Redes de Computadores 2

Perguntas-treino para teste 2

2018/2019

1. Qualidade de Serviço na Internet

Enumerar as limitações das redes "best-effort"no que respeita à qualidade de serviço e ao suporte a serviços multimédia.

As principais limitações das redes "best-effort" são:

- O único requisito a satisfazer é o da conectividade sendo apenas necessário garantir que as entidades conseguem comunicar entre si;
- A qualidade da comunicação fica a depender obviamente do estado da rede (melhor em situações de pouca carga, pior em situações de congestão);
- Os recursos são partilhados de forma equitativa entre os vários fluxos de tráfego não existindo mecanismos que permitam tratar melhor alguns tipos de tráfego em detrimento de outros.

Estas limitações podem originar problemas tais como: atraso excessivo de fim-a-fim, variações no atraso (jitter), perdas excessivas de pacotes.

Além disso, este tipo de rede é insuficiente na Internet atual, devido a:

- As novas aplicações (multimédia e tempo real) têm outro tipo de requisitos;
- A Internet é actualmente uma infra-estrutura comercial.

Descrever as métricas relevantes para a qualidade de serviço: atraso fim-a-fim, variações no atraso (jitter), largura de banda e taxa de perdas.

- Atraso excessivo de fim-a-fim: Os atrasos de fim-a-fim são a acumulação da transmissão, processamento, atraso de filas em routers e propagação de atrasos nas conexões;
- Variações no atraso (jitter): Este problema está relacionado com o facto de o tempo de quando o pacote é gerado até este ser recebido poder variar de pacote para pacote;
- Largura de banda: A largura de banda está relacionada com a quantidade em bits/s que a rede suporta (por exemplo, uma rede com uma largura de banda 1 Mbps, isso significa poder transferir cerca de 1 megabit por segundo).
- Taxa de perdas: Esta métrica refere-se à percentagem de pacotes perdidos numa transmissão, é calculada dividindo o número de pacotes perdidos pelo número de pacotes enviados.

Explicar as diferenças entre as abordagens de Serviços Integrados (IntServ) e de Serviços Diferenciados (DiffServ).

Serviços Integrados (IntServ):

O IntServ ou Serviços Integrados é um modelo de implementação da QoS que propõe que as garantias de Qualidade de Serviço sejam disponibilizadas de forma individualizada por cada sessão estabelecida. As características principais são:

- Implica software novo e complexo nos encaminhadores;
- As aplicações reservam largura de banda fim-a-fim, o que implica alterações fundamentais no modo de funcionamento actual da Internet;
- Mais difícil de implementar, sendo que se o router for abaixo, perde as configurações e as ligações entre routers perdem a largura de banda que tinham alocada;
- Recorre à reserva de recursos: Os encaminhadores mantém informação de estado como se se tratasse de um circuito virtual com os recursos alocados por cada sessão estabelecida;
- Tem uma componente responsável por admitir ou negar o estabelecimento de novas sessões (controlo de admissão), conseguindo desta forma garantir que o aparecimento de novas sessões não vai prejudicar as anteriormente estabelecidas.

Serviços Diferenciados (DiffServ):

O DiffServ ou Serviços Diferenciados trabalha a QoS em relação aos grandes fluxos de dados. Desta forma o DiffServ requer a negociação dos recursos para todos os pacotes de uma rede. As principais características são:

- Poucas alterações à infraestrutura actual de forma a disponibilizar diferentes classes de serviços.
- São mais fáceis de implementar, sendo que aquilo que fazem é verificar de que tipo é o pacote e tratá-lo de acordo com o tipo, além disso neste tipo de serviço, caso o router seja reiniciado, mantém as configurações.
- Marcação e classificação de tráfego em classes de serviço;
- Tratamento diferenciado dos pacotes (de acordo com a classe).

Neste tipo de serviço, são necessários routers encaminhadores:

- Encaminhadores de periferia:
  - Perfil de tráfego permitido ao utilizador;
  - Marcação de pacotes.
- Encaminhadores interiores "stateless":
  - Sem noção de sessão;
  - Expedição: tráfego dentro do perfil tem prioridade sobre o fora de perfil

Parâmetro de comparação	SD	SI
Escalabilidade	Melhor	Pior
Protocolo de sinalização	Não	Sim
Granularidade de alocação de recursos	Classe	fluxo
Informação de estado	O(classes)	O(fluxos)
Agregação de fluxos em classes	Sim	Prevista
Acções de controlo de tráfego	Periferia da rede	Toda a rede
Alteração na base instalada	Mínima	Grande
Implementação global	Acessível	Difícil
Garantia de QoS	Pior	Melhor

Explicar a necessidade de ser classificar e policiar o tráfego que entra na rede.

A classificação do tráfego da rede consiste em distinguir os diferentes serviços e protocolos disponíveis na rede. Normalmente, assim que os pacotes são classificados como pertencentes a determinado serviço ou protocolo é lhes atribuída uma *flag*(policiamento), que ajuda os *routers* a decidirem quais as melhores políticas a usar para esses tráfegos.

