

FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
LICENCIATURA EM ENGENHARIA INFORMÁTICA
REDES DE COMPUTADORES I

TEMA: Protocolos TCP e UDP

Grupo Docente:

- Eng^o. Felizardo Munguambe (MSc)
- Eng^o. Délcio Chadreca (MSc)

Tópicos da Aula

- ▶ Introdução;
- ▶ Conceitos fundamentais;
- ▶ Camada de Transporte
- ▶ TCP
- ▶ UDP

Introdução

Nas aulas anteriores teve oportunidade de estudar sobre o modelo TCP/IP, onde teve a oportunidade de conhecer as diferentes camadas da arquitetura TCP/IP. Na presente aula vamos abordar sobre os dois principais protocolos da camada de transporte, o TCP e o UDP.

É sabido que os serviços oferecidos pelo protocolo IP não oferecem confiabilidade. E que problemas comuns como congestionamento, perda ou ordenação de pacotes não são tratados, entretanto, as aplicações HTTP, FTP, SMTP, etc., necessitam prover um serviço de qualidade aos seus usuários, é justamente nesse ponto onde vamos enquadrar a camada de transporte, que pode oferecer um serviço confiável de entrega de dados das aplicações. E para implementar as competências atribuídas a camada de transporte, foram desenvolvidos os protocolos.

Conceitos fundamentais

Para perceber melhor sobre os protocolos da camada de transporte, é necessário fazer a distinção de alguns conceitos fundamentais que vão ajudar na percepção.

- **Orientação a conexão;**
- **Confiabilidade;**
- **Controle de fluxo;**
- *Handshake;*

Conceitos fundamentais

Orientação a conexão.

Um serviço diz – se orientado a conexão quando é estabelecida uma conexão, um vínculo, ou uma ligação antes que qualquer outra actividade seja desenvolvida. A aplicação envia um pedido de conexão para o destino e faz uso da “conexão” para poder transferir dados. Portanto, se faz necessário o estabelecimento de uma conexão, por meio de uma sequência de passos previamente definida pelo protocolo para que os dois pontos da conexão passam interagir entre si.

Um serviço ou protocol diz-se não orientado a conexão quando não existe apresentação prévia entre os terminais. Os pacotes são enviados simplesmente.

cont.

Handshake - o *handshake* é o mecanismo de estabelecimento e finalização de conexão, ou seja, o *handshake* é o processo pelo qual duas ou mais máquinas afirmam reconhecer umas às outras e que estão aptas para iniciar a comunicação.

A comunicação entre duas aplicações em diferentes máquinas é chamada de conexão. A abertura da conexão é feita através de um processo de *handshake* (aperto de mão).

Confiabilidade – A confiabilidade consiste em garantir que os dados são todos entregues na ordem correcta, em ser erros e com QoS.

cont.

Controle de fluxo – o controle de fluxo regula a taxa que são transmitidos os dados. O serviço de controle de fluxo é usado nos casos em que o descarte de dados é provocado pela congestão de tráfego. Por exemplo, se um transmissor estiver a enviar dados a uma taxa superior àquela a que o receptor os pode processar ou armazenar, os dados serão necessariamente descartados. A ideia do controle de fluxo é evitar o descarte de dados devido a congestão de tráfego.

Camada de transporte

A camada de Transporte se localiza entre as camadas de Aplicação e de Internet (ou Rede) na pilha TCP/IP, e corresponde à camada de Transporte no Modelo OSI. Ela é responsável por fornecer serviços à camada de aplicação.

A comunicação entre processos de aplicações de diferentes aplicações distintas possam compartilhar dados usando a mesma implementação de protocolos das camadas mais baixas.

Camada de transporte

Os protocolos da camada de transporte devem saber distinguir as diferentes aplicações, e combinar esses dados a um fluxo de dados que será enviado as camadas mais baixas da pilha de protocolos, e conseguir efectuar o processo inverso no *host* de destino. Além do que já foi referido, a camada de transporte deve dividir, ou seja, segmentar grandes quantidades de dados que devem ser transmitidos em fragmentos (ou segmentos) menores, de modo que a sua transmissão seja possível.

