

Fundamentos de Redes de Computadores

Aula 6 - Protocolos de Transporte

INTRODUÇÃO



Quando enviamos um pacote de uma máquina para outra, desejamos ter certeza de que ele chegou a seu destino, se chegou íntegro, e sem erro.

Esse tipo de controle é responsabilidade da camada de transporte da Arquitetura TCP/IP que iremos estudar nesta aula.

OBJETIVOS



Conhecer os protocolos TCP e UDP.

CAMADA DE TRANSPORTE

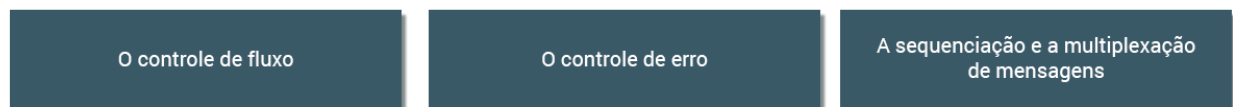
Essa camada reúne os protocolos que realizam as funções de transporte de dados fim a fim, ou seja, considerando apenas a origem e o destino da comunicação, sem se preocupar com os elementos intermediários.

Na arquitetura TCP/IP, a camada de transporte possui dois protocolos que são o UDP (User Datagram Protocol) e TCP (Transmission Control Protocol).

O protocolo UDP realiza apenas a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente.

O protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino mais confiável.

São responsabilidades do protocolo TCP:



Atenção

, A camada de transporte oferece para o nível de aplicação um conjunto de funções e procedimentos para acesso ao sistema de comunicação de modo a permitir a criação e a utilização de aplicações de forma independente da implementação. Dessa maneira, as interfaces socket (ambiente Unix) e Winsock (ambiente Windows) fornecem um conjunto de funções-padrão para permitir que as aplicações possam ser desenvolvidas independentemente do sistema operacional no qual rodarão.

PROTOCOLO UDP

O protocolo UDP fornece uma forma simples de acesso ao sistema de comunicação, provendo um serviço sem conexão, sem confiabilidade e sem correção de erros.

Atenção

, A **principal função** do nível de transporte implementada em UDP é a **capacidade de multiplexação de acesso ao sistema de comunicação**., Essa função permite que vários processos ou programas executados em um computador possam acessar o sistema de comunicação, para que o tráfego de dados respectivo a cada um deles seja corretamente identificado, separado e utilize buffers individuais.

Um processo	É o programa que implementa uma aplicação do sistema operacional, e que pode ser uma aplicação do nível de aplicação TCP/IP.
A forma de identificação	A forma de identificação de um ponto de acesso de serviço (SAP) do modelo OSI é a porta de protocolo em TCP/IP.
A porta	É a unidade que permite identificar o tráfego de dados destinado a diversas aplicações.

A identificação única de um processo acessando os serviços TCP/IP é, então, o endereço IP da máquina e a porta (ou portas) usadas pela aplicação.

Cada processo pode utilizar mais de uma porta simultaneamente, mas uma porta só pode ser utilizada por uma aplicação em um dado momento. Uma aplicação que deseje utilizar os serviços de comunicação deverá requisitar uma ou mais portas para realizar a comunicação. A mesma porta usada por uma aplicação pode ser usada por outra, desde

que a primeira tenha terminado de utilizá-la.

PORTAS

Tanto o TCP quanto o UDP usam números de porta (ou soquete) para passar as informações às camadas superiores. Os números de portas são usados para manter registro de diferentes conversações que cruzam a rede ao mesmo tempo. Os desenvolvedores de aplicações de software concordaram em usar os números de portas bem conhecidos, que estão definidos no RFC1700. Toda conversação destinada à aplicação HTTP usa o número de porta padrão 80. Conversações, que não envolvem aplicações com números de portas bem conhecidos, recebem números de porta que foram selecionados aleatoriamente em um conjunto específico. Esses números de portas são usados como endereços de origem e destino no segmento TCP.

