



UNIVERSIDADE DE  
**VASSOURAS**



# UNIVERSIDADE DE VASSOURAS

Curso de Graduação em Engenharia de Software

Aula - TCP

## Redes de Computadores

**Prof. André Saraiva**

Mestre em Sistemas de Computação

Especialista em Arquitetura e Projeto de Cloud Computing

Analista Sênior Blue Team em Cibersegurança pela Kimoshiro

Tutor EaD pela Universidade Federal Fluminense - UFF



## Tópicos

- 💡 Ementa da Disciplina
  - Camadas TCP/IP

## Rede de Computadores

### 📍 Modelos – OSI vs. TCP

- Este modelo de camadas de rede surgiu em 1984, e foi criada pela ISO (International Organization for Standardization).
- A Pilha de **Protocolos OSI** (Open Systems Interconnection) é um modelo de referência amplamente utilizado para entender como os protocolos de rede funcionam em conjunto.
- A pilha é dividida em sete camadas, cada uma com sua função específica, de modo a organizar o processo de comunicação em redes.

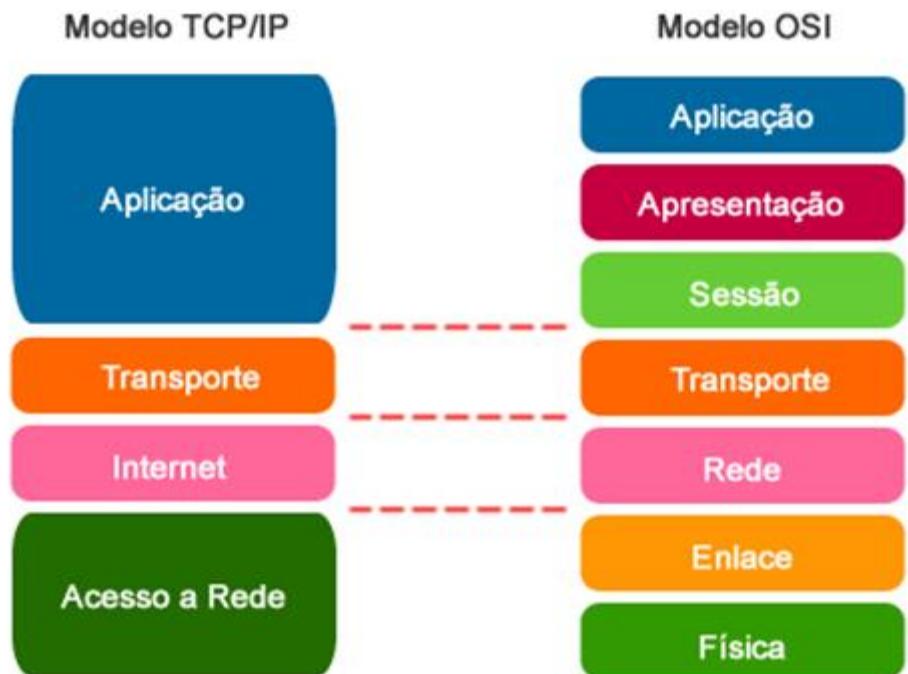
## Rede de Computadores

### 📍 Modelos – OSI vs. TCP

- O modelo TCP/IP surgiu em 1969, criado pelo Departamento de Defesa dos Estados Unidos (DoD)
- O Modelo TCP/IP é formado por protocolos, responsáveis pelo controle de diferentes funções, garantindo uma boa comunicação entre o transmissor e o receptor, e é dividido em quatro camadas: aplicação, transporte, internet e acesso à rede.
- Este modelo é utilizado para comunicação entre diferentes computadores, SO e apps. Para isso, cada computador precisa ter um módulo do TCP/IP em seu SO.

## Rede de Computadores

### 📍 Modelos – OSI vs. TCP



## Rede de Computadores

 Pilha de Protocolos da Internet

- **Aplicação:** suporte à aplicações de rede.
- Definem formato, ordem, semântica das mensagens trocadas pelas aplicações.

Exemplo da web:

Cliente (browser) gera mensagem de requisição de conteúdo.

Servidor envia mensagem de resposta.

Exemplos de protocolos:

HTTP, FTP, SMTP, ...

