

DCA-125 Sistemas de Tempo-Real

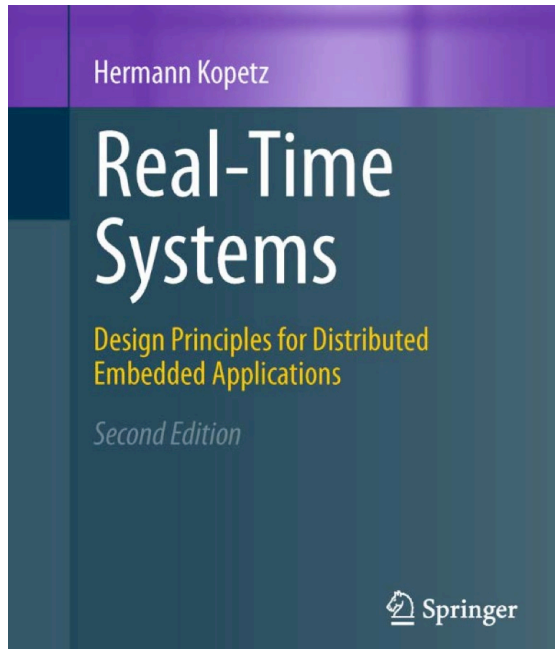
Luiz Affonso Guedes
www.dca.ufrn.br/~affonso
affonso@dca.ufrn.br



Comunicação em Rede

Referências Bibliográfica

H. Kopetz. Real-Time Systems, Second Edition, Springer. Capítulo 7- Real-Time Communication



Comunicação em Rede

Objetivo do Capítulo

- ❑ Contextualizar a necessidade de comunicação em rede.
- ❑ Discutir os principais modelos de comunicação.
- ❑ API Sockets

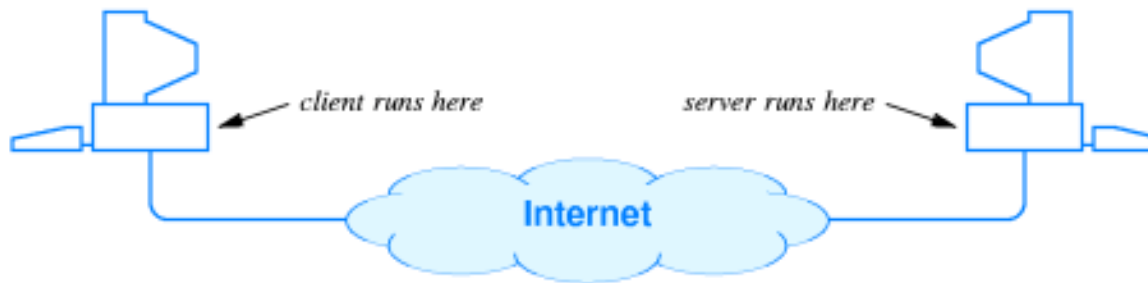
Fatos de Indeterminismo Temporal

□ No nó:

- Aplicação
- SO

□ Na Comunicação

- Rede
- Acesso ao meio
- Mecanismo de comunicação



Outros Tipos de Sistemas Operacionais

- ❑ Sistemas Operacionais de Tempo-real
 - Aplicações que exigem garantias determinísticos de tempo
 - Controle de processos industriais
 - Tempo-real soft e tempo-real hard
- ❑ Sistemas Operacionais Embarcados
 - Altamente compactos
 - Smart cards
- ❑ SO Distribuídos, de tempo-real e embarcados, seguros, ...

Comunicação de Tempo-Real

❑ Requisitos gerais:

- Baixa latência dos protocolos,
- Entrega de dados confiável,
- Atraso fim-a-fim determinístico,
- Jitter fim-a-fim- baixo,
- Sincronização de tempo global

Comunicação de Tempo-Real

❑ Requisitos de tempo:

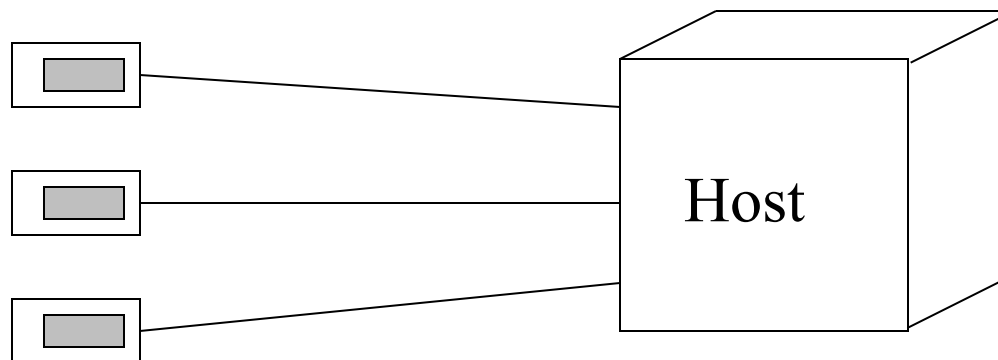
- Mensagens em geral têm tamanho muito pequeno quando comparadas às das aplicações de não tempo-real.
 - Monitoramento de temperatura X transferência de arquivo
- Jitter Mínimo
 - Em geral as leis de controle assumem amostragem constante dos sinais (transformada Z).
- As mensagens devem carregar consigo os tempos associados às variáveis
 - Os tempo devem ser sincronizados entre os diversos dispositivos distribuídos ao longo da rede de comunicação.