Algumas portas são reservadas no TCP e no UDP, embora possa não haver aplicações para suportá-los. Os números de portas têm os seguintes conjuntos atribuídos:

Portas Conhecidas (números 0 a 1023)

Esses números estão reservados para serviços e aplicações. Eles são comumente usados para aplicações como o HTTP (servidor web), SMTP (envio de e-mail) e DNS (resolução de nomes de domínio). Através da definição dessas portas conhecidas para aplicações de servidor, aplicações de clientes podem ser programadas para solicitar uma conexão com essa porta específica e seu serviço associado.

Portas Registradas (números 1024 a 49151)

Esses números de portas são designados para processos ou aplicações de usuário. Esses processos são principalmente aplicações individuais que um usuário escolheu para instalar, em vez de aplicações comuns que receberiam uma Porta Conhecida. Quando não usadas para um recurso de servidor, essas portas também podem ser dinamicamente selecionadas por um cliente como sua porta de origem.

Portas Dinâmicas ou Privadas (números 49152 a 65535)

Elas são geralmente designadas dinamicamente a aplicações de cliente quando se inicia uma conexão. Não é muito comum um cliente se conectar a um serviço usando uma Porta Dinâmica ou Privada (embora alguns programas de compartilhamento de arquivos peer-to-peer o façam). Os sistemas finais usam números de portas para selecionar as aplicações corretas. Os números de portas de origem são atribuídos dinamicamente pelo host de origem, normalmente são números maiores do que 1023.

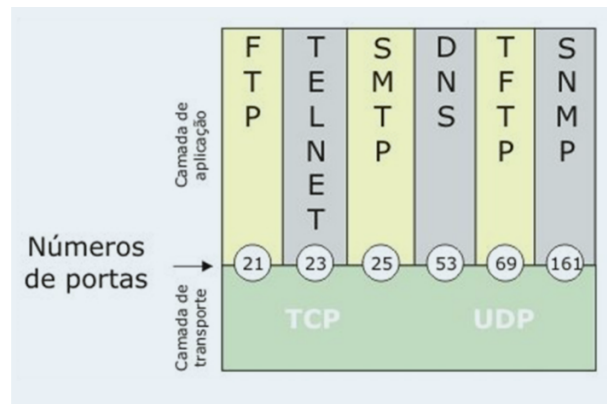


Figura 1 Portas

Fonte: autor adaptado de Tannenbaum 2007.

A forma de utilização de portas mostra uma distinção entre a parte cliente e a parte servidora de uma aplicação TCP/IP. O programa cliente pode utilizar um número de porta qualquer, já que nenhum programa na rede terá necessidade de enviar uma mensagem para ele. Já uma aplicação servidora deve utilizar um número de porta bem-conhecido (Well-known ports) de modo que um cliente qualquer, querendo utilizar os serviços do servidor, tenha que saber apenas o endereço IP da máquina em que este está executado.

Se não houvesse a utilização de um número de porta bem-conhecido, a arquitetura TCP/IP deveria possuir um mecanismo de diretório para que um cliente pudesse descobrir o número da porta associado ao servidor. Para evitar esse passo intermediário, utiliza-se números de porta bem-conhecidos e o cliente já possui pré-programado em seu código o número de porta a ser utilizado.

A figura 2 ilustra a multiplexação/demultiplexação realizada pelo protocolo UDP, camada de transporte:

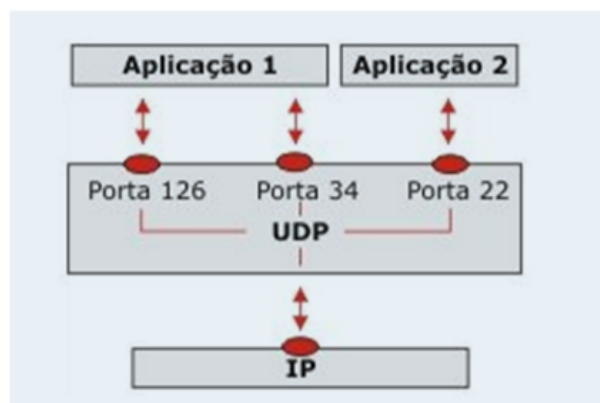


Figura 2 Endereçamento de Porta

Fonte: autor adaptado de Tannenbaum 2007.

