Redes de Computadores

Camada de Transporte

Prof. Renê Pomilio de Oliveira

Slides baseados nas aulas da Profa. Dra. Kalinka Castelo Branco (ICMC/USP)
Prof. Dr. Anderson Chaves Carniel (UTFPR)

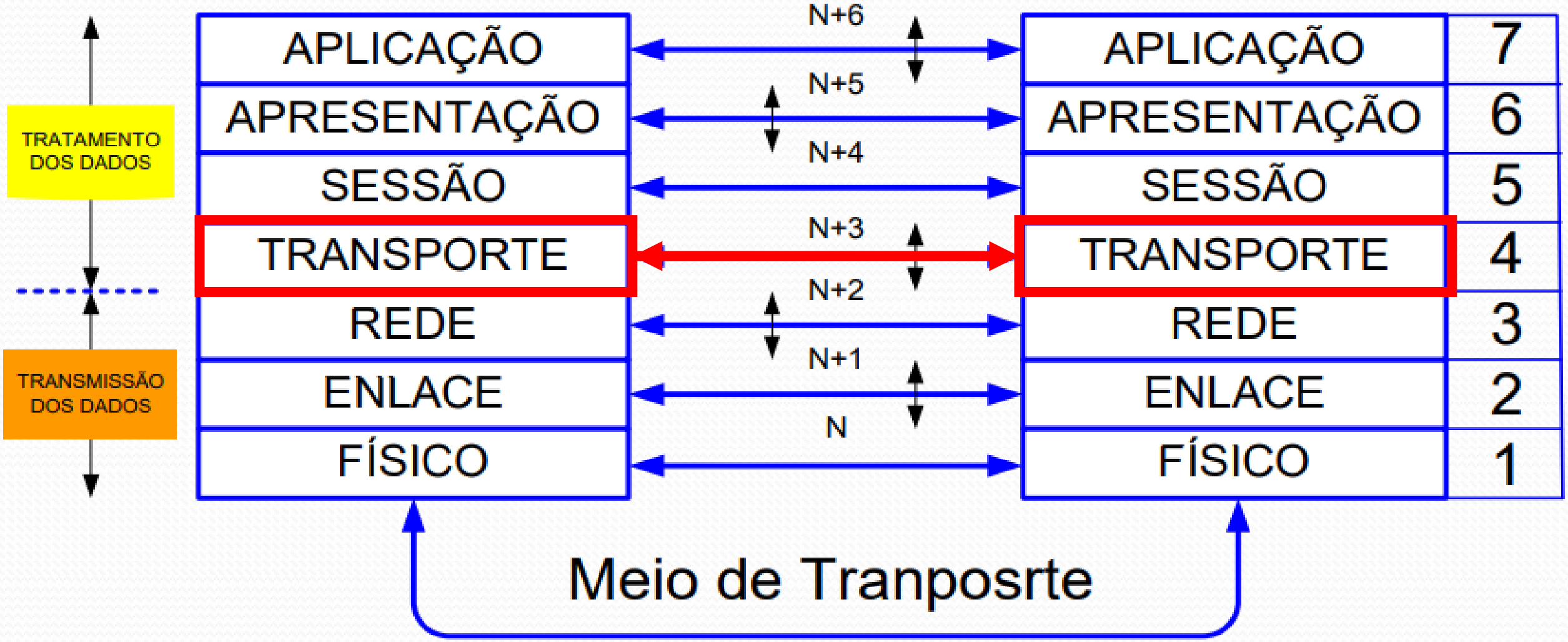
Câmpus Dois Vizinhos

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS DOIS VIZINHOS

MINISTÉRIO DA EDUCAÇÃO

BRASIL
GOVERNO FEDERAL

Entidades da Camada



Camada de Transporte - Funções

1) A função básica da camada de transporte é aceitar dados da camada de aplicação, dividi-los em unidades menores em caso de necessidade, passá-los para a camada de rede e garantir que todas essas unidades cheguem corretamente à outra extremidade.

2) Tudo deve ser feito com eficiência de forma que as camadas superiores fiquem isoladas das mudanças na tecnologia de hardware. (*Independente da aplicação se é desktop, mobile ou web*)

Camada de Transporte - Funções

3) A camada de transporte é uma camada fim a fim, que liga a origem ao destino.

4) Um programa da máquina de origem mantém uma conversa com um programa semelhante instalado na máquina de destino, utilizando cabeçalhos de mensagem e mensagens de controle.
   - ❖ Nessas camadas de transporte de diferentes hosts são trocadas TPDUs (Transport Protocol Data Units) chamados de segmentos. Um segmento é composto pelo cabeçalho da camada de transporte e os dados da camada de aplicação.

Protocolos da Camada de Transporte

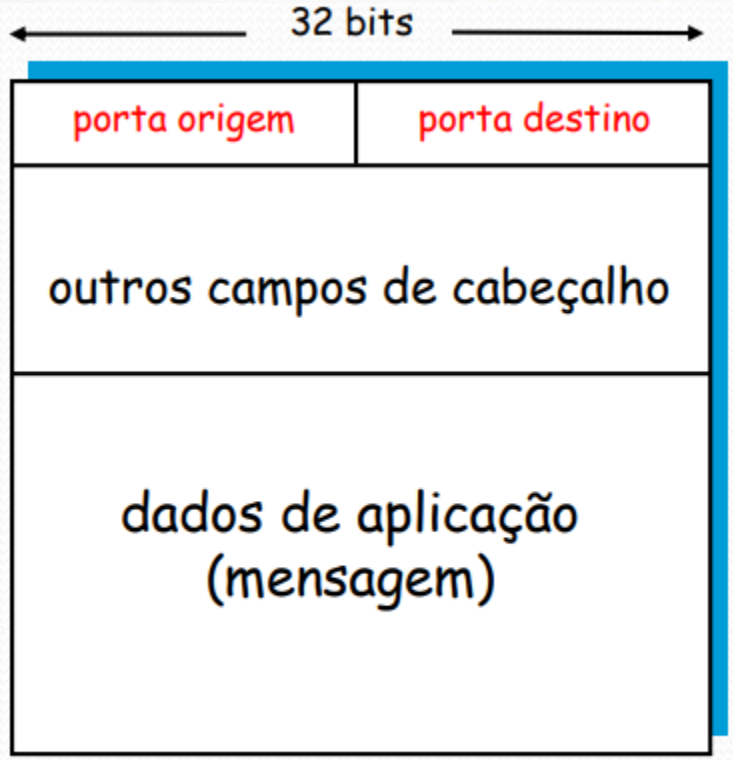
- Serviços de Transporte da Internet:
- **Confiável** (TCP)
  - congestão
  - controle de fluxo
  - orientado à conexão
- **não confiável** UDP
- serviços não disponíveis:
  - tempo-real
  - garantia de banda

Multiplexação de Aplicações

- **Segmento**: unidade de dados trocada entre entidades da camada de transporte
- **TPDU**: transport protocol data unit (unidade de dados do protocolo de transporte)
- **Demultiplexação**: entrega de segmentos recebidos aos processos de aplicação corretos