Portanto, a necessidade de classificar e policiar o tráfego da rede, está sobretudo relacionada com o facto de numa rede, o tráfego existente ser heterogéneo e consistir num conjunto de pacotes pertencentes a diferentes serviços e aplicações, cada um com necessidades de largura de banda, e por isso, serem necessários procedimentos para gerir essa mesma largura de banda, limitando o tráfego para que não exceda os parâmetros acordados. Sem estes procedimentos, a qualidade e usabilidade das aplicações e serviços da rede será comprometida. Exemplos de mecanismos de policiamento, são: token bucket e a medição contínua da taxa de transmissão.

Explicar a necessidade de mecanismos de escalonamento e controlo de admissão.

Os mecanismos de escalonamento, que determina a ordem pela qual os pacotes dos diferentes fluxos são retransmitidos nas ligações de saída de um encaminhadores são essenciais, uma vez que condicionam o atraso e a taxa de transmissão experimentados por cada fluxo.

Os mecanismos de controlo de admissão são necessários, uma vez que não é possível suportar tráfego para além da capacidade das ligações. Implementando este mecanismo, o fluxo de aplicação declara as suas necessidades e a rede pode bloquear a comunicação por falta de recursos.

Descrever o funcionamento dos mecanismos de escalonamento: FIFO, Round Robin, Weighted Fair Queuing

FIFO:

- Os pacotes são enviados pela ordem de chegada
- Política de descartagem: se quando chega um novo pacote a fila de espera está cheia é necessário descartar um pacote.

Escalonamento com base em prioridades:

- Os pacotes com maior prioridade são transmitidos primeiro;
- A classe é determinada através de uma marca destinada especificamente a esse efeito ou através de outros campos do cabeçalho (endereço fonte, endereço destino, números de porta, etc.).

Escalonamento Round Robin:

- Múltiplas Classes;
- As diferentes filas são "investigadas" ciclicamente sendo enviado um pacote de cada classe, à vez (se estiver disponível).

Weighted Fair Queuing:

- Generalização do algoritmo de Round Robin;
- Cada classe tem direito a uma percentagem do tempo de cada ciclo, tempo esse de que dispõe para transmitir os pacotes que estão na fila (se existirem).

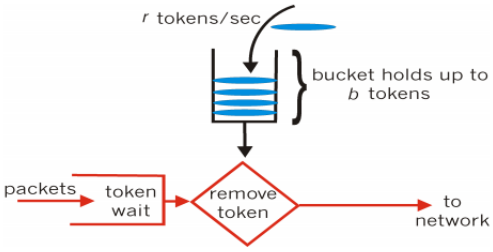
Para que estes mecanismos funcionem de forma eficiente devem haver outros para gerir filas de espera, que têm como objetivos:

- Controlar o comprimento das filas de espera;
- Evitar que elas atinjam o seu comprimento máximo com o objectivo de evitar a congestão:
  - Descartar os pacotes antes das filas encherem;
  - As fontes detectam perdas e abrandam a transmissão.

Mecanismos: RED (Random Early Detection), RIO (Random Early Decetection with In and Out)

Descrever o funcionamento dos mecanismos de policiamento: Token Bucket.

O Token Bucket é um mecanismo usado fundamentalmente para limitar o tráfego de uma rede, para que este não exceda os parâmetros acordados.



Este mecanismo limita a entrada tendo em conta o tamanho do burst (número máximo de pacotes enviados consecutivamente) e a taxa média especificada (quantos pacotes pode ser enviados por unidade de tempo).

Para o pacote passar, tem que haver tokens no bucket, funcionando o token como um "bilhete".

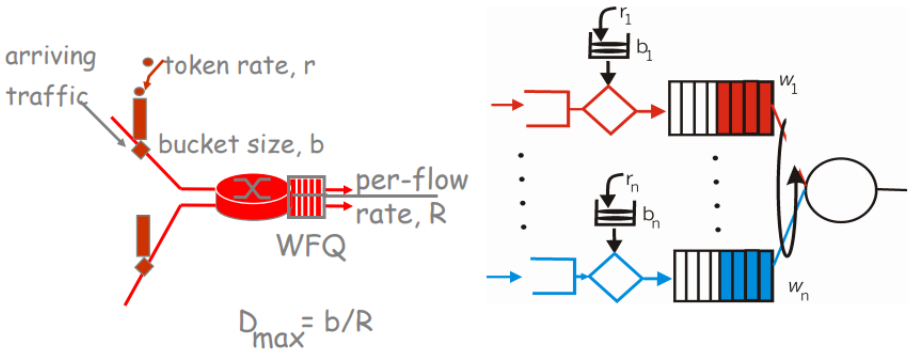
Se a taxa de transmissão do pacote for inferior à do bucket, o pacote passa.

Se a taxa de transmissão do pacote for superior à do bucket, os pacotes são descartados, ou ficam na fila de espera.

Este mecanismo tem as seguintes características:

- Bucket pode conter b tokens;
- Tokens gerados à taxa r token/seg a menos que o bucket esteja cheio;
- No intervalo de tempo t o número de pacotes admitidos é menor ou igual a (r t + b).