Funções da Camada de Transporte

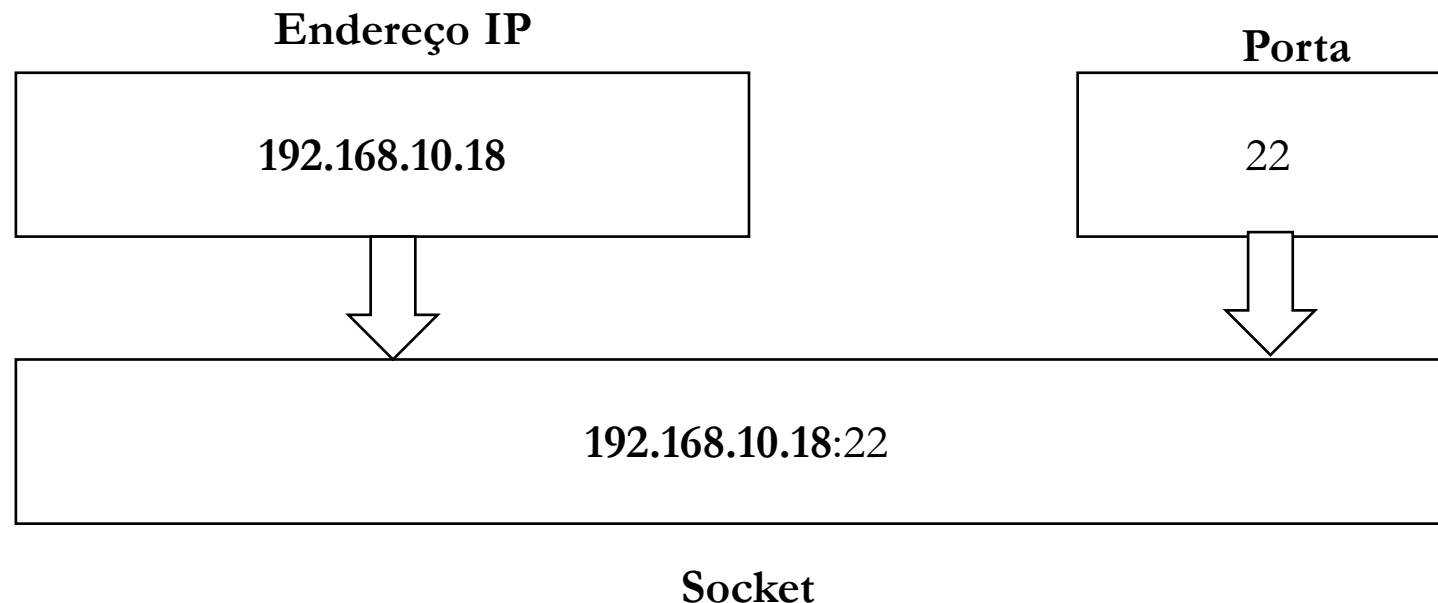
Constituem principais funções da camada de transporte, as seguintes:

- Comunicação entre processos;
- Controle de fluxo;
- Controle de erros;
- Multiplexação e Desmultiplexação;
- Controle de Congestionamento de rede;
- Estabelecer e gerir conexões;
- etc.

Comunicação entre processos

A comunicação entre aplicações em diferentes máquinas, deverá utilizar serviços da camada de transporte. A entrega da informação a aplicação adequada é possível graças ao uso de identificadores denominados **Números de Porta (*Port Numbers*)**. E para que seja possível identificar o *host* que irão se comunicar, usamos o endereço IP.

Fig. 1



Transmission Control Protocol (TCP)

TCP (*Transmission Control Protocol*)

O TCP é o protocolo da camada de transporte da arquitetura Internet responsável em oferecer confiabilidade na transmissão. Alguns autores consideram-no como a chave do sucesso da Internet. O TCP fornece um serviço orientado ca conexão, confiável e *full-duplex* para os serviços de aplicação.

Antes de iniciar a transferência de dados, o TCP começa a estabelecer uma sessão, com um protocolo de navegação de três vias, que é terminada no final da transferência. Adicionalmente, o protocolo realiza o controle de fluxo com janela, a partilha de ligações e a recuperação de erros.