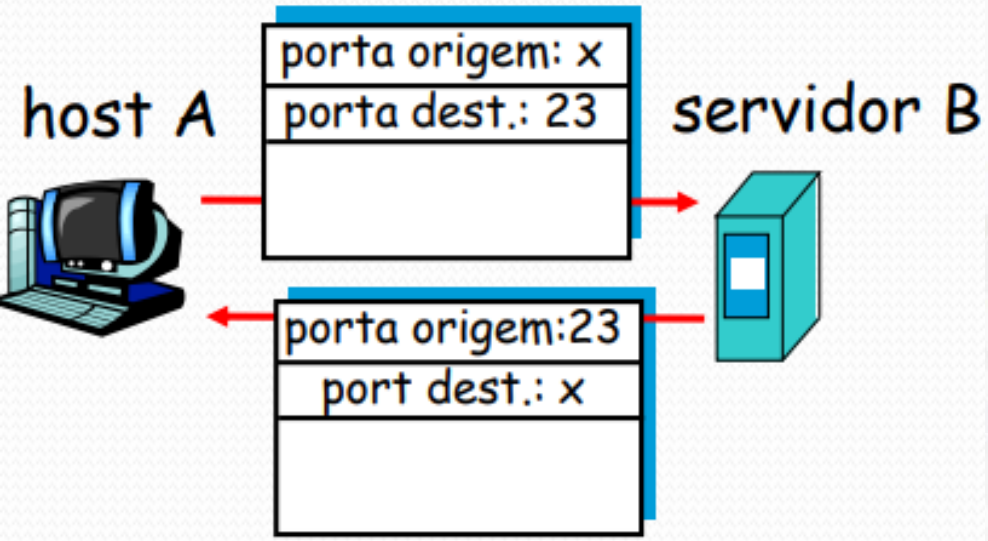
Multiplexação de Aplicações

- **Multiplexação:** reunir dados de múltiplos processo de aplicação, juntar cabeçalhos com informações para demultiplexação
- multiplexação/demultiplexação:
  - baseada nos número de porta do transmissor, número de porta do receptor e endereços IP
- números de porta origem e destino em cada segmento
- **Lembre-se**: portas com números bem-conhecidos são usadas para aplicações específicas

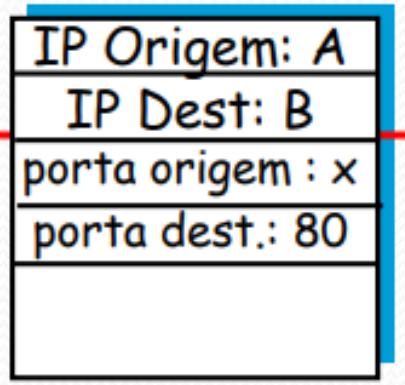


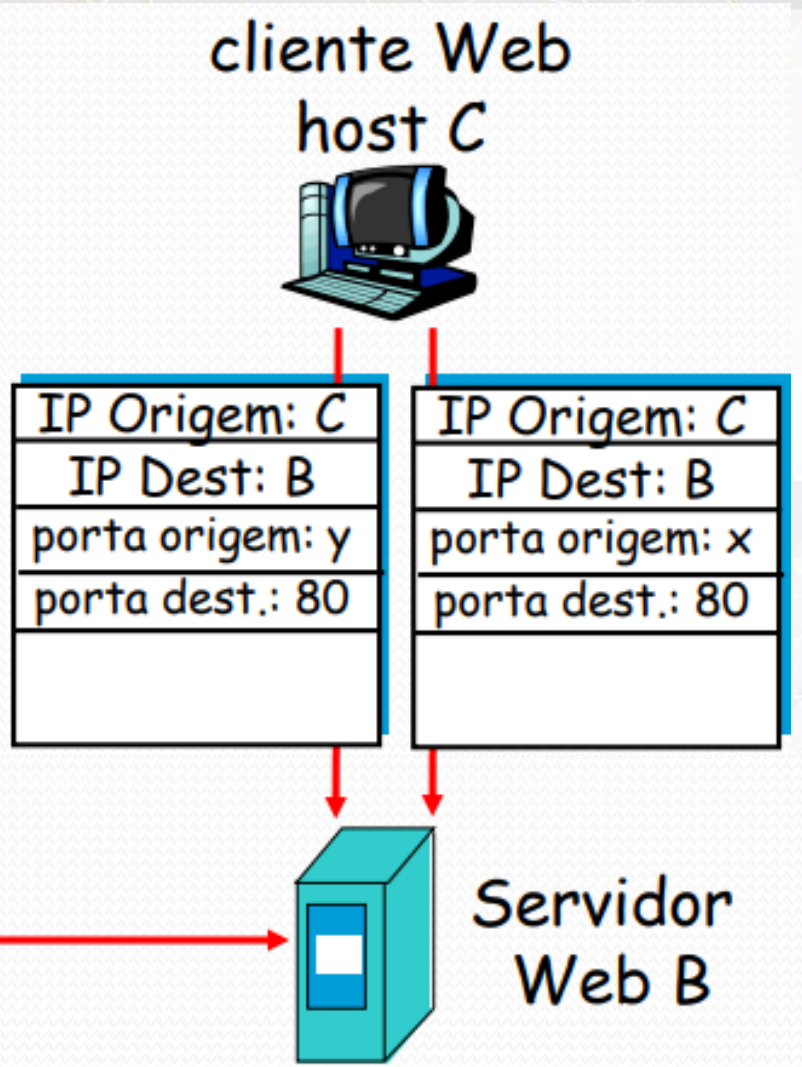
formato do segmento TCP/UDP

Multiplexação - exemplos



aplicação Telnet





aplicação: servidor Web

Multiplexação

- Muitos hosts são **multiprogramados**;
  - isso significa que muitas conexões estarão entrando e saindo de cada host.
- É responsabilidade da camada de transporte multiplexar todas as comunicações em um único canal determinando a qual conexão uma mensagem pertence.
- É responsabilidade desta camada estabelecer conexões, encerrá-las e controlá-las de forma que um host muito rápido não possa sobrecarregar um host muito lento (controle de fluxo). Em redes IP são utilizados dois protocolos para a implementação destas funções: o TCP e o UDP.

Endereçamento da Camada de Transporte

- Da mesma forma que em outras camadas, a camada de transporte também possui um endereçamento.
- Quando um processo de aplicação deseja estabelecer uma conexão com um processo de aplicação remoto, é necessário especificar a aplicação com a qual ele irá se conectar.
  - O método utilizado é definir os endereços de transporte que os processos podem ouvir para receber solicitações de conexão.

Endereçamento da Camada de Transporte

- Os processos utilizam os **TSAP** (*Transport Service Access Point* – Ponto de Acesso de Serviços de Transporte) para se intercomunicarem.
- Em redes IP, o TSAP é um número de 16 bits chamado de porta. O endereço da camada de transporte é um número de 48 bits, composto pela agregação do endereço IP do host e o número da porta.
- Os serviços da camada de transporte são obtidos por meio da comunicação entre os *sockets* do transmissor e do receptor.
  - *socket* é um ponto final de um fluxo de comunicação entre processos através de uma rede de computadores. **Exemplo no quadro PROTOCOLO://IP:PORTA**

Endereçamento da Camada de Transporte

- No mundo das redes, um *socket* é o fim da linha de uma comunicação entre hosts. E este fim da linha precisar ter todos os dados que o identificam, para que seu "endereço" não seja confundido com o de outro host.
- Muitos acham que o IP é uma "chave primária – *primary key*" no mundo da comunicação via TCP/IP (aqui falamos de camada de rede [IP] e camada de transporte [TCP]), e não é o caso (para entender melhor os protocolos e as camadas do modelo OSI
- A especificação do protocolo de rede e da porta são necessárias para definir o fim da linha de uma comunicação entre dois hosts

Endereçamento da Camada de Transporte

- Para uma melhor organização de serviços, algumas portas foram definidas pela **IANA** *(Internet Assigned Numbers Authority)* como “portas bem conhecidas” (well-known ports). Estas são as portas abaixo de **1024**, para aplicações não padronizadas são utilizadas portas acima deste valor.
- A IANA coordenação global da raiz (root) do DNS, endereçamento IP e outros recursos do protocolo da Internet para o melhoramento da performance

Endereçamento da Camada de Transporte

- Os *sockets* são diferentes para cada protocolo de transporte, desta forma mesmo que um socket TCP possua o mesmo número que um socket UDP, ambos são responsáveis por aplicações diferentes.
- Os sockets de origem e destino são responsáveis pela identificação única da comunicação. Desta forma é possível a implementação da função conhecida como multiplexação.
- A multiplexação possibilita que haja várias conexões partindo de um único host ou terminando em um mesmo servidor.