PROTOCOLO TCP

O protocolo TCP trabalha no mesmo nível que o protocolo UDP, mas oferece serviços mais complexos, que incluem controle de erros e fluxo, serviço com conexão e envio de fluxo de dados.

O protocolo TCP oferece as seguintes características:

- Controle de Fluxo e Erro fim a fim;
- Serviço confiável de transferência de dados;
- Comunicação full-duplex fim a fim;

- A aplicação basta enviar um fluxo de bytes;
- Ordenação de mensagens;
- Opção de envio de dados urgentes.

As aplicações mais comuns que usam TCP são: **TELNET, FTP, SMTP, HTTP.**

A conexão TCP é ilustrada na figura 3:

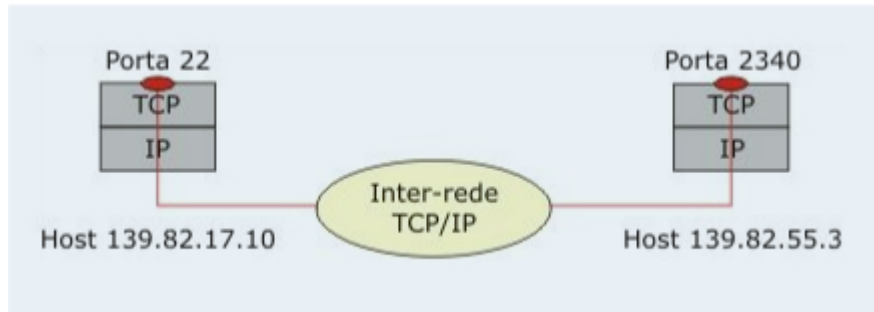


Figura 3 Conexão TCP

Fonte: autor adaptado de Tannenbaum 2007.

Uma conexão TCP é formada por três fases:

1. o **estabelecimento** de conexão;
2. a **troca** de dados, e
3. a **finalização** da conexão.

Veja a ilustração dessas fases na figura a seguir:

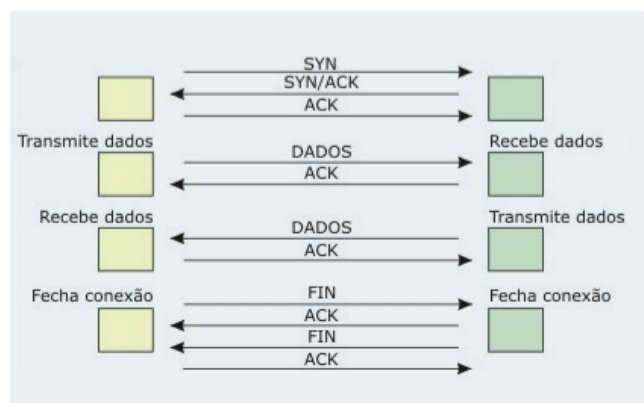


Figura 4 Conexão TCP

Fonte: autor adaptado de Tannenbaum 2007.

Vejamos um exemplo:

Observe a figura a seguir.

Quando a estação de trabalho A solicita uma sessão de emulação de terminal TELNET com o host Z, ela envia um segmento de início de estabelecimento de sessão SYN com os campos **source port** e **destination port** preenchidos da seguinte forma:

- **Destination port** identifica o processo servidor e para Telnet o valor é 23;
- **Source port** contém um endereço gerado randomicamente que identifica o processo cliente, no caso 1028.

