

# REDES DE COMPUTADORES

Prof. Priscilla Cunha

[pcunha@uni9.pro.br](mailto:pcunha@uni9.pro.br)

# Agenda

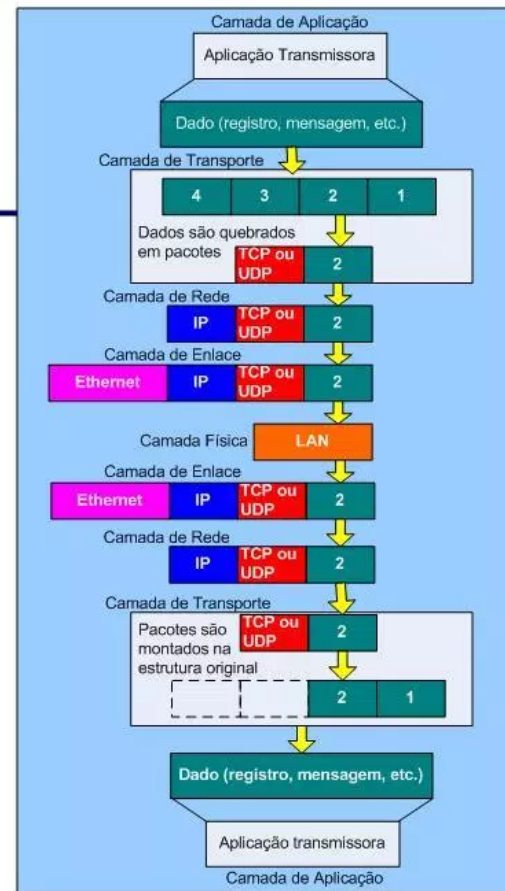




# **CAMADA DE TRANSPORTE**

# Modelo OSI

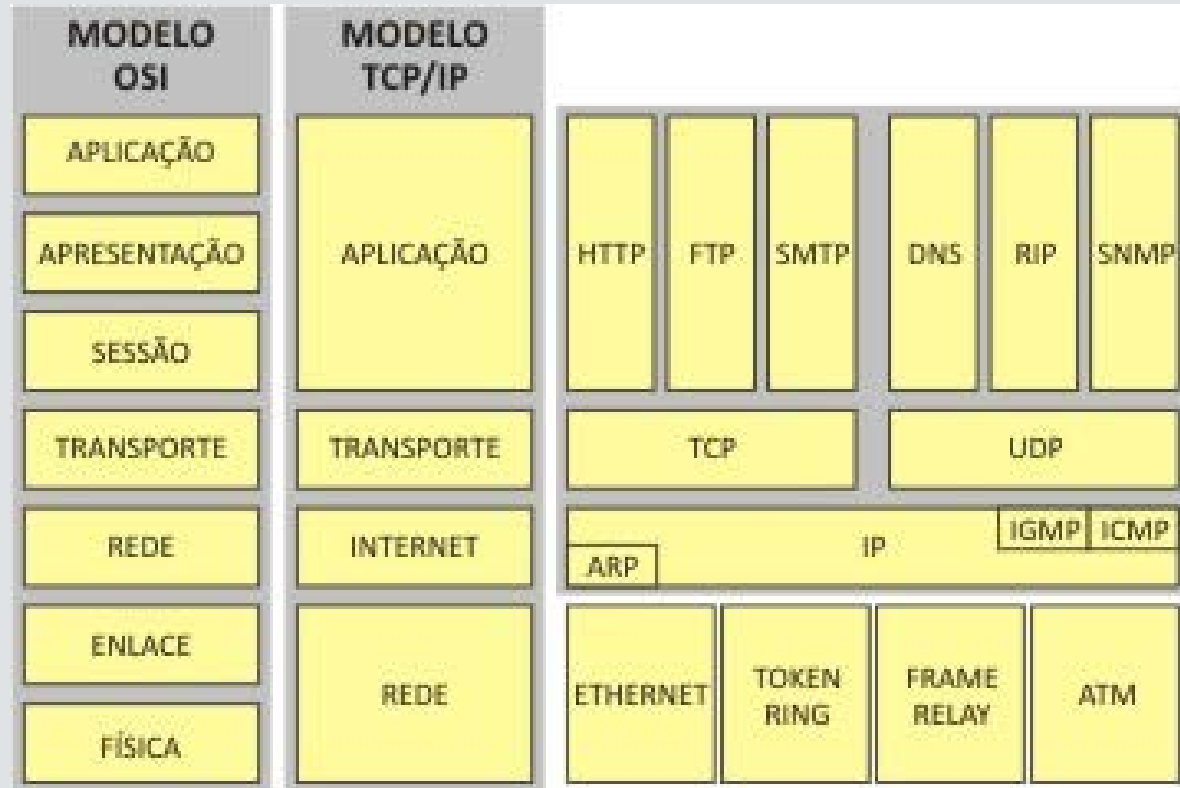
- **Camada de Aplicação**
  - Tipo de comunicação
  - E-mail; transferência de arquivos, cliente/servidor
- **Camada de Apresentação**
  - Criptografia
  - Conversão de código (ASCII para EBCDIC)
- **Camada de Sessão**
  - Início e término de sessão
  - Controle de sequência
- **Camada de Transporte**
  - Assegura a transmissão fim-a-fim dos arquivos e mensagens
- **Camada de Rede**
  - Encaminha os dados para diferentes LANs e WANs baseado no endereçamento da rede
- **Camada de Enlace de Dados**
  - Transmite os pacotes de um nó de rede para outro baseado no endereço da estação
- **Camada Física**
  - Sinais elétricos e características mecânicas da transmissão de bits