Endereçamento da Camada de Transporte

- A formação do socket de origem e destino se dá da seguinte forma:
  - Ao iniciar uma comunicação é especificado para a aplicação o endereço IP de destino e a porta de destino;
  - A porta de origem é atribuída dinamicamente pela camada de transporte. Ela geralmente é um número sequencial randômico acima de 1024;
  - O endereço IP de origem é atribuído pela camada 3.

UDP: User Datagram Protocol [RFC 768]

- protocolo de transporte da Internet “sem gorduras”
- serviço “best effort”, segmentos UDP podem ser:
  - perdidos
  - entregues fora de ordem para a aplicação

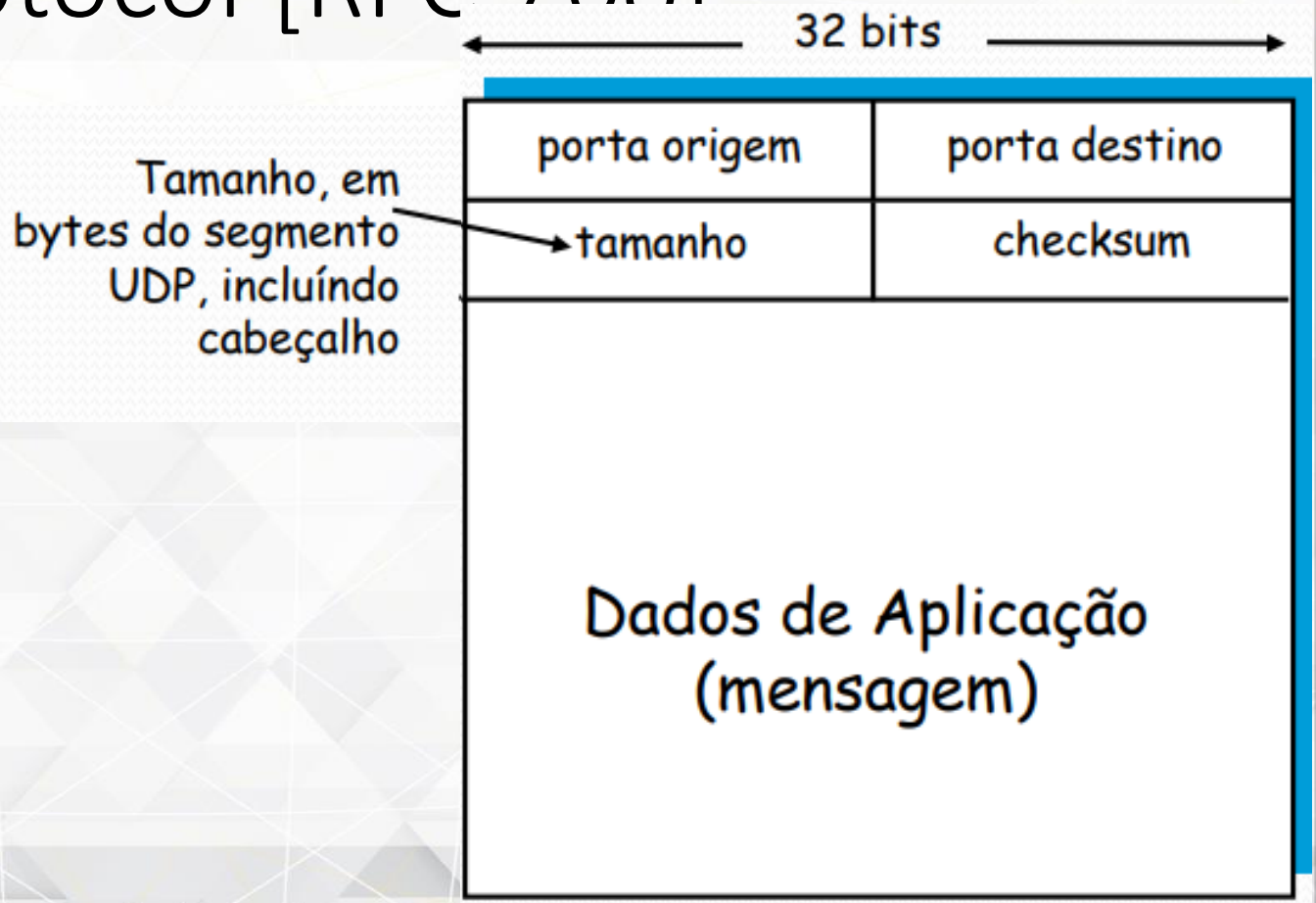
UDP: User Datagram Protocol [RFC 768]

Porque existe um UDP?

- não há estabelecimento de conexão (que pode resultar em atrasos)
- simples: não há estado de conexão nem no transmissor, nem no receptor
- cabeçalho de segmento reduzido
- não há controle de congestionamento: UDP pode enviar segmentos tão rápido quanto desejado

UDP: User Datagram Protocol [RFC 768]

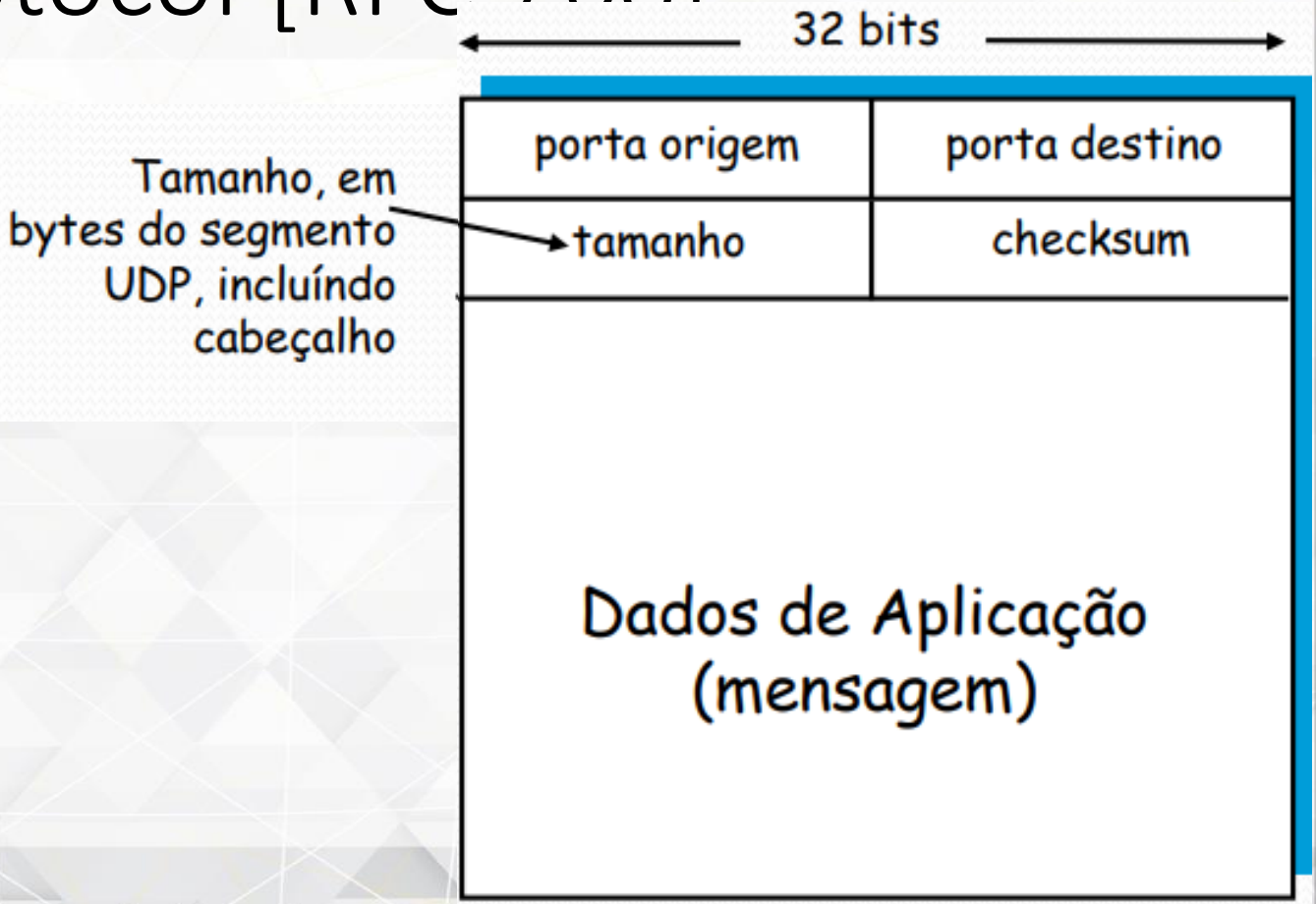
- muito usado por aplicações de multimídia contínua (Voz e vídeo):
  - tolerantes à perda
  - sensíveis à taxa
- outros usos do UDP
  - DNS
  - SNMP



