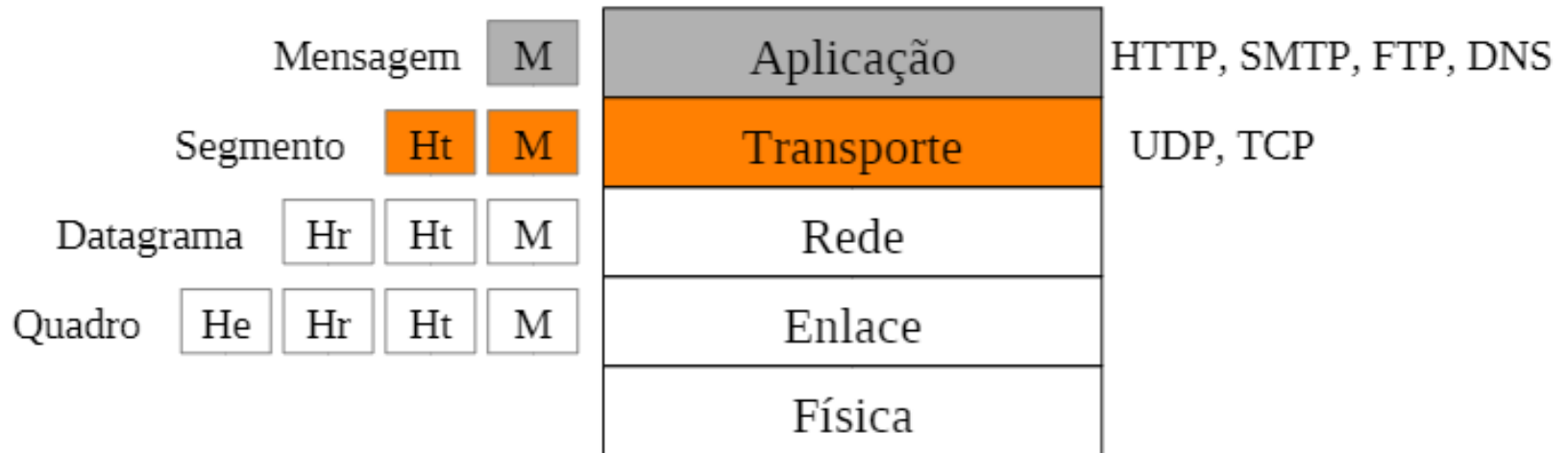


- Princípios e serviços da camada de transporte
- Protocolos de transporte : TCP e UDP

Camada de Transporte



Protocolos e serviços da camada de transporte

- Entender os princípios por trás dos serviços da camada de transporte:
 - *Multiplexação/demultiplexação*
 - *Transferência de dados confiável*
 - *Controle de fluxo*
 - *Controle de congestionamento*
- Protocolos de transporte na Internet:
 - *UDP: transporte não orientado à conexão*
 - *TCP: transporte orientado à conexão*
 - *Controle de congestionamento do TCP*

Serviços da Camada de Transporte

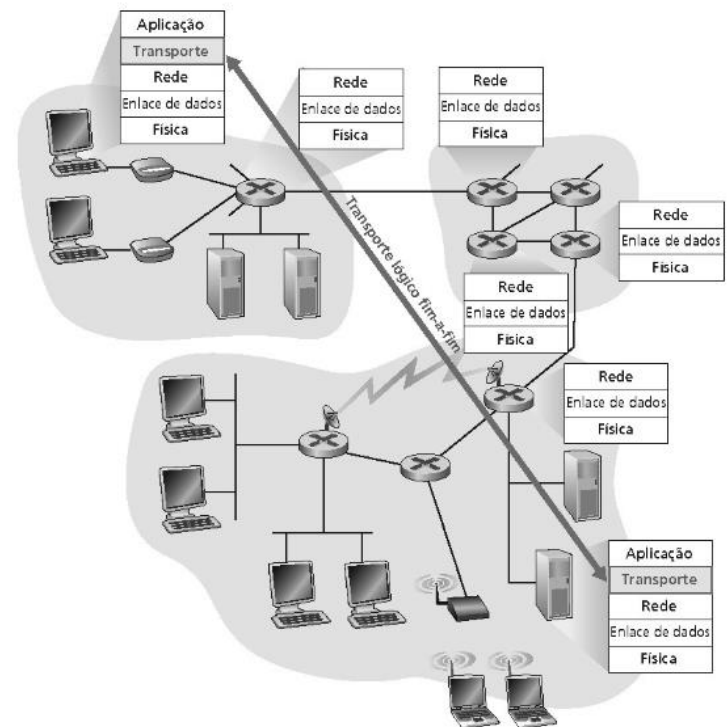
- Multiplexação e demultiplexação
- Transporte não orientado à conexão: UDP
- Princípios de transferência confiável de dados
- Transporte orientado à conexão: TCP
- Estrutura do segmento
- Transferência confiável de dados
- Controle de fluxo
- Gestão de conexão
- Princípios de controle de congestionamento
- Controle de congestionamento do TCP

