

INSTITUTO FEDERAL
Rio Grande do Sul

Redes de Computadores

Luis Augusto Dias Knob
luis.knob@sertao.ifrs.edu.br

Camada de transporte

- A camada de transporte possibilita a **comunicação entre os programas de aplicação**, de uma forma **fim-a-fim**, ou seja, o transporte garante que os dados transmitidos/recebidos estarão corretos.
- Os protocolos de transporte típicos da internet são o **TCP** e o **UDP**.

Camada de transporte

- é a camada responsável pela transferência:
 - eficiente e confiável
- * independente do tipo, topologia ou configuração das redes físicas existentes entre elas, garantindo ainda que os dados cheguem sem erros e na seqüência correta.

Os principais serviços

- Controle de conexão
 - a camada de transporte possui protocolos que oferecem **serviços orientados à conexão e não orientados à conexão.**
 - os serviços **orientados à conexão** primeiramente **estabelece uma comunicação entre usuários finais** e só depois começa a transmissão.

Os principais serviços

- Fragmentação

- Exemplo:

- imagine um e-mail (SMTP) enviado com um texto simples como:

- “Prezada Radegondes, vamos sair hoje? Beijos, Jonielshon”.

- Agora pense num e-mail com todas as fotos de um passeio ou com três arquivos de MP3.

Nota-se que uma mensagem SMTP pode conter de alguns bytes até dezenas de megabytes. Logo **é necessário fragmentar tais informações em pacotes de tamanhos menores** para serem encapsulados pela camada de rede.

Os principais serviços

- Endereçamento

- Sabemos que a identificação de um host na Internet é única.
- ... mas imagine duas situações:
 - um servidor web (HTTP) também é servidor FTP;
 - você abre duas janelas do navegador acessando dois links diferentes de um mesmo site.
- como identificar dois serviços em um mesmo host?
 - para resolver estas perguntas a camada de transporte oferece à camada de aplicação a função de endereçamento, onde os serviços são identificados pela sua porta e uma conexão entre sua estação e outro host é feita através de um socket (IP+porta)

Os principais serviços

- **Endereçamento**

- As portas de origem e destino devem ser identificadas no estabelecimento de conexões TCP ou no envio de datagramas usando UDP
- As portas são identificadas no cabeçalho das mensagens por um campo de 16 bits
 - Inteiros de 0 a 65535
 - Distintas para cada protocolo

Os principais serviços

- **Endereçamento**

- Portas de 0 a 1023 são reservadas para uso de serviços padrão da rede [RFC 1700]
- Exemplos

Protocolo	Porta	Serviço
TCP	21	FTP
TCP	22	SSH
TCP	80	HTTP
UDP	53	DNS

Os principais serviços

- **Confiabilidade**

- **controle de fluxo;**

- Através de um sistema de **bufferização** denominada *janela deslizante* (ou não), o TCP envia uma série de pacotes sem aguardar o reconhecimento de cada um deles. Na medida em que recebe o reconhecimento de cada bloco enviado, atualiza o buffer (caso reconhecimento positivo) ou reenvia (caso reconhecimento negativo ou não reconhecimento após um *timeout*);

Os principais serviços

- **Confiabilidade**
 - controle de erros;
 - Além da numeração dos segmentos transmitidos, vai junto com o header uma soma verificadora dos dados transmitidos (*checksum*). Assim o destinatário verifica a soma efetuando o cálculo dos dados recebidos, a fim de evitar perdas, duplicação ou entrega fora de ordem;
 - controle de congestionamento e Qualidade de Serviço

UDP

- **O protocolo UDP é bastante simples**
 - Orientado a datagrama
 - Não orientado à conexão
 - Não executa controle de fluxo, controle de erro e sequenciamento
 - Não tem reconhecimento dos datagramas (ACK/NACK)
- **Devido a sua simplicidade é considerado não confiável**

Cabeçalho UDP

Porta Origem	Porta Destino
Tamanho	Checksum
Dados	

Onde,

- Porta Origem e Porta Destino identificam o processo de aplicação que está enviando dados e o processo de aplicação que irá receber os dados.
- Tamanho é representa o tamanho total do frame UDP
- Checksum é calculado usando o header UDP e também a área de dados, e destina-se a verificação de erros de transmissão.