formato do segmento UDP

UDP: User Datagram Protocol [RFC 768]

- Porta de Origem e Porta de Destino – Indicam os pares de porta que estão executando a comunicação;
- Comprimento – Indica o comprimento de todo o datagrama isto é, cabeçalho e dados;
- Checksum – Verificação de integridade do datagrama;



formato do segmento UDP

UDP: User Datagram Protocol UDP checksum

- **Objetivo**: detectar “ erros ” (ex.,bits trocados) no segmento transmitido

- Transmissor:
  - trata o conteúdo do segmento como seqüência de inteiros de 16 bits
  - checksum: soma (complemento de 1 da soma) do conteúdo do segmento
  - transmissor coloca o valor do checksum no campo de checksum do UDP

- Receptor:
  - computa o checksum do segmento recebido
  - verifica se o checksum calculado é igual ao valor do campo checksum:
    - ✓ NÃO - erro detectado
    - ✓ SIM - não há erros

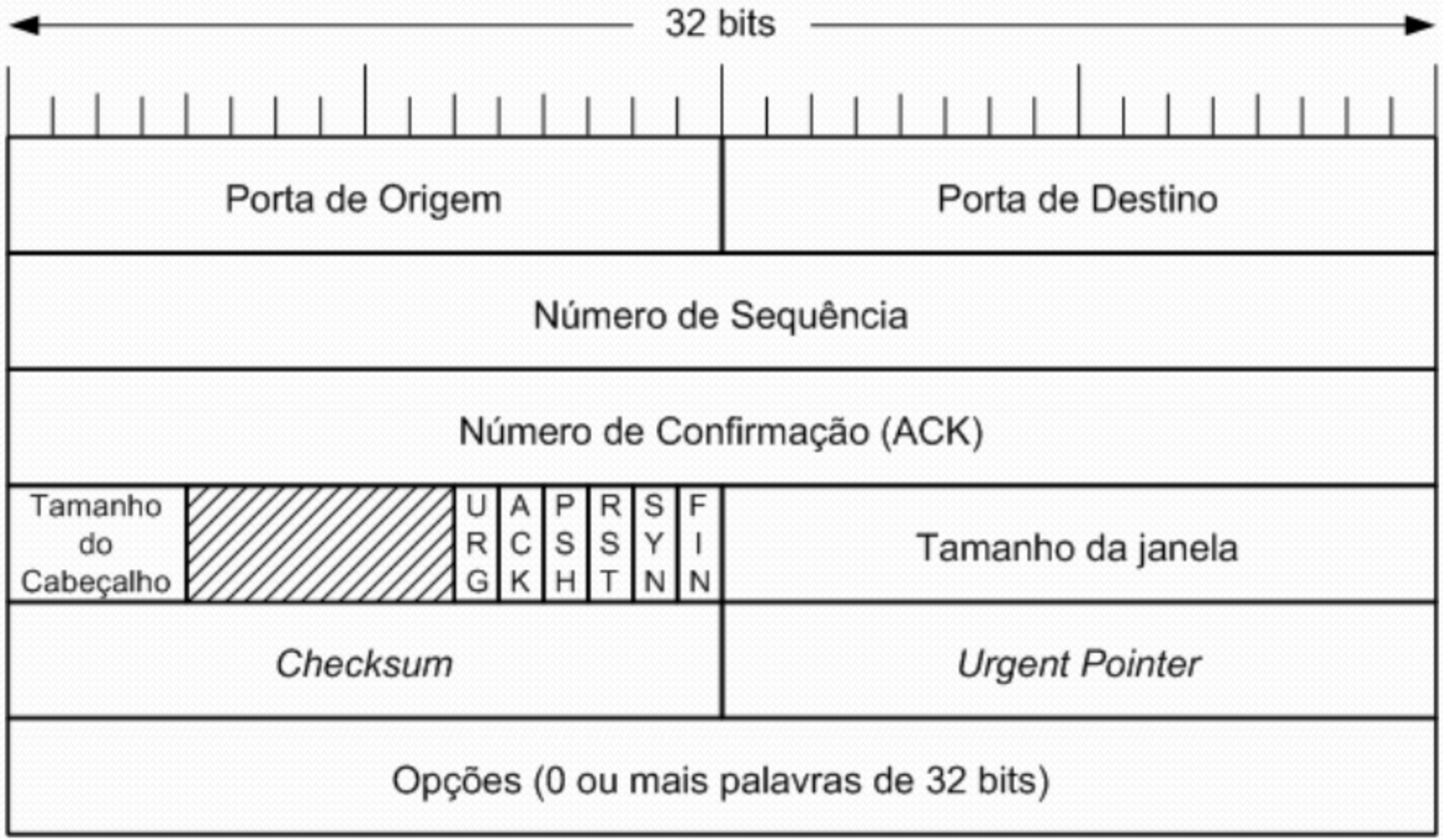
TCP – Transport Control Protocol

- O TCP (Protocolo de Controle de Transmissão) foi projetado para oferecer um fluxo de bytes fim a fim confiável em uma inter-rede não confiável.
- Ele é um protocolo orientado a conexão que permite a entrega sem erros de um fluxo de bytes originado de uma determinada máquina para qualquer computados da rede.
- Fragmenta o fluxo de entrada em mensagens e passa cada uma delas para a camada de redes. No destino, o processo TCP remonta as mensagens recebidas gerando o fluxo de saída. O TCP foi projetado para se adaptar dinamicamente às propriedades da camada de rede e ser robusto diante dos muitos tipos de falhas que podem ocorrer

