999	A	193.136.62.2
frodo	99999999	TXT	"Servidor de mail e web"
mafalda2	99999999	TXT	"Servidor de contas de utilizador"
mafalda	99999999	CNAME	mafalda2
www	99999999	CNAME	frodo
pop	99999999	CNAME	frodo
mail	99999999	CNAME	mafalda2
pdc	99999999	CNAME	mafalda2
smb	99999999	CNAME	mafalda2
samba	99999999	CNAME	mafalda2

Apelidos (Alias)

Nomes canónicos (nomes reais / próprios)

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".



DNS RR – NS E “GLUE RECORDS”:

Os registos NS são fundamentais no sistema DNS, são eles que garantem a ligação descendente entre os domínios. Para manter a ligação, cada domínio tem de conhecer os servidores de nomes dos seus sub-domínios.

```
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

Os “glue records” (destacados a verde) são registos de endereço (A) que não pertencem ao domínio, mas são necessários para obter os endereços dos servidores de nomes.

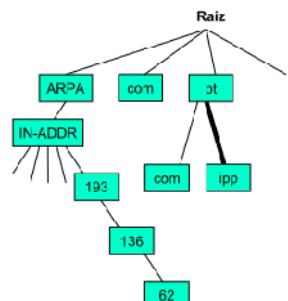


Sem o “glue record” a última resolução não seria possível, pois quem a deveria realizar seriam os servidores de nomes do domínio “dei.isep.ipp.pt”.

DNS RR – PTR E DOMÍNIO IN-ADDR.ARPA.:

O domínio especial “IN-ADDR.ARPA” contém um registo da estrutura de endereços IPv4. Cada nível de sub-domínio corresponde a um octeto do endereço IPv4, o octeto da esquerda corresponde ao sub-domínio 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.



Exemplo “in-addr.arpa.” - configuração de zona no BIND

```
2.62.136.193.in-addr.arpa IN PTR frodo.dei.isep.ipp.pt
3.62.136.193.in-addr.arpa IN PTR picasso.dei.isep.ipp.pt
4.62.136.193.in-addr.arpa IN PTR mafalda2.dei.isep.ipp.pt
```

Exemplo “in-addr.arpa.” com directiva “\$ORIGIN” - configuração de zona no BIND

```
$ORIGIN 62.136.193.in-addr.arpa

2 IN PTR frodo.dei.isep.ipp.pt
3 IN PTR picasso.dei.isep.ipp.pt
4 IN PTR mafalda2.dei.isep.ipp.pt
5 IN PTR srv1.dei.isep.ipp.pt
6 IN PTR srv2.dei.isep.ipp.pt
```

Os registos PTR têm de ser consistentes com os registos A.

CORREIO ELECTRÓNICO:

Os domínios DNS são usados pelo correio electrónico 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”).

Uma solução possível é recorrer ao domínio superior e criar um registo “A” associado ao nome do domínio que representa o endereço IP do servidor de correio:

Exemplo com RR A para os sub-domínios - 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
dei.isep.ipp.pt      IN      A      193.136.62.2
dee.isep.ipp.pt     IN      A      193.136.63.3
```

Servidor de mail de cada sub-domínio
(MTA – Mail Transfer Agent ; Mail server;
Mail Exchanger)

Por exemplo, o envio de uma mensagem para "username@dei.isep.ipp.pt", começa pela resolução de "dei.isep.ipp.pt", obtendo-se "193.136.62.2". De seguida é usado o protocolo SMTP para enviar a mensagem ao nó 193.136.62.2, esse é o servidor de correio.

DNS RR – MX:

Os registos MX (MailExchange) 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).

Exemplo de registos MX - configuração de zona no BIND

```
dei.isep.ipp.pt      IN      MX      10      frodo.dei.isep.ipp.pt
dei.isep.ipp.pt      IN      MX      20      picasso.dei.isep.ipp.pt
picasso.dei.isep.ipp.pt IN      A      193.136.62.3
frodo.dei.isep.ipp.pt IN      A      193.136.62.110
```

Utilizando a informação do exemplo, quem pretender enviar correio para o domínio "dei.isep.ipp.pt" vai obter uma lista de dois servidores, em primeiro lugar vai tentar usar o "frodo.dei.isep.ipp.pt".

É possível definir vários servidores de correio com a mesma preferência, nesse caso o BIND aplica o algoritmo "round-robin" à ordem dos registos devolvidos aos clientes, como nesta situação os clientes usam o primeiro registo, consegue-se distribuir os pedidos pelos vários servidores.

DNS RR – SRV:

Os registos SRV (RFC 7282) têm como objectivo 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.

Exemplo de registos SRV - configuração de zona no BIND

```
_ldap._tcp.dei.isep.ipp.pt      IN      SRV      5 4 389      mafalda2.dei.isep.ipp.pt
_ldap._tcp.dei.isep.ipp.pt      IN      SRV      5 6 389      frodo.dei.isep.ipp.pt
_ldap._tcp.dei.isep.ipp.pt      IN      SRV      20 20 389      picasso.dei.isep.ipp.pt
```

No exemplo, quando um cliente LDAP pretende encontrar um servidor no domínio "dei.isep.ipp.pt" pede ao servidor de nomes o registo SRV do nome "_ldap._tcp.dei.isep.ipp.pt", recebendo 3 respostas, destas selecciona desde logo as de mais elevada prioridade (no caso 5) de entre elas vai entrar em consideração com os pesos (4 e 6), por isso existe 40% de probabilidade de usar o 1º e 60% de probabilidade de usar o 2º.

DNS RR – 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 electrónico 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.

Exemplo de registos SPF - configuração de zona no BIND

```
dei.isep.ipp.pt      IN  SPF      "v=spf1 +mx -all"  
dei.isep.ipp.pt      IN  TXT      "v=spf1 +mx -all"
```

O texto associado define quem está autorizado a enviar em nome do domínio, começa por identificar a versão (v=spf1) e depois define quem está autorizado (+) e quem não está (-). No caso apenas os MX do domínio estão autorizados a enviar.

O texto de autorização suporta um grande número de possibilidades (ver RFC 4408).