# Arquitetura TCP/IP



# Modelo OSI x Modelo TCP/IP



# Camada de Transporte

- O principal objetivo da camada de transporte é oferecer um serviço confiável, eficiente a seus usuários que são processos presentes na camada de aplicação.
- Para atingir esse objetivo, a camada de transporte utiliza vários serviços oferecidos pela camada de rede.
- Unidade de transmissão: segmento

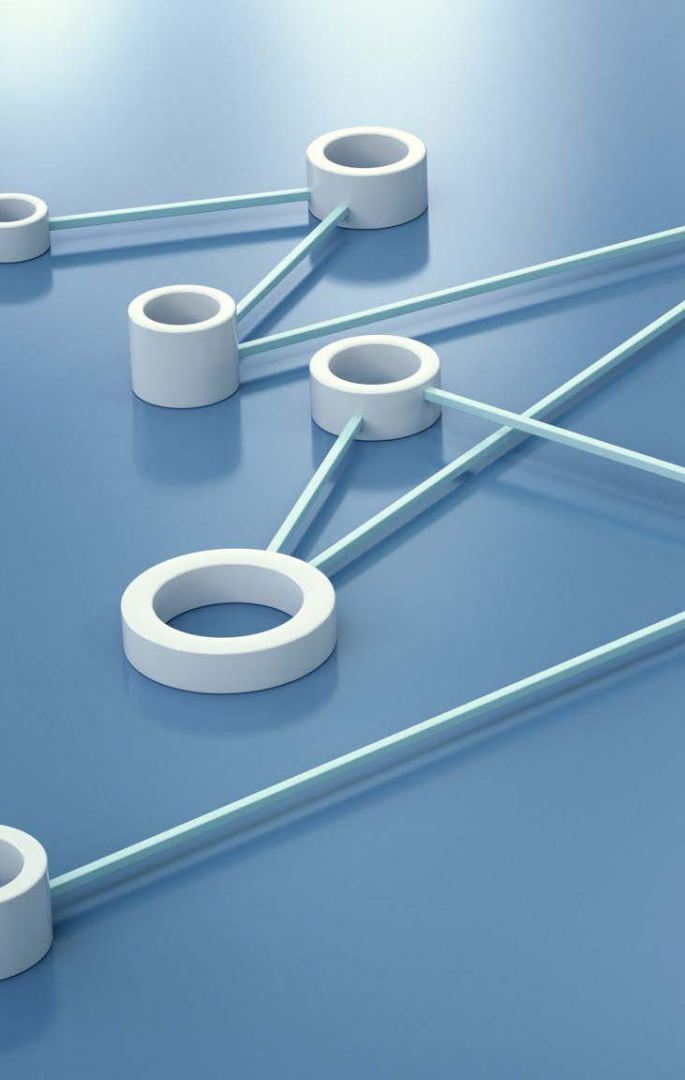
- A camada de transporte trata da qualidade do serviço, transportando as informações e regulando seu fluxo, fazendo entregas mais confiáveis.
- No modelo OSI podemos dizer que as primeiras camadas são provedores de transporte e as camadas superiores são usuárias do serviço de transporte.



- Existem dois tipos de serviço de transporte:
  - Orientado à conexões: onde o envio dos pacotes se dá somente após a negociação dos parâmetros, e as conexões são realizadas em três fases: o estabelecimento, a transferência de dados e o encerramento.
  - Não orientado a conexão: onde o envio de pacotes se dá sem qualquer negociação entre origem e destino.