Modelo TCP/IP



## Rede de Computadores

 Pilha de Protocolos da Internet

- **Transporte:** comunicação entre processos em computadores (potencialmente) diferentes.
- Transfere dados de um processo específico para o outro, em computadores diferentes.
- Dois exemplos clássicos de protocolos: **UDP e TCP**.

Protocolos diferentes proveem **modelos de serviço diferentes**.

TCP provê garantias mais fortes.

*Responde a pergunta: para qual processo devemos enviar o conteúdo de um pacote que chegou a este host?*



## Rede de Computadores

 Pilha de Protocolos da Internet

- **Internet (antigamente chamada de rede):** comunicação entre hosts.
- Transfere dados de um host para outro. Diferença (aparentemente) sutil em relação à camada de transporte.
  - Aquela comunicava processos específicos, diferenciando-os.
  - Esta comunica hosts indiscriminadamente.
- Provê o serviço de roteamento dos pacotes.
- *Responde a pergunta: por qual caminho devemos enviar este pacote para que chegue ao host de destino?*

Modelo TCP/IP



## Rede de Computadores

 Pilha de Protocolos da Internet

- Acesso a rede (**Enlace e Física**)
- Enlace: comunicação nós (hosts, comutadores) diretamente conectados por um enlace. Exemplo: nós “vizinhos”.
- Diferente da camada de rede, preocupação apenas com **próximo salto**.
- Em enlaces susceptíveis a falhas, verifica se houve bits errados na recepção pelo enlace.
- ***Responde a pergunta: como envio o pacote para o próximo dispositivo no caminho?***

Modelo TCP/IP



## Rede de Computadores

 Pilha de Protocolos da Internet

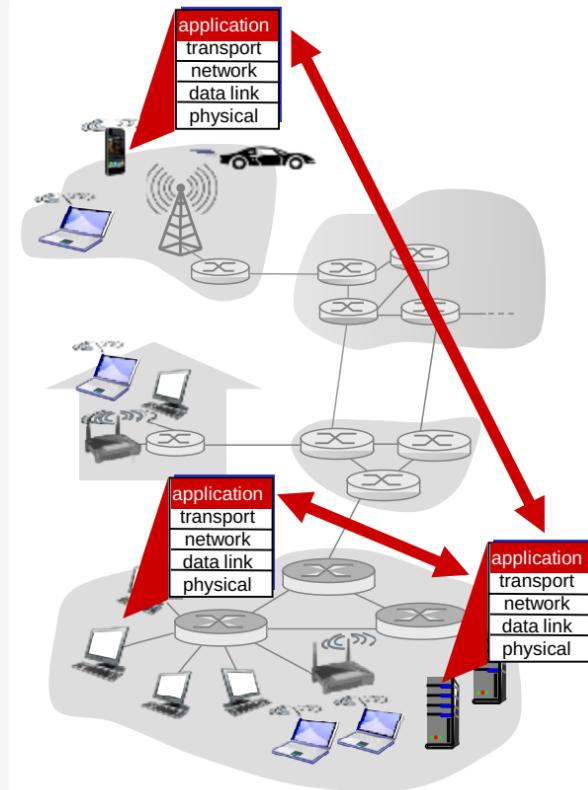
- **Acesso a rede (Enlace e Física)**
- **Física:** representação do pacote no meio físico de transmissão.
- Traduz bits para sinais.
  - Pulso elétricos, ondas acústicas, pulsos de luz, ...
- Se preocupa com a forma mais eficiente de representação. Exemplo:
  - como transmitir mais bits em menos tempo no canal?
  - como minimizar os erros na transmissão?



## Rede de Computadores

 Aplicações de Rede

- Escrever programas que:
  - Rodem em (diferentes) sistemas finais.
  - Se comuniquem via rede.
  - e.g., software de um servidor web que se comunica com o software de um browser.
- Não é necessário se preocupar com a escrita de software para dispositivos do núcleo.
  - Dispositivos do núcleo não rodam aplicações.
- Implantação apenas nos hosts permite rápido desenvolvimento, popularização das aplicações.



## Rede de Computadores

 Comunicação entre Processos

- Aplicações são compostas por **processos**.
  - Executados dentro de um host.
- Dentro de um mesmo host, processos se comunicam utilizando mecanismos de **comunicação inter-processos**.
  - Definidos pelo SO.
  - e.g., leitura/escrita de arquivos.
  - e.g., memória compartilhada.
- Em hosts diferentes, processos se comunicam através de **troca de mensagens**.