Checksum UDP

- O Checksum no UDP é opcional
 - Campo de checksum = 0, não executa verificação
 - Campo de checksum \neq 0, executa verificação
- O cálculo do checksum utiliza o header, os dados e também o Pseudo-Header
 - Este pseudo-header é utilizado para verificação adicional e confirmação de que o datagrama chegou ao destino correto

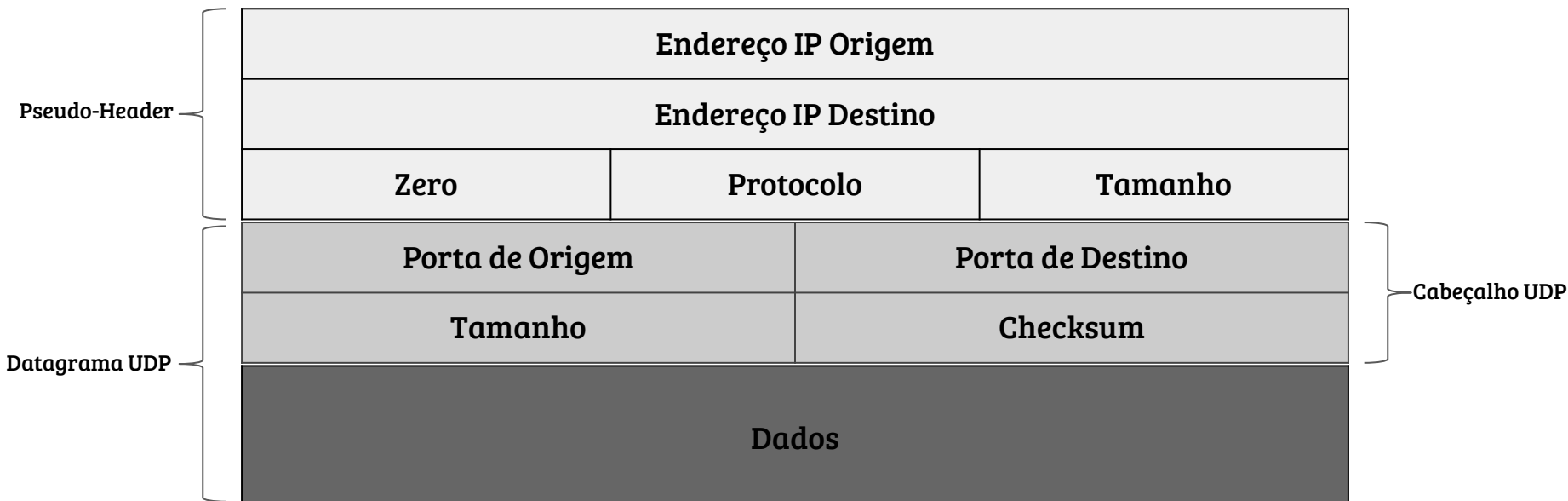
Pseudo-Header

Endereço IP Origem		
Endereço IP Destino		
Zero	Protocolo	Tamanho

Onde,

- Endereço IP Origem e Endereço IP destino são do nível de rede (protocolo IP) utilizadas para a segunda validação do destino do datagrama.
- Zero é um campo com valor zero para complementar a estrutura do pseudo-header.
- Protocolo indica qual o protocolo de transporte (TCP ou UDP), pois o pseudo-header é utilizado para os dois protocolos.
- Tamanho indica o tamanho do frame de transporte (UDP ou TCP)

Ordem de Header para o Checksum do UDP



Atenção!

O Pseudo-Header não é transmitido junto com o datagrama UDP, ele é utilizado apenas para cálculo do Checksum.

