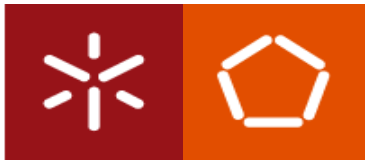


Universidade do Minho
Escola de Engenharia

Comunicações por Computador

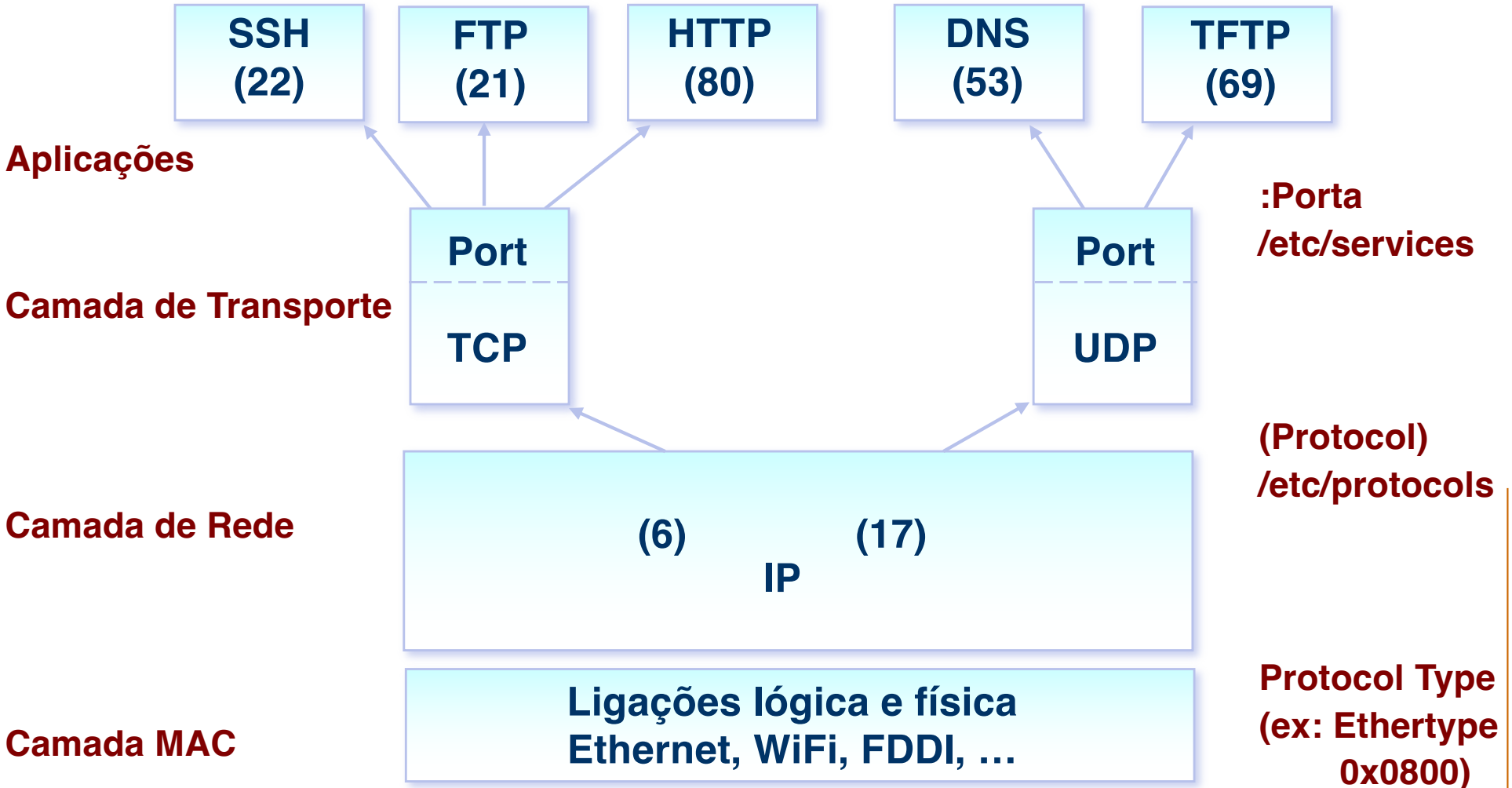
Cap 5 - Protocolos TCP/IP

Universidade do Minho
Grupo de Comunicações por Computador
Departamento de Informática

TCP/IP

Protocolos de Transporte: UDP e TCP

Universidade do Minho
Escola de Engenharia
Departamento de Informática



TCP/IP
UDP - *User Datagram Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Funções do User Datagram Protocol (UDP)