Como já foi referido anteriormente, o TCP é orientado a conexão, para poder exercer controle sobre os pacotes que são enviados e conseguir efectuar a fragmentação, o TCP precisa garantir que os usuários finais tenham controle sobre o que está sendo enviado. O protocolo TCP especifica três fases durante uma conexão:

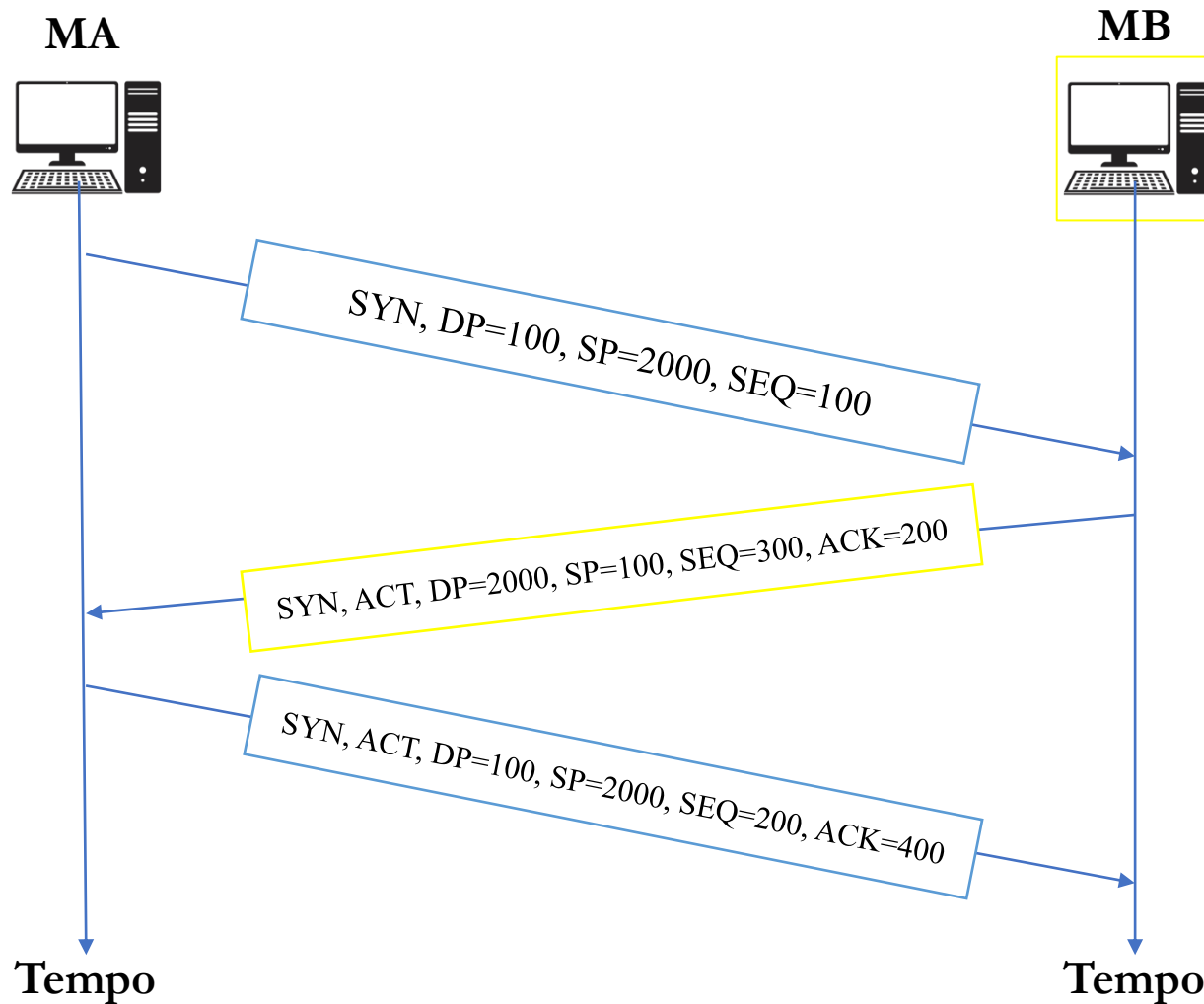
1. Estabelecer a ligação;
2. Transferência; e
3. Terminar da ligação;

1. Estabelecer a ligação

Para poder estabelecer a ligação da conexão TCP é necessário que o cliente inicie a ligação enviando um pacote TCP com flag SYN activa e espera-se que o servidor aceite a ligação enviando um pacote SYN+ACK. Se, durante um determinado espaço de tempo, esse pacote não for recebido ocorre um *timeout* e o pacote SYN é reenviado. O estabelecimento da ligação é terminada por parte do cliente, confirmando a aceitação do servidor respondendo-lhe com um pacote “ACK”.