DNS RR – 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 = 41° 10' 39,782" Norte; longitude = 8° 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
```

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, por exemplo com os registos SRV, os actuais 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 actualizações dinâmicas de registos DNS. É usado pelo comando "nsupdate".

Os processos de alteração da base de dados estão protegidos por mecanismos de segurança dos quais se destaca o "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)" descrito no documento RFC 3645 e adoptado pela Microsoft ("Microsoft DNS").

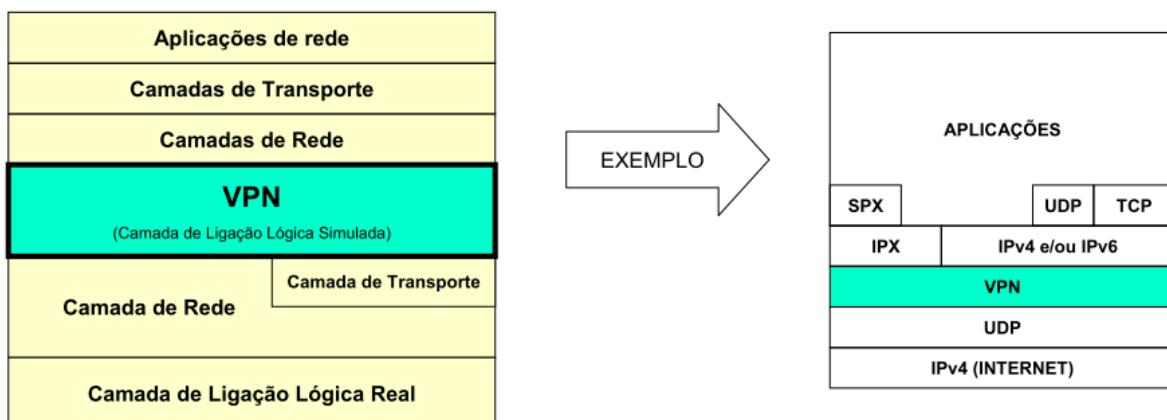
Os servidores DNS também podem ser actualizados por outras vias paralelas, alteram os ficheiros de configuração DNS e obrigam o servidor DNS a reler os mesmos, por exemplo o comando "ddclient" funciona deste modo usando pedidos HTTP.

T9

Redes Privadas Virtuais (VPN)

VIRTUAL PRIVATE NETWORK (VPN):

Uma VPN é uma infra-estrutura de comunicação de nível 2 (camada de ligação lógica) que é simulada sobre uma outra rede, tipicamente uma infra-estrutura de nível 3 (camada de rede).



A designação VIRTUAL tem origem no facto de se tratar de uma infra-estrutura simulada (não real), normalmente uma ligação "ponto-a-ponto"; PRIVADA advém do facto de serem usados mecanismos de segurança que garantem a confidencialidade dos dados que circulam na VPN.

VPN LAN-LAN ("SITE-TO-SITE VPN"):

Embora idênticas sob o ponto de vista de funcionamento, podem considerar-se dois tipos de aplicação diferente das VPN:

LAN-LAN e HOST-LAN

As VPN LAN-LAN destinam-se a interligar redes, são implementadas pelos administradores das redes como outro qualquer tipo de ligação entre duas redes.

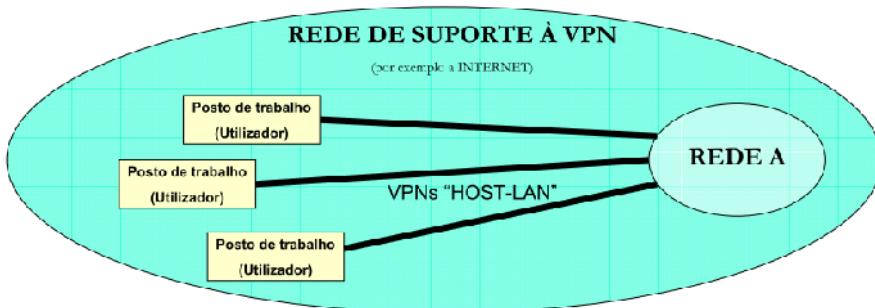


Os utilizadores das redes usufruem destas ligações VPN sem necessidade de conhecerem a sua existência. Dadas as suas características, devem ter um carácter permanente, podendo ser estabelecidas e recuperadas automaticamente sem intervenção manual.

VPN HOST-LAN ("REMOTE-ACCESS VPN"):

As VPN HOST-LAN servem para ligar nós individuais a uma rede remota.

Tipicamente uma VPN deste tipo 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.

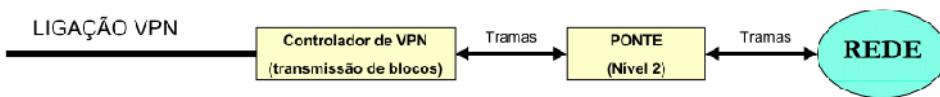


Neste tipo de VPN, o controlo de acesso baseia-se na autenticação do utilizador, após esse processo o posto de trabalho recebe um endereço de rede pertencente à rede remota que lhe permite operar como se estivesse directamente ligado a essa rede.

INTERLIGAÇÃO DE REDES POR VPN – NÍVEL 2:

Uma VPN tem como objectivo simular uma ligação física “ponto-a-ponto”. A forma como esta ligação pode ser usada para interligar redes remotas depende dos objectivos e muitas vezes das restrições impostas pelo próprio protocolo da VPN.

A interligação por VPN no nível 2 consiste na retransmissão de tramas de nível 2 através da VPN, a VPN comporta-se 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.

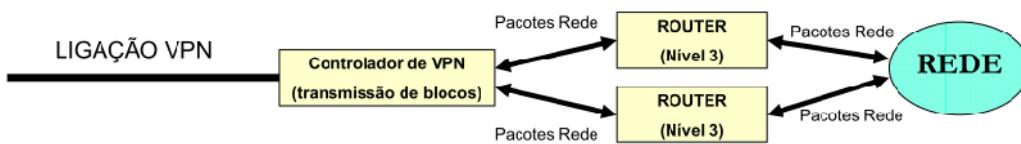
A “Rede 1” e a “Rede 2” funcionam como dois segmentos de uma mesma rede local, interligados por uma ponte. Por exemplo sob o ponto de vista IP, são apenas uma rede.



INTERLIGAÇÃO DE REDES POR VPN – NÍVEL 3:

A interligação por VPN no nível 3 consiste na retransmissão de pacotes de rede através da VPN, a VPN comporta-se 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.

Por exemplo, sob o ponto de vista IP, a “Rede 1” e a “Rede 2” são duas redes distintas, cada uma exigindo o seu espaço de endereçamento independente.

No exemplo a própria ligação VPN pode exigir endereços de rede para a ligação ponto a ponto entre os dois encaminhadores.



SEGURANÇA DAS VPN:

Tratando-se transferências de dados que usam infra-estruturas potencialmente inseguras e nas quais é possível todo tipo de intervenções de terceiros, a introdução de mecanismos de segurança é fundamental.

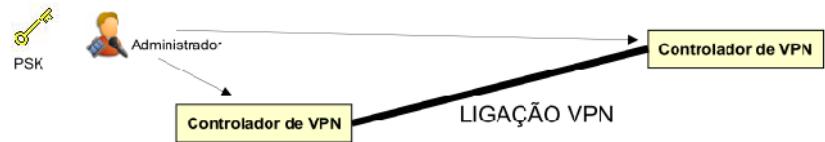


Autenticação: Garantir a autenticidade dos intervenientes (nós da VPN) ou seja, máquinas/servidores ou utilizadores.

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).

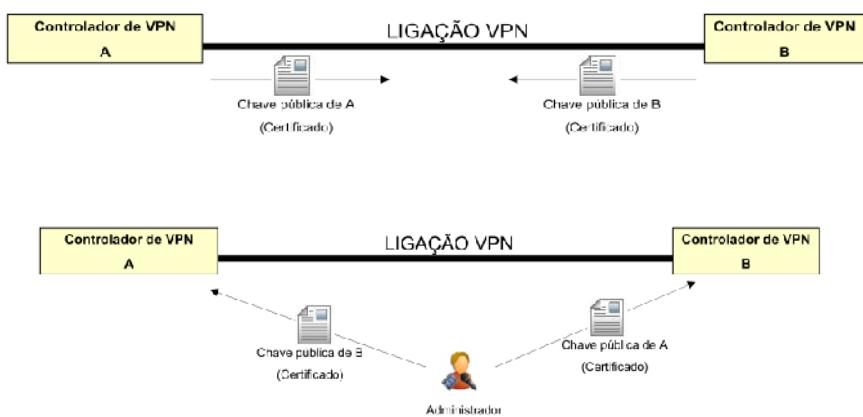
Sendo um segredo pré-partilhado, este é um bom mecanismo de autenticação. Se uma das partes não possui a chave correcta não vai poder comunicar. A operação de distribuição da chave pode ser complicada, na sua versão mais simples é realizada pelo administrador das duas extremidades da VPN.



VPN – CHAVES PÚBLICAS:

A técnica de cifragem conhecida por criptografia de chave pública veio resolver de forma radical as dificuldades na distribuição das chaves.

Ao contrário dos algoritmos simétricos anteriores, em que existe uma única chave que tem de ser mantida secreta a todo o custo, agora 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.



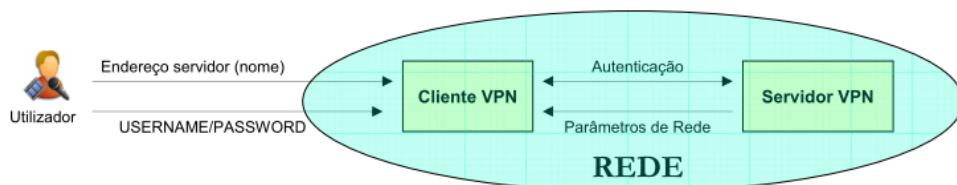
Os certificados são um elemento importante porque garantem a autenticidade dos intervenientes. Se os nós forem controlados pelo mesmo administrador (VPN LAN-LAN) os certificados podem ser instalados manualmente para maior segurança.

VPN DE UTILIZADOR:

Uma VPN de utilizador é tipicamente uma VPN HOST-LAN e caracteriza-se por ser criada por iniciativa do utilizador.

Para esse efeito 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 + PASSWORD". Posteriormente o servidor fornece ao cliente os parâmetros de configuração de rede para o cliente poder funcionar.



O par "NOME-DE-UTILIZADOR + PASSWORD" 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".

VPN DE UTILIZADOR – AUTENTICAÇÃO COM CHAVE PÚBLICA:

As VPN de utilizador caracterizam-se por:

- Autenticação de utilizador por UTILIZADOR/PASSWORD
- Servidor sem conhecimento prévio da existência do cliente
- Número elevado de clientes/utilizadores para cada servidor.

Sendo a autenticação do utilizador garantida por PASSWORD, também é necessário garantir a autenticidade do servidor e um manuseamento seguro da PASSWORD do utilizador.

Dadas as características particulares, alguns métodos não são praticáveis. É o caso do PSK devido ao elevado número de clientes/utilizadores.

Embora possam ser combinados de várias formas existem duas abordagens em uso: com chave pública e com chave secreta.

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.

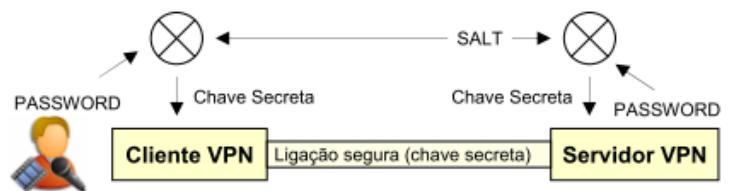


VPN DE UTILIZADOR – AUTENTICAÇÃO COM CHAVE SECRETA:

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 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.

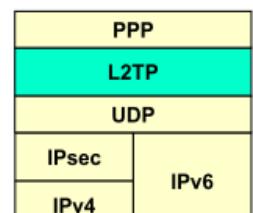
L2TP - LAYER 2 TUNNELING PROTOCOL:

O L2TP é um dos tipos de VPN com mais utilização, foi desenvolvido pela Microsoft e pela Cisco, mas está normalizado em várias RFC's.

O L2TP 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.

O IPsec, 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 respectivo tráfego.



O L2TP cria os túneis 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.

Protocolo PPP

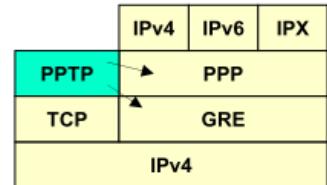
PPTP - POINT-TO-POINT TUNNELING PROTOCOL:

O protocolo PPTP é um predecessor do L2TP, mas ainda é bastante usado pela Microsoft e também pela Cisco. Uma vez que 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 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 directo 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, tanto a privacidade como a autenticação são asseguradas pelo protocolo PPP.



O protocolo PPP pode suportar diversos mecanismos de autenticação e privacidade, no contexto actual a Microsoft usa o protocolo de autenticação MSCHAPv2 (CHAP = Challenge Handshake Authentication Protocol) que além da autenticação do 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").

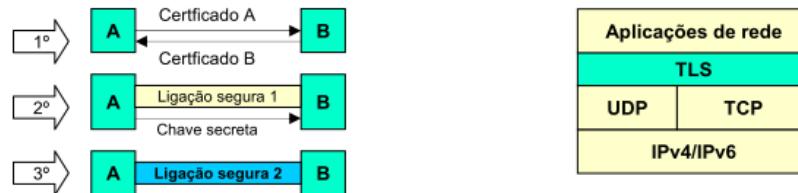
OUTROS PROTOCOLOS DE VPN:

Existe uma grande variedade de protocolos de VPN, alguns proprietários, outros de domínio público, com clara vantagem destes últimos sob o ponto de vista de inter-operacionalidade.

Tal como o protocolo L2TP se baseia e usa o protocolo IPsec, também existem alguns tipos de VPN apoiadas no protocolo SSL/TLS.

O protocolo SSL – Secure Sockets Layer (nome original da Netscape), actualmente em fase final de normalização com a designação TLS - Transport Layer Security tem como finalidade proporcionar segurança, quer sob o ponto de vista de privacidade, quer sob o ponto de vista de autenticação dos interlocutores.

Embora seja bastante flexível e modular, nas aplicações mais correntes usa certificados de chave pública para autenticação e distribuição de uma chave secreta e de seguida transfere os dados usando criptografia convencional que é muito mais rápida:



Tal como a sigla TLS indica, situa-se na camada de transporte (o IPsec encontra-se na camada de rede), assim sendo pode usar ligações TCP ou pacotes UDP. Os protocolos de aplicação podem tornar-se seguros sem grandes modificações se passarem a usar a camada TLS, actualmente quase todos os protocolos de aplicação dispõem de uma versão segura ("s") que usa TLS: http/https; pop3/pop3s; smtp/smtps; etc. Não sendo uma implementação de VPN a camada TLS pode ser usada para esse efeito.

UMA VPN SOBRE TLS - OPENVPN:

O OpenVPN é uma das implementações "Open Source" de VPN sobre TLS que mais se destaca actualmente.

A privacidade pode ser assegurada para o TLS, 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).

A autenticação pode usar vários métodos de acordo com os intervenientes, desde logo certificados de chave pública, esta é a técnica ideal para uma VPN LAN-LAN. Outra possibilidade em configurações LAN-LAN é usar PSK.

Quando há utilizadores envolvidos (VPN HOST-LAN) é possível a autenticação por PASSWORD. Neste tipo de configuração a PASSWORD é enviada directamente 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, a preferência vai para a UDP pois a implementação de VPN sobre TCP é problemática (TCP sobre TCP):

Uma vez que o TCP faz a retransmissão de segmentos quando não recebe o respectivo ACK, isso significa que quando há uma perda de conectividade o TCP faz retransmissões constantes, normalmente isso não é problema porque estas retransmissões perdem-se, mas se forem efectuadas sobre uma VPN a funcionar sobre TCP não se vão perder, pelo contrário vão acumular-se sucessivamente e pode provocar uma falha.

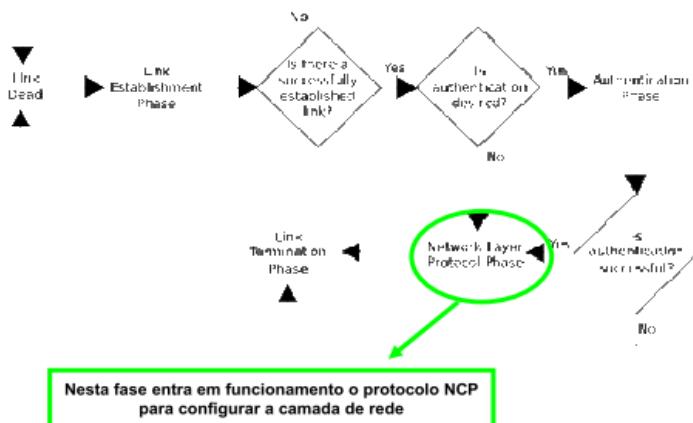
PPP – POINT TO POINT PROTOCOL:

O protocolo PPP deriva directamente 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.



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.



PPP – NCP (NETWORK CONTROL PROTOCOL):

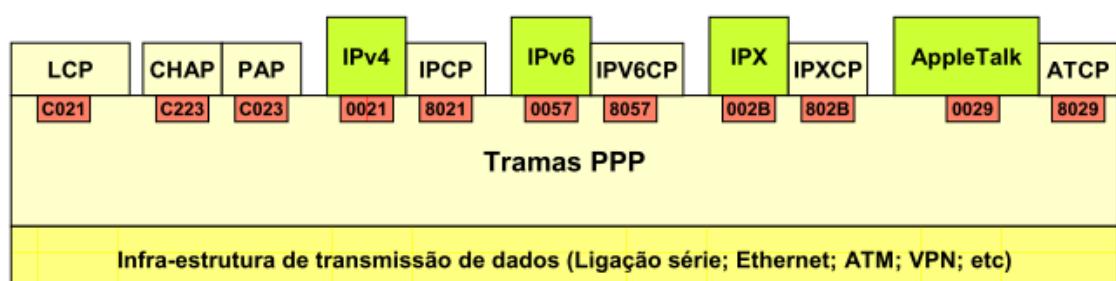
Depois de o LCP estabelecer o funcionamento do nível de ligação de dados, o processo segue ao nível de rede para cada um dos protocolos de rede pretendidos.

O protocolo NCP (Network Control Protocol) usa os identificadores de protocolo “0x8?”. ONCP é o responsável pela interacção 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, por exemplo, para o IPv4 o protocolo NCP é o IPCP (Internet Protocol Control Protocol – 0x8021). Entre outras funcionalidades o IPCP é o responsável pela configuração automática dos parâmetros de rede nos nós ao estilo DHCP.

Os identificadores de protocolo “0x0?” são usados para transportar os dados dos protocolos de rede propriamente ditos, para o IPv4 usa-se o identificador 0x0021

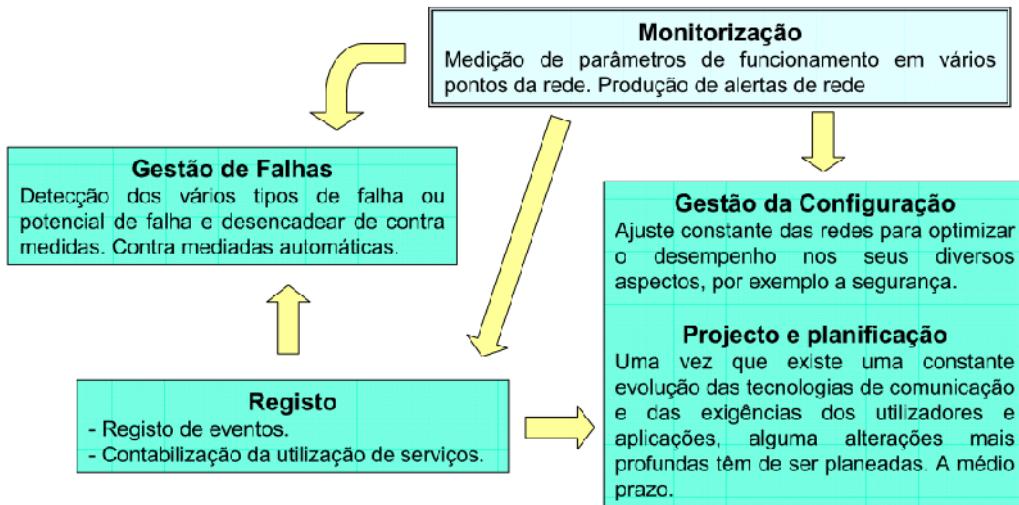
A figura apresenta alguns dos protocolos usados sobre a camada ppp e os respectivos identificadores de protocolo em notação hexadecimal.



Gestão de redes

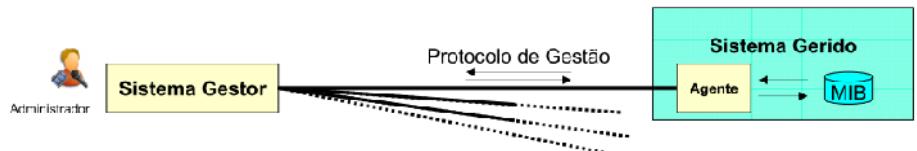
GESTÃO DE REDES:

Gestão de redes refere-se neste contexto às actividades relacionadas com a manutenção do bom funcionamento de um conjunto de redes.



MODELO “AGENTE – GESTOR”:

A maioria dos sistemas de gestão de rede adopta 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").

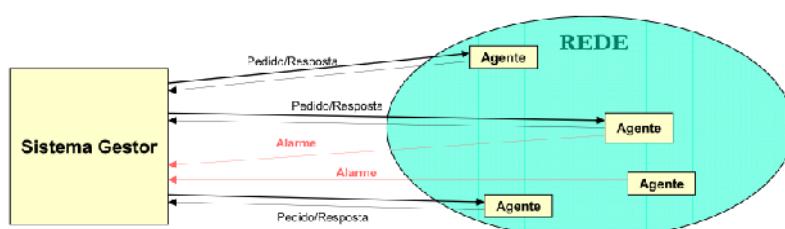
O agente é um serviço de rede residente no sistema gerido e que permite o acesso à MIB usando um protocolo de gestão através da rede.

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.

PROTOCOLO DE GESTÃO:

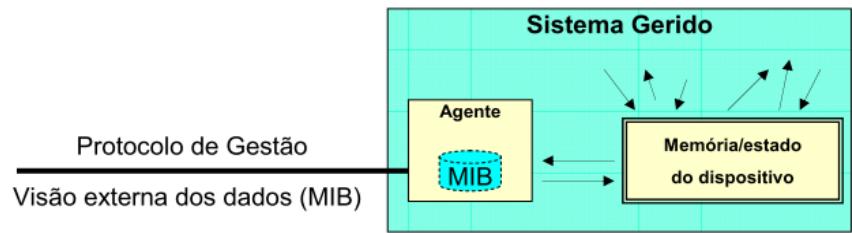
O protocolo de gestão envolve dois tipos de transacções:

- 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.
- 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 actividades.



MIB (“MANAGEMENT INFORMATION BASE”):

A MIB é o conjunto de dados associados a cada dispositivo de rede com capacidade de gestão.



Na realidade a MIB é 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 objectos, sendo o paradigma dos objectos levado mais ou menos longe conforme a implementação em causa.

Numa MIB de objectos pura, cada objecto de gestão é uma instânciação de uma classe (definida numa estrutura de classes e subclasses com herança). Cada classe define os métodos apropriados para interagir com o objecto.

Protocolo SNMP

SNMP ("SIMPLE NETWORK MANAGEMENT PROTOCOL"):