TCP – Transport Control Protocol

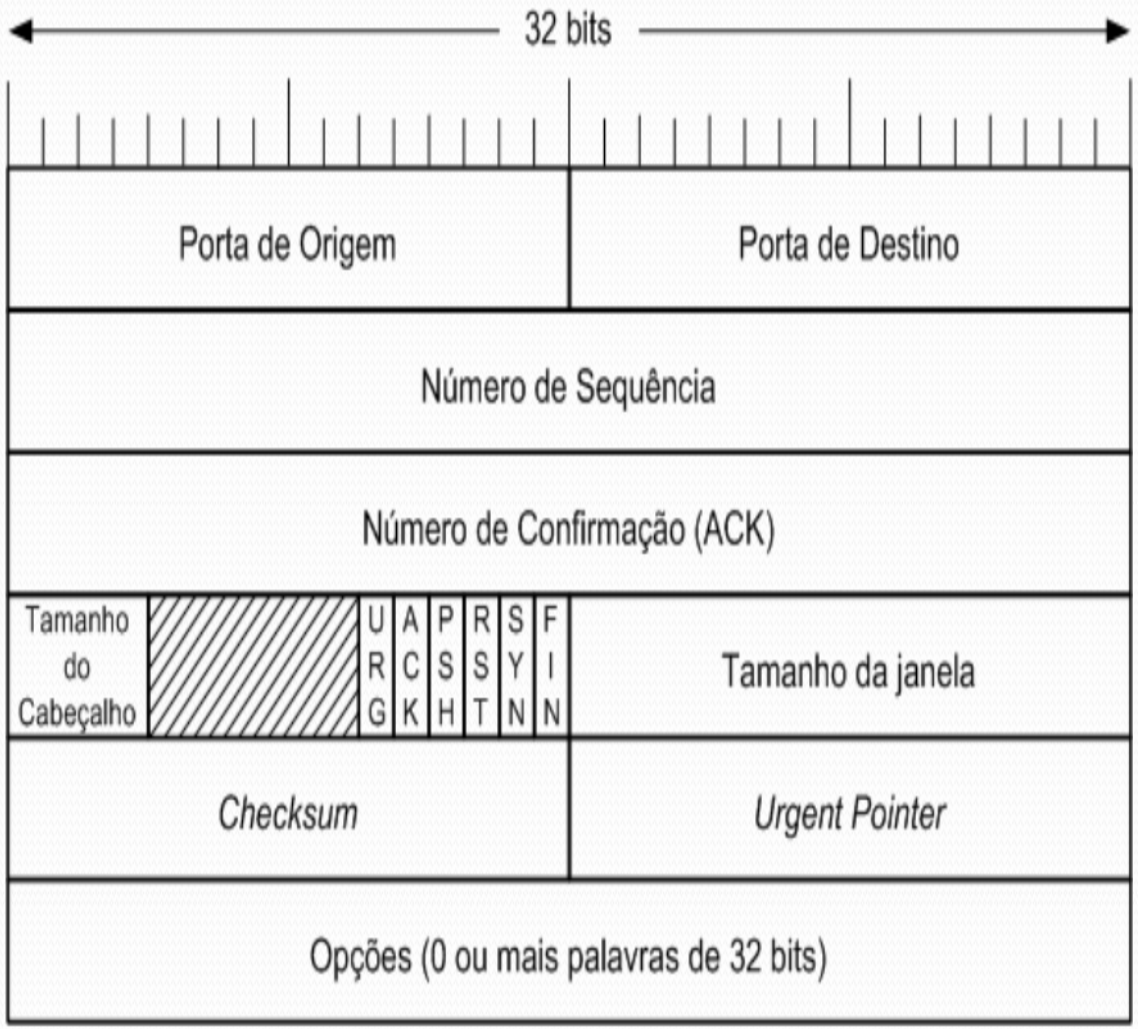
- O TCP foi formalmente definido na RFC 793, posteriormente alguns erros foram corrigidos e o TCP foi definido na RFC 1122.
- O TCP define um cabeçalho para suas mensagens composto dos seguintes campos:

TCP – Transport Control Protocol - *proof*



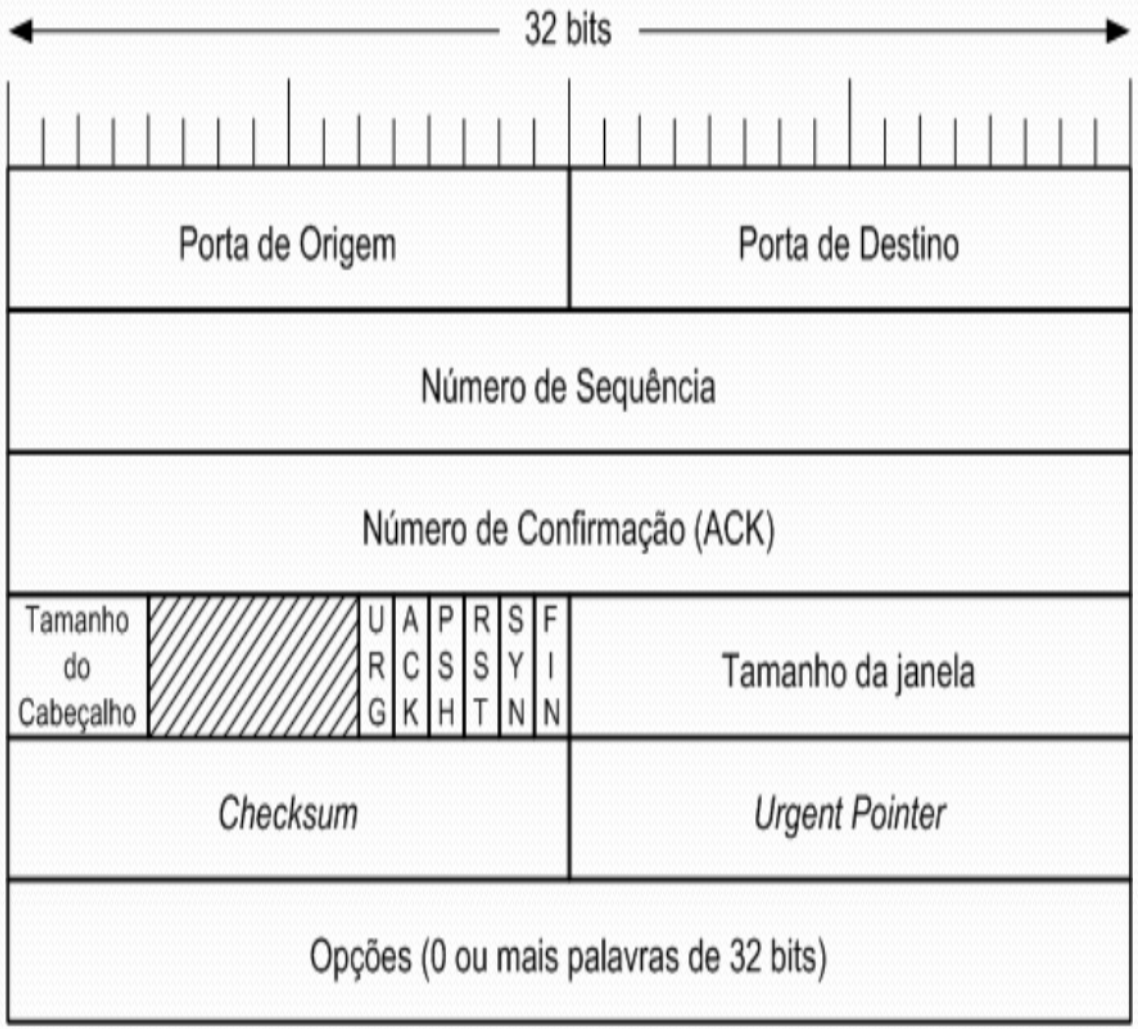
TCP – Transport Control Protocol - *proof*