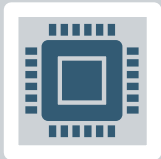
- Essa camada realiza 2 tarefas:
  - Controle de fluxo
    - Efetua o estabelecimento de chamada, controle no envio e recebimento de segmentos e finalização de uma chamada
  - Multiplexação
    - Encaminha a informação para a aplicação correspondente por meio de um identificador conhecido como porta

- Nessa camada temos 2 protocolos:
  - TCP: com funções para tráfego de arquivos que necessitam de garantia de entrega e que seja confiável
  - UDP: sem garantia de entrega, mais simples, muito voltado a aplicações em tempo real como voz e vídeo ao vivo

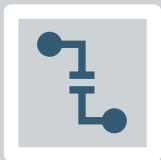


# Serviço Orientado a Conexão

- Na fase de estabelecimento da conexão, um único caminho entre a origem e o destino é determinado.
- Os recursos são normalmente reservados nesse momento para garantir um nível consistente de serviço.



Durante a fase de transferência de dados, os dados são transmitidos em sequência pelo caminho estabelecido, chegando ao destino na sequência como foram enviados.

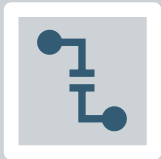


A fase de encerramento da conexão consiste em terminar a conexão entre a origem e o destino quando não for mais necessária.



Os hosts TCP estabelecem uma sessão orientada a conexão de forma confiável, chamado de aperto de mão de três vias (three way handshake).

- Uma sequência de conexão handshake triplo/aberta sincroniza a conexão nas duas extremidades antes dos dados serem transferidos.
- A troca de números de sequência de introdução, durante a sequência de conexão, é importante, pois garante que dados perdidos, devido a problemas de transmissão que possam ocorrer mais adiante, possam ser recuperados.



A retransmissão e confirmação positiva, ou PAR (Positive Acknowledgment and Retransmission), é uma técnica comum que muitos protocolos usam para fornecer confiabilidade.



Com a PAR, a origem envia um pacote, aciona um timer e espera por uma confirmação antes de enviar o próximo pacote.



Se o timer expirar antes da origem receber uma confirmação, a origem retransmitirá o pacote e iniciará novamente o timer.

O tamanho da janela determina a quantidade de dados que pode ser transmitida de uma vez antes de receber uma confirmação do destino.

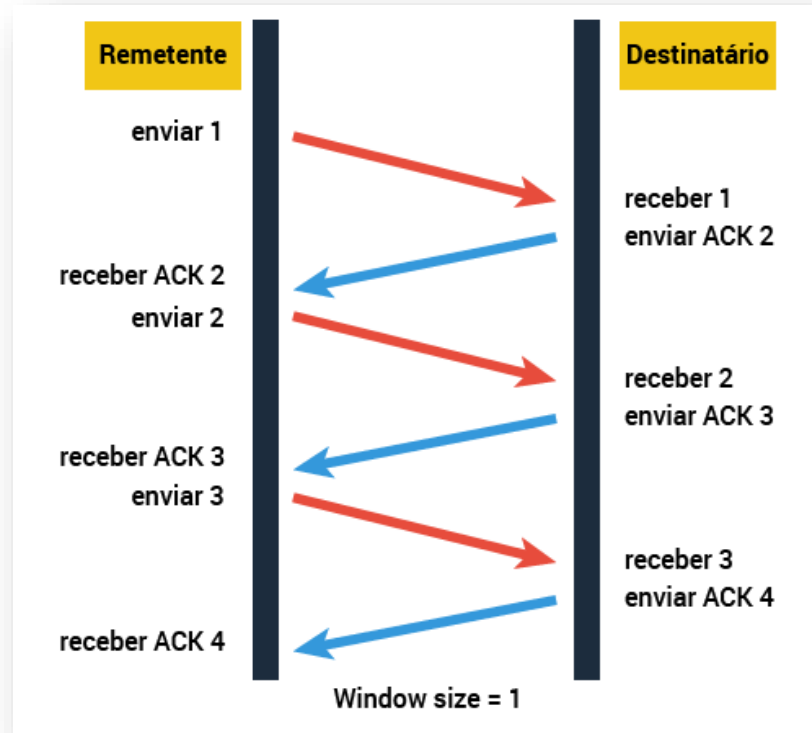
Quanto maior o tamanho da janela (bytes), maior a quantidade de dados que o host pode transmitir.