## Clientes, Servidores

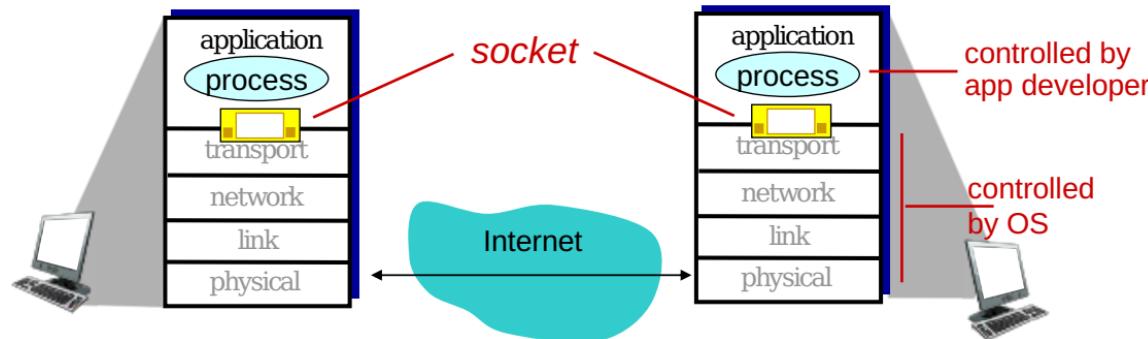
- **Processo cliente:** inicia a comunicação.
- **Processo servidor:** aguarda pedidos de comunicação.

**Nota:** aplicações P2P possuem tanto processos servidores, quanto processos clientes.

## Rede de Computadores

 Sockets

- Processos enviam/recebem mensagens utilizando um **socket**.
- Socket é análogo a uma janela.
  - Processo transmissor passa dados pela janela.
  - Do outro lado, implementação da camada de transporte se encarrega em entregar mensagem.
  - Na recepção, procedimento é análogo: processo espera que transporte entregue dados pelo socket.



## Rede de Computadores

 Sockets

- Para receber mensagens, processos devem ser **endereçados**.
- i.e., devem possuir algum **identificador**.
- Hosts possuem endereços únicos.
- Endereço IP
- **Pergunta:** endereço IP do host é suficiente para identificar o processo?
- **Resposta:** não, vários processos de rede podem ser executados simultaneamente no mesmo host.
- Solução: identificação de um processo usa tanto IP do host, quanto **números de porta**.
- Exemplos de números de porta:
  - Servidor HTTP: 80
  - Servidor de e-mail (SMTP): 25
- Para enviar mensagem HTTP para servidor web do IC:
  - Endereço IP: 200.20.15.48
  - Porta: 80

## Rede de Computadores

# Protocolo de Camada de Aplicação

- Tipos de mensagens trocadas.
  - e.g., requisição, resposta.
- Sintaxe das mensagens.
  - Quais os campos da mensagem, como dados são representados nestes campos.
- Semântica das mensagens.
  - Significado dos valores nos campos.
- Regras para como e quando gerar e responder a mensagens.
- Protocolos abertos:
  - Definidos em RFCs.
  - Permitem interoperabilidade.
  - e.g., HTTP, SMTP, DNS.
- Protocolos proprietários:
  - e.g., Skype.

## Rede de Computadores

# 📍 Domain Name System (DNS)

- **Pessoas:** muitos identificadores.
  - CPF, RG, número de passaporte, ...
- **Hosts e roteadores na Internet:**
  - Endereço IP (número de 32 bits) usado para endereçar datagramas.
  - “Nome”, e.g., www.yahoo.com, usado por humanos.
- **Pergunta:** como mapear nomes para seus respectivos IPs e vice-versa?
- **Domain Name System (DNS):**
- **Base de dados distribuída:** implementada em uma hierarquia de múltiplos servidores de nomes.
- **Protocolo da camada de aplicação:** hosts e servidores de nome se comunicam para **resolver** nomes (tradução entre nome e endereço IP).
  - Note: função fundamental da Internet implementada como protocolo de aplicação.
  - Complexidade nas bordas.