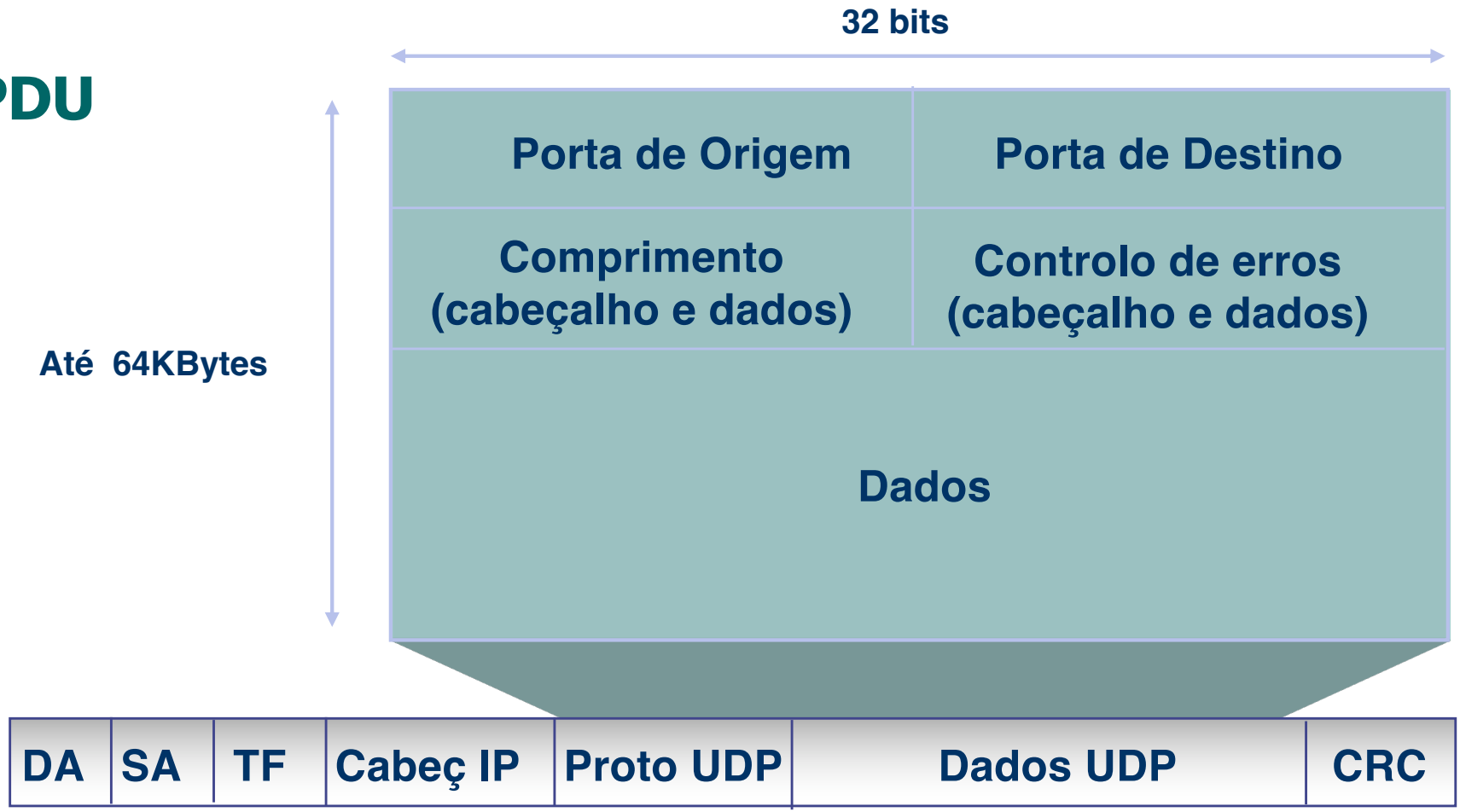
- protocolo de transporte fim-a-fim, não fiável
- orientado ao datagrama (sem conexão)
- actua como uma interface da aplicação com o IP para multiplexar e desmultiplexar tráfego
- usa o conceito de porta / número de porta
  - forma de direccionar datagramas IP para o nível superior
  - portas reservadas: 0 a 1023; portas registradas: 1024 a 49151; portas dinâmicas: 49152 a 65535
- é utilizado em situações que não justificam o TCP
  - exemplos: TFTP, RPC, DNS

TCP/IP

UDP - *User Datagram Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

PDU



TCP/IP
UDP - *User Datagram Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Controlo de erros (checksum) no UDP

- complemento para 1 da soma de grupos de 16 bits
- cobre o datagrama completo (cabeçalho e dados)
- o cálculo é facultativo mas a verificação é obrigatória
- Soma = **0** significa que o cálculo não foi efectuado
- se Soma ≠ **0** e o receptor detecta erro na soma:
  - o datagrama é ignorado (descartado);
  - não é gerada mensagem de erro para o transmissor;
  - protocolo não inclui mecanismos de retransmissão ...

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Funções do Transmission Control Protocol