Depois que um host transmitir o número de bytes da janela dimensionada, ele tem de receber uma confirmação de que os dados foram recebidos antes de poder enviar mais mensagens.




- Esse processo é conhecido também como janelamento, que é um mecanismo de controle de fluxo que exige que o dispositivo de origem receba uma confirmação do destino depois de transmitir uma determinada quantidade de dados.


- Por exemplo, com um tamanho de janela 1, cada segmento individual (1) tem de ser confirmado antes que o próximo segmento possa ser enviado.
- A parte "móvel" da janela móvel, refere-se ao fato de que o tamanho seja negociado dinamicamente durante a sessão TCP.



Em outro exemplo, uma janela de tamanho três, o dispositivo da origem pode enviar três segmentos ao destino.



Ele deve então, aguardar por uma confirmação.



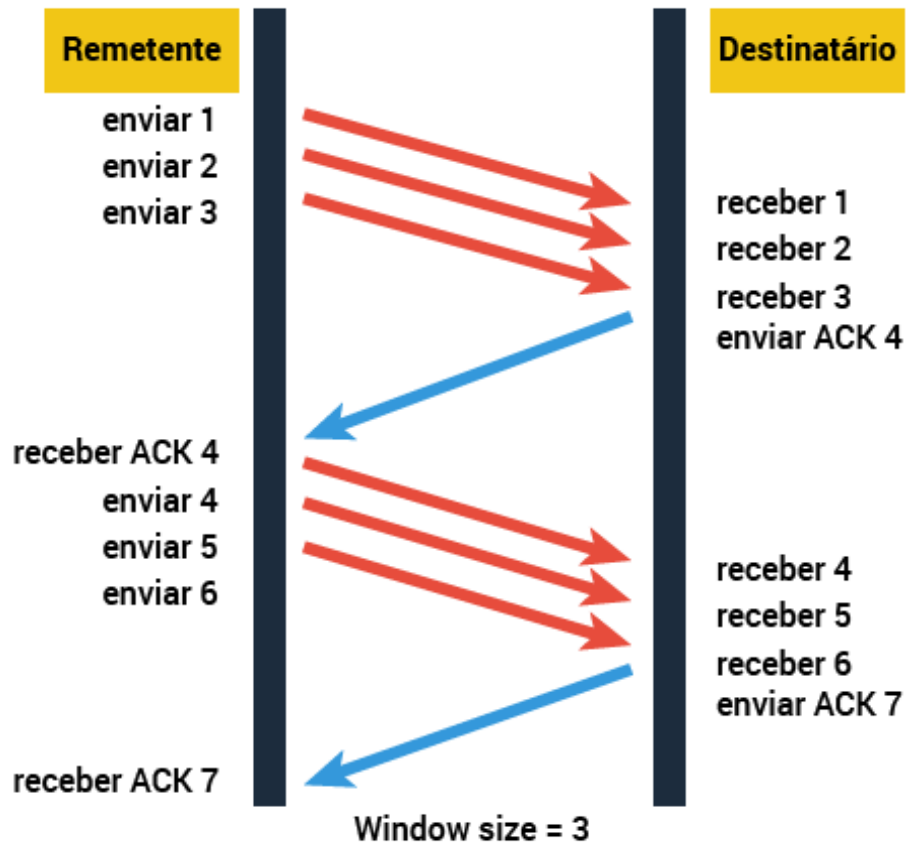
Se o destino receber os três, ele enviará uma confirmação ao dispositivo da origem, que agora poderá transmitir mais três segmentos.



Se, por algum motivo, o destino não receber os três segmentos, por exemplo, devido à sobrecarga de buffers, ele não enviará uma confirmação.



Por não receber a confirmação, a origem saberá que os mesmos deverão ser retransmitidos e que a taxa de transmissão deverá ser diminuída.





# Controle de Fluxo

- A camada de transporte provê um serviço de controle de fluxo às suas aplicações, para eliminar a possibilidade de o remetente estourar o buffer do destinatário
- É um serviço de compatibilização de velocidades e oferece serviço de controle de fluxo fazendo que o remetente mantenha uma variável denominada janela de recepção.
- O controle de fluxo é implementado pelo TCP.

O controle de fluxo é feito através do janelamento, já explicado anteriormente.

Antes de iniciar a conexão o host de origem envia um flag (1 bit) conhecido como SYN com um número de sequência (x) para o host de destino.