Para garantir a QoS, o token bucket pode ser combinado com o mecanismo de escalonamento WFQ disponibilizando um limite superior no atraso que pode ser configurável.



2. Nível de transporte

Explicar o conceito de ligação fim-a-fim.

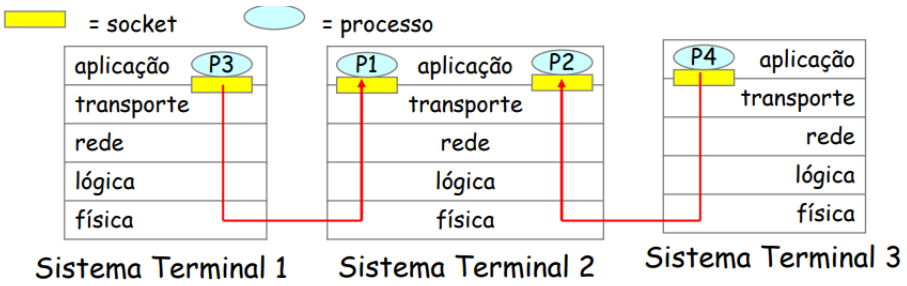
Neste tipo de ligações, os recursos específicos das aplicações, residem nos nós finais da rede, em vez de nos intermediários (routers e gateways), sendo que estes intermediários transmitem dados aleatoriamente (sem descriminação de pacotes), e por esta falta de descriminação, qualquer router intermediário pode ser substituído por outro, sem que ocorram falhas.

Nestas ligações:

- O emissor parte a mensagem gerada pela aplicação em segmentos que passa à camada de rede;
- O receptor junta os diferentes segmentos que constituem uma mensagem que passa à respectiva aplicação.

Pode-se por isso considerar os conceitos de multiplexagem e desmultiplexagem:

- Multiplexagem no emissor: Recolher os dados de diferentes sockets e delimitá-los com os respectivos cabeçalhos construindo os respectivos segmentos
- Desmultiplexagem no recetor: Entregar os diferentes segmentos ao socket correcto.



Descrever as diferenças entre serviços orientados à conexão e não orientados à conexão.

- Serviços orientados à conexão: Neste tipo de serviços, quando os dispositivos comunicam, existe um *handshaking*, no qual são definidos os parâmetros da comunicação. Este tipo de sistemas, só podem funcionar em ambientes de comunicação bidireccional.Um exemplo deste serviço é o protocolo TCP.

  As principais características deste serviço são:

  - Estabelece, mantém e finaliza a conexão lógica entre processos;
  - Tem uma vasta variedade de aplicações;
  - Mais comum;
  - Fornece um serviço fiável.

- Serviços não orientados à conexão: Neste tipo de comunicação, a informação é transmitida da origem para o destino, sem verificar se o destino ainda está lá, ou se está preparada para receber dados. Exemplos deste tipo de serviço, são os protocolos ICMP, UDP e SNMP.

  A comparação entre os dois tipos de serviço pode ser vista na figura seguinte.

BASE DE COMPARAÇÃO	SERVIÇO ORIENTADO A CONEXÃO	SERVIÇO NÃO ORIENTADO A CONEXÃO
Requisito de conexão prévia	Necessário	Não requerido
Confiabilidade	Garante a transferência confiável de dados.	Não garantido.
Congestionamento	Improvável	Ocorre provavelmente.
Retransmissão de dados perdidos	Possível	Praticamente, não é possível.
Adequabilidade	Adequado para comunicação longa e estável.	Adequado para transmissão em rajada.
Sinalização	Usado para estabelecimento de conexão.	Não há conceito de sinalização.
Encaminhamento de pacotes	Os pacotes viajam sequencialmente até o nó de destino e seguem a mesma rota.	Os pacotes chegam ao destino aleatoriamente sem seguir o mesmo caminho.
Atraso	Há um atraso na transferência de informações, mas quando a conexão é estabelecida, uma entrega mais rápida pode ser alcançada.	Devido à ausência de fase de estabelecimento da conexão, a transmissão é mais rápida.
Alocação de recursos	Precisa ser alocado.	Nenhuma alocação prévia do recurso é necessária.

Comparar os serviços de transporte com os serviços de rede.

- Serviços de rede: Este tipo de serviços destinam-se a levar pacotes de um host até outro.
- Serviços de transporte: Permitem distinguir entre diferentes aplicações/processos a serem executados no mesmo sistema terminal (host), além de que este tipo usa e melhora os serviços disponibilizados pela camada de rede, tais como:
  - a troca de dados fiável e ordenada (TCP), como por exemplo, o controlo de congestionamento, o controlo de fluxo, ou o estabelecimento de ligação;
  - troca de dados não fiável e desordenada (UDP).

  No entanto, não estão disponíveis serviços como a garantia de atraso máximo e largura de banda mínima.

Descrever o conceito de porta de protocolo e explicar a sua função.