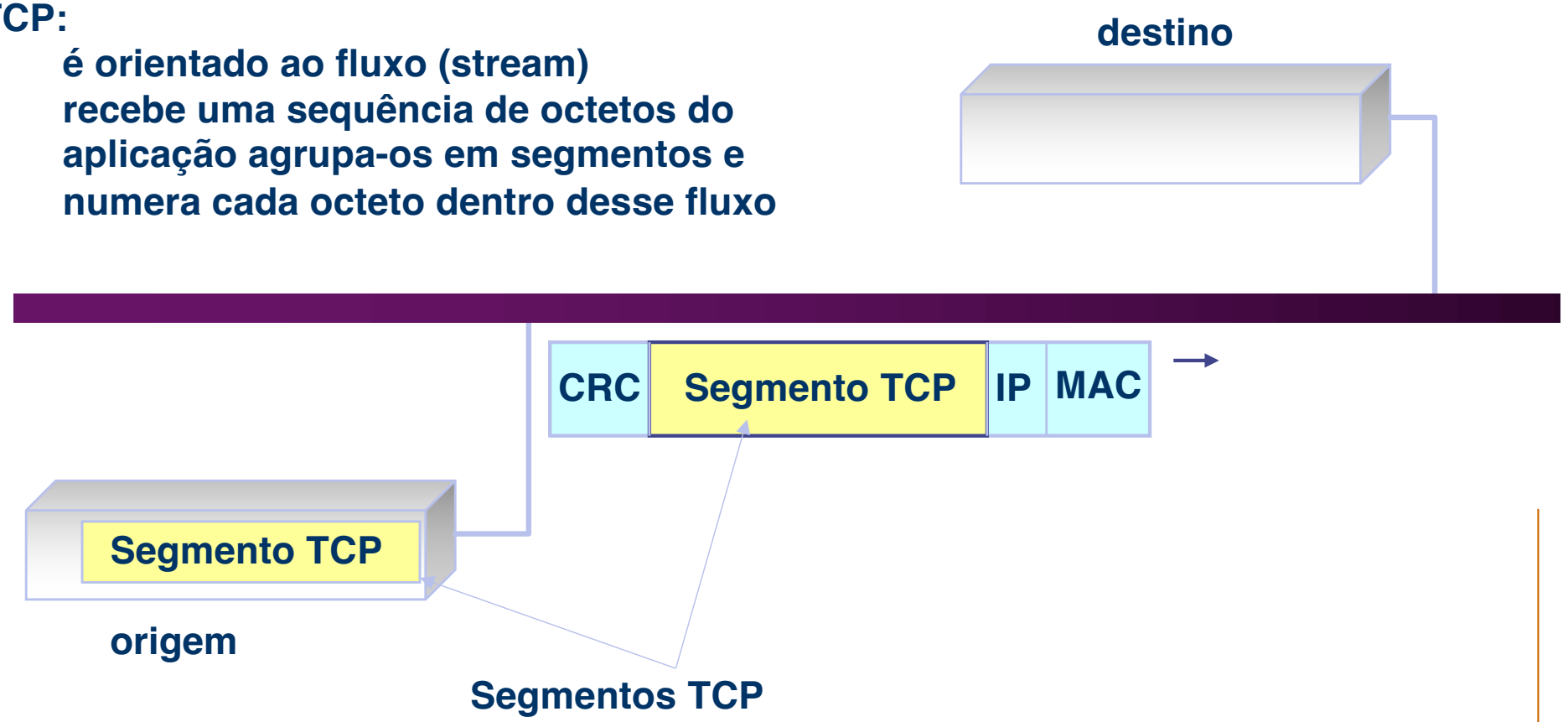
- transporte fiável de dados fim-a-fim (aplicações)
- efectua associações lógicas fim-a-fim: **conexões**
  - cada conexão é identificada por um par de sockets: *(IP_origem:porta_origem,IP_destino:porta_destino)*
  - uma conexão é um circuito virtual entre portas de aplicações (também designadas portas de serviço)
- multiplexa os dados de várias aplicações através de número de porta
- efectua controlo de fluxo, de congestão e de erros

TCP/IP
TCP - *Transmission Control Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

TCP:

- **é orientado ao fluxo (stream)**
- **recebe uma sequência de octetos do aplicação agrupa-os em segmentos e numera cada octeto dentro desse fluxo**