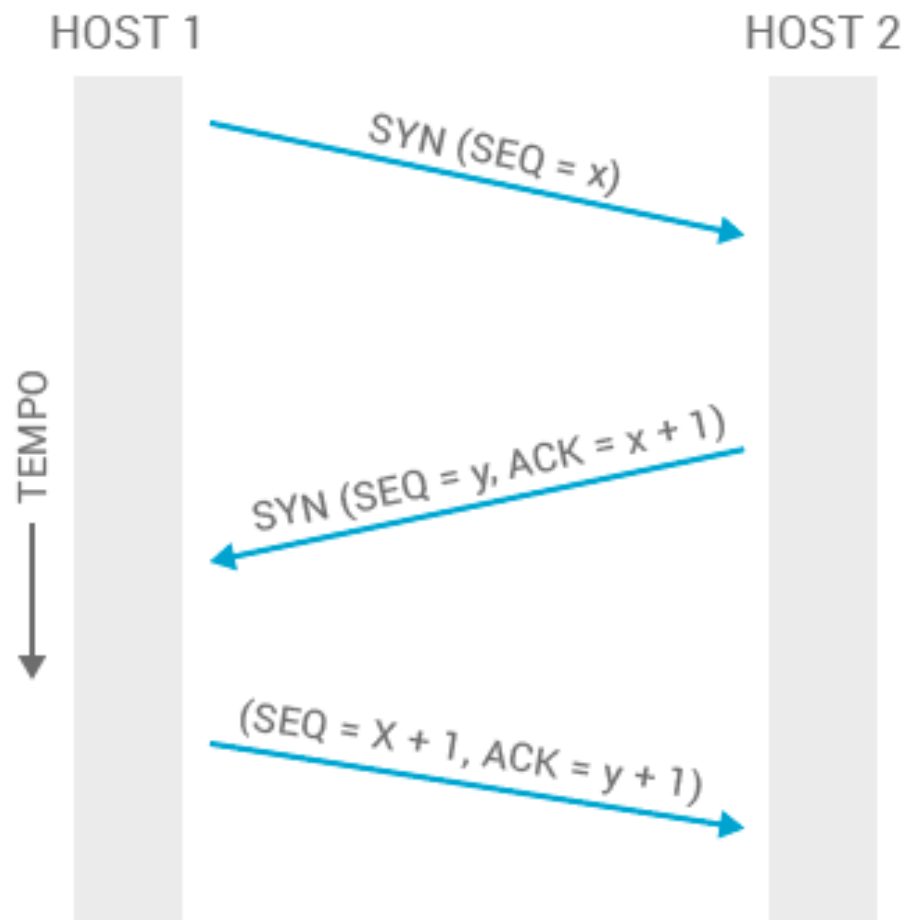


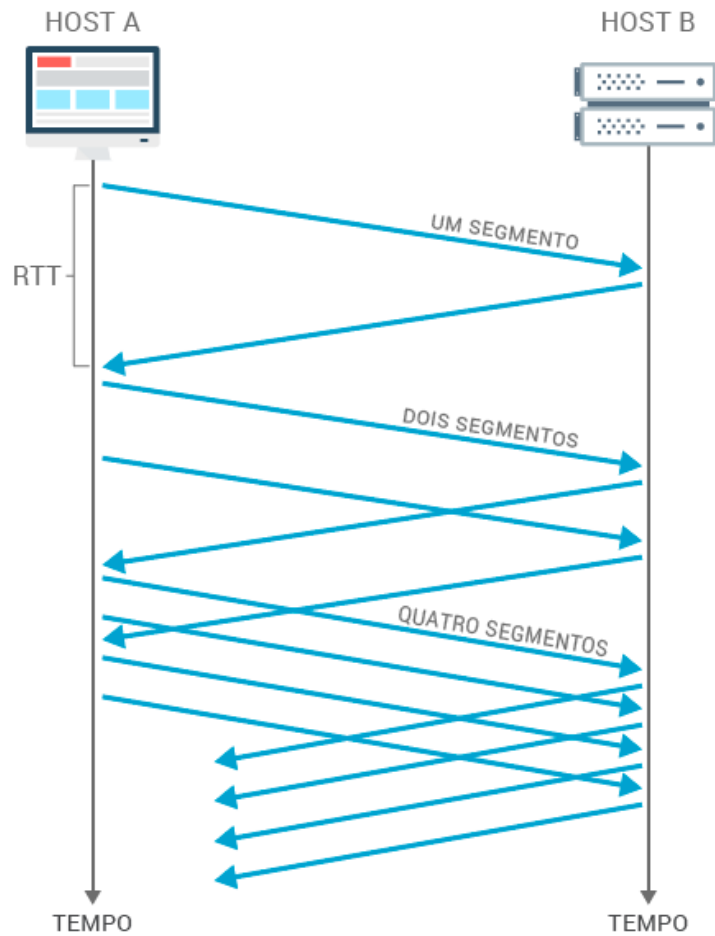
- Assim que o datagrama IP contendo o segmento TCP SYN chega ao host de destino, esse extrai o segmento TCP SYN do datagrama, aloca buffers e variáveis TCP à conexão e envia um segmento de aceitação de conexão ao TCP cliente,  $ACK = x+1$ , onde  $x$  refere-se ao número de sequência enviado host de origem.