## Rede de Computadores

# Domain Name System (DNS)

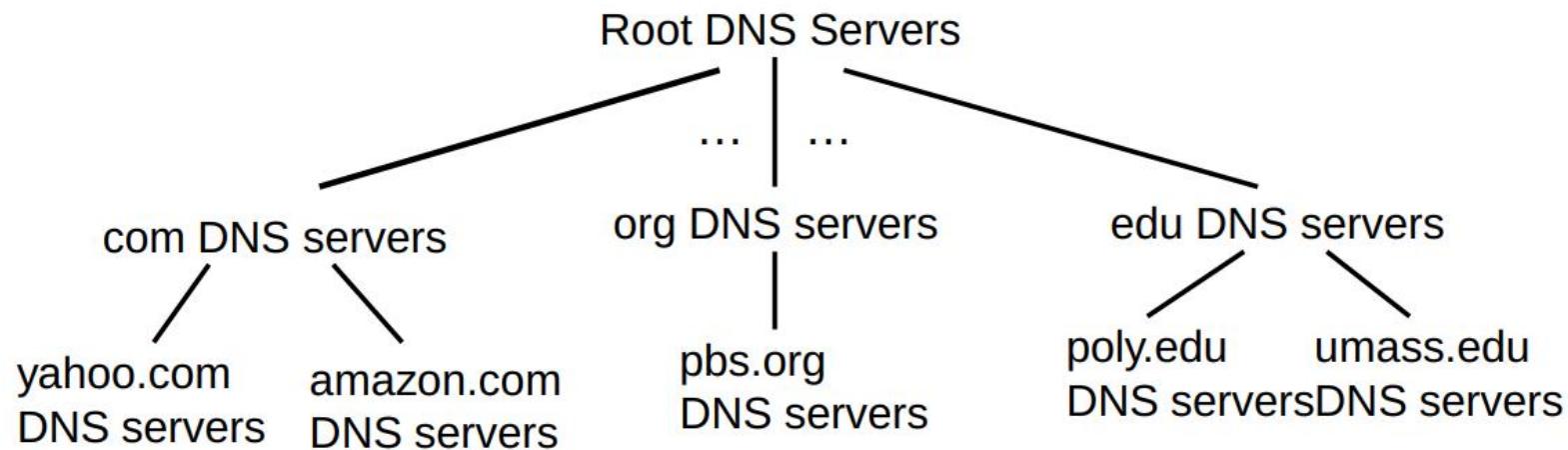
- **Serviços do DNS:**
- Tradução de nomes de hosts para endereços IP.
- Host aliasing.
  - Atribuição de “apelidos”.
  - Host possui **nome canônico** e, possivelmente, vários apelidos.
- Aliasing de servidores de e-mail.
- Balanceamento de carga.
  - Servidores web replicados.
  - Cada servidor com seu IP.
  - Mas o mesmo nome associado a vários IPs.
- **Por que não um DNS centralizado?**
  - Ponto único de falha.
  - Concentração de grande volume de tráfego.
  - Base de dados centralizada distante.
  - Manutenção.

Em suma: **não escala!**

## Rede de Computadores

 Domain Name System (DNS)