Embora existam outros protocolos de gestão, destacando-se o CMIP ("Common Management Information Protocol ") proposto pelo modelo OSI, a verdade é que associado à pilha de protocolos TCP/IP o protocolo usado é o SNMP. A maioria dos dispositivos de rede actuais suporta o protocolo SNMP.

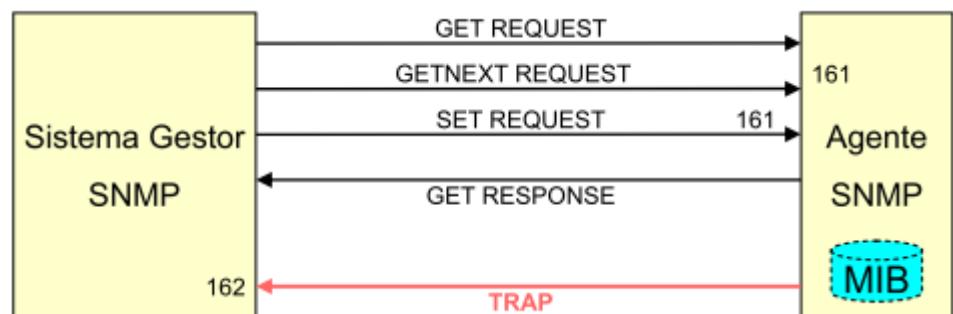
Estando as redes cada vez mais centradas na pilha TCP/IP e atendendo às várias evoluções que o protocolo SNMP tem sofrido no sentido do seu enriquecimento, não é de esperar a sua substituição no futuro.

A versão 1 (SNMPv1) ainda é muito usada devido à quantidade de dispositivos que não suporta outras versões. No SNMPv1 existem apenas 5 mensagens possíveis. 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 objectos da MIB são simples variáveis.

As mensagens "get request" permitem obter os valores de objectos da MIB através da mensagem "get response". A mensagem "set request" permite alterar o valor dos objectos, sendo confirmada por uma "get response".

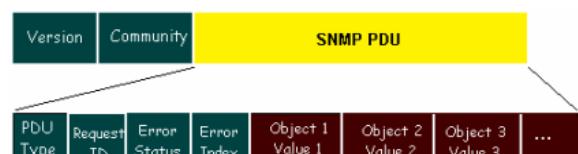
Na MIB os objectos são armazenados em sequência, a mensagem "getnext" permite uma consulta sequencial.



SNMPv1 - MENSAGENS:

Todas as 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.

0 - GetRequest
1 - GetNextRequest
2 - GetResponse
3 - SetRequest
4 - Trap

"Error Status" identifica o resultado da operação:

Em caso de erro, "Error Index" pode servir para indicar relativamente a qual dos objectos ocorreu esse erro.

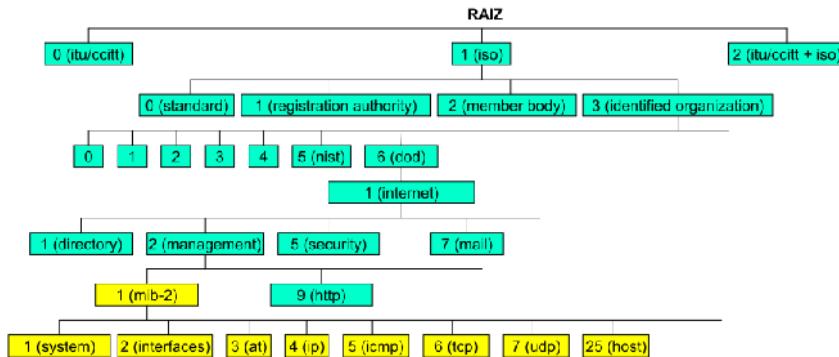
As mensagens tipo 0, 1, e 3 usam sempre o valor zero nos campos de erro.

A mensagem "Trap" possui um formato diferente.

0 - noError
1 - tooBig
2 - noSuchName
3 - badValue
4 - readOnly
5 - genErr

OID – OBJECT IDENTIFIER:

Os objectos (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.

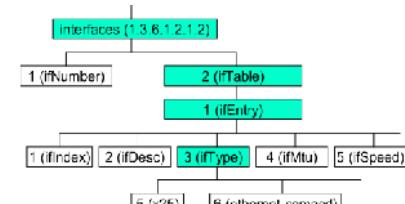


Cada tipo de objecto (classe) é identificado pela sequência de números até à raiz, por exemplo o objecto "internet" é identificado por "1.3.6.1", todos os objectos abaixo começam por esta sequência, os objectos relativos à gestão de rede começam por "1.3.6.1.2.1".

EXEMPLO: "INTERFACES", OID = 1.3.6.1.2.1.2:

No ramo das interfaces (1.3.6.1.2.1.2) está definido o tipo de objecto "ifNumber" que contém o número de interfaces existentes.

O tipo ifTable é uma tabela de objectos do tipo ifEntry (1.3.6.1.2.1.2.2.1), cada objecto do tipo ifEntry contém o objecto 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 objectos e não instâncias de objectos 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 1.3.6.1.2.1.2.2.1.8 representa a descrição da oitava interface que o sistema possui.

Outras classes de objectos 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 SNMP:

No SNMPv1 e SNMPv2c os mecanismos de segurança são muito elementares, normalmente é possível definir dois tipos de acesso: "leitura apenas" e "leitura e escrita". Os 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 protecção.

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 actualizações na MIB, o SNMPv2 introduz novas mensagens:

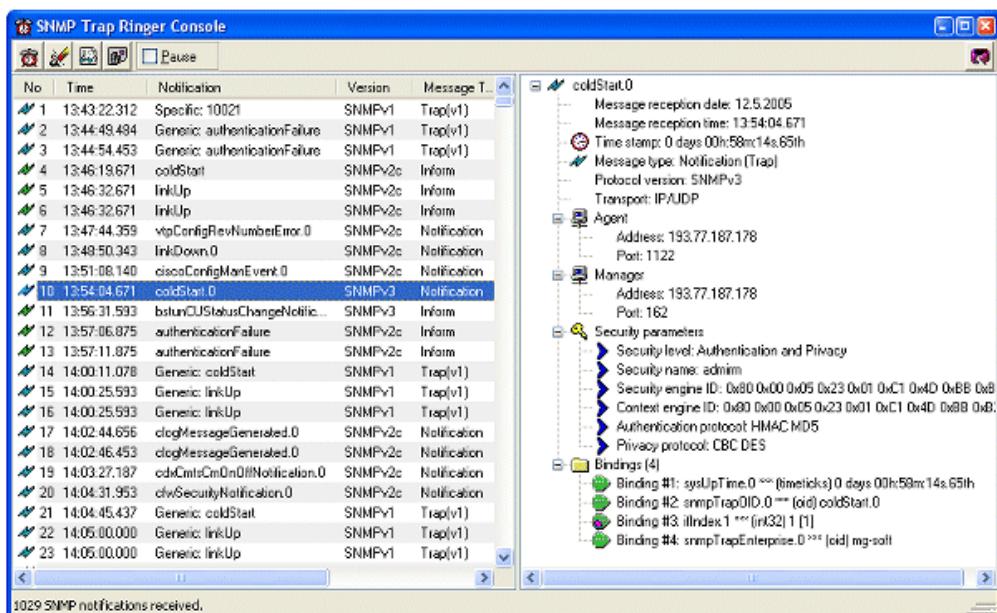
- “Getbulk Request”: serve para obter volumes de informação da MIB superiores aos permitidos pelo “Get Request”.
- “Info Request”: equivalente à mensagem “Trap”, mas com confirmação da recepção.

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 directamente 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.

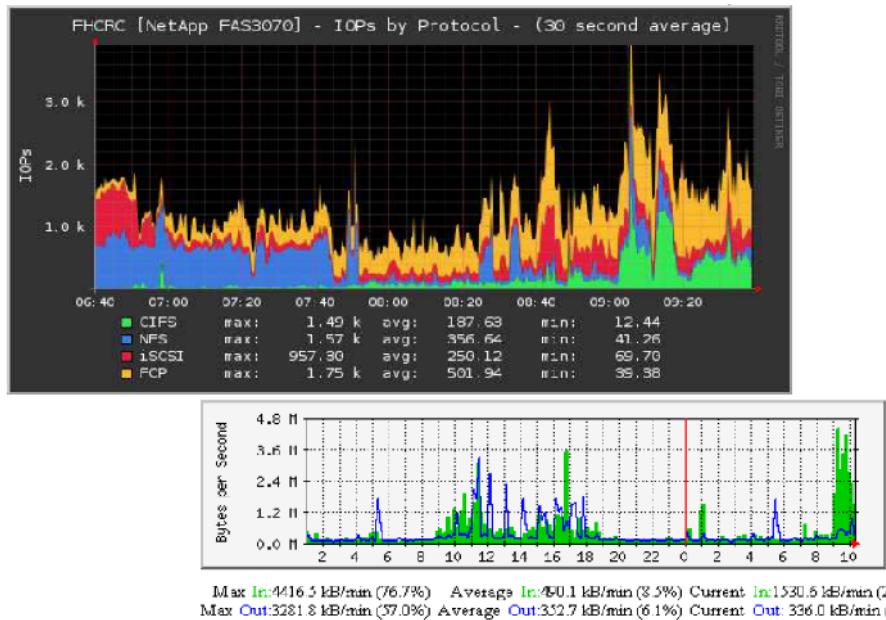
EXEMPLOS SNMP – “TRAPS”:



EXEMPLOS SNMP – MIB:

The screenshot shows a network management interface with several panels. The top-left panel displays the 'MIB Tree' with nodes like ipNetToMediaEntry, ipNetToMediaIndex, ipNetToMediaPhysAddress, ipNetToMediaNetAddress, and ipNetToMediaType. The top-right panel shows 'LiveGrid' tables for 'ipNetToMediaNetAddress' and 'ipNetToMediaEntry'. The bottom-left panel shows a 'Variable Grid' table with columns for Object, OID, Module, Object Type, and Value. The bottom-right panel shows 'MIB Info' for 'ipNetToMediaType' with fields for Syntax, Enumeration, and MAX ACCESS. The enumeration values listed are other[1], invalid[2], dynamic[3], and static[4].

EXEMPLOS SNMP – GRÁFICOS DESENHADOS PELO GESTOR:



T11

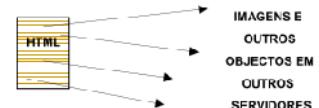
Protocolo HTTP

TRANSFERÊNCIA DE FICHEIROS EM REDE:

Com o surgimento dos documentos de hiper 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.

O FTP é demasiado complexo e torna-se lento quando se pretende realizar muitas transferências de ficheiros de dimensão reduzida, envolvendo diversos servidores, situação típica em documentos de hiper 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.



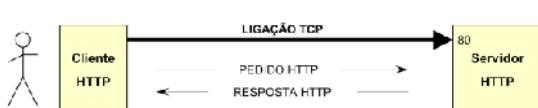
O FTP, com a necessidade de uma ligação de controlo de sessão e autenticação de utilizador não é de todo adequada para este tipo de aplicação. Muitas vezes demora mais tempo a estabelecer a sessão FTP do que a transferir o ficheiro.

HTTP - HYPertext Transfer Protocol:

Embora existissem versões anteriores em uso, a primeira versão completamente funcional e compatível com as seguintes surgiu em 1996, o "HTTP 1.0".

O objectivo 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.

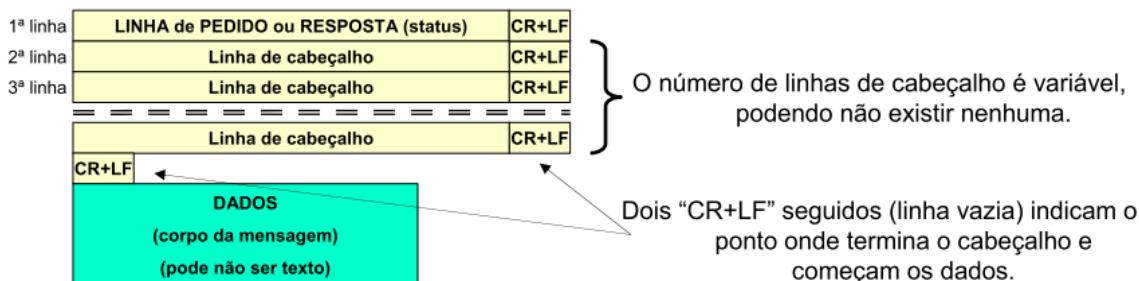
Normalmente o cliente começa por estabelecer uma ligação TCP com o servidor que está à espera no porto número 80, depois de estabelecida a ligação o protocolo HTTP usa-a para comunicação entre as duas entidades. Seguindo o modelo cliente / servidor, o cliente envia um "pedido HTTP" e servidor devolve 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 HTTP:

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):



HTTP – PEDIDOS E RESPOSTAS:

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

Método (Tipo de pedido)	Espaço	Argumento (URI)	Espaço	Nome da versão HTTP	CR+LF
OPTIONS					
GET				HTTP/1.0	
HEAD		Identificação do recurso, <u>não pode conter</u>			
POST		espaços, nem CR, nem LF.		HTTP/1.1	
PUT					
DELETE		O significado pode variar de acordo com		HTTP/1.2	
TRACE		o método, o valor "*" significa que			(...)
CONNECT		não é aplicável no método usado.			

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

Nome da versão HTTP	Espaço	Código	Espaço	Texto de descrição do código	CR+LF
HTTP/1.0					
HTTP/1.1				Código de estado / resultado.	
HTTP/1.2				É sempre um número inteiro de 3 dígitos.	
(...)				Por exemplo “200” significa sucesso da operação e o texto de descrição correspondente é “OK”.	

HTTP – LINHAS DE CABEÇALHO (PARÂMETROS DE CABEÇALHO):

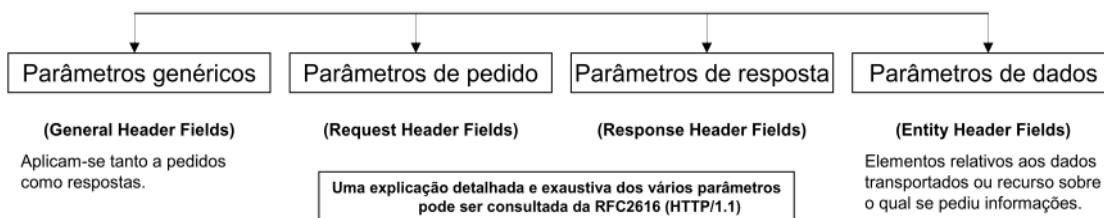
As linhas de cabeçalho servem para implementar diversas funcionalidades do protocolo HTTP, a sua forma geral é:

Nome do parâmetro : **Valor do parâmetro** | CR+LF

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. Seque-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.

Tanto os pedidos como as respostas podem conter parâmetros de cabecalho:



HTTP – GENERAL HEADER FIELDS:

Alguns dos parâmetros genéricos de cabeçalho mais usados são:

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.

HTTP – 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 associada 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.

HTTP – REQUEST HEADER FIELDS:

Parâmetros que podem acompanhar os pedidos HTTP, os mais usados são:

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.

HTTP – RESPONSE HEADER FIELDS:

Alguns dos parâmetros específicos que podem acompanhar as respostas HTTP são:

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)"

	Acompanha a resposta “401 Unauthorized”. Contém uma identificação do método de autenticação que o cliente deve usar e parâmetros associados ao processo.
WWW-Authenticate	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	Contém um par “nome=valor” que o cliente deve guardar e fornecer em todos os pedidos subsequentes com este servidor.
Set-Cookie2	

HTTP/1.1 – MÉTODOS OPTIONS E GET:

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

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

Serve para obter o documento identificado por “URI”.

Se o URI 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 objecto 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. Por um lado 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.

HTTP/1.1 – MÉTODOS HEAD, POST, PUT E DELETE:

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.

HTTP/1.1 – 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.

“201 Created” – indica que um recurso foi criado, por exemplo com o método PUT.

“202 Accepted” – indica que o pedido foi aceite, mas poderá não ter sido executado de imediato.

HTTP/1.1	Espaço	3XX	Espaço	Texto de descrição do código	CR+LF
----------	--------	-----	--------	------------------------------	-------

Os códigos 3XX indicam uma falha e necessidade de redireccionar a operação. Exemplos:

“300 Multiple Choices” – indica que há várias possibilidades para executar o pedido. Fornece uma lista.

“301 Moved Permanently” – indica que um recurso foi deslocado, nova localização no campo “Location:”.

“307 Temporary Redirect” – situação temporária, nova localização no campo “Location:” do cabeçalho.

HTTP/1.1 – CÓDIGOS DE RESPOSTA (4XX E 5XX):

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.

T12

Correio electrónico

CORREIO ELECTRÓNICO:

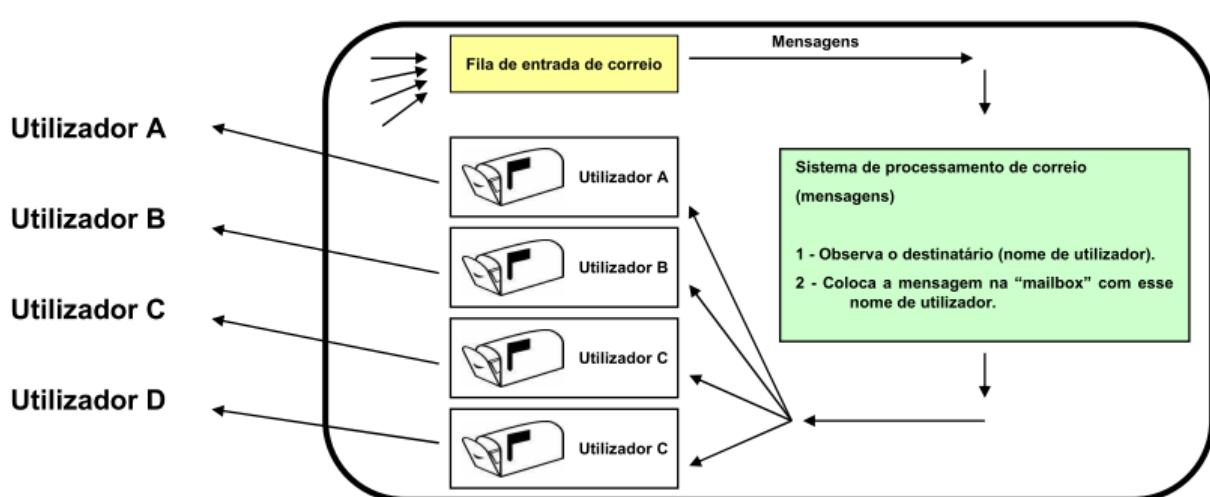
O objectivo do correio electrónico é o envio de mensagens “off-line” (não interativo) entre utilizadores. A única diferença relativamente a um sistema de correio tradicional é que as mensagens não são escritas em papel.

Num sistema de correio, o destinatário pode não estar presente no momento em que a mensagem chega (“off-line”). Consequentemente é necessário um local onde o “carteiro” possa colocar as mensagens até que o destinatário “apareça” para as recolher.

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 (“mail-box”).

CAIXAS DE CORREIO – “MAILBOXES”:

O sistema de correio electrónico típico tem uma estrutura bastante simples:



Cada mensagem tem um remetente e um destinatário que são nomes de utilizador. Cada utilizador possui uma mailbox, apenas o utilizador e o sistema podem aceder à mailbox. O sistema é implementado usando um sistema de ficheiros com permissões de utilizador.

CORREIO ELECTRÓNICO BASEADO EM SISTEMA DE FICHEIROS:

Os sistemas de correio electrónico desenvolveram-se usando simples sistemas de ficheiros partilhados. Tanto as mailboxes dos utilizadores como a fila de entrada de correio são objectos do sistema de ficheiros, ficheiros e/ou directórios. Um sistema deste tipo está totalmente contido num único servidor e não utiliza a rede directamente.



Uma vez que o sistema está limitado a um único servidor/sistema, a identificação dos utilizadores faz-se recorrendo apenas ao nome do utilizador.

CORREIO ELECTRÓNICO EM REDE:

Com o progressivo desenvolvimento das redes de computadores, surgiu a necessidade de alargar o funcionamento dos sistemas de correio electrónico existentes de tal modo que, utilizadores de sistemas centrais diferentes possam também comunicar uns com os outros.

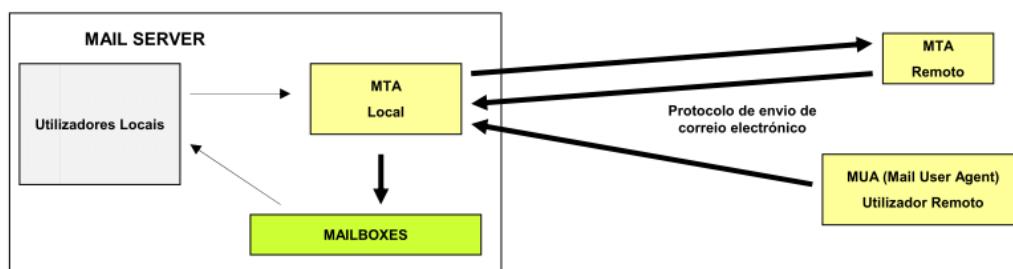


A comunicação entre sistemas centrais de correio recorre a uma infra-estrutura de rede e será realizada segundo um protocolo de aplicação reconhecido pelos dois intervenientes.

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

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 directamente o sistema de ficheiros para enviarem correio e lerem o correio das respectivas mailboxes.

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 adequado designado de MUA ("Mail User Agent").

Protocolos SMTP, POP3 e IMAP

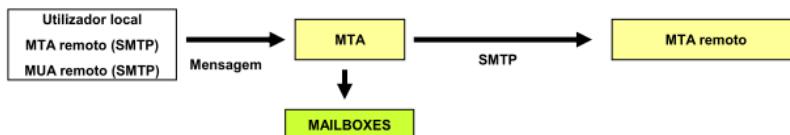
SMTP – SIMPLE MAIL TRANSFER PROTOCOL:

Na Internet, o protocolo de aplicação mais usado para transferir correio entre sistemas é o SMTP. 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.



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.

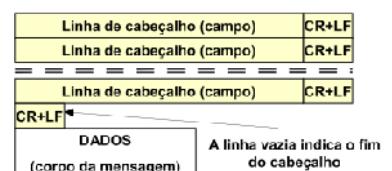
Na prática torna-se mais cômodo identificar utilizadores em domínios DNS e não em servidores. Para conseguir isso pode-se recorrer ao domínio acima e criar um registo A que provoque a resolução do nome de domínio para o endereço do servidor de correio.

Actualmente o sistema DNS implementa registos apropriados para resolver este problema de uma forma mais eficiente, os registos MX ("Mail Exchanger"). Os registos MX associam directamente o nome do domínio a um ou vários endereços dos servidores de correio desse domínio.

Os MTA actuais resolvem o nome do domínio pedindo o respectivo registo MX e não o registo A.

SMTP – FORMATO DAS MENSAGENS:

As mensagens de correio electrónico são constituídas por um cabeçalho seguido do corpo da mensagem.



Cada linha de cabeçalho contém um identificador de campo e o respectivo valor, separado por ":".

Os valores dos campos podem ocupar mais do que uma linha, nesse caso as linhas de continuação devem começar com um espaço em branco.

EXEMPLOS