Como a conexão é bidirecional, o host de destino aproveita o envio do ACK para enviar também a solicitação de conexão SYN com número de sequência  $y$ .

Da mesma forma o host de origem devolve o  $ACK = y + 1$  e o número de sequência  $x$  acrescido de  $+ 1$ .

- Ao receber o segmento SYNACK, o host de origem também reserva buffers e variáveis para a conexão.
- Completadas as três etapas, conhecido como three-way-handshake, o elemento de rede de origem e o servidor podem enviar segmentos contendo dados um ao outro.







# TCP – TRANSMISSION CONTROL PROTOCOL

---

Orientado a conexão (se conecta com o destino antes de transmitir)

---

Confiável (faz controle de fluxo e verifica se tudo chegou ao destino)

---

Full duplex (envia e recebe pacotes simultaneamente)

---

Faz parte do TCP/IP

---

Divide as mensagens em partes menores e atribui números de sequência a elas

---

Reagrupa a mensagem no destino

---

Reenvia o que não foi recebido



O TCP fornece a sequência de segmentos com uma confirmação de referência de encaminhamento.



Cada datagrama é numerado antes da transmissão.



Na estação receptora, o TCP reagrupa os segmentos em uma mensagem completa.



Se um número de sequência estiver faltando na série, aquele segmento será retransmitido.



Os segmentos que não forem confirmados dentro de um dado período de tempo serão retransmitidos.



# Cabeçalho TCP

| Bit 0                         |  | Bit 15        |  | Bit 16                |  | Bit 31       |  |
|-------------------------------|--|---------------|--|-----------------------|--|--------------|--|
| Porta de Origem (16)          |  |               |  | Porta de Destino (16) |  |              |  |
| Número de Sequência (32)      |  |               |  |                       |  |              |  |
| Número de Confirmação (32)    |  |               |  |                       |  |              |  |
| Comprimento do Cabeçalho (4)  |  | Reservado (6) |  | Bits de Código (6)    |  | Janela (16)  |  |
| Checksum (16)                 |  |               |  |                       |  | Urgente (16) |  |
| Opções (0 ou 32 caso existam) |  |               |  |                       |  |              |  |
| Dados (variam)                |  |               |  |                       |  |              |  |

↑

20 Bytes

↓

- O tamanho do cabeçalho TCP é medido em palavras de 32 bits.
  - O tamanho mínimo é de 5 palavras (20 bytes, sem opções).
  - O tamanho máximo é de 15 palavras (60 bytes, com opções).

# Campos do TCP

- Porta de origem e de destino - contém os números das portas TCP definidos para programas aplicativos.
- Número de sequência - é orientado a byte e corresponde à sequência do segmento anteriormente transmitido, somado ao número de bytes transmitidos.
- Número de confirmação - Orientação a byte e corresponde à sequência do segmento que está sendo confirmado, somado ao número de bytes recebidos.

- HLEN - Tamanho do cabeçalho do TCP (termina onde os dados começam).
- Reservado - Reservado para uso futuro.
- Bits de código:
  - URG: Envio de dados urgentes.
  - ACK: Confirmação dos dados enviados anteriormente.
  - PSH: Envia rapidamente os dados depois que lê o segmento.
  - RST: Reset de conexão.
  - SYN: Inicia uma conexão.
  - FIN: Finaliza uma conexão.

- Janela - indica os buffers (memória) disponíveis no receptor (controle de fluxo).
- Checksum (soma de verificação) - inclui o cabeçalho TCP, os dados e um pseudo cabeçalho para permitir a máxima confiabilidade.
- Ponteiro de urgente - identifica o número de sequência do octeto seguinte ao dado de urgência.
- Opções - para recursos não previstos originalmente.
- Preenchimento Dados - Dados da camada superior.