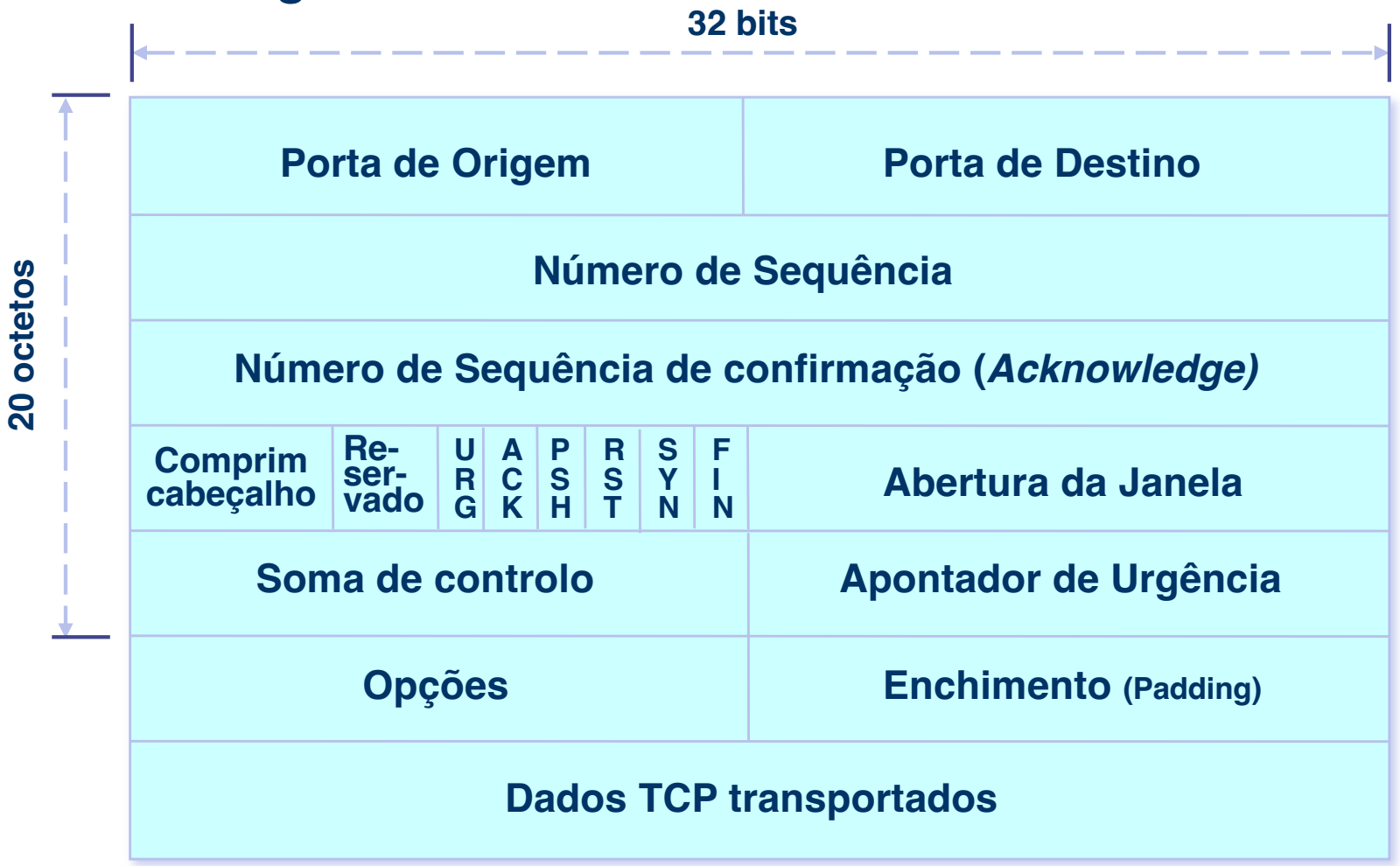


TCP/IP

TCP - *Transmission Control Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Estrutura do Segmento TCP



TCP/IP
TCP - *Transmission Control Protocol*

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- **Porta Orig/Dest** – Nº da porta TCP da aplicação de Origem/Destino
- **Número de Sequência** - ordem do primeiro octeto de dados no segmento (se SYN = 1, este número é o *initial sequence number*, ISN)
- **Número de Ack** (32 bits) - o número de ordem do octeto seguinte na sequência que a entidade TCP espera receber.
- **Comprimento Cabeçalho** (4 bits) - número de 32 bits no cabeçalho.
- **Flags** (6 bits) - indicações específicas.
- **Janela** – nº de octetos em trânsito sem confirmação (controlo fluxo)
- **Soma de controlo** (16 bits) – soma para detecção de erros (segm)
- **Apontador de Urgência** (16 bits) – adicionado ao nº de sequência dá o nº de sequência do último octeto de dados urgentes.
- **Opções** (variável) - especifica características opcionais (**ex. MSS**)

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Flags TCP (1 bit por flag)

ACK - indica se o nº de sequência de confirmação é válido

PSH - o receptor deve passar imediatamente os dados à aplicação

RST - indica que a conexão TCP vai ser reinicializada

SYN - indica que os números de sequência devem ser sincronizados para se iniciar uma conexão

FIN - indica que o transmissor terminou o envio de dados

URG - indica se o apontador de urgência é válido

Os segmentos SYN e FIN consomem um número de sequência

Universidade do Minho
Escola de Engenharia
Departamento de Informática

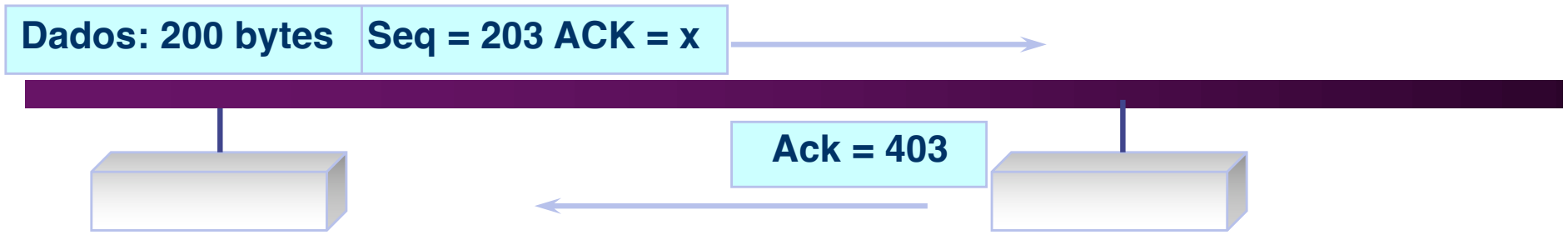
Segmentos TCP

- sequenciação necessária para ordenação na chegada
- o *número de sequência* é incrementado pelo número de bytes do campo de dados
- cada segmento TCP tem de ser confirmado (ACK), contudo é válido o ACK de múltiplos segmentos
- o campo ACK indica o próximo byte (*sequence*) que o receptor espera receber (*piggyback*)
- o emissor pode retransmitir por *timeout*:

  o protocolo define o tempo máximo de vida dos segmentos ou MSL (maximum segment lifetime)