```
From: Utilizador <user@dei.isep.ipp.pt>
Subject: Mensagem de Teste
Date: Wed, 21 May 2008 15:54:50 +0100
Reply-To: <user@ipp.pt>
To: <admin@dei.isep.ipp.pt>
Cc: <root@isep.ipp.pt>
Return-Path: <errors@dei.isep.ipp.pt>
Message ID: <011701c6bb52$1ca5f10$e55f4d30$@dei.isep.ipp.pt>
Tn-Reply-To: <8AB511FF5C834F8F2308F52E6437D5DP@ipp.pt>
```

SMTP – PROTOCOLO:

O SMTP usa uma ligação TCP para transferir a mensagem de correio electrónico, 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+LF, seguindo um conjunto de comandos suportado (RFC 821). No exemplo seguinte o texto enviado pelo cliente encontra-se a "negrito":

```
220 frodo.dei.isep.ipp.pt ESMTP Mailer DFTNET-1.1; Wed, 21 May 2008 18:15:30 +0100
HELO frodo.dei.isep.ipp.pt
250 frodo.dei.isep.ipp.pt Hei! I'm pri 4ppp.dei.isep.ipp.pt [193.36.62.213], pleased to meet you
MAIL FROM:<andre@dei.isep.ipp.pt>
250 2.1.0 <andre@dei.isep.ipp.pt>... Sender ok
RCPT TO:<asc@isep.ipp.pt>
250 2.1.5 <asc@isep.ipp.pt>... Recipient ok
DATA
254 Enter mail, end with "." on a line by itself
From: "Andre Moreira" <andre@dei.isep.ipp.pt>
To: <asc@isep.ipp.pt>
Subject: Teste