As portas, em redes de computadores, podem ser definidas como canal de comunicação que identifica um processo ou um tipo de serviço de rede específicos.

Todos os dispositivos de rede usam portas, e devem poder alterar que porta usam, em qualquer momento.

Inicialmente foram criadas para permitir vários programas usarem o mesmo endereço de IP.

As portas são numeradas de 1 a 65000, sendo algumas das portas mais conhecidas as portas 80 (serviço HTTP), 21 (serviço FTP), 22 (serviço SSH).

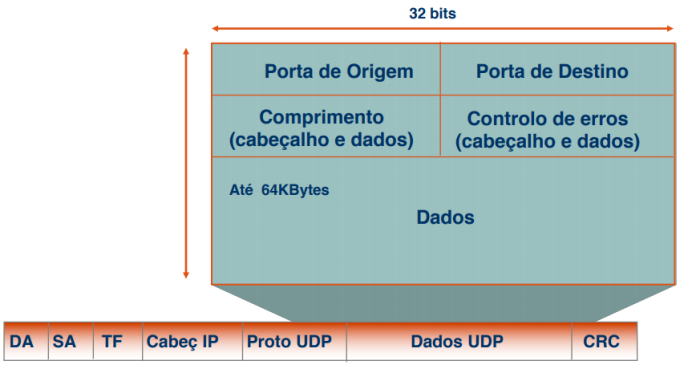
As portas são componentes essenciais das redes IP, sendo a sua função principal permitir que as aplicações presentes numa rede, partilhem recursos, sem interferirem umas com outras.

Descrever as funções do protocolo UDP.

As principais funções do protocolo UDP são:

- Protocolo de transporte fim-a-fim (não fiável);
- Orientado ao datagrama (sem conexão);
- Actua como uma interface da aplicação com o IP para multiplexar e desmultiplexar tráfego;
- Usa o conceito de porta / número de porta;
- Utilizado em situações que não justificam o TCP(por exemplo,TFTP, RPC, DNS);
- Também é utilizado em situações em que o tempo é crítico (exemplos: streaming de áudio e vídeo).

Identificar os vários campos de um datagrama UDP.



Descrever as características do serviço UDP e dar exemplos de aplicações para as quais este serviço é adequado.

As características principais do UDP são:

- Serviço "best effort", sendo que os datagramas UDP podem ser:
  - perdidos;
  - entregues fora de ordem.
- Não orientado à conexão:
  - não há handshaking entre emissor e o receptor;
  - cada segmento UDP é processado de forma independente.
- Serviço não fiável:
  - sem garantia de entrega dos datagramas;
  - sem garantia de entrega ordenada dos datagramas.
- Com detecção de erros (checksum);
- Sem correcção de erros;
- Não há estabelecimento de conexão (menos atraso);
- Simples: não há variáveis de estado;
- Tamanho do cabeçalho é pequeno;
- Não há controlo do congestionamento: UDP envia sempre que precisa.

As aplicações para as quais o UDP é adequado são:

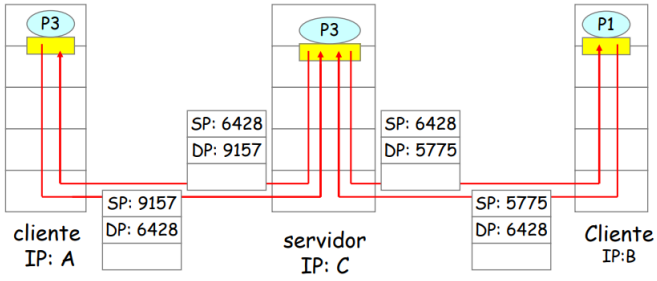
- Domain Name Service (DNS);
- IPTV;
- SNMP.

Em relação à desmultiplexagem no UDP:

1. O socket UDP é identificado através de dois números: endereço IP destino, e número de porta destino (dest IP address, dest port number);

2. Quando um Sistema Terminal recebe um datagrama UDP verifica qual o número da porta destino que consta do datagrama UDP e redirecciona o datagrama para o socket com esse número de porta;

3. Datagramas com diferentes endereços IP origem e/ou portas origem podem ser redireccionados para o mesmo socket.



Descrever as características do serviço TCP e compará-las com as do serviço UDP.

As principais características do serviço TCP são:

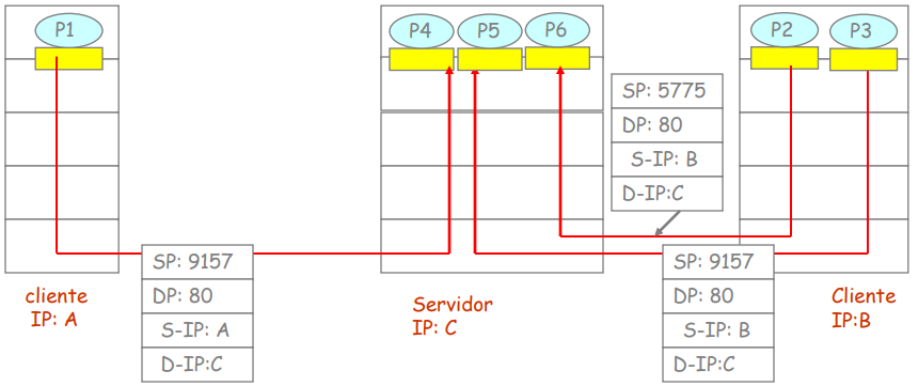
- Serviço full-duplex;
- Serviço fiável devido a:
  - Garantia de entrega dos dados;
  - Garantia de entrega ordenada;
  - Detecção de erros;
  - Correcção de erros por retransmissão.
- Orientado à conexão, seguindo este processo:
  1. Estabelecimento de conexão entre dois processos;
  2. Transmissão de dados (nos dois sentidos);
  3. Finalização da conexão.
- Com controlo de fluxo fim-a-fim:
  - Buffers nos dois extremos da ligação.

Para além disto, as funções do TCP são:

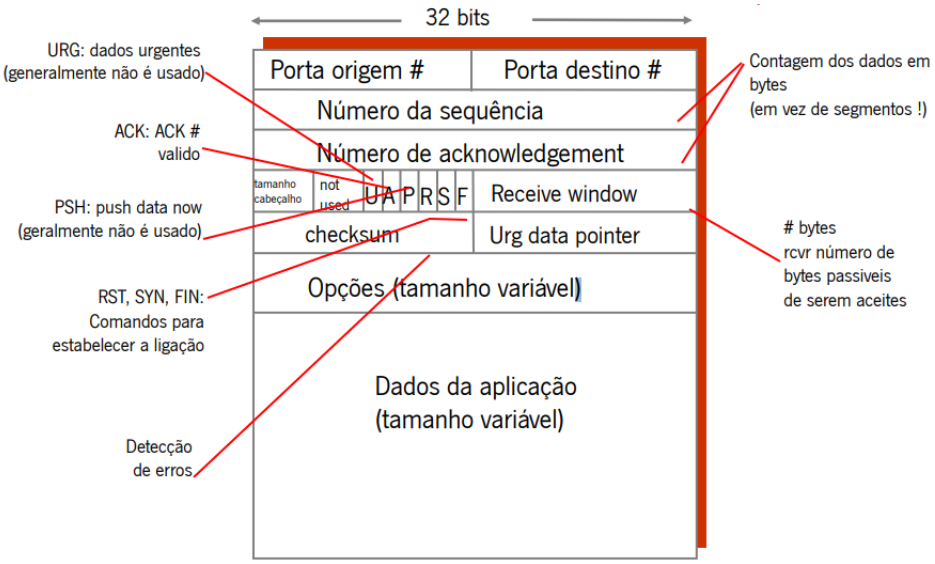
- Transporte fiável de dados fim-a-fim (aplicações);
- Efectua associações lógicas fim-a-fim: conexões;
- Cada conexão é identificada por um par de sockets: (IP_origem:porta_origem, IP_destino:porta_destino);
- Uma conexão é um circuito virtual entre portas de aplicações (também designadas portas de serviço);
- Multiplexa os dados de várias aplicações através de número de porta;
- Efectua controlo de fluxo, de congestionamento e de erros.

Em relação à desmultiplexagem no TCP:

1. O socket TCP é identificado:
   - Endereço IP de origem;
   - Número de porta de origem;
   - Endereço IP de destino;
   - Número de porta de destino.
2. O sistema terminal que recebe os segmentos usa estes 4 parâmetros para encaminhar os segmentos para o socket apropriado:
   - Um Servidor suporta múltiplos sockets em simultâneo;
   - Servidores Web têm diferentes sockets para cada conexão do cliente.



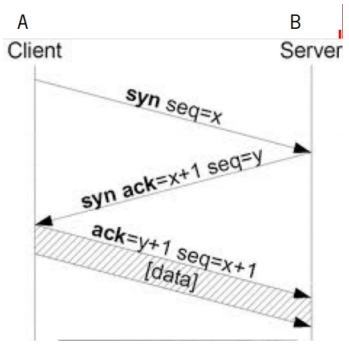
Identificar as funções dos vários campos de um segmento TCP.



Descrever o processo de criação e fecho de ligações TCP.

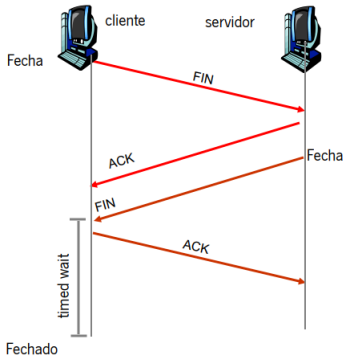
Para criar as ligações ocorre inicialmente um processo chamado ***handshake***, na seguinte ordem:

1. O cliente envia segmento SYN para o servidor:
   - Especifica o número inicial da sequência - seq #;
   - Não contém dados.
2. O servidor recebe o SYN e responde com um segmento SYNACK:
   - Aloca espaço nos buffers;
   - Especifica o número de sequência inicial - seq. #.
3. O cliente recebe o segmento SYNACK, e responde com um segmento ACK que pode conter dados.