Camada de Transporte

- A camada de transporte serve para verificar se a informação foi entregue sem erros ao destinatário. Para isso, a informação que é composta por um grande segmento de dados é dividida em pequenos segmentos que são agrupados novamente quando chegam ao destino passando-se o inverso.
- Esta camada permite ainda efetuara resolução de nomes de computadores em endereços, ou seja, podemos saber o nome de um computador através do seu endereço de IP e vice-versa

Serviços da camada de transporte

- Fornecem **comunicação lógica** entre processos de aplicação em diferentes hospedeiros
- Os protocolos de transporte são executados nos sistemas finais
- Lado emissor: divide as mensagens da aplicação em segmentos e envia para a camada de rede
- Lado recetor: restaura os segmentos em mensagens e passa para a camada de aplicação



Protocolos da camada de transporte

- A camada de transporte no modelo TCP/IP realiza a transferência de dados entre os extremos da comunicação.
- No lado emissor, os dados recebidos da camada de aplicação são divididos em segmentos antes de serem transmitidos.
- No lado recetor, os segmentos são reagrupados e entregues à camada de aplicação.
- Os protocolos usados são:
 - *TCP – transmission control protocol*
 - *UDP – user datagram Protocol*

Camada de transporte vs. camada de rede

- Camada de rede: comunicação lógica entre os hosts
- Camada de transporte: comunicação lógica entre os processos .
 - *Depende dos serviços da camada de rede*

Correção de erros de controlo

- Para existir correção de erros e controlo sobre o fluxo de informação são utilizados dois serviços de ligação:
 - *Ligação orientada à conexão (connection – oriented)*
 - *Ligação livre ou não orientada à conexão – (connectionless)*

Controlo de fluxo com janela

- Considerando que na camada de rede o IP não oferece qualquer garantia de entrega dos datagramas, cabe ao protocolo TCP administrar os temporizadores e retransmitir os datagramas sempre que necessário.
- Os datagramas também podem chegar fora de ordem e cabe ao TCP reorganizá-los em mensagens na sequência correta, fornecendo a confiabilidade que o IP não oferece.
- O TCP é um protocolo orientado à conexão que permite a entrega sem erros de um fluxo de bytes originados de uma determinada máquina para qualquer outra máquina da inter-rede.
- Esse protocolo atua na camada de transporte oferecendo um serviço de transferência de bytes fim a fim, de modo confiável, em uma inter-rede não-confiável.
- O protocolo básico utilizado pelas entidades TCP é o **protocolo de janelas** deslizantes usado para manter um registro do quadro de sequências enviadas e os respectivos reconhecimentos recebidos pelos utilizadores.
- Nas transmissões com controle de fluxo, temos uma janela de duração variável, a qual permite que um remetente possa transmitir determinado número de unidades de dados antes que uma confirmação seja recebida ou que um evento especificado ocorra. Consequentemente, o objetivo da janela deslizante é aumentar a taxa de transferência de pacotes melhorando a utilização do meio de transmissão.