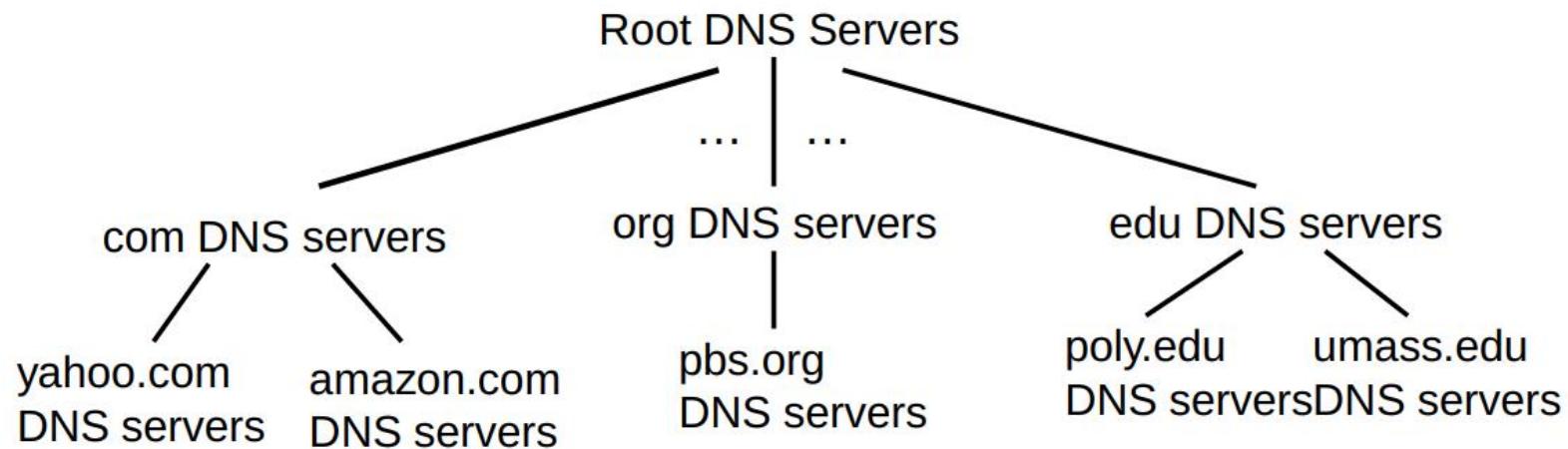
- Cliente quer IP de **www.amazon.com**. Primeira abordagem:
- Cliente pergunta ao servidor raiz a localização do DNS do domínio .com.
- Cliente pergunta ao servidor DNS do domínio .com a localização do servidor DNS do domínio amazon.com.
- Cliente pergunta ao servidor DNS do domínio amazon.com pelo endereço IP do host www.amazon.com.



## Rede de Computadores

 Domain Name System (DNS)

- **Servidores Top-Level Domain (TLD):**
- **TLD:** .org, .net, .com, .edu, .br, .us, .uk, .jp, ...
- Cada TLD tem seu servidor DNS específico.
- A Registro.br mantém o DNS para o TLD .br.
- **Servidores autoritativos:**
- Servidor de DNS de uma organização específica.
- Provê mapeamentos para os endereços IP da organização e seus nomes de host.



📍 Camada de Transporte

# Camada de Transporte

## Rede de Computadores

 Camada de Transporte

- **Transporte:** comunicação entre processos em computadores (potencialmente) diferentes.
- Transfere dados de um processo específico para o outro, em computadores diferentes.
- Dois exemplos clássicos de protocolos: **UDP e TCP**.

Protocolos diferentes proveem **modelos de serviço diferentes**.

TCP provê garantias mais fortes.

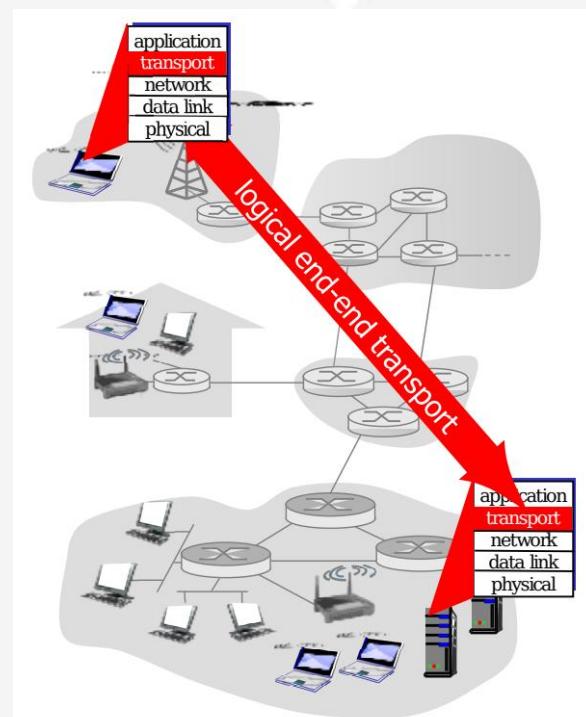
*Responde a pergunta: para qual processo devemos enviar o conteúdo de um pacote que chegou a este host?*