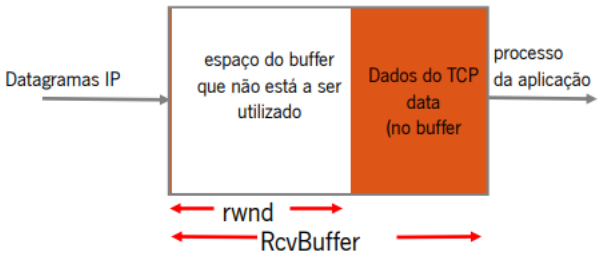
Para terminar a conexão:

1. O cliente envia um segmento de controlo TCP FIN para o servidor.
2. O servidor recebe o FIN, e responde com um ACK. Fecha a conexão e envia um FIN.
3. O cliente recebe o FIN, e responde com um ACK.
4. O servidor, recebe o ACK. A conexão fica fechada.



Explicar como funciona o processo de controlo de fluxo fim-a-fim.

O processo de controlo de fluxo existe para o transmissor não sobrecarregar o buffer do receptor com o envio de demasiados dados num curto espaço de tempo. Sendo que o lado do recetor de uma conexão TCP tem um buffer e supõe-se que o receptor TCP descarta os segmentos que chegam desordenados.



Espaço do buffer não utilizado: **rwnd = RcvBuffer-[LastByteRcvd - LastByteRead]**

O funcionamento do processo de controlo de fluxo é o seguinte:

O receptor: informa o transmissor do espaço do buffer que ainda não está a ser utilizado, incluindo o valor do rwnd no cabeçalho do segmento

O transmissor: limita o número de bytes não confirmados (unACKed) para rwnd:

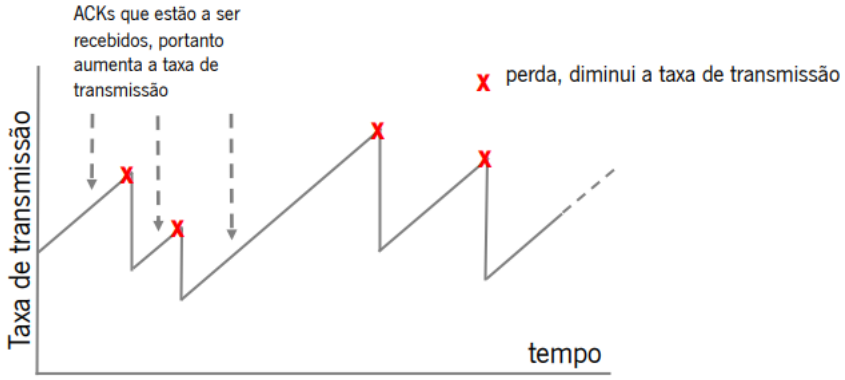
– Isto garante que o buffer do receptor não fica sobrecarregado.

Descrever a finalidade do processo de controlo de congestionamento.

O processo de controlo de congestionamento serve fundamentalmente, para o transmissor TCP poder enviar tão rápido quanto possível, mas sem causar o congestionamento da rede.

Num modelo descentralizado cada transmissor TCP configura a sua pópria taxa de transmissão com base no feedback implícito recebido:

- ACK: segmento recebido, a rede não está congestionada, portanto pode aumentar a taxa de transmissão;
- Segmento perdido: assume que a perda se deve ao congestionamento da rede, portanto diminui a taxa de transmissão;
- "Sonda a largura de banda": aumenta a taxa de transmissão se um ACK for recebido, até, eventualmente, ocorrer uma perda, depois diminui a taxa de transmissão (Continua a aumentar quando recebe um ACK, diminui quando ocorre uma perda).



O transmissor limita a taxa de transmissão limitando o número de bytes não confirmados em trânsito (in pipeline): **LastByteSent-LastByteAcked** $\leq$ **cwnd**

A cwnd é dinâmica, varia em função da percepção que o TCP tem do estado da rede.

Um segmento pode-se perder por:

- timeout: o receptor não responde – Reduz-se a cwnd para 1 ;
- 3 ACKs duplicados: Ao receber três ACKs duplicados, a entidade TCP retransmite o respetivo segmento (Fast Retransmit), diminui para metade o valor da janela de congestionamento (cwd) e entra numa fase que se designa por Fast Recovery:
  - A janela de congestionamento é incrementada por cada ACK duplicado recebido (começando logo pelos três ACKs duplicados);

  - Quando chega o ACK que confirma todos os dados, nomeadamente o segmento retransmitido, a janela volta ao valor inicial (que tinha quando entrou na fase de Fast Recovery), igual a metade do valor da janela de congestionamento antes do Fast Retransmit.

Quando um ACK é recebido, aumenta o cwnd:

- Fase slowstart: Aumenta exponencialmente, acontece quando iniciamos a conexão, ou quando ocorre um timeout;
- Fase congestion avoidance: Aumenta linearmente.