TCP é constituído pelos campos seguintes:

Porto de origem - Número do porto da aplicação que enviou os dados;

Porto de destino - Número do porto 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 recetor espera receber;

Tamanho do cabeçalho - Tamanho do cabeçalho em palavras de 32 *bits*. É usado para 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:

TCP é constituído pelos campos seguintes:

Bits de controlo - Bits usados no controlo e gestão da sessão:

- **URG** — Quando activo, indica que os dados são prioritários. Os dados prioritários são processados 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 à 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.

Janela - Tamanho em bytes da janela usada no protocolo de controlo de fluxo;

Verificação de erros - Código CRC de verificação de erros. O código incide sobre Cabeçalho e os dados;

Indicador de prioridade - Indica qual o último byte numa sequência de dados prioritários. Assim, o recetor sabe o volume de dados prioritários que vai receber;

Opções - Permite indicar parâmetros de controlo. Por exemplo, o tamanho máximo de ~ segmento.

Transferência de Dados e Partilha de Ligação

O protocolo TCP garante a transferência de dados entre aplicações.

A aplicação do emissor gera os 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 de **porto**.

O conjunto formado pelo **número do porto**, pelo **endereço destino** e pelo nome do **protocolo de transporte** que deve ser usado na transmissão designa-se por **SOCKET**.

Numa comunicação de dados, existe um socket origem e um socket destino

Sockets

A definição correta dos sockets é fundamental para a partilha de uma ligação por várias aplicações. Exemplo:

- Admitindo que uma organização tem um servidor que responde a pedidos FTP e HTTP. Se um determinado cliente aceder ao servidor para realizar uma transferência 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 de um cliente.
- Se por outro lado, existirem dois clientes em máquinas diferentes a acederem ao mesmo servidor FTP, o servidor consegue distinguir os pedidos através do endereço lógico dos clientes e através dos portos das aplicações dos clientes que realizaram o pedido FTP.
- Se os pedidos viessem da mesma máquina, a forma única de os distinguir seria através dos números dos portos origem, uma vez que o endereços lógicos de destino e origem seriam os mesmos.

Portos

- Os portos desempenham um papel fundamental para que as aplicações diferentes partilhem a mesma ligação TCP.
- A atribuição de portos às aplicações de servidores obedece a um mapeamento predefinido.
- As conversações que não envolvem uma aplicação com número de porto conhecido recebem números de porto aleatórios num intervalo específico acima de 1023.
- Alguns portos são reservadas no TCP e no UDP, embora possa haver aplicações que não os suportem. Os números de portos têm os seguintes intervalos atribuídos:
- Números **abaixo de 1024** são considerados números de portos conhecidos, **aplicações de servidor**.
- Números **acima de 1023** recebem números de porto **atribuídos dinamicamente**.
- **Números de porto registados** são aqueles registados para aplicações específicas de fabricantes. A maioria desses números é superior a 1024.

Portos Lógicos

- Numa máquina existem 65.536 portos TCP e 65.536 UDP que podem ser usadas pelas mais diversas aplicações/serviços, o que (teoricamente) poderíamos ter 65.536 aplicações/serviços distintos a correr em simultâneo na nossa máquina.

Mas como saber quais o portos abertos num PC ?

TCPView

- O [TCPView](#) é um programa do Windows que mostra listagens detalhadas de todos os pontos de extremidade TCP e UDP no seu sistema, incluindo os endereços locais e remotos e o estado das conexões TCP.
- Usar a linha de comandos para saber quais os portos abertos
 - *Para saber todas as ligações TCP e UDP estabelecidas entre a nossa máquina e outras basta que use o comando:*

netstat -a

Correção de erros de controlo

Ligação orientada (connection – oriented)

- No caso da ligação orientada, a mesma vai utilizar serviços que permitem a ligação e verificação de erros entre os dispositivos emissor e recetor e manter a ligação de forma a que não seja necessário voltar a repetir o envio da informação.
- Esta manutenção é feita por segmentos de informação designados de ACKs ou acknowledgments. O protocolo usado é o TCP

Correção de erros de controlo

Ligação livre – (connectionless)

- O serviço connectionless não possui verificação e correção de erros nem controlo sobre o fluxo de informação
- Por outro lado, é muito mais rápido transmitir a informação
- Se houver perda de alguma parte da informação é necessário repetir o envio de todo o conteúdo
- Nesta situações é usado o protocolo UDP – User Datagram Protocol