TCP/IP
TCP - *Transmission Control Protocol*

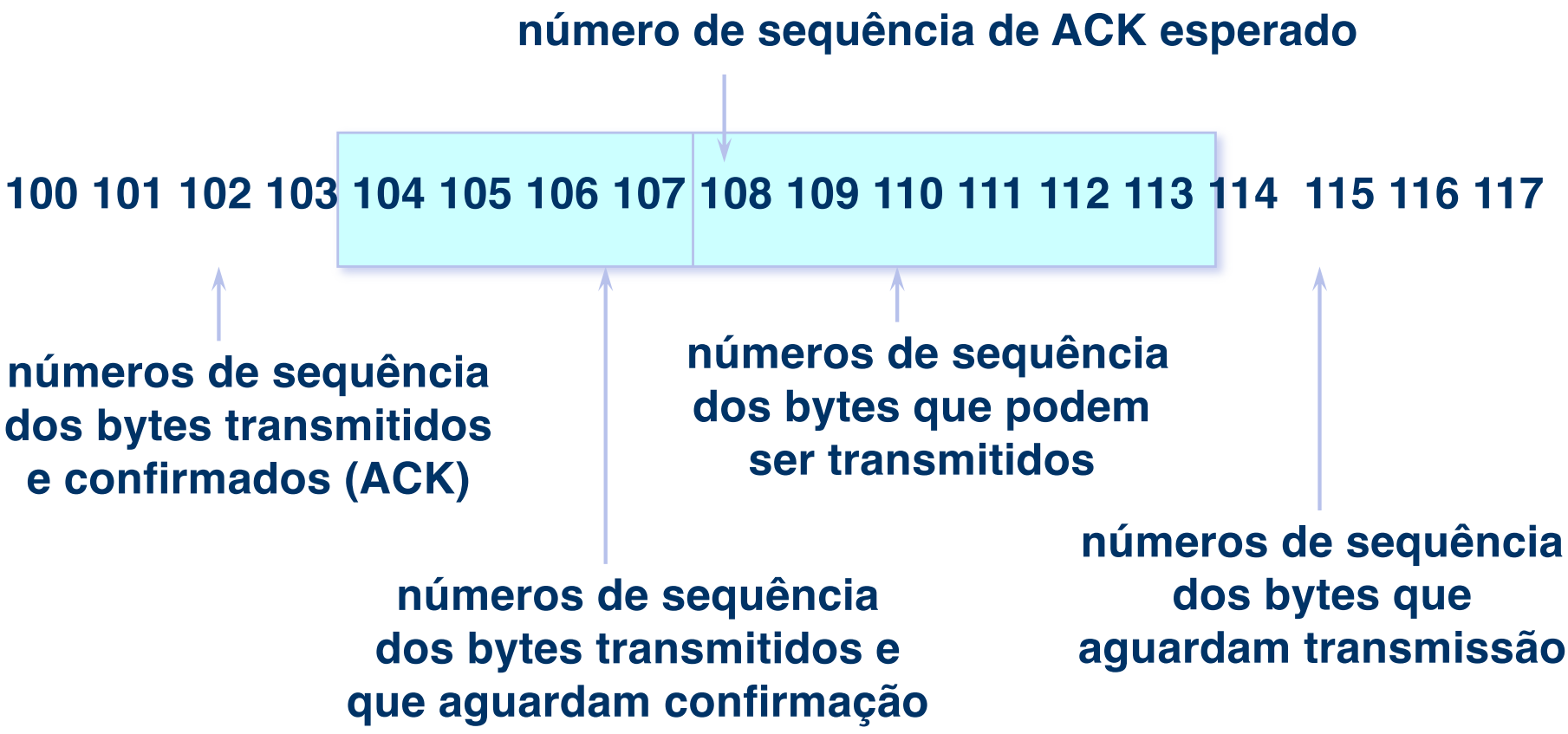
Universidade do Minho
Escola de Engenharia
Departamento de Informática



- Cada *sistema-final (end-system)* mantém o seu próprio *Número de Sequência*: 0 .. 2^{32} -1
- Numeração orientada ao byte
- Nº de ACK = Número de Sequência + bytes lidos no segmento = próximo byte a receber

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Controlo de fluxo baseado na abertura da janela anunciada no segmento recebido do parceiro