3. Nível de Aplicação

Distinguir entre aplicações baseadas no paradigma cliente-servidor e no paradigma peer-to-peer.

Arquitetura Cliente-Servidor:

É uma estrutura distribuída de aplicações que divide tarefas ou cargas de trabalho entre os provedores de um recurso ou serviço, chamados de servidores , e solicitantes de serviço, chamados de clientes.

Exemplos: Páginas Web, e-mail, network printing.

- Servidor:
  - Sempre ligado;
  - Com um endereço IP permanente;
  - Data centers para "escalar".
- Clientes:
  - Comunicam com o servidor;
  - Podem estar ligados de forma intermitente;
  - Podem possuir um endereço IP dinâmico;
  - Não comunicam diretamente uns com os outros.

Arquitetura P2P:

- Não há servidores permanentemente ligados;
- Os peers comunicam diretamente uns com os outros;
- Requerem serviços de peers e em troca disponibilizam serviços a outros peers:
  - escalável (novos peers trazem uma maior capacidades ao serviço, assim como novas necessidades).
- Os peers estão ligados de forma intermitente e mudam frequentemente de endereço IP;
- Gestão complexa.
- Os peers funcionam tanto como servidores ou como clientes.
- Exemplos: VoIP, Bitcoin, Streaming de áudio/vídeo, mensagens instantâneas.

Descrever a função e explicar o funcionamento do serviço DNS.

O serviço DNS é fundamentalmente uma base de dados distribuída implementada numa hierarquia de muitos servidores de nomes.

A função deste serviço é associar nomes de domínios mais facilmente memorizáveis a endereços IP numéricos, necessários à localização e identificação de serviços e dispositivos, processo esse denominado por resolução de nome (name resolution).

O DNS não pode ser um serviço centralizado/escalável porque:

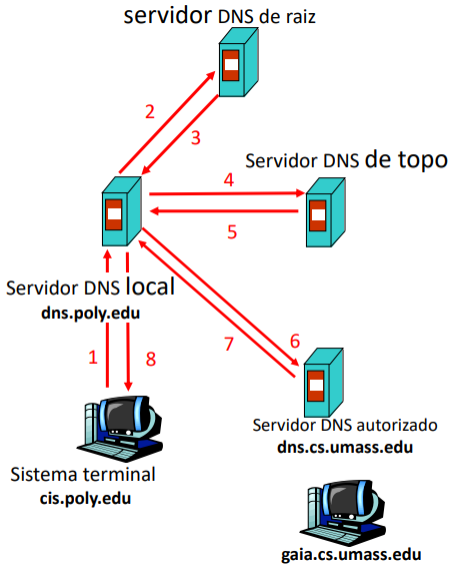
- Disponibilidade: se o único servidor de DNS falhasse, o serviço se tornaria indisponível para o mundo inteiro;
- Volume de tráfego: o servidor deveria tratar os pedidos DNS do planeta inteiro;
- Distância: grande parte dos utilizadores estaria muito distante do servidor, onde quer que ele fosse instalado, gerando grandes atrasos para resolver pedidos DNS;
- Difícil manutenção;

A solução encontrada foi usar uma base de dados distribuída e hierárquica. Os servidores DNS dividem-se nas seguintes categorias:

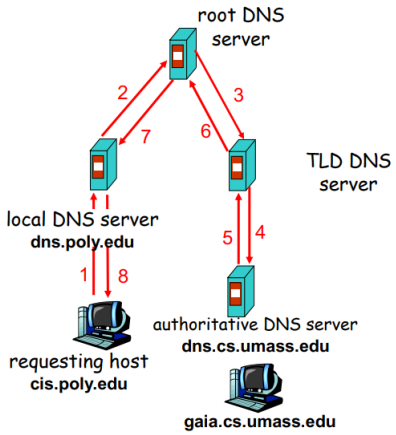
- Servidores-raiz;
- Servidores de domínio de topo;
- Servidores com autoridade.

O DNS pode funcionar de 2 formas:

- Pedido interactivo: o servidor contactado retorna o nome do servidor que deverá ser contactado: "Eu não sei este nome, mas pergunta a este servidor"



- Pedido recursivo: o servidor contactado contacta com outros servidores para retornar a resposta pretendida.



Depois de aprender o mapeamento, o servidor de nomes guarda em cache o mapeamento:

- Os registos em cache expiram passado algum tempo e são eliminados;
- os servidores TLD normalmente estão na cache dos servidores de nomes locais (evitando por isso pedidos aos servidores de raiz).