Protocolo UDP

- O UDP é um protocolo mais simples e por si só não fornece garantia na entrega dos pacotes. No entanto, esse processo de garantia de dados pode ser simplesmente realizado pela aplicação em si (que usa o protocolo UDP) e não pelo protocolo.
- Os protocolos, como o TFTP e o NFS, necessitam de velocidade na transferência de dados, utilizam o protocolo UDP

Portas TCP/UDP

- Numa máquina existem (teoricamente) 65.536 portas TCP que podem ser usadas pelas mais diversas aplicações/serviços, o que (teoricamente) poderíamos ter 65.536 aplicações/serviços distintos a correr em simultâneo na nossa máquina.
- Relembrando: o IP identifica a máquina e o porto identifica a aplicação/serviço. Além das portas TCP também existem 65.536 portas UDP(teoricamente).

[Portas](#)

TCP/UDP ?

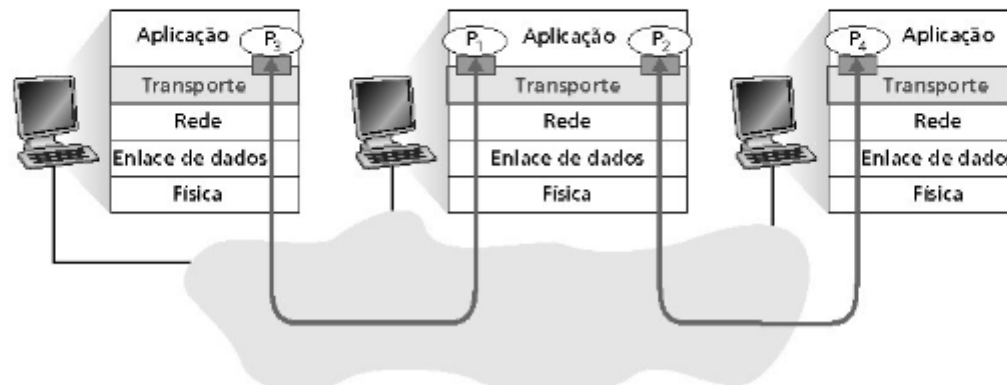
Aplicação prática

- A escolha entre o uso do protocolo TCP ou UDP cabe ao desenvolvedor de cada aplicação, que deve decidir o que ela precisa. Embora muitos prefiram a segurança e confiabilidade oferecidas pelo TCP, outros se beneficiam mais optando pela velocidade de transmissão gerada pelo UDP

Multiplexação/demultiplexação

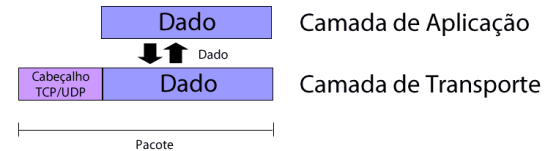
coleta dados de múltiplos sockets,
envelopa os dados com cabeçalho
(usado depois para
demultiplexação)

entrega os segmentos
recebidos ao socket correto



Síntese

Protocolos da camada de transporte



TCP

Conexão confiável utilizada para controlar o gerenciamento das aplicações a nível de serviços entre computadores. Faz tanto o transporte em sequência do dado quanto a checagem da integridade dos mesmos.

UDP

Conexão não confiável utilizada para controlar o gerenciamento das aplicações a nível de serviços entre computadores e é utilizado para o transporte de algum dados onde a própria aplicação faz a verificação da integridade dos dados;

ICMP

Internet control message protocol - fornece a gestão e o relatório de erros para ajudar no gerenciamento de dados durante a comunicação entre computadores. Esta conexão é utilizada para reportar o status do computador que está sendo conectado ao computador que está tentando conectar como por exemplo reportar que o computador de destino não está acessível.

IGMP

Internet Group Management Protocol utilizado para suportar mensagens multicasting e rastrear grupos de utilizadores na rede de computadores.

Referências

- <https://www.citisystems.com.br/protocolo-tcp-ip/>
- <https://docplayer.com.br/3825192-Redes-de-computadores-e-a-internet-capitulo3-camada-de-transporte.html>