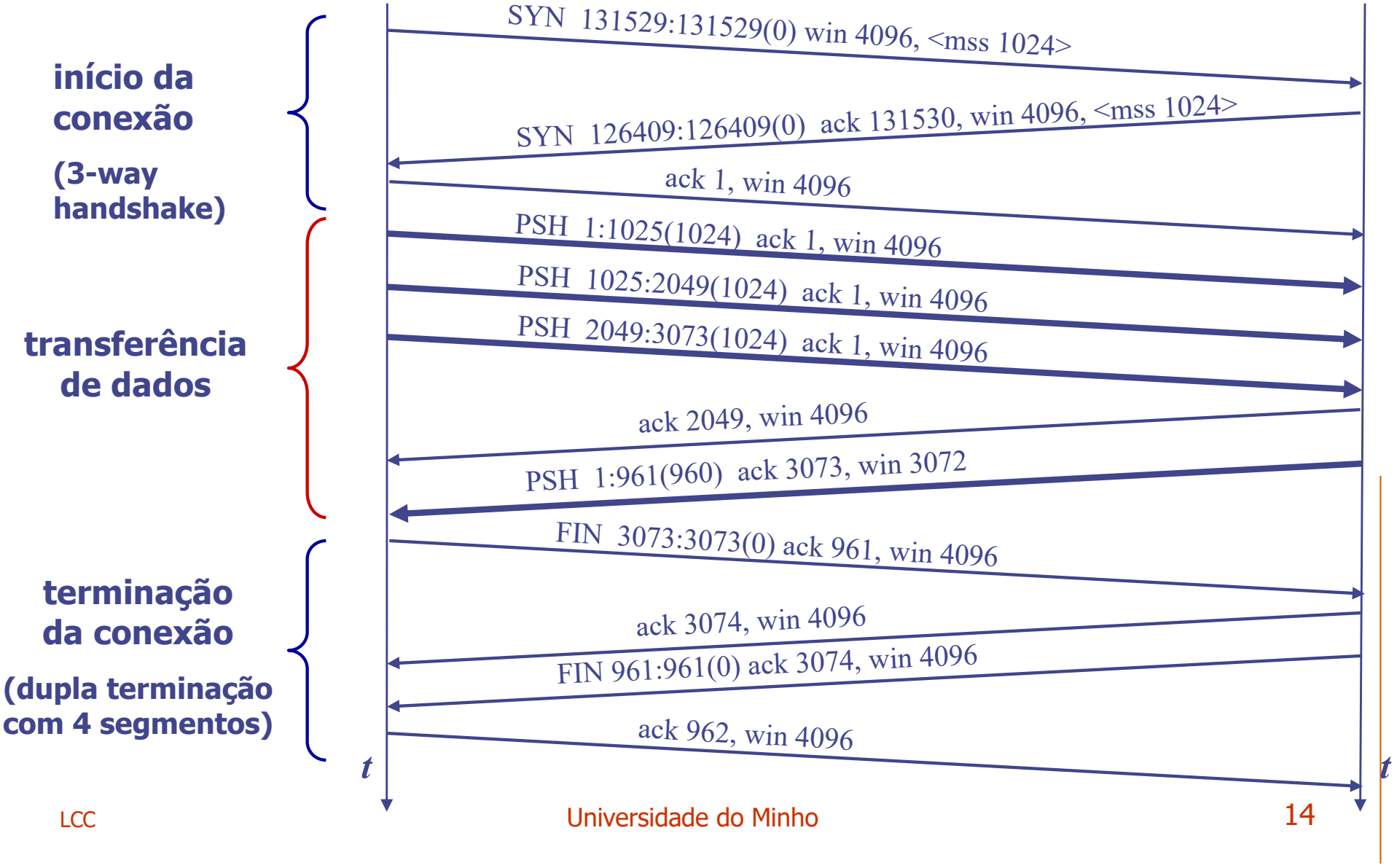


TCP/IP

TCP - *Transmission Control Protocol*

operação

Universidade do Minho
Escola de Engenharia
Departamento de Informática



Universidade do Minho
Escola de Engenharia
Departamento de Informática

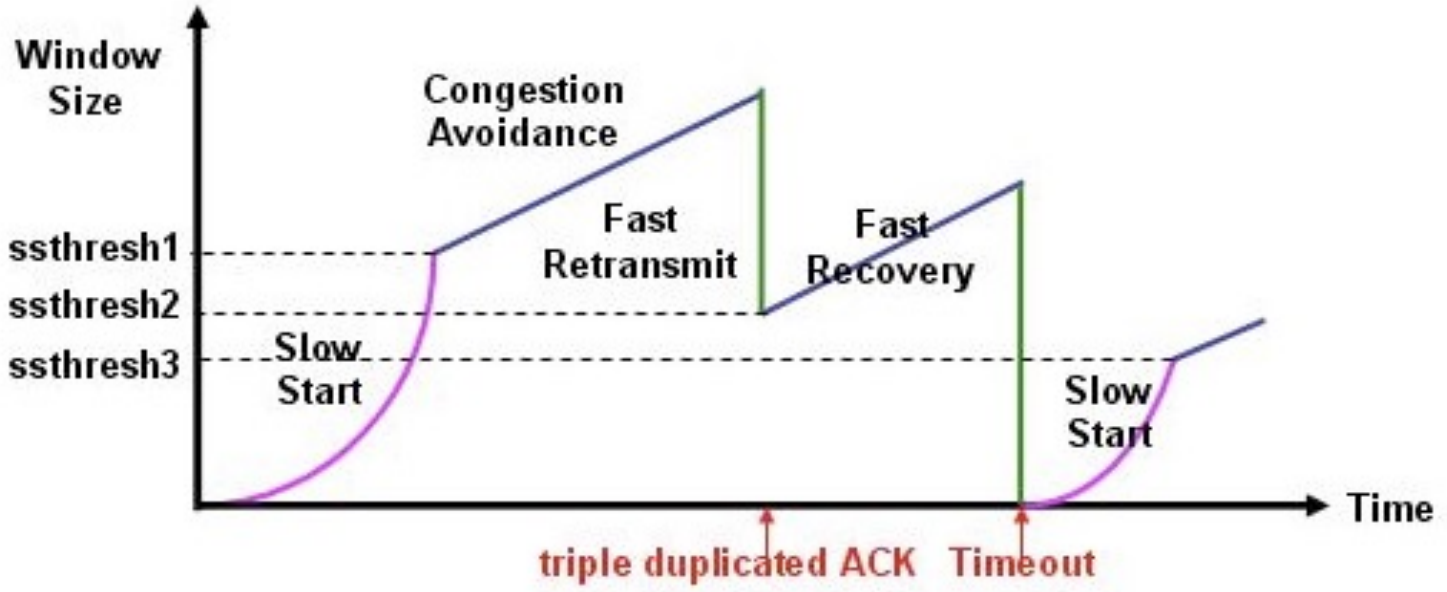
Uma entidade TCP:

- mantém um *timer* por segmento
- regista o instante de tempo do início da transmissão e da recepção das confirmações por forma a estimar um valor médio do **Round-Trip Time (RTT)**
- pode tomar a iniciativa de retransmitir
- começa por enviar até ao máximo permitido pela janela até que seja detectada congestão (*timeout* ou ACK duplicados)
- se os ACK tardam, assume uma janela de congestão implícita, mais fechada, que passa a ser incrementada linearmente (**slow start algorithm**)

TCP - *Transmission Control Protocol*

Controlo de congestão

Universidade do Minho
Escola de Engenharia
Departamento de Informática



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Maximum Segment Size (MSS) do TCP

- opção TCP que apenas aparece em segmentos SYN
- o MSS é o maior bloco de dados da aplicação que o TCP enviará na conexão
- ao iniciar-se uma conexão, cada lado tem a opção de anunciar ao outro o MSS que espera receber
- o maior MSS possível é igual ao MTU do interface menos os comprimentos dos cabeçalhos TCP e IP:

  Exemplos:
  - sobre Ethernet o maior MSS é 1460 bytes
  - sobre IEEE 802.3 o maior MSS é 1452 bytes

TCP/IP
Outros protocolos de transporte

Universidade do Minho
Escola de Engenharia
Departamento de Informática

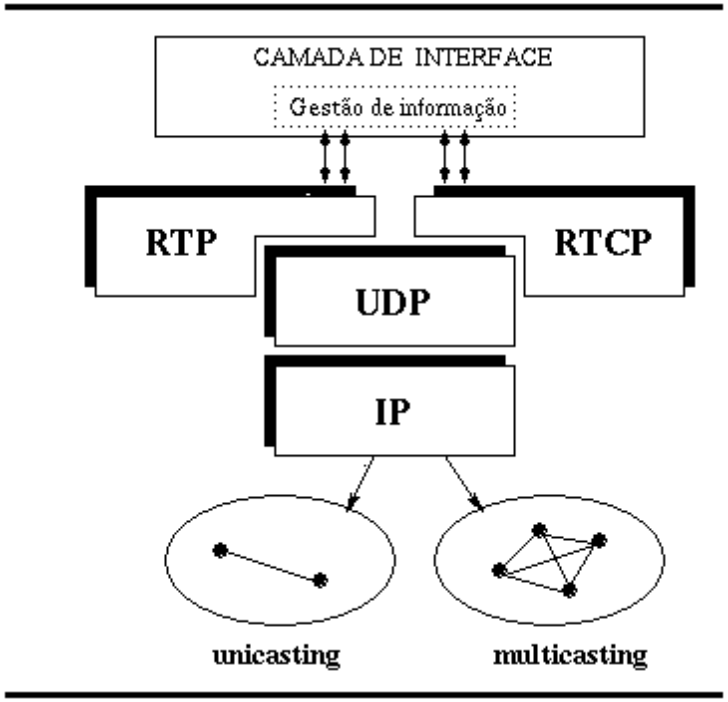
Protocolo RTP (Real Time Protocol) [RFC1889]

- Objectivo: suporte para aplicações adaptativas (real-time)
- Fornece mecanismos de transporte fim-a-fim a dados gerados por aplicações de tempo real (audio, video)
- Pode ser aplicado em cenários *unicast* ou *multicast*
- Estabelece um canal de dados + canal de controlo (RTCP)
- RTCP (Real Time Control Protocol)
  - fornece mecanismos de notificação do estado de operação do canal de dados (ex.perdas, atrasos, sincronização)

TCP/IP
Outros protocolos de transporte

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- Níveis protocolares com o RTP



- Fornecer um mecanismo de transmissão a dados com requisitos T.R.
- Fornecer mecanismos para controlo dos dados por parte da aplicação

Aplicações de rede
Modelo Cliente/Servidor

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- A comunicação é iniciada ao nível das aplicações.
- Aplicações de rede:
  - smtp – simple message transfer protocol
  - domain (dns) – domain name server
  - snmp – simple network management protocol
  - ftp – file transfer protocol, etc
- As aplicações usam os serviços de transporte locais

Muitas das aplicações de rede são baseadas no paradigma de interacção ***cliente/servidor***

Aplicações de rede
Modelo Cliente/Servidor

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Cliente

- aplicação invocada pelo utilizador e mantida como cliente temporariamente
- corre localmente
- interage com o servidor de forma activa
- pode contactar múltiplos servidores durante a mesma sessão
- não necessita de recursos sofisticados

Servidor

- dedicado à prestação de um serviço
- pode atender em simultâneo múltiplos clientes
- é invocado de forma automática quando o sistema arranca
- em geral, corre em sistemas partilhados
- permanentemente "à escuta"
- requer mais recursos de *hardware* e *software*;