## Rede de Computadores

 Camada de Transporte

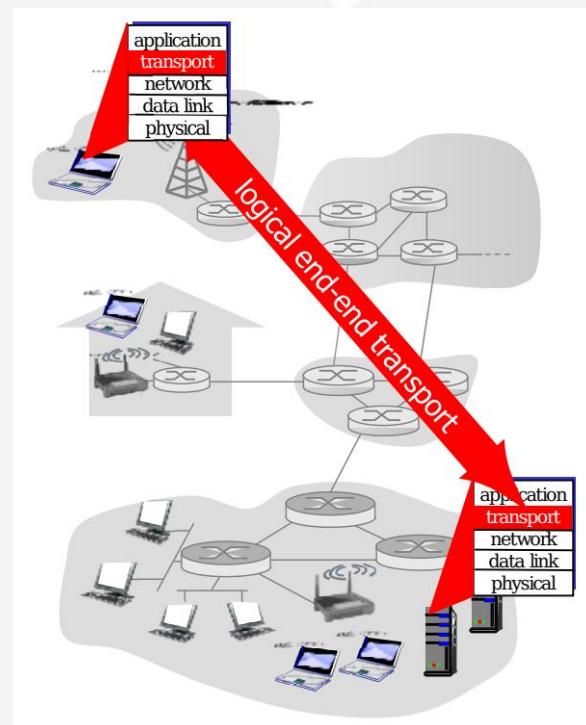
- Provê **comunicação lógica** entre **processos** da aplicação rodando em hosts diferentes.
- Protocolos de transporte **são executados nos sistemas finais**.
- Lado transmissor: quebra mensagens da aplicação em **segmentos**, passa segmentos para camada de rede.
- Lado receptor: remonta segmentos para formar mensagens originais, passa mensagens para a camada de aplicação.
- Mais de um protocolo disponível para as aplicações.
- Na Internet: TCP e UDP.



## Rede de Computadores

 Camada de Transporte

- 12 crianças na casa do Bob enviam cartas a 12 crianças na casa da Alice
- hosts = casas.
- processos = crianças.
- mensagens da aplicação = cartas nos envelopes.
- protocolo de transporte = Bob e Alice que demultiplexam cartas para as crianças.
- protocolo de camada de rede = correios.



## Rede de Computadores

 User Datagram Protocol (UDP)

- “Serviço básico”, “mínimo” da camada de transporte da Internet.
- Segmentos UDP podem ser: Perdidos e/ou Entregues, porém fora de ordem para a aplicação.
- **Sem conexão:**
  - Não há comunicação inicial entre UDP do transmissor e do receptor.
  - Datagramas são simplesmente enviados.
  - Cada segmento UDP é tratado de forma completamente independente dos demais.
- Usos do UDP:
  - **Aplicações de Streaming multimídia (tolerantes a perda, sensíveis a taxa).**
  - **DNS.**
  - **SNMP.**
- Transferência confiável sobre UDP: Possível, mas depende da aplicação.
  - Adição de confiabilidade da própria aplicação e métodos de recuperação de erros específicos de cada aplicação.

## 📍 User Datagram Protocol (UDP) - Checksum

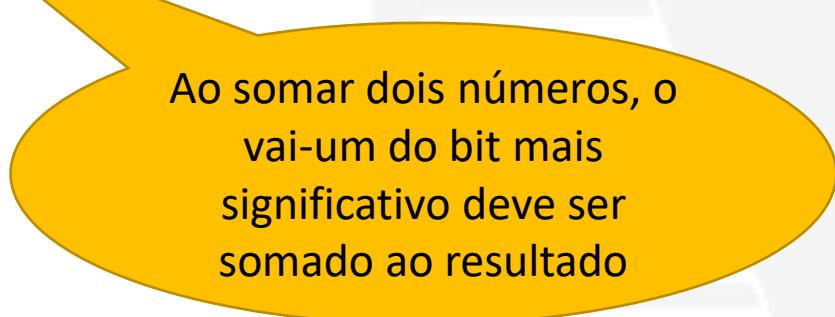
- **Objetivo:** detectar “erros” (e.g., bits com valor trocado) no segmento transmitido.
- **Transmissor:**
  - Trata conteúdo do segmento, incluindo campos de cabeçalho, como uma sequência de inteiros de 16 bits.
  - Checksum: soma, em complemento a 1, do conteúdo do segmento.
  - Transmissor coloca valor do checksum no campo do cabeçalho UDP.
- **Receptor:**
  - Computa o checksum do segmento recebido.
  - Checksum computado é igual ao indicado pelo cabeçalho?
  - Não: erro detectado.
  - Sim: nenhum erro detectado.