# UDP – USER DATAGRAM PROTOCOL

---

Mais simples e rápido

---

Não orientado a conexão (não se comunica com o destino antes de enviar)

---

Não confiável (não confirma a entrega de nada)

---

Tratamento de erros é feito por outros protocolos

---

Não faz controle de fluxo

---

Usado por outros protocolos, como: TFTP, SNMP, DHCP e DNS

- Pensando nessas características do protocolo UDP, usá-lo seria besteira certo?
- Mas essa questão de não negociar a comunicação antes do envio, e não confirmar entregas, agrega algo essencial ao protocolo: VELOCIDADE!



- O UDP não é um protocolo para serviços que precisam de confiabilidade!
- A vantagem do UDP é seu uso para serviços cuja velocidade é fundamental e a perda mínima de dados não muito desvantajosa, como em jogos online, por exemplo.
  - É normal alguns bytes se perderem na comunicação, mas que é sempre importante que a aplicação continue rodando com rapidez (sem se importar tanto com as perdas e falhas), para que não ocorra o famigerado lag.

- O UDP não realiza controle de fluxo, controle de erros ou retransmissão após a recepção de um segmento incorreto.
- Tudo isso cabe aos processos do usuário.
- O que ele faz é fornecer uma interface para o protocolo IP com o recurso adicional de multiplexação de vários processos que utilizam as portas.

- Uma área na qual o UDP é especialmente útil é a de situações cliente/servidor.
- Com frequência, o cliente envia uma pequena solicitação ao servidor e espera uma pequena resposta de volta.
- Se a solicitação ou a resposta se perder, o cliente simplesmente chegará ao timeout e tentará de novo.
- Não só o código é simples, mas é necessário um número menor de mensagens (uma em cada sentido) do que no TCP.

| UDP  | TCP   |
|--|---|
| Serviço sem conexão; nenhuma sessão é estabelecida entre os hosts.   | Serviço orientado por conexão; uma sessão é estabelecida entre os hosts.                                    |
| UDP não garante ou confirma a entrega ou seqüência os dados.   | TCP garante a entrega através do uso de confirmações e entrega seqüenciada dos dados.                       |
| Os programas que usam UDP são responsáveis por oferecer a confiabilidade necessária ao transporte de dados.              | Os programas que usam TCP têm garantia de transporte confiável de dados.                                    |
| UDP é rápido, necessita de baixa sobrecarga e pode oferecer suporte à comunicação ponto a ponto e ponto a vários pontos. | TCP é mais lento, necessita de maior sobrecarga e pode oferecer suporte apenas à comunicação ponto a ponto. |

Fonte: [http://juliobattisti.com.br/artigos/windows/tcpip\\_p11.asp](http://juliobattisti.com.br/artigos/windows/tcpip_p11.asp)



# PORTAS DE COMUNICAÇÃO



As portas de comunicação são usadas pelo TCP e pelo UDP para passar informações às camadas superiores e servem para gerenciar as diversas comunicações dos hosts.



Elas permite que diversas sessões de comunicação ocorram simultaneamente em um mesmo host.