- **Porta de Origem e Porta de Destino** – identifica os pontos terminais locais da conexão;
- **Número de Sequência** – Identifica o fragmento dentro de todo o fluxo gerado;
- **Número de Confirmação** – Indica qual o próximo byte esperado;
- **Tamanho do Cabeçalho** – Informa quantas palavras de 32 bits compõem o cabeçalho TCP;



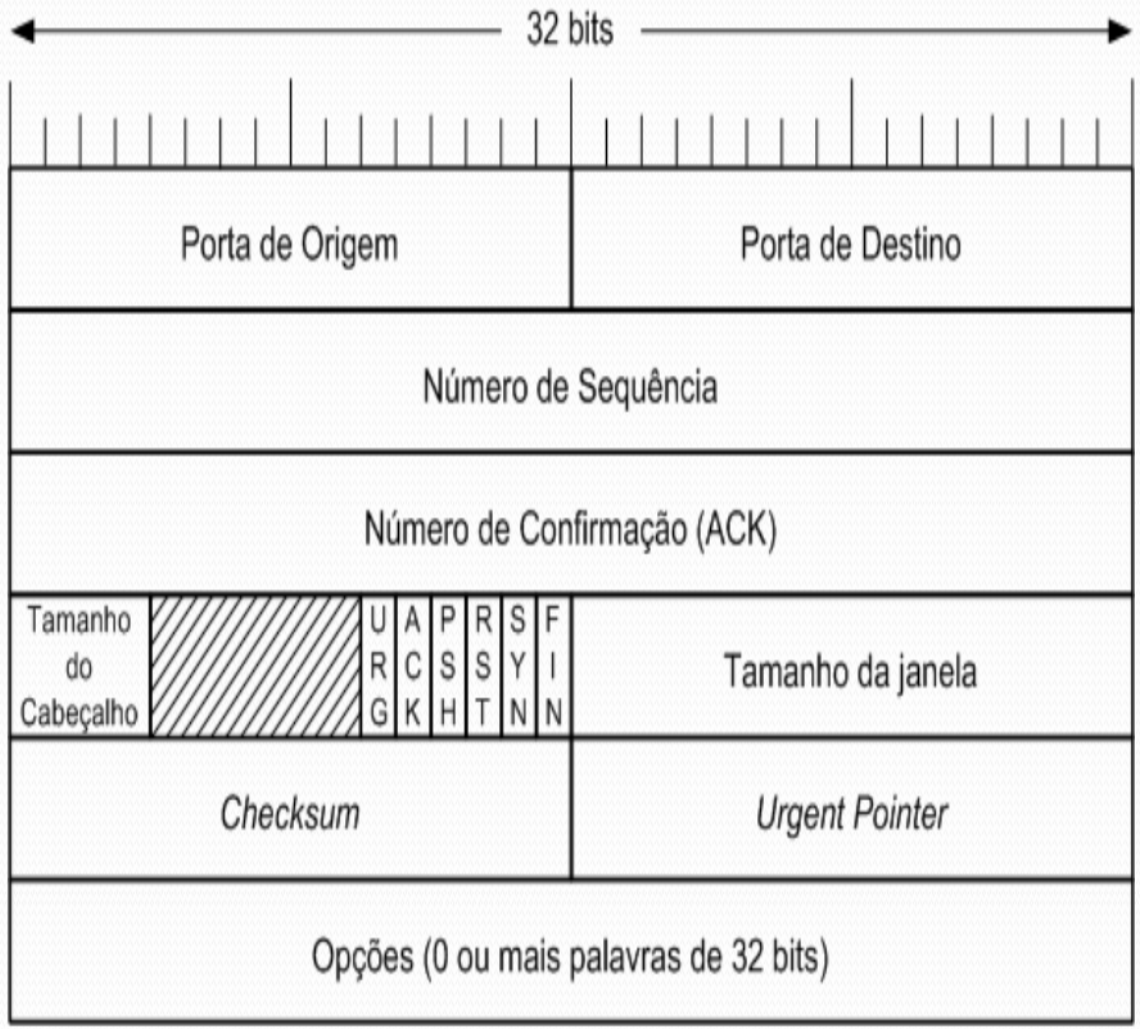
TCP – Transport Control Protocol - *proof*

- **URG** – Indica a utilização do *urgent pointer*;
- **ACK** – É utilizado para indicar que este segmento é um ACK e que o campo Número de Confirmação deve ser interpretado;
- **PSH** – Indica que este segmento não deve ser enfileirado como todos os outros, mas sim posto à frente na fila;



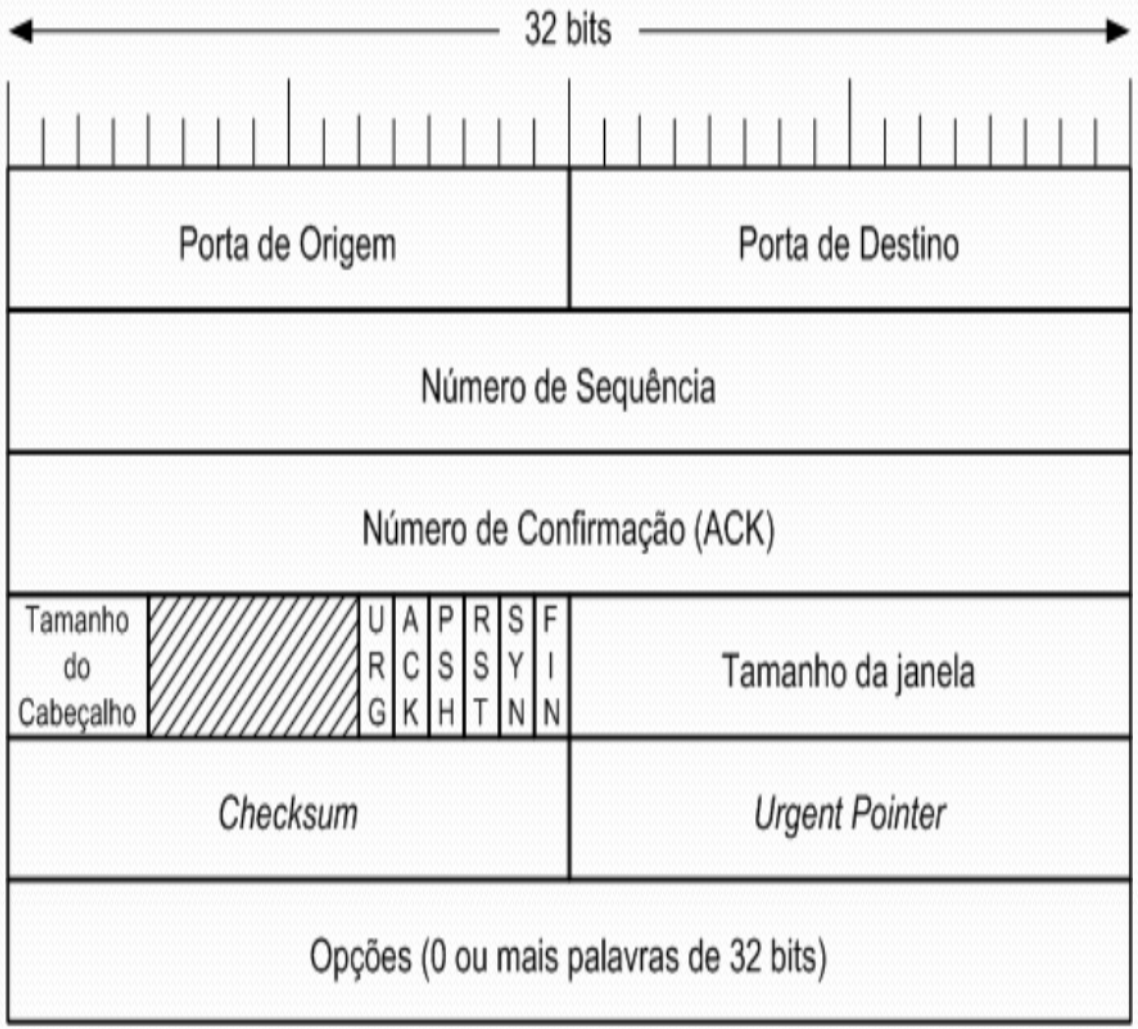
TCP – Transport Control Protocol - *proof*