Exemplos de um Handshake

Fig. 2



SYN – Synchronize
ACK – Acknowledge

Explicação da fig.2

No exemplo da figura, a máquina **MA** começa por enviar um SYN juntamente com o número de sequência (SEQ=100) para que o receptor se sincronize com este número. De seguida, a máquina **MB** responde com outro sinal SYN juntamente com o seu número de sequência de MA com ACK=200 (número de confirmação, significa que recebeu o número 100 e espera o 200). Finalmente, MA confirma a sequência de MB, com um sinal SYN, e um ACK=400 (Indicando que recebeu o segmento 300 e espera o 400). O *bit* de controlo, ACK (**ver figura 4**), é usado para indicado para indicar que o número de confirmação do segmento é válido. Ainda em relação ao exemplo, a máquina MA usa o porto (ou porta) 2000 e a máquina MB usa o porto 100, de acordo com os campos porto destino (DP) e porto origem (SP).

2. Transferência de dados e partilha de ligação.

O protocolo TCP garante a transferência de dados entre aplicações. A aplicação do emissor gera dados e envia-os para o TCP juntamente com o endereço lógico e a identificação da aplicação a quem se destinam os dados. As aplicações são identificadas com um número designado **porta** (ou **porto**). O conjunto formado pelo número do porto, pelo endereço de destino e pelo nome do protocolo de transporte que deve ser usado na transmissão designa-se *socket*.(veja fig.1). Numa comunicação de dados, existe um *socket* origem e um *socket* destino.

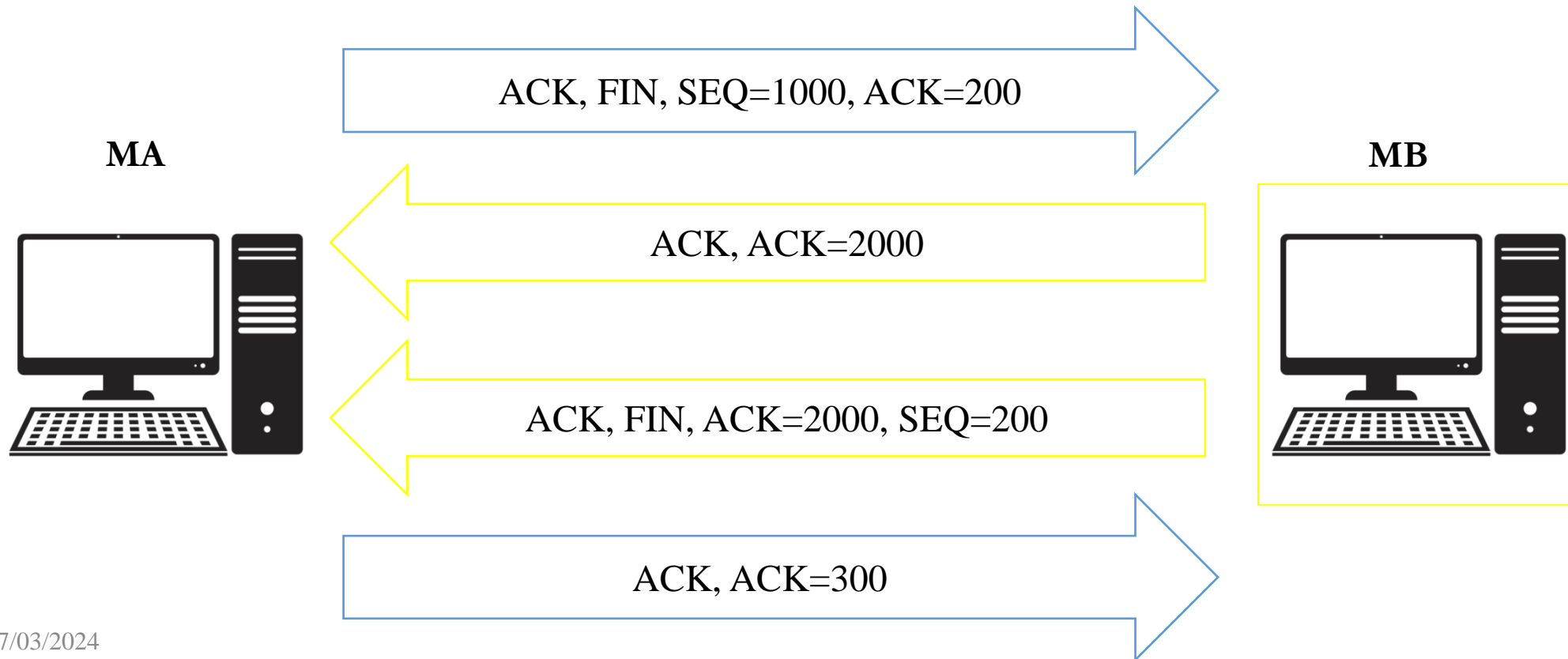
A definição correcta dos *sockets* é fundamental para a partilha de uma ligação por várias aplicações.