Tamanho Máximo do Datagrama

- Teoricamente o tamanho máximo é de 64Kb
 - Porque no IPv4 o campo tamanho total é de 16 bits
 - Mas deve-se considerar que no IP estão sendo calculado
 - Tamanho do Header do IP (20 bytes)
 - Datagrama UDP (8 bytes)
 - Assim, o tamanho máximo é de **65507** bytes

TCP

- Oferece confiabilidade na transmissão
- Considerado o “sucesso da Internet”
- Orientado à conexão
- Confiável
- Full-duplex

Cabeçalho TCP

Porta de Origem			Porta de Destino		
Número de Sequência					
Acknowledgment					
Tamanho	Reservado	Flags	Janela Deslizantes		
Checksum			Urgent Pointer		
Opções					
Dados					

Cabeçalho TCP

Onde,

- **Porta Origem e Porta Destino** identificam o processo de aplicação que está enviando dados e o processo de aplicação que irá receber os dados.
- **Número de sequência** identifica os bytes enviados. Na prática ele é a identificação do primeiro byte de dados contido no segmento enviado. Os demais são sequenciados a partir deste byte.
- **Acknowledgement** identifica os bytes que foram recebidos e tratados sem erro pelo destino, bem como a sequência do próximo byte esperado
- **Tamanho** é representa o tamanho total do frame TCP
- **Reservado** é um campo ainda não utilizado
- **FLAGS** identifica as flags (syn, fin, psh, rst, ack, urg)
- **Window** identifica o tamanho da janela para o controle de fluxo
- **Checksum** destina-se a verificação de erros de transmissão. É calculado usando o pseudo header, o header TCP e também a área de dados
- **Urgent Pointer** é um ponteiro para dados urgentes, contidos na área de dados.

TCP

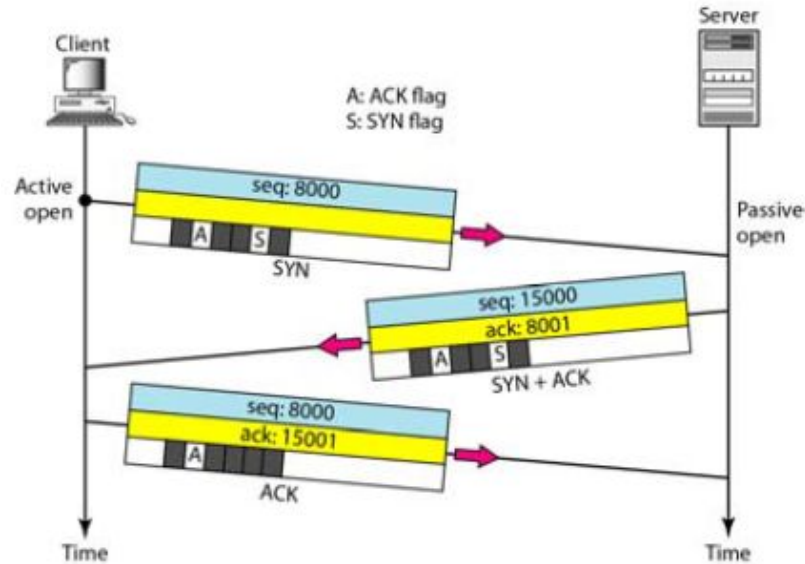
- Para ter controle dos pacotes enviados e conseguir efetuar a fragmentação, o TCP precisa que os usuários finais tenham o controle do que está sendo enviado.
- Três fases durante a conexão:
 - Estabelecimento da ligação
 - Transferência
 - Término da ligação

TCP

- Para o estabelecimento da conexão o TCP necessita que:
 - o cliente **inicia a ligação** enviando um pacote TCP com a *flag* **SYN** ativa;
 - espera-se que o **servidor aceite a ligação** enviando um pacote **SYN+ACK**.
 - *se durante um determinado tempo esse pacote SYN não for recebido, ocorre um **timeout** e o pacote SYN é reenviado.*
 - o estabelecimento da **ligação é concluído** por parte do cliente, confirmando a aceitação do servidor através de um pacote **ACK** como resposta.

TCP

- Estabelecimento de conexão



TCP

- A conexão só é encerrada depois da entrega dos dados ao receptor