- **RST** – É utilizado para reiniciar uma conexão que tenha ficado confusa devido a falhas no host ou por qualquer outra razão;
- **SYN** – Este bit é utilizado para indicar um pedido de conexão e a confirmação da conexão;
- **FIN** – Utilizado para indicar que o emissor não possui mais dados para enviar e deseja finalizar a conexão;



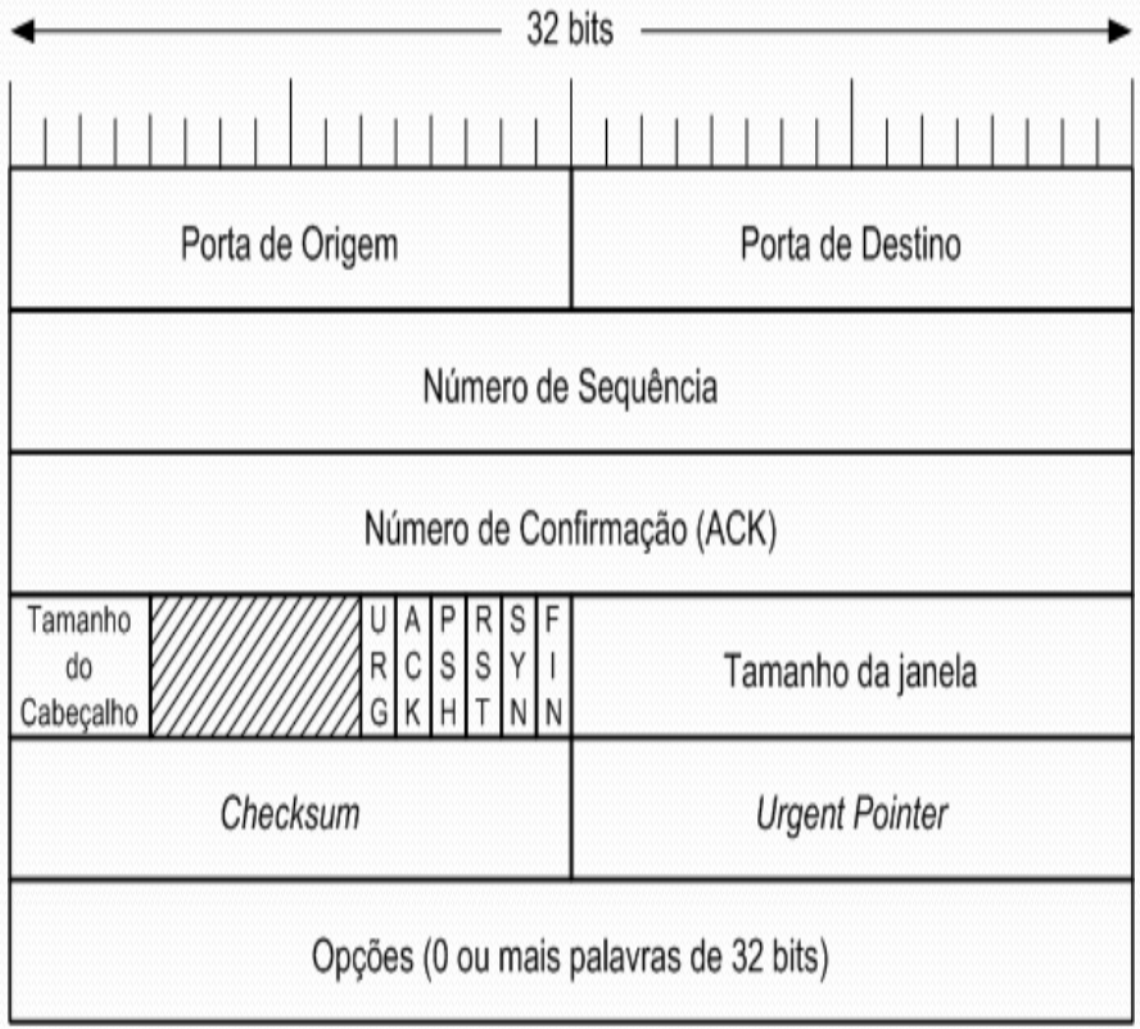
TCP – Transport Control Protocol - *proof*

- **Tamanho da Janela** – Indica quantos bytes podem ser enviados a partir do byte confirmado. Este campo é utilizado no controle de fluxo do TCP;
- **Checksum** – Indicador de integridade do segmento;



TCP – Transport Control Protocol - *proof*

- **Urgent Pointer** – Indica um deslocamento de bytes a partir do número de sequência atual em que os dados urgentes devem ser encontrados;
- **Opções** – Projetado para que o TCP possa oferecer recursos extras que não foram previstos em seu protocolo;



Estabelecimento de Conexões

- O estabelecimento de uma conexão TCP ocorre antes que qualquer outro recurso TCP possa começar seu trabalho.
- O estabelecimento da conexão se fundamenta no processo de inicialização dos campos referentes à sequência, aos ACKs e na troca dos números de sockets usados.
- As conexões são estabelecidas no TCP por meio do *three way handshake* (handshake de três vias).

Estabelecimento de Conexões

- O estabelecimento da conexão é feito usando dois bits na cabeçalho TCP: SYN e ACK.
- Um segmento que possua a flag SYN ativa sinaliza uma requisição de sincronia do número de seqüência. Essa sincronização é necessária em ambos os sentidos, pois origem e destino utilizam números de seqüência distintos.
- Cada pedido de conexão é seguido de uma confirmação utilizando o bit ACK.
- O segundo segmento do *three way handshake* exerce as duas funções ao mesmo tempo: Confirma a sincronização do servidor com o cliente e requisita a sincronização do cliente com o servidor.

Estabelecimento de Conexões

- Basicamente o *Three way handshake* 'simula' um acordo.
  - ❖ O Cliente pergunta pro servidor: "Você está ai?" e;
  - ➢ servidor responde: "Sim estou...".
  - ❖ Depois o servidor pergunta: "Você está ai?" e;
  - ➢ o cliente responde: "Sim estou...".
  - Mas espere! Como tem 4 sentenças em apenas 3 trocas de mensagens?? Simples, a segunda mensagem contém uma resposta e uma pergunta.

Estabelecimento de Conexões

- **Cliente**: Servidor, você está ai?? (SYN)

  Servidor: Sim estou... (ACK) E você, está ai? (SYN)

  Cliente: Sim, estou... (ACK)
- O servidor precisa perguntar se o cliente está lá pela simples necessidade de sincronização do número de sequência. O número de sequência é utilizado para garantir a entrega de todas as mensagens.

Estabelecimento de Conexões

- **Cliente**: Alô servidor, mensagem 2000 (Número de sequência do cliente), o senhor está disponível (SYN)?

- ✓ **Servidor**: Sim, claro cliente! Mensagem 1450 (Numero de sequência do servidor) Prossiga com a mensagem 2001 (ACK=2001).

- **Cliente**: Ok, servidor! Mensagem 2001 (numero de sequência), confirmando número da próxima mensagem: 1451!