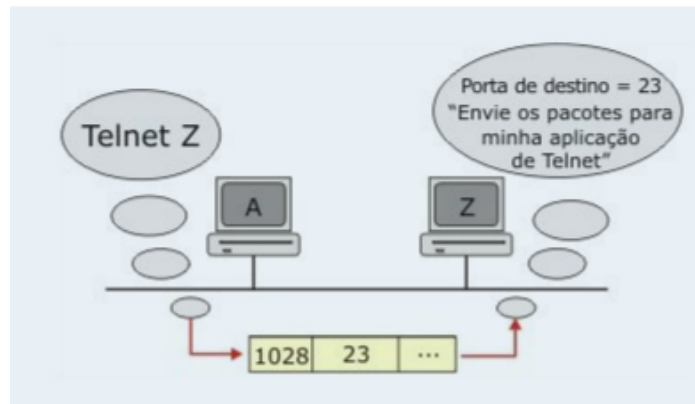


Figura 5 Funcionamento TCP
Fonte: autor adaptado de Tannenbaum 2007.

O **host Z**, ao receber esse segmento, irá verificar se a aplicação está ativa antes de dar o aceite do pedido de estabelecimento de sessão através do segmento SYN-ACK.

Cada **segmento TCP** enviado, tem um número de sequência para que o **módulo TCP** no host destino possa reordená-los na chegada.

Observe agora a figura a seguir.

Quando o **nó destino** recebe um segmento, envia uma confirmação através de um segmento TCP com o campo **acknowledgement** preenchido. Nesse campo, está o número de sequência do próximo segmento esperado, indicando para o **nó origem** o correto recebimento, pelo nó destino, dos pacotes anteriores.

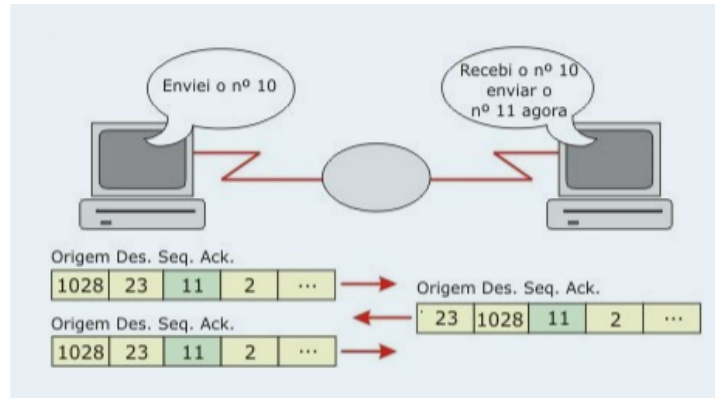


Figura 6 Funcionamento TCP
Fonte: autor adaptado de Tannenbaum 2007.

O sequenciamento dos pacotes TCP é orientado a byte.

O número de sequência do segmento é sempre igual ao número de sequência do segmento anteriormente transmitido, somado ao número de bytes transmitidos. O primeiro número de sequência é gerado randomicamente, e é determinado no estabelecimento da conexão TCP.

A confirmação de entrega dos segmentos TCP é também orientada a byte. O valor do campo acknowledgement de um segmento é sempre igual ao número de sequência do segmento que está sendo confirmado somado ao número de bytes recebidos (gráfico à esquerda).

Esse valor indica para o nó origem o número de sequência do próximo segmento que o nó destino espera receber. (figura 7)

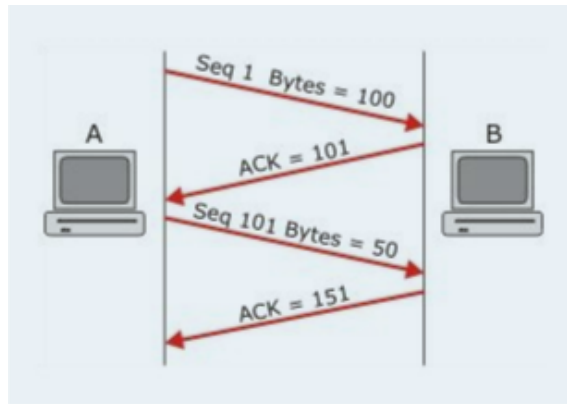


Figura 7 Funcionamento TCP
Fonte: autor adaptado de Tannenbaum 2007.