Aplicações de rede
Modelo Cliente/Servidor

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- Identificação de cada serviço é efectuada a nível dos protocolos de transporte
  - *número de porta* único associado a cada serviço:

    ftp (21), ssh (22), smtp (25), http (80), domain (53)...
- Em geral, um servidor tem a possibilidade de atender, em simultâneo, vários clientes
  - *Cada cliente possui também um identificador (port number). O software no servidor usa ambos os números de porta e os endereços IP do cliente e do servidor para identificar cada conexão.*
  - **socket** ::= <endereço_IP>:<número_de_porta>

Aplicações de rede
Modelo Cliente/Servidor

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- Uma aplicação cliente pode:
  - escolher o protocolo de transporte a usar de acordo com as suas necessidades

    exemplo: fiável (TCP), não fiável (UDP) ou outros protocolos com características específicas de acordo com os media manipulados pelas aplicações

  - contactar vários servidores, um para cada serviço (eventualmente em sistemas diferentes)

    exemplo: cliente WWW, cliente e-mail

Aplicações de rede
Modelo Cliente/Servidor

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- Um servidor pode
  - ser cliente de outro serviço

    exemplo: serviço de resolução de um nome em que o servidor questiona outros servidores

- Num ambiente com múltiplos servidores deve ser evitada a criação de dependências circulares entre eles
  - Servidor A depende do servidor B que depende do servidor C que por sua vez depende do servidor A

Aplicações de rede
Utilização de sockets

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- A interface entre aplicações cliente/servidor e a pilha protocolar de comunicações é designada

  Sockets Application Program Interface

- A API é definida ao nível do sistema operativo
  - envolve a definição do nome das várias funções e respectivos argumentos
  - usa o conceito de descriptores
  - tornou-se norma de facto (surgiu c/ Unix BSD)
  - disponível Linux, Solaris, Windows, MacOS, ...

Aplicações de rede
Utilização de sockets

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Funções da *API sockets*

- sd = socket (protofamily, type, protocol) **criar socket**
- close (sd) **fecha a conexão (se existir); termina uso**
- bind (sd, localaddr, addrlen) **associação de endereços/portas**
- listen (sd, queuesize) **colocar o socket em modo passivo**
- nsd = accept (sd, clientaddr, addrlen) **aceitar pedido de conexão**
- connect (sd, serveraddr, addrlen) **efectuar pedido de conexão,**
- send (sd, data, length, flags) **enviar dados sobre uma conexão**
- sendto (sd, data, length, flags, dstaddr, addrlen) **enviar dados** (CL)
- sendmsg (sd, msgstruct, flags) **enviar dados** (CL)
- recv (...); recvfrom (...); recvmsg (...) **receber dados**
- **....**

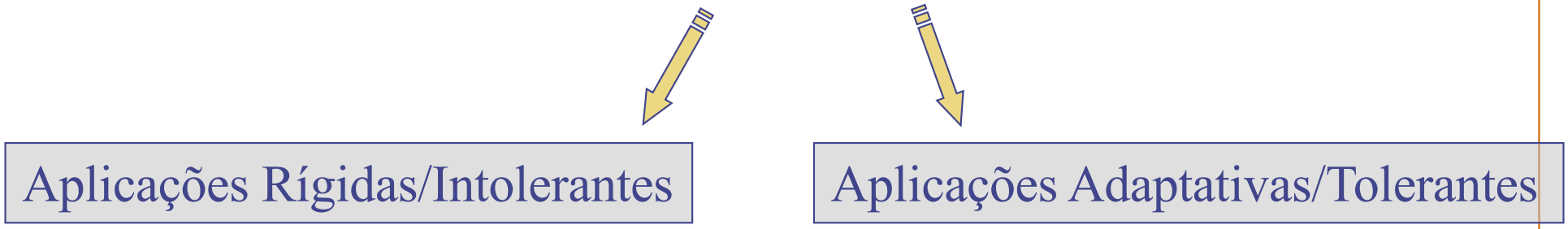
Classificação Geral das Aplicações de Rede

- Aplicações Elásticas
  - ftp, e-mail, http, ...
- Aplicações que não são extremamente sensíveis a factores como:
    - Atrasos, variação dos atrasos, variações de largura de banda, perdas, etc...
    - apesar desses mesmos factores afectarem o seu desempenho

- Preocupação sob o ponto de vista de integridade dos dados
- Protocolos de transporte tradicionais tais como: TCP, UDP, ...

Classificação Geral das Aplicações de Rede

- Aplicações de Tempo Real (audio e vídeo em tempo real, vídeo-conferência...)
- Requisitos: garantia de isocronismo dos dados gerados, sensibilidade a atrasos, sensibilidade a perda de pacotes, necessidades de sincronização, largura de banda etc...
- Duas classes de aplicações de T.R:



Aplicações de Tempo Real

Universidade do Minho
Escola de Engenharia
Departamento de Informática

- Aplicações Rígidas
  - Requerem suportes que garantam limites para os diversos parâmetros que as condicionam (largura de banda, perdas, atrasos de pacotes...)

- Aplicações Tolerantes/Adaptativas
  - Assumem um determinado grau de inadequação do meio
  - não se baseiam em limites fixos para os parâmetros de funcionamento
  - admitem um grau de tolerância às condições de operação
  - observação vs adaptação