Estabelecimento de Conexões

- Dessa forma eles trocam o número de sequência, que tem como função “enumerar” as mensagens de cada um.
- Por exemplo, se a última mensagem foi a **2001** e a mensagem que chegou pro servidor foi a **2003**, ele tem completa certeza que uma mensagem (**2002**) se perdeu no caminho! Então basta somente solicitar uma retransmissão.
- O número de sequência nem sempre é incrementado por 1, ele pode ser incrementado com base no número de bytes enviados pela origem. O ACK tem como objetivo solicitar a continuidade das mensagens. Pode-se interpretar um ACK=210 como sendo: “Pronto, recebi até a 209, pode mandar a 210”.

Recuperação de Erros

- O TCP proporciona uma transferência confiável de dados, o que também é chamado de confiabilidade ou recuperação de erros.
- Para conseguir a confiabilidade o TCP enumera os bytes de dados usando os campos referentes à sequência e aos ACKs no cabeçalho TCP.
- O TCP alcança a confiabilidade em ambas as direções, usando um campo referente ao número de sequência de uma direção, combinado com o campo referente ao ACK na direção oposta.

Portas de Serviços

- A numeração de portas utiliza 16 bits. Logo (2^16) existem ao todo **65536** portas a serem utilizadas.
- Isso para cada protocolo da camadas de transporte. Logo 65536 portas para o TCP, e 65536 portas para o UDP.
- Conhecer essas portas é fundamental para operar um *Firewall* de forma satisfatória.

Portas de Serviços

Com tantas porta como saber todas elas???

- Não precisa conhecer todas, uma vez que a maior parte delas não são especificadas.
- Apenas as primeiras 1024 são especificadas.
- Para um administrador de rede é imprescindível saber pelo menos as portas dos serviços básicos de Rede: telnet, SSH, FTP, SMTP, POP, HTTP, HTTPS... Não são muitas, mas antes de ver isso, vamos entender que controla essas portas.
- O uso das portas de 1 a 1024 é padronizada pela IANA (Internet Assigned Numbers Authority).
  - Essa entidade é responsável por alocar portas para determinados serviços. Essas portas são chamadas de well-known ports.

Portas de Serviços

- Lista de algumas principais portas TCP:

Porta	Descrição	Protocolos	Oficial?
1	TCP Port Service Multiplexer	TCP/UDP	Oficial
21	FTP – control (command)	TCP	Oficial
22	Secure Shell (SSH)–used for secure logins, file transfer...	TCP/UDP	Oficial
25	Simple Mail Transfer Protocol (SMTP)–used for e-mail r...	TCP	Oficial
53	Domain Name System (DNS)	TCP/UDP	Oficial
110	Post Office Protocol 3 (POP3)	TCP	Oficial
111	ONC RPC (SunRPC)	TCP/UDP	Oficial
143	Internet Message Access Protocol (IMAP)–used for retr...	TCP/UDP	Oficial
443	HTTPS (Hypertext Transfer Protocol over SSL/TLS)	TCP	Oficial
465	Cisco protocol	TCP	Não Oficial
465	SMTP over SSL	TCP	Não Oficial

Serviço	Porta
http	80
ftp	20 e 21
telnet	23
dhcp	67
dns	53
snmp	161 e 162
nfs	2049
smb	137, 138, 139 e 445
smtp	25
pop3	110

http://www.portalchapeco.com.br/jackson/portas.htm

https://www.hardware.com.br/livros/redes/portas-tcp-udp.html

Portas de Serviços – FTP

- **FTP 20-21:** é um dos protocolos de transferência de arquivos mais antigos e ainda assim um dos mais usados. O ponto fraco do FTP é a questão da segurança: todas as informações, incluindo as senhas trafegam em texto puro e podem ser capturadas por qualquer um que tenha acesso à transmissão.

Portas de Serviços – SSH

- **SSH 22:** é o canivete suíço da administração remota em servidores Linux. Inicialmente o SSH permitia executar apenas comandos de texto remotamente; depois passou a permitir executar também aplicativos gráficos e, em seguida, ganhou também um módulo para transferência de arquivos, o **SFTP**. A vantagem do SSH sobre o Telnet e o FTP é que tudo é feito através de um canal encriptado, com uma excelente segurança.

Portas de Serviços – Telnet

- **Telnet 23:** é provavelmente o protocolo de acesso remoto mais antigo. A primeira demonstração foi feita em 1969, com o acesso de um servidor Unix remoto (ainda na fase inicial de implantação da **Arpanet**), muito antes de ser inventado o padrão Ethernet e antes mesmo da primeira versão do TCP/IP.
- Uma curiosidade, é que o sistema usado pelo Telnet para a transmissão de comandos é usado como base para diversos outros protocolos, como o SMTP e o HTTP. De fato, você pode usar um cliente Telnet para mandar um e-mail (se souber usar os comandos corretos), ou mesmo acessar um servidor web, desde que consiga simular uma conexão HTTP válida, como faria um navegador.

Portas de Serviços – SMTP

- **SMTP 25:** é o protocolo padrão para o envio de e-mails. Ele é usado tanto para o envio da mensagem original, do seu micro até o servidor SMTP do provedor, quanto para transferir a mensagem para outros servidores, até que ela chegue ao servidor destino. Tradicionalmente, o Sendmail é o servidor de e-mails mais usado, mas, devido aos problemas de segurança, ele vem perdendo espaço para o Qmail e o Postfix.

Portas de Serviços – DNS

- **DNS 53:** Os servidores DNS são contatados pelos clientes através da porta 53, UDP. Eles são responsáveis por converter nomes de domínios como “guiadohardware.net” nos endereços IP dos servidores.
- Existem no mundo 13 servidores DNS principais, chamados “root servers”. Cada um deles armazena uma cópia completa de toda a base de endereços. Estes servidores estão instalados em países diferentes e ligados a links independentes. A maior parte deles roda o Bind, mas pelo menos um deles roda um servidor diferente, de forma que, mesmo que uma brecha grave de segurança seja descoberta e seja usada em um cyberataque, pelo menos um dos servidores continue no ar, mantendo a Internet operacional.

Portas de Serviços – HTTP

- **HTTP 80:** O HTTP é o principal protocolo da Internet, usado para acesso às paginas web. Embora a porta 80 seja a porta padrão dos servidores web, é possível configurar um servidor web para usar qualquer outra porta TCP. Neste caso, você precisa especificar a porta ao acessar o site, como em: http://200.19.73.47:80.

Portas de Serviços – POP3

- **POP3 110:** Servidores de e-mail, como o Postfix, armazenam os e-mails recebidos em uma pasta local. Se você tiver acesso ao servidor via SSH, pode ler estes e-mails localmente, usando Mutt (no Linux). Entretanto, para transferir os e-mails para sua máquina, é necessário um servidor adicional. É aí que entra o protocolo POP3, representado no Linux pelo courier-pop e outros servidores.

Portas de Serviços – HTTPS

- **HTTPS 443:** O HTTPS permite transmitir dados de forma segura, encriptados usando o SSL. Ele é usado por bancos e todo tipo de site de comércio eletrônico ou que armazene informações confidenciais

➢ **SSL** significa **Secure Sockets Layer**, um tipo de segurança digital que permite a comunicação criptografada entre um site e um navegador. Atualmente a tecnologia se encontra **depreciada** e está sendo completamente substituída pelo TLS.

➢ **TLS** é uma sigla que representa **Transport Layer Security** e certifica a proteção de dados de maneira semelhante ao SSL. Como o SSL não está mais de fato em uso, esse é o termo correto que deveria ser utilizado.