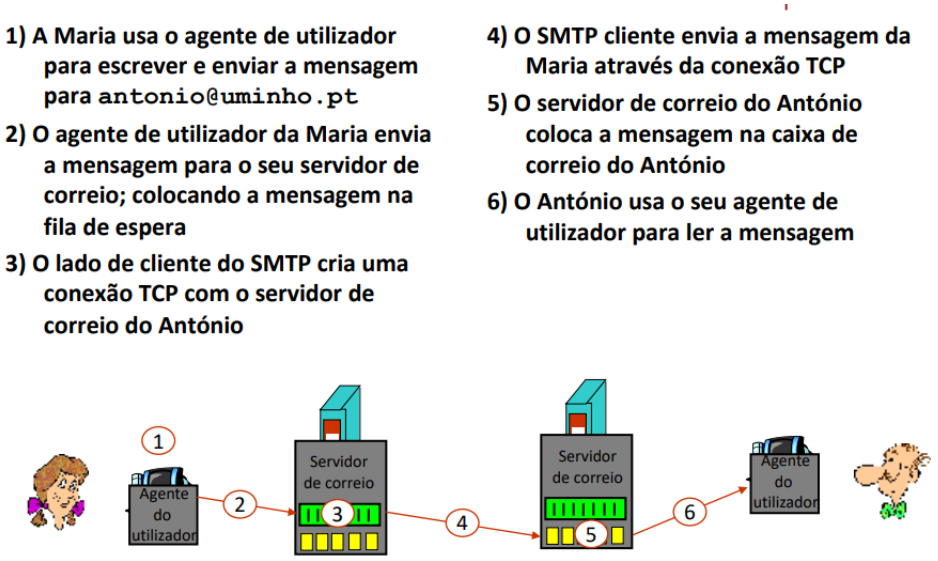
Descrever o funcionamento do sistema de correio electrónico da Internet, incluindo a sua arquitectura, o formato das mensagens e as funções dos protocolos SMTP, POP3 e IMAP.

O sistema de correio eletrónico, é constituído pelos 3 componentes:

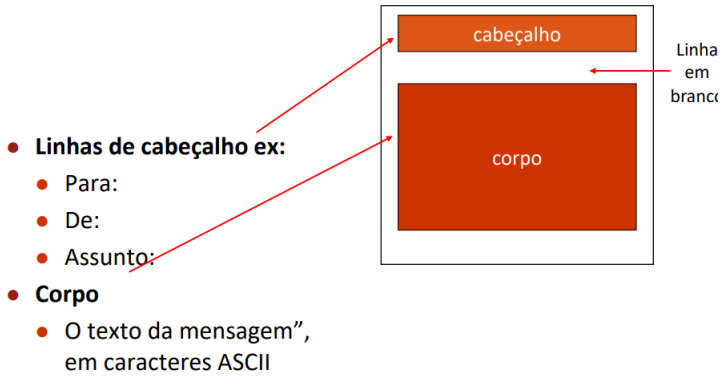
- Agentes do utilizador:
  - "Faz a leitura do correio";
  - Composição, edição e leitura da mensagens de correio;
  - Exemplos: Eudora, Outlook, Mozilla Thunderbird.
- Servidores de correio;
- Simple mail transfer protocol: utilizado pelo servidores de correio para trocarem mensagens de correio entre si, e para as receberem dos agentes SMTP.

O protocolo SMTP, usa o TCP para fazer a transmissão fiável das mensagens desde o cliente até ao servidor (porta 25), havendo uma transferência directa de dados do servidor emissor para o servidor receptor, que passa pelas seguintes fases:

1. handshaking;
2. transferência das mensagens;
3. término.



O formato das mensagens é o seguinte.



Em relação aos protocolos de acesso ao correio eletrónico:

- SMTP: Entrega/armazenamento no servidor receptor;
- Protocolo de acesso ao correio - entrega das mensagens do servidor receptor ao agente do utilizador:
  - POP: Post Office Protocol: Serve para autorização (agente <–> servidor) e download ;
  - IMAP:
    - Internet Mail Access Protocol;
    - Mais funcionalidades (mais complexo);
  - HTTP: gmail, Hotmail, Yahoo!, Sapo, etc.

Explicar o funcionamento do protocolo HTTP, nomeadamente nas suas versões persistente e não-persistente, com e sem pipelining.

O protocolo HTTP funciona da seguinte forma:

1. O cliente inicia uma conexão TCP (cria um socket) ao servidor, porta 80;
2. O servidor aceita a conexão TCP do cliente
3. Mensagens HTTP (mensagens do protocolo do nível da aplicação) são trocadas entre o browser (cliente HTTP) e o servidor Web (servidor HTTP);
4. O TCP fecha a conexão.

HTTP não guarda variáveis de estado (O servidor não mantém informação sobre os pedidos dos clientes anteriores). As conexões HTTP, podem ser:

- Não persistentes (Um objecto é enviado através de uma conexão TCP):
  - O Sistema Operativo tem que reservar recursos para cada ligação TCP estabelecida;
  - Muitos browsers abrem ligações TCP paralelas para irem buscar os objectos referenciados
- Persistentes (Múltiplos objectos podem ser enviados numa única conexão TCP entre o cliente e o servidor):
  - O servidor deixa a ligação aberta depois de enviar a mensagem de resposta;
  - Os pedidos HTTP posteriores são enviados através da mesma ligação.
  - Pode ser persistente:
    * Sem pipelining: O cliente envia um novo pedido apenas quando recebe a resposta ao anterior;
    * Com pipelining: O cliente envia os pedido assim que os encontra no objecto referenciado.