cont.

Exemplo: Se um determinado cliente aceder a um servidor que responde pedidos FTP e outra HTTP, irá usar o mesmo endereço lógico do servidor, mas o número dos portos será diferente. Só assim é possível saber a que aplicação se destina um determinado pedido do cliente. Se por outro lado, existirem dois clientes em máquinas diferentes a aceder ao mesmo servidor FTP, o servidor consegue distinguir os pedidos através do endereço lógico dos clientes e através do número de portas das aplicações dos clientes que realizaram o pedido FTP. Se os pedidos FTP viessem da mesma máquina, a única forma de distinguir seria através dos números das portas de origem, uma vez que os endereços lógicos de origem e de destino, bem como a porta de destino, seriam os mesmos.

3. Terminar Ligação

Fig.3



Explicação da Fig.3

Para terminar a ligação, o TCP usa um protocolo de quatro caminhos baseado no *bit* de controlo FIN (*Finish*). No exemplo da figura 3, a máquina MA pretende terminar a ligação. Para tal, envia um segmento com *bit* FIN activo, com o número de sequência do segmento e a confirmação do último segmento recebido. A máquina MB notifica a aplicação de que o emissor pretende terminar a ligação TCP e fica à espera que a aplicação confirme que pode terminar a ligação, o TCP envia o sinal FIN juntamente com a confirmação do segmento recebido (ACK) e com o último segmento SEQ=200. Finalmente, MA envia a confirmação do segmento recebido (ACK=300) e termina a ligação TCP.

Fig. 4 – Formato do segmento TCP

Porta de Origem (16)					Porta de Destino (16)				
Número de sequência (32)									
Número de Confirmação (32)									
Tamanho do cabeçalho (4)	Reservado (6)	Bit de Controlo						Janela (16)	
		URG	ACK	PSH	RST	SYN	FIN		
Verificação de erros (16)					Indicador de prioridade (16)				
Opções (0 ou 32)									
Dados (Tamanho variável)									

Descrição da fig.4

Nota: Os valores entre parenteses indicam o número de *bits* do campo respectivo.

O segmento TCP é constituído pelos seguintes campos:

- **Porta de origem** – Número da porta da aplicação que envia os dados;
- **Porta de destino** – Número da porta da aplicação a que se destinam os dados;
- **Número de sequência** – Número de sequência do primeiro byte dos dados do segmento;
- **Número de confirmação** – Número de sequência do próximo byte que o receptor espera receber;

- **Tamanho do cabeçalho** – Tamanho do cabeçalho em palavras de 32 *bits*. É usado para determinar onde começam os dados;
- **Reservado** – Campo reservado para uso futuro. É colocado a 0;
- **Bits de controlo** – *Bits* usados no controlo e gestão da sessão:
 - **URG** – Quando activo, indica que os dados são prioritários. Os dados prioritários são passados em primeiro lugar;
 - **ACK** – Quando activo, indica que o número do campo de confirmação é válido;
 - **PSH** – Quando activo, indica ao receptor que os dados não prioritários devem ser entregues imediatamente a aplicação para processamento;
 - **RST** – Quando activo, reinicializa a sessão;
 - **SYN** – Usado para estabelecer uma sessão;
 - **FIN** – Quando activo, indica que são os últimos dados recebidos;

Recuperação de erros

O TCP garante a fiabilidade da transmissão de dados através de um mecanismo de recuperação de erros. Nas situações em que determinados segmentos não são recebidos ou são recebidos com erros, detectado através do código de verificação de erros, o TCP inicia um processo de recuperação dos segmentos perdidos. Para tal, todos os *bytes* dos segmentos são numerados com um número de sequência e um número de confirmação.