Revisão de Redes

- ❑ Definição de Redes de Computadores
 - É um conjunto de computadores autônomos e interconectados.



Rede



Não é
Rede

Revisão de Redes

❑ Autonomia

- Em rede, nenhum computador **obedece** a comandos de outro.
 - Possui autonomia até para se desconectar da rede

❑ Interconexão: Meio de Comunicação

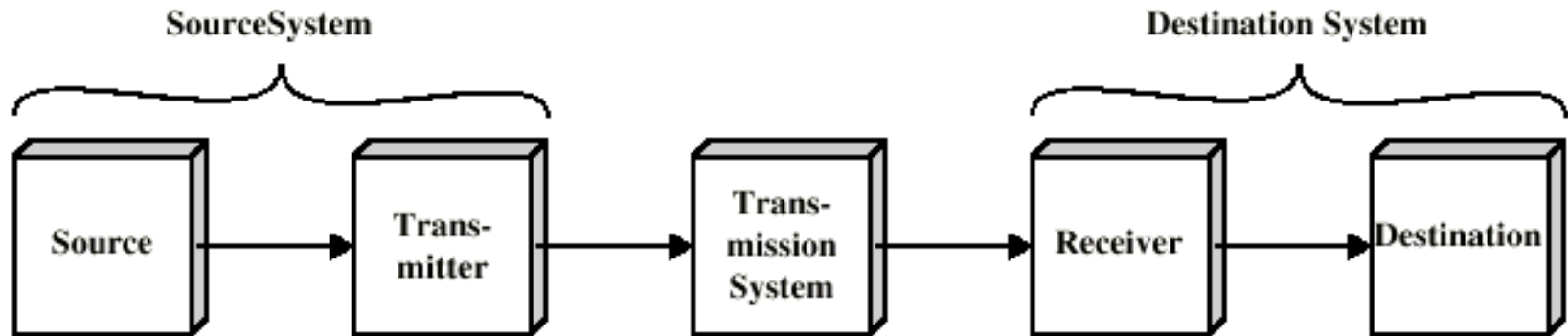
- Cabo de cobre
 - Coaxial
 - Par Trançado
- Fibra óptica
- Rotas de microondas
- Radiodifusão

Definições Básicas

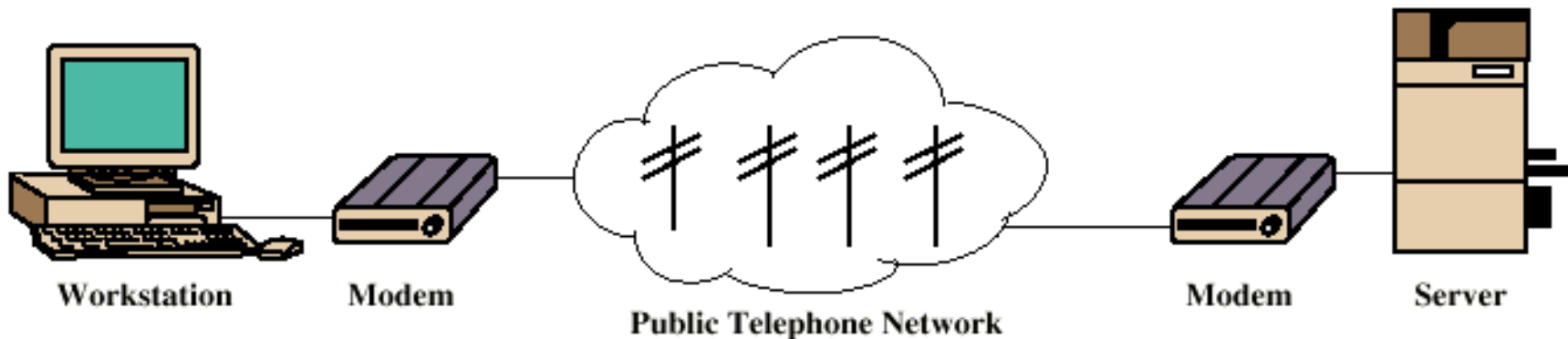
- ❑ Uma rede é um conjunto de computadores e outros dispositivos interligados por um meio físico.
 - Estes podem enviar e receber dados entre si, mais ou menos em tempo real.
- ❑ Redes de Pacotes
 - O canal de comunicação é compartilhado entre os diversos computadores.
 - Os computadores de reversam na utilização do canal, enviando mensagens: pacotes.