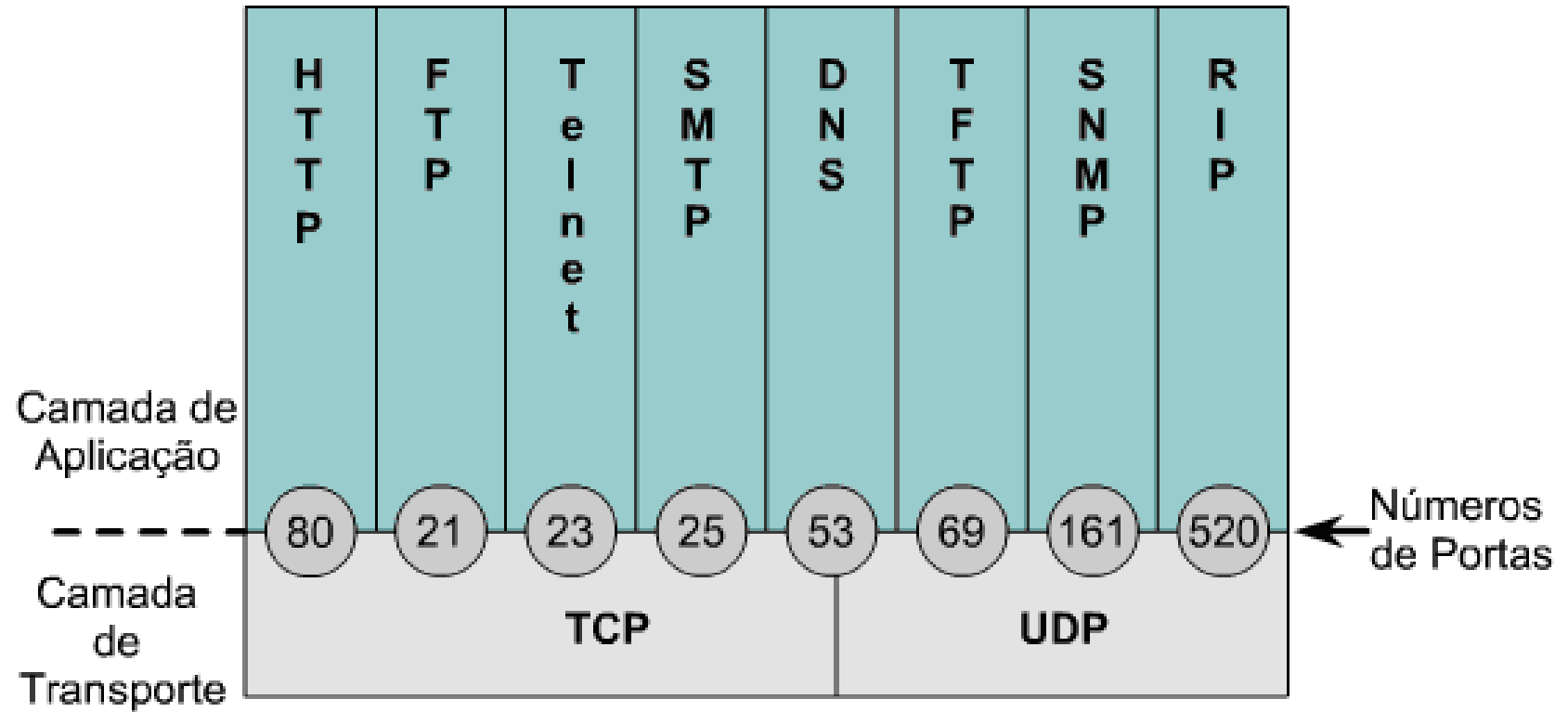
Uma porta direciona a requisição para um serviço específico da rede e são atribuídas pela IANA.

- O mecanismo de portas utilizado no TCP/IP permite que o computador suporte múltiplas sessões de comunicação com outros computadores ou softwares.
- Uma porta direciona a requisição para um serviço específico da rede.

- Os serviços mais comuns usam números de portas pré definidos:
  - Portas de 0 a 1023: reservados para aplicações de domínio público (Well Know Ports – Portas bem conhecidas).
    - 1 a 255 – portas públicas
    - 256 a 1023 – portas designadas a empresas
  - Portas de 1024 a 49151: reservados para aplicações comerciais registradas.
  - Portas de 49152 a 65535: portas dinâmicas ou privadas.



- Algumas portas conhecidas:
  - HTTP: porta 80
  - FTP: porta 20 e 21
  - SSH: porta 22
  - TELNET: porta 23
  - SMTP: porta 25
  - DNS: porta 53



| Porta | Protocolo | Descrição | Porta | Protocolo | Descrição |
|-------|-----------|-----------|-------|-----------|-----------|
| 20    | TCP       | FTP-data  | 21    | TCP       | FTP       |
| 23    | TCP       | Telnet    | 25    | TCP       | SMTP      |
| 53    | TCP/UDP   | DNS       | 69    | UD        | TFTP      |
| 80    | TCP       | HTTP      | 110   | TCP       | POP3      |
| 161   | UDP       | SNMP      | 443   | TCP/UDP   | HTTPS     |

- Um soquete é formado por um endereço IP associado a um número de porta (do protocolo correspondente ao serviço usado).
- Trabalhando em modo cliente-servidor, o servidor espera pelo pedido dos clientes ouvindo uma porta específica.



# VÍDEOS

PRODUCTION \_\_\_\_\_

DIRECTOR \_\_\_\_\_

CAMERA \_\_\_\_\_

SCENE \_\_\_\_\_

TAKE \_\_\_\_\_



# Vídeos

- Entenda os protocolos TCP e UDP - <https://www.youtube.com/watch?v=cy-ITN-ODM>
- Redes - Protocolos TCP e UDP - <https://www.youtube.com/watch?v=U-ExLyklla0>
- Protocolo TCP e UDP - <https://www.youtube.com/watch?v=5fiXrjL1o7Q>
- TCP vs UDP Comparison - <https://www.youtube.com/watch?v=uwoD5YsGACg>



# ATIVIDADE AVALIATIVA 3

Disponível no classroom



**Dúvidas?  
Não mais..**