Mensagem de teste
.
250 2.0.0 m/LHFUWx004991 Message accepted for delivery
QUIT
221 2.0.0 frodo.dei.isep.ipp.pt closing connection
```

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".

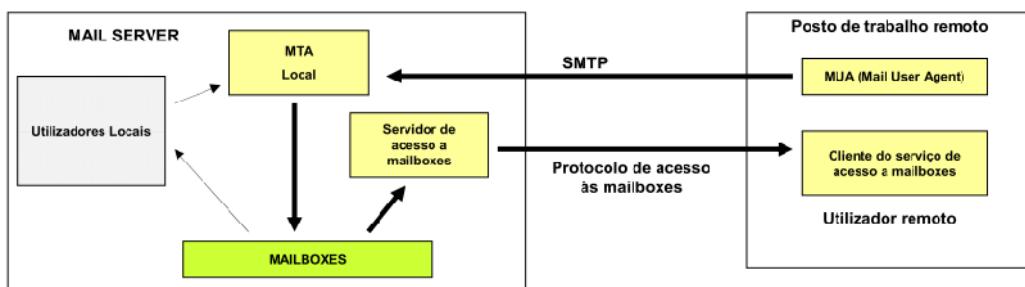
Se o servidor suporta ESMTP responde com um código de sucesso (250), caso contrário responde com um código de erro (5xx), nesse caso o cliente terá de enviar um "HELO" e limitar-se ao SMTP normal.