Mas pode haver erros mesmo assim? Sim...

 User Datagram Protocol (UDP) - Checksum

- Soma de dois valores de 16 bits

Valor 1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
Valor 2	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Soma	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1



Ao somar dois números, o  
vai-um do bit mais  
significativo deve ser  
somado ao resultado

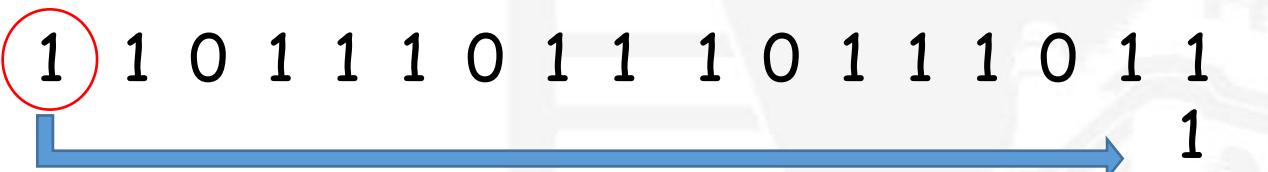
 User Datagram Protocol (UDP) - Checksum

- Soma de dois valores de 16 bits

Valor 1      1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0  
Valor 2      1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

---

Soma      1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1

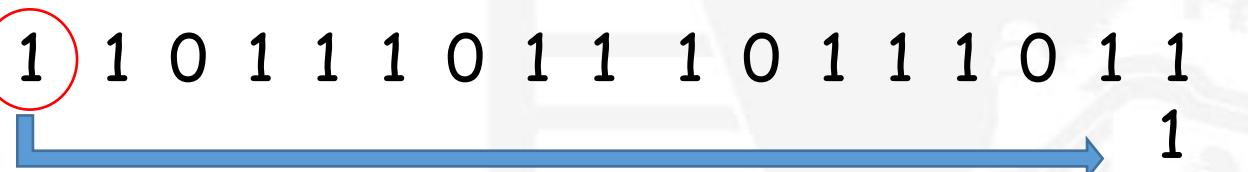


---

Soma      1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0

 User Datagram Protocol (UDP) - Checksum

- Resultado em complemento a 1

Valor 1	1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
Valor 2	1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
	<hr/>
Soma	1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
	
	<hr/>
Soma checksum	1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0
	0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

## 📍 User Datagram Protocol (UDP) - Checksum

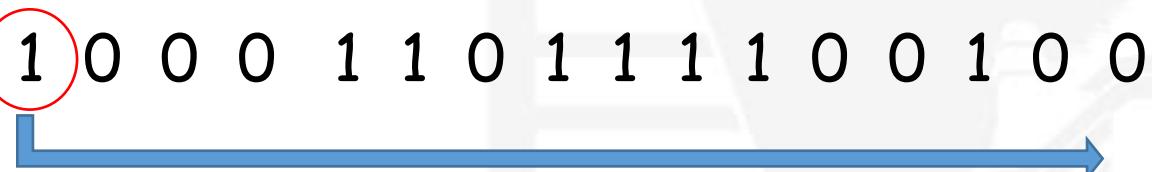
- **Exercício para fazer agora. Descubra o checksum da soma abaixo**

0	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1	0	0	0	1

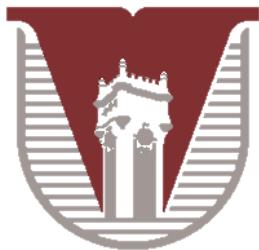
---

 User Datagram Protocol (UDP) - Checksum

- Exercício para fazer agora

Valor 1	0 1 0 0 1 1 0 0 1 1 0 1 0 1 1 1
Valor 2	1 1 0 0 1 1 1 0 1 1 1 1 0 0 0 1
	<hr/>
Soma	1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0
	
	<hr/>
Soma checksum	0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 1
	1 1 1 0 0 1 0 0 0 0 1 1 1 0 1 1 0

## Contato



**Professor:**  
André Saraiva, MSc



**E-mail:**  
[andre.saraiva@univassouras.edu.br](mailto:andre.saraiva@univassouras.edu.br)