Introdução

○ Modelo de Comunicação simplificado



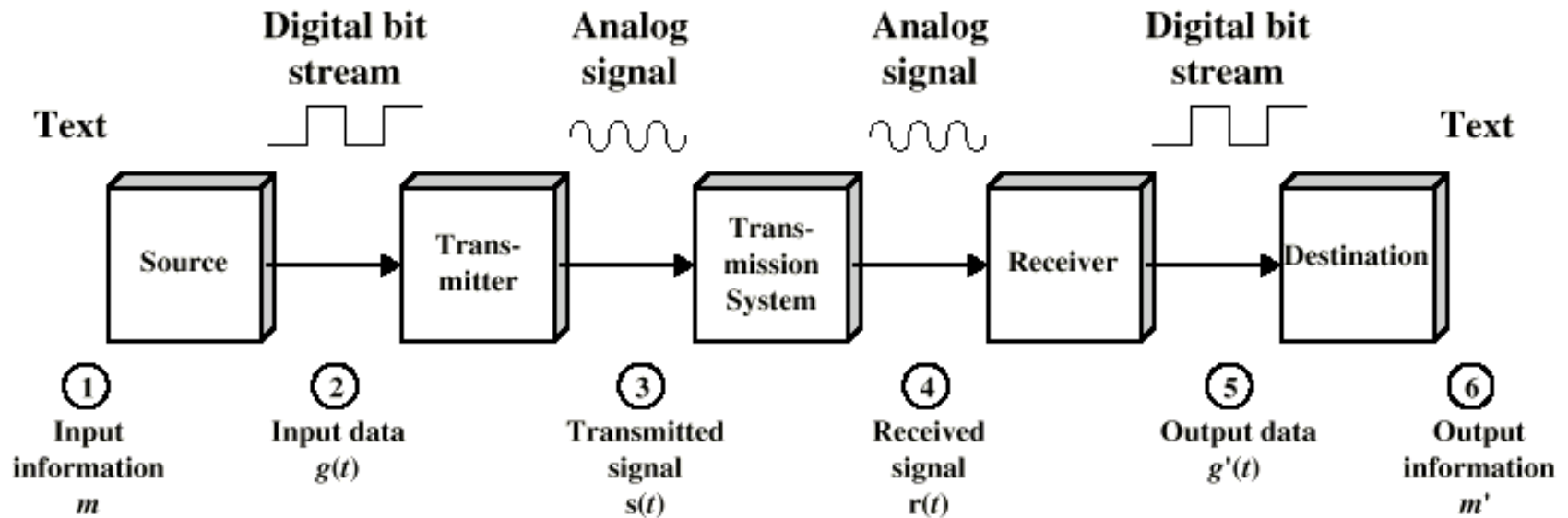
(a) General block diagram



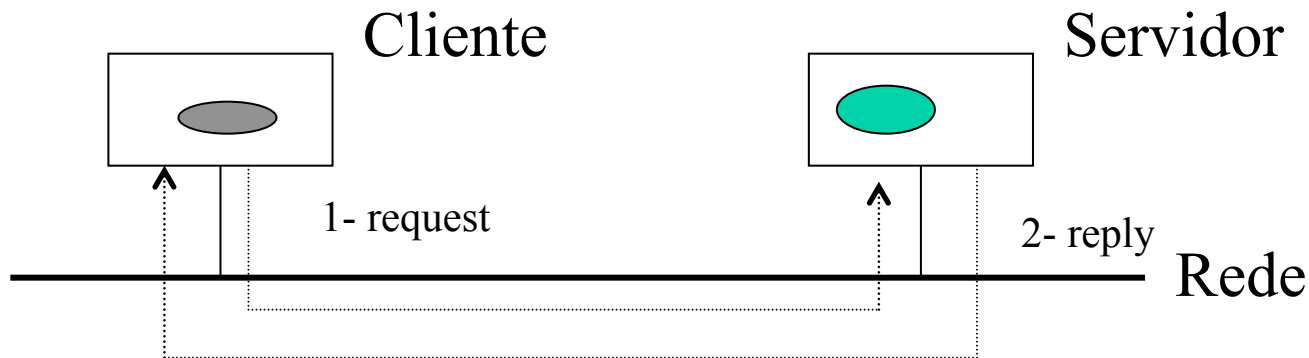