```
220 frodo.dei.isep.ipp.pt ESMTP Mailer DEINET-1.1; Wed, 21 May 2008 10:46:30 +0100
EHLO frodo.dei.isep.ipp.pt
250-frodo.dei.isep.ipp.pt Hello pcu14ppp.dei.isep.ipp.pt [193.136.62.213], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 33554432
250-DSN
250-ETRN
250-AUTH DIGEST-MD5
250-DELIVERBY
251 HKIP
QUIT
```

A identificação das extensões suportadas pelo servidor é fornecida ao cliente juntamente com a resposta ao "EHLO".

CORREIO ELECTRÓNICO – ACESSO REMOTO:

Os utilizadores remotos de um sistema de correio podem recorrer ao protocolo SMTP para emitir mensagens, mas para poderem aceder às respectivas mailboxes torna-se necessário um protocolo adicional.



Actualmente os dois protocolos mais usados para acesso a mailboxes remotas são o IMAP4 e o POP3.

POP3 – POST OFFICE PROTOCOL VERSION 3:

O protocolo POP3 (RFC 1939) usa uma ligação TCP dirigida ao porto 110 do servidor, as mensagens trocadas entre o cliente e o servidor são em texto simples, com 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 "senha" enviadas pelo cliente:

```
+OK POP3 frodo.dei.isep.ipp.pt 2004.89rdk server ready
USER andré
+OK User name accepted, password please
PASS xxxxxxxx
+OK Mailbox open, 0 messages
STAT
+OK 0 0
LIST
+OK Mail_box 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 POP3 é bastante limitado, normalmente serve apenas para obter o conteúdo da mailbox do utilizador e a ligação com o servidor é terminada após essa consulta integral.

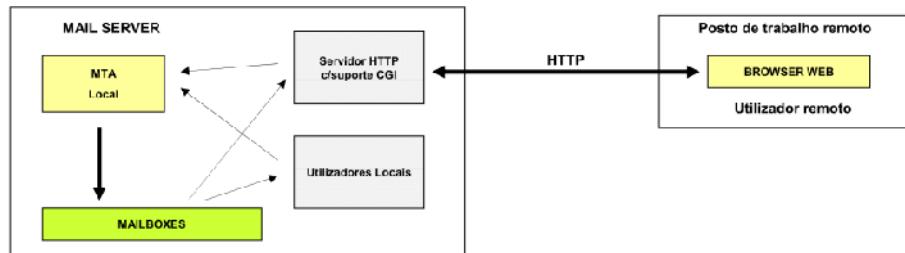
O protocolo IMAP4 (IMAP4rev1 - RFC 3501) é bastante mais interativo, também usa uma conexão TCP, nesta caso para o porto 143, mas normalmente a ligação do cliente com o servidor mantém-se activa constituindo uma sessão interactiva.

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:

Muitas das vantagens equivalentes às do IMAP4 estão actualmente disponíveis usando sistemas conhecidos por 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 electrónico do mesmo modo que os utilizadores locais. Deste modo eliminam todos os inconvenientes do acesso remoto sem necessidade de clientes especiais.

Formato MIME

MIME - MULTIPURPOSE INTERNET MAIL EXTENSIONS:

O MIME é um formato de mensagens que permite ultrapassar as limitações das mensagens de texto simples. Embora tenha sido desenvolvido para o correio electrónico é actualmente usado em muitos outros protocolos como por exemplo o HTTP.

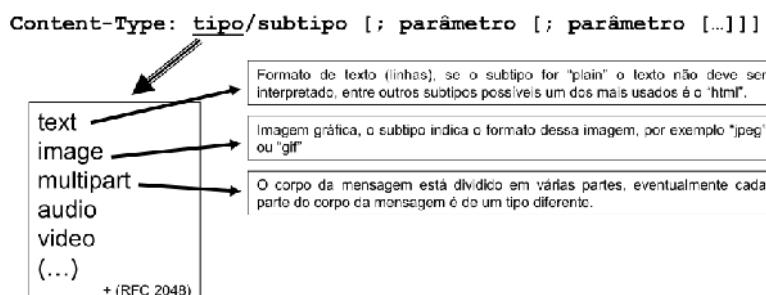
Uma mensagem de correio electrónico é identificada como estando em formato MIME através da presença do campo "MIME-Version" no cabeçalho, a versão actualmente em uso é a "1.0".

As mensagens 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.

MIME – “Content-Type”:

O campo de cabecalho "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, por exemplo:

Content-Type: text/html; charset=us-ascii

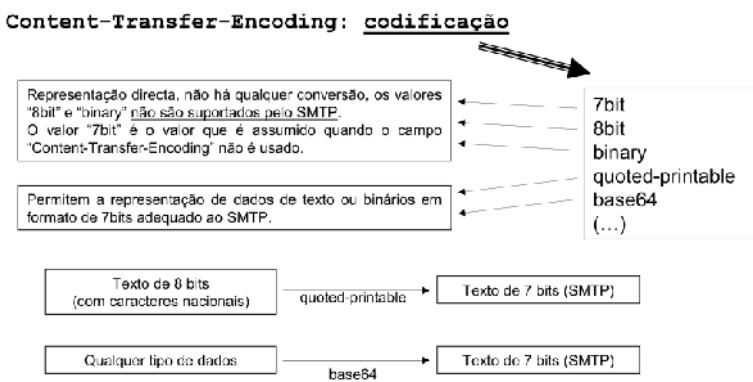
O campo "Content-Type" é importante porque permite à aplicação de destino interpretar os dados para os apresentar correctamente ao utilizador.

MIME = "Content-Type: multipart";

O conteúdo "multipart" é bastante usado porque permite transportar numa única mensagem vários tipos de conteúdos diferentes em partes separadas. Exemplo:

MIME – “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.



MIME – “CONTENT-TRANSFER-ENCODING: QUOTED-PRINTABLE”:

O objectivo da representação em formato “quoted-printable” 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 excepção 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.

MIME – “CONTENT-TRANSFER-ENCODING: BASE64”:

A representação em formato “base64” pode ser usada para qualquer tipo de dados. A sua maior desvantagem é que os dados codificados ocupam cerca de 33% mais espaço do que os dados originais.

Foi escolhido um conjunto de 64 caracteres adequado: “A..Za..z 0..9 + / ”

Com 64 caracteres pode-se representar qualquer conjunto de 6 bits, $\log_2(64) = 6$.

Os octetos de entrada são agrupados em conjuntos de 3, cada 3 octetos (24 bits) vão produzir 4 caracteres no texto codificado. O texto codificado está limitado a linha de 76 caracteres, mas na descodificação as mudanças de linha são ignoradas.

O sinal “=” é usado para indicar que o alinhamento do fim da mensagem não corresponde a 24 bits, nesse caso procede-se a um enchimento (“padding”) com bits 0, serão usados um ou dois sinais “=” conforme tenham sido necessário um enchimento de 8 ou 16 bits zero.

T13

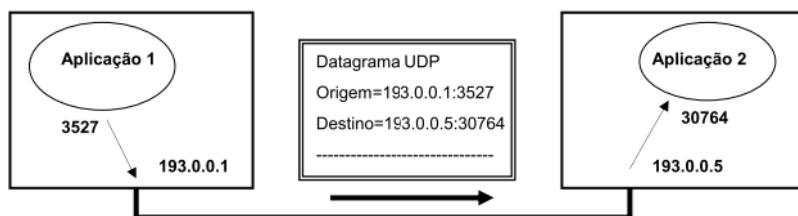
Desenvolvimento de aplicações de rede UDP e TCP

PROTOCOLO UDP (“USER DATAGRAM PROTOCOL”):

Trata-se de um serviço de “datagramas”, ou seja, que 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.