No caso de existir erro num dos segmentos, o TCP não confirma a recepção do segmento cliente e reenvia-o.

Controlo de fluxo com Janela

O TCP realiza o controle de fluxo com base numa janela de *bytes*. A janela indica o número máximo de *bytes* que podem ser enviados sem confirmação do receptor. O tamanho da janela ajusta-se de acordo com a percentagem de erros verificados na transmissão. Inicialmente, a janela tem um tamanho pequeno e vai crescendo até que comecem a ocorrer erros.

Quando o emissor envia tantos *bytes* quanto os suportados pela janela, diz-se que a janela está cheia e o emissor não pode enviar mais segmentos.

User Datagram Protocol (UDP)

UDP

O protocolo UDP, tal como o TCP, disponibiliza serviços de transferência de dados e de partilha de ligação por várias aplicações. No entanto, o UDP é um protocolo da camada de transporte não confiável e não orientado a conexão. Fornece apenas serviços de endereçamento e fragmentação, não provendo confiabilidade (controle de fluxo, erro, congestionamento).

O UDP por ser mais simples e possuir um cabeçalho menor, gera menor *overhead*. Ideal para algumas aplicações onde a velocidade é mais útil que a confiabilidade como aplicações multimédia. Afinal não faz sentido algum receber um trecho de um ficheiro de música que já passou, não?

cont.

Além de aplicações multimédia, o UDP é utilizado também pelo TFTP (*Trivial File Transfer Protocol*), RIP (*Routing Information Protocol*), SNMP (*Simple Network Management Protocol*) e DNS(*Domain Name System*).

Fig. 5 Formato do segmento USP

Porta de origem (16)	Porta de destino (16)
Tamanho do Segmento (16)	Verificação de erros (16)
Dados	

O cabeçalho UDP é constituído pelos campos seguintes:

- **Porta de origem** – Número da porta da aplicação que enviou os dados;
- **Porta de destino** – Número da porta da aplicação a que se destinam os dados;
- **Tamanho do segmento** – Tamanho de todo o segmento, incluindo o cabeçalho de dados, em palavras de 32 *bits*;

A transferência de dados e a partilha de informação baseia-se no uso de *sockets*, como no protocolo TCP. A única diferença é que os *sockets* UDP fazem referência ao protocolo de transporte UDP em vez de TCP. Um servidor pode responder na mesma porta ao protocolo TCP e ao UDP.

Bibliografia consultada

- Barrett, D., & King, T. (2010). *Redes de Computadores*. Rio de Janeiro: LTC Livros Técnicos e Científicos Editora.
- Boavida, F., Bernardes, M., & Vapi, P. (2011). *Administração de Redes de Informáticas*. Lisboa: FCA - Editora de Informática, LDA.
- Leon-Garcia, A., & Widjaja, I. (2001). *Communication Networkd - Fundamental Concepts and Key Architectures*. The McGraw-Hill Companies.
- Peterson, L. L., & Davie, B. S. (2011). *Computer Networks: a systems approach*. San Francisco: Morgan Kufmann.
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer Networks* . Boston,: Pearson Education, Inc.,.
- Torres, G. (2001). *Redes de Compotadores Curso Completo*. Axcel Books do Brasil Editora.
- Véstias, M. (2009). *Redes Cisco - Para profissionais*. Lisboa: FCA - Editora de Informática, LDA.

Questões de reflexão

1. Explique como funciona o mecanismo de Janela deslizante, e diga onde este é usado.
2. O UDP é, geralmente, usado em aplicações tolerantes a perdas de dados. Especifique pelo menos 3 aplicações diferentes das apresentadas na aula.
3. Estabeleça a distinção do mecanismo usado para detecção e correcção de erros na camada de transporte e na camada de enlace de dados.

OBRIGADO !!!