Após o envio dos dados, a sessão TCP pode ser encerrada por qualquer uma das partes (cliente ou servidor) elegantemente através de um segmento FIN. Esse segmento não possui dados, sendo reconhecido por ter o bit FIN do campo flag do cabeçalho TCP "ligado" (figura 8).

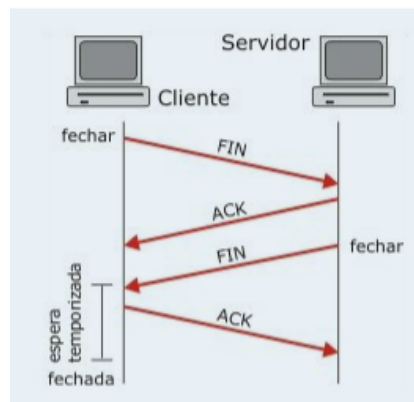


Figura 8 Funcionamento TCP – Fechamento de Conexão
Fonte: autor adaptado de Tannenbaum 2007.

O controle de fluxo do TCP é implementado através de um mecanismo de janela.

A janela define quantos bytes podem ser enviados sem a necessidade do recebimento de uma confirmação.

Essa janela está relacionada ao tamanho do buffer de recepção do destinatário, e o seu valor inicial é informado quando do estabelecimento da sessão TCP.

RESUMO DAS CARACTERÍSTICAS DOS PROTOCOLOS DE TRANSPORTE

O TCP



Fornecer um circuito virtual entre aplicações do usuário final;



Orientado para conexão;



Confiável;



Divide as mensagens enviadas em segmentos;



Reagrupa as mensagens na estação de destino;



Reenvia tudo o que não foi recebido;



Reagrupa as mensagens a partir de segmentos recebidos.

O UDP



Transporta dados sem confiabilidade entre hosts;



Sem conexão;



Transmite mensagens (chamado de datagramas do usuário);



Não fornece verificação de software para a entrega da mensagem (não é confiável);



Não reagrupa as mensagens de entrada;



Não usa confirmações;



Não fornece controle de fluxo.

ATIVIDADES

Acesse os sites e responda questões online de redes:

<https://www.qconcursos.com/questoes-de-concursos/disciplinas/tecnologia-da-informacao-redes-de-computadores> (glossário)

<https://www.aprovaconcursos.com.br/questoes-de-concurso/disciplina/redes-de-computadores> (glossário)

<https://www.gabaritou.com.br/Questao?DisciplinalD=11> (glossário)

Acesse os sites e explore os recursos online:

Redes de computadores e a internet. Disponível em:

http://wps.aw.com/br_kurose_redes_3/40/10271/2629589.cw/index.html (glossário)

Animações do livro do Forouzan. Disponível em:

http://highered.mheducation.com/sites/0072967722/student_view0/animations.html# (glossário)

Questão 1 - A Porta 80 utilizada pelo HTTP é denominada:

☐

Porta Registrada

☐

Porta Dinâmica

☐

Porta Conhecida

☐

Porta Privada

☐

Porta de Protocolo

☐☐☐☐☐

Justificativa

Questão 2 - O protocolo de transporte confiável da pilha TCP/IP é o:

☐

IP

☐

UDP

☐

HTTP

☐

TCP

☐

DNS

☐☐☐☐☐

Justificativa

Questão 3 - O FLAG que sinaliza o início de uma conexão do TCP é o:

☐

SYN

☐

FIN

☐

PUSH

☐

ACK

☐

SEQ



Justificativa

Glossário