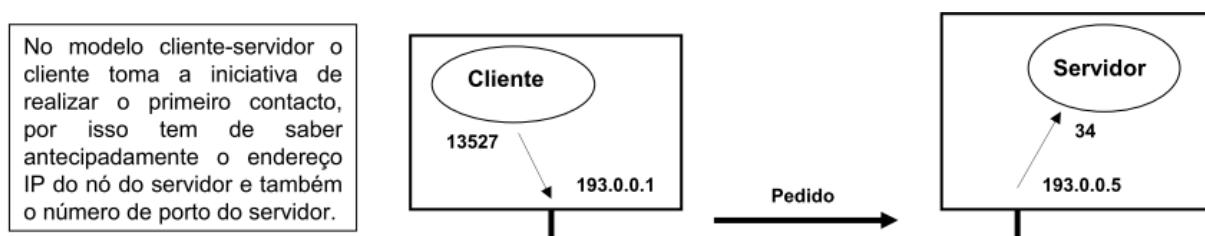
"SOCKETS" DE REDE EM UDP:

Sob o ponto de vista de desenvolvimento de aplicações, as interacções com a rede baseliam-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 efectuando 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 RECEPÇÃO DE DATAGRAMAS UDP:

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.

Tratando-se de um serviço de "datagramas" é possível enviar em "broadcast", basta especificar como destino o endereço de "broadcast" de uma rede IPv4, ou "255.255.255.255" para "broadcast" na rede local.

Sob o ponto de vista da aplicação, a recepção é normalmente síncrona, ou seja a operação de recepção bloqueia a execução da aplicação (processo ou thread) até que seja recebido um datagrama.

Após a recepção de um "datagrama" UDP a aplicação receptora 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 excepção 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.

PROTOCOLO TCP ("TRANSMISSION CONTROL PROTOCOL"):

O protocolo TCP proporciona um serviço de qualidade significativamente superior ao do protocolo UDP.

O TCP permite criar ligações lógicas bidireccionais entre aplicações residentes em nós de rede distintos. Estas ligações lógicas 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 transacções 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.

As transacções de dados via TCP apenas são possíveis após uma fase prévia de estabelecimento da ligação TCP (“conexão TCP”).

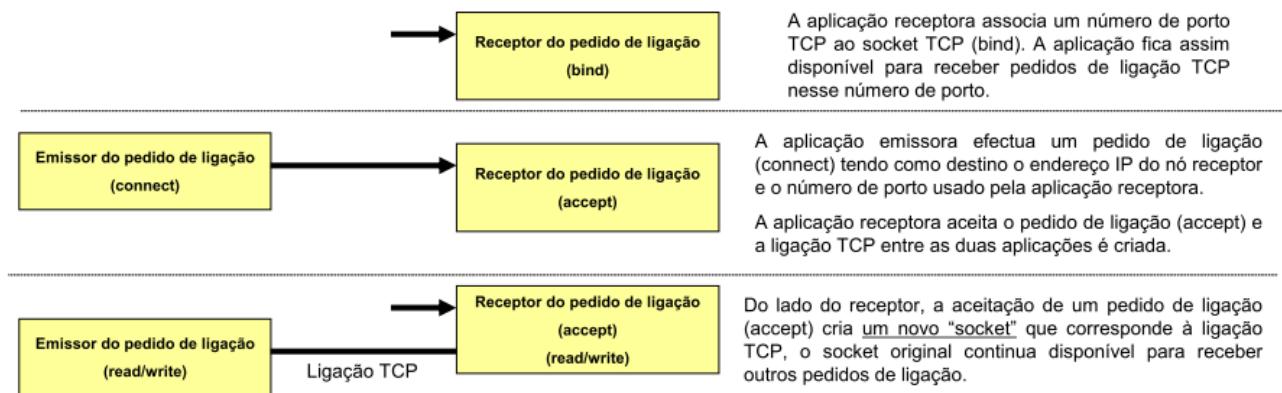
Para ser possível estabelecer uma conexão TCP entre duas aplicações cada uma delas tem de assumir um papel diferente:

- Uma das aplicações aceita o estabelecimento da ligação TCP (Servidor TCP)
- A outra aplicação pede o estabelecimento da ligação TCP (Cliente TCP).

ESTABELECIMENTO DA LIGAÇÃO TCP:

Para ser possível estabelecer a ligação TCP entre duas aplicações, uma delas assume o papel de receptor 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.



As operações de envio e recepção de dados através de ligações TCP são realizadas em fluxo de bytes, normalmente são usadas funções semelhantes às usadas nas operações de leitura e escrita em ficheiros e “pipes”.

A aplicação receptora associa um número de porta TCP ao socket TCP (bind). A aplicação fica assim disponível para receber pedidos de ligação TCP nesse número de porta.

A aplicação emissora efectua um pedido de ligação (connect) tendo como destino o endereço IP do nó receptor e o número de porta usado pela aplicação receptora.

A aplicação receptora aceita o pedido de ligação (accept) e a ligação TCP entre as duas aplicações é criada.

No lado do receptor, a aceitação de um pedido de ligação (accept) cria um novo “socket” que corresponde à ligação TCP, o socket original continua disponível para receber outros pedidos de ligação.

As duas extremidades da ligação TCP são acessíveis, no emissor através do “socket” usado para efectuar o pedido de ligação e no receptor através do novo “socket” criado quando o pedido de ligação foi aceite.

LIGAÇÕES TCP – CANAIS DEDICADOS:

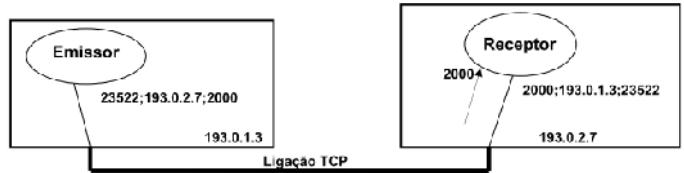
Uma ligação TCP constitui um canal de comunicação dedicado entre duas aplicações, para todos os efeitos pode ser tratado como um “pipe” bidireccional.

Antes de a ligação TCP ser estabelecida, cada um dos “sockets” tem associado a si apenas o número de porta local.



Após o estabelecimento da ligação TCP, os “sockets” correspondentes à ligação têm associados também o endereço IP remoto e o número de porta remoto.

Pode observar-se que no receptor existem “sockets” diferentes associados ao mesmo número de porta 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 porta de destino dos dados corresponde ao porta local, o endereço IP remoto corresponde ao endereço de origem dos dados e o número de porta remoto corresponde ao número de porta de origem dos dados.

UDP – ENVIO E RECEPÇÃO DE “DATAGRAMAS”:

Uma vez que não se trata de um serviço com ligação, assim que é associado um número de porta local ao “socket” (bind), ele encontra-se disponível para receber “datagramas” da rede e enviar “datagramas”.

Todas as operações de envio e recepção de rede são geridas pelo sistema operativo através de filas FIFO. Isto significa que mesmo que a aplicação não solicite explicitamente a recepção 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 recepção.

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 recepção de um "datagrama" devolve um bloco de dados com determinada dimensão e o endereço de origem do "datagrama" (endereço IP + número de porto UDP).

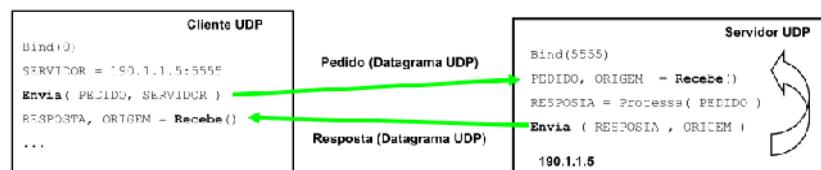
Normalmente as recepções de dados são bloqueantes, ou seja, se não existir nenhum "datagrama" na fila, a operação de recepção bloqueia o processo/thread até que chegue um "datagrama".

UDP – SERVIÇO NÃO FIÁVEL:

O serviço UDP é de utilização muito simples, mas não oferece qualquer tipo de garantias, cabe às aplicações (protocolo de aplicação) resolver os problemas que daí possam resultar.

Não havendo qualquer garantia que os "datagramas" emitidos cheguem ao destino, nem sequer qualquer "feedback" relativamente à entrega, as aplicações pouco cuidadosas podem facilmente ficar comprometidas.

A figura seguinte representa a interacção normal entre um cliente e um servidor UDP:



Neste caso as consequências da falta de fiabilidade do UDP manifestam-se no cliente que confia que vai chegar uma resposta. Se a resposta não chegar o cliente fica "eternamente" bloqueado na "system-call" Recebe(). Isto pode acontecer por vários motivos: o pedido não foi entregue; o servidor falhou; a resposta não foi entregue.

CLIENTES UDP – TOLERÂNCIA A FALHAS:

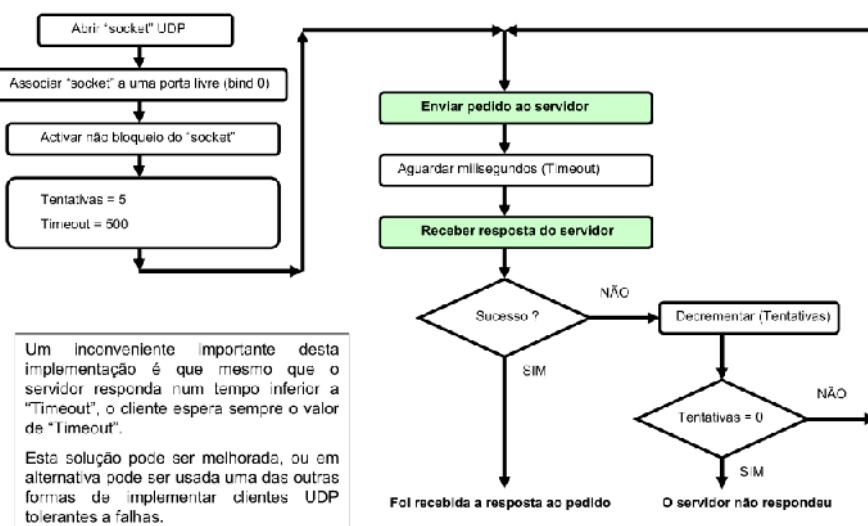
No contexto do modelo cliente-servidor, as limitações do protocolo UDP levam a que cada pedido seja tratado independentemente de pedidos anteriores (servidor idempotente). Sob o ponto de vista de falhas na entrega de "datagramas" o servidor fica numa posição cómoda pois o seu funcionamento nunca será afectado 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 recepção da resposta definindo um tempo máximo para essa operação ("timeout").

Existem várias formas de implementar esta funcionalidade:

- 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 recepção em sucessivos intervalos de tempo, perfazendo no total o "timeout".
- Threads – uma outra possibilidade é criar um "thread" para efectuar a recepção 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 – alguns 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.

CLIENTE UDP TOLERANTE A FALHAS – SOCKET NÃO BLOQUEANTE:



UDP – ENVIO EM “BROADCAST”:

Algo que não é possível num serviço com ligação como o TCP, mas é possível num serviço de “datagramas” como o UDP é o envio de dados para um endereço de “broadcast”.

Quando um “datagrama” é enviado para o endereço de “broadcast” todos os nós da rede correspondente vão receber o “datagrama” no número de porto especificado como destino.

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 máscara 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.

Uma das aplicações mais importantes do envio de “datagramas” UDP em “broadcast” é permitir a um cliente contactar um servidor que se encontra na rede local, mas cujo endereço desconhece.

Esta facilidade é usada em muitos ambientes de rede local como por exemplo Windows/NetBIOS para localizar de forma expedita servidores na “vizinhança da rede”.

UDP – ENVIO EM “BROADCAST” – LOCALIZAÇÃO DE APLICAÇÕES:

Mesmo que o protocolo de aplicação utilize TCP, nada impede que numa fase prévia, para efeito de localização das aplicações (descoberta do endereço IPv4) se use UDP em “broadcast”.

Numa arquitectura cliente-servidor pode adopta-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 respectivos endereços IPv4.
- Pedido de servidores – O cliente envia em “broadcast”, um “datagrama” UDP para um número de porto pré combinado, onde solicita servidores. Os servidores presentes na rede respondem, permitindo igualmente ao cliente a construção de uma lista de servidores disponíveis e respectivos endereços IPv4.

Em qualquer um dos modelos o cliente fica a conhecer uma lista de servidores disponíveis e respectivos endereços IPv4 podendo depois escolher um e comunicar com ele usando UDP ou TCP.

UDP – TAMANHO DOS “DATAGRAMAS”:

Para além da falta de garantias ou “feedback” da entrega dos “datagramas” UDP, o desenvolvimento de aplicações pode deparar-se com um outro problema:

Que volume de informação pode ser colocado em cada “datagrama” UDP?

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, actualmente 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).

Quando as transacções ultrapassam volumes de 512 bytes só há duas alternativas possíveis:

- Esquecer os “datagramas” UDP e optar antes por conexões TCP.
- Dividir a informação por vários “datagramas” UDP.

UDP – TRANSACÇÕES EM MÚLTIPLOS “DATAGRAMAS”:

Normalmente quando se verifica que as transacções de um protocolo de aplicação envolvem volumes de dados superiores a 512 bytes, opta-se por usar TCP e não UDP.

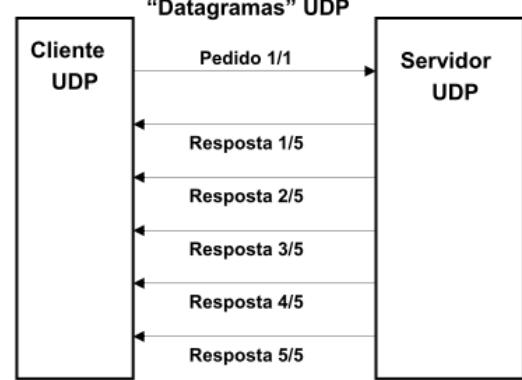
Sendo o UDP um protocolo sem ligação sem qualquer garantia de entrega ou ordem de entrega, a divisão de um bloco de dados por vários “datagramas” UDP vai obrigar as aplicações a realizarem uma série de procedimentos de verificação.

No mínimo, o receptor terá de saber o número total de “datagramas” e estes terão de ser numerados.

Exemplo:

O protocolo de aplicação poderá ser mais ou menos refinado em termos de recuperação de falhas:

- A falha de entrega de um "datagrama" pode levar à nulidade de toda a transacção e levar à sua repetição.
- O cliente, ao detectar a falha de entrega de um "datagrama", pode solicitar ao servidor o envio apenas do "datagrama" em falta.



SOCKETS UDP – ASSOCIAÇÃO A ENDEREÇOS REMOTOS:

O UDP é 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 o "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 recepção:

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".

RECEPÇÃO – 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 recepção. Os servidores UDP multi-processo aproveitam esta propriedade.

Ligações TCP – Envio e recepção de dados:

O protocolo TCP tem a vantagem de garantir a entrega fiável dos dados na ordem exacta em que são emitidos, sem qualquer limitação relativamente ao volume de dados enviados ou recebidos. O envio e recepção de dados através de uma ligação TCP são realizados byte a byte.

Tem de haver uma correspondência exacta 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.

LIGAÇÕES TCP - ENVIO E RECEPÇÃO - PROTOCOLO DE APLICAÇÃO:

Um protocolo de aplicação é um conjunto de regras que duas aplicações de rede usam para poderem dialogar sem ambiguidades. Define os procedimentos de cada uma das aplicações e as trocas de informação que têm lugar nas várias fases dos procedimentos.

Quando o protocolo de aplicação usa uma ligação TCP é da sua responsabilidade garantir que vai existir uma correspondência exacta entre as operações de escrita e operações de leitura, realizadas nas extremidades opostas da ligação.

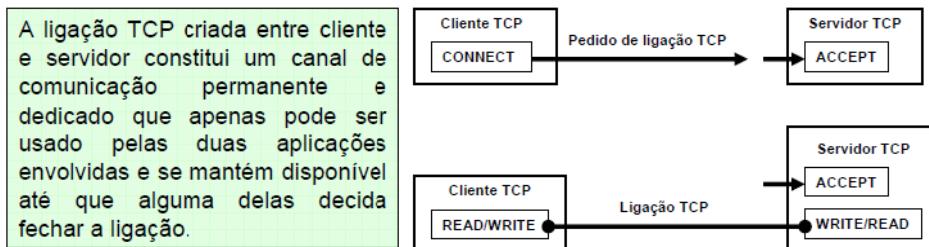
Existem três abordagens ao problema que podem ser usadas isoladamente ou combinadas:

- 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 desperdício 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 receptor do tamanho da mensagem que se segue. Com esse conhecimento o receptor pode solicitar a leitura do número exacto 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 receptor efectua 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.

SERVIDORES TCP:

Um servidor TCP é antes de mais uma aplicação que aceita pedidos de ligação TCP num número de porto definido no protocolo de aplicação.

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.

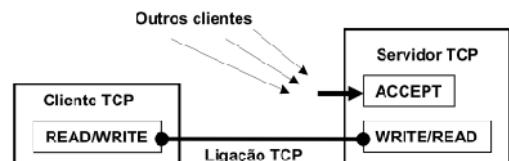


O formato das mensagens trocadas entre cliente e servidor através da ligação TCP é definido pelo protocolo de aplicação, mas graças ao carácter persistente da ligação TCP constitui uma sessão no sentido em que estas interacções podem não se limitar apenas a um pedido e respectiva resposta.

SERVIDORES TCP MULTI-PROCESSO:

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.

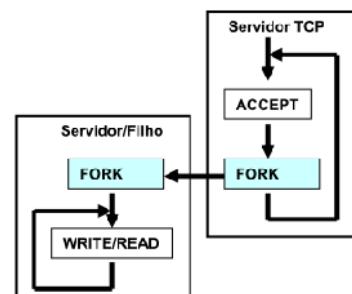


Ou seja o servidor tem de lidar com dois "sockets" aceitando novos clientes no "socket" original (accept) e dialogando com o cliente no novo "socket" correspondente à ligação TCP.

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.

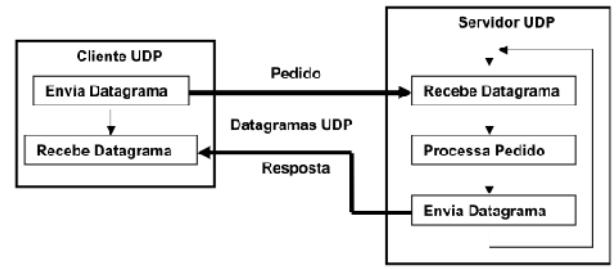


SERVIDORES UDP:

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, um pedido só é atendido pelo servidor depois de os pedidos anteriores terem sido processados e respondidos.



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 seleccionar o contexto correcto tendo como critério o endereço do cliente (endereço + número de porto).

SERVIDORES UDP MULTI-PROCESSO:

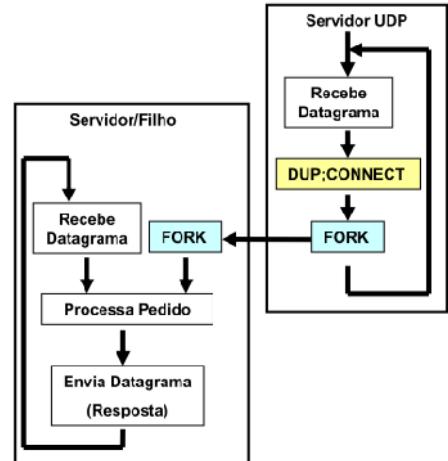
Os servidores UDP normalmente são implementados num único processo, cada porto UDP é um ponto de recepção único que não pode ser usado simultaneamente por vários processos sem conflitos. Com vários processos a ler do mesmo "socket" nunca se saberia qual dos processos receberia o "datagrama".

Contudo através da associação do "socket" UDP a um endereço remoto obtém-se um ponto distinto de recepção pois esse "socket" deixa de receber "datagramas" de outros endereços (endereço + porto) que não o que foi associado.

Torna-se então possível uma implementação multi-processo ao estilo TCP:

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".



PROTOCOLO DE APLICAÇÃO:

Sob o ponto de vista das redes de computadores, um protocolo é um conjunto de regras que visam permitir a troca de informação sem ambiguidades entre duas entidades que comunicam através de uma infra-estrutura 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 exacta, possível de:

- Todos os formatos de mensagens usados nas diferentes fases dos diálogos.
- Todas as diálogos e acções possíveis, os respectivos objectivos e resultados possíveis.
- Procedimentos de detecção e recuperação de erros.

O desenvolvimento de um protocolo de aplicação deve ter em consideração os seus objectivos iniciais, como por exemplo que tipos de dados vão ser transferidos.

Os protocolos de aplicação devem ser flexíveis permitindo a implementação de novas funcionalidades mantendo a compatibilidade com versões anteriores:

- O primeiro elemento das mensagens deve identificar a versão do protocolo.
- O formato geral das mensagens deve ser suficiente flexível para comportar novos formatos específicos.

RECEPÇÃO 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 recepção 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.
- Necessitam de executar outras tarefas enquanto não chegarem dados.

Soluções (recepção 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” (Ex.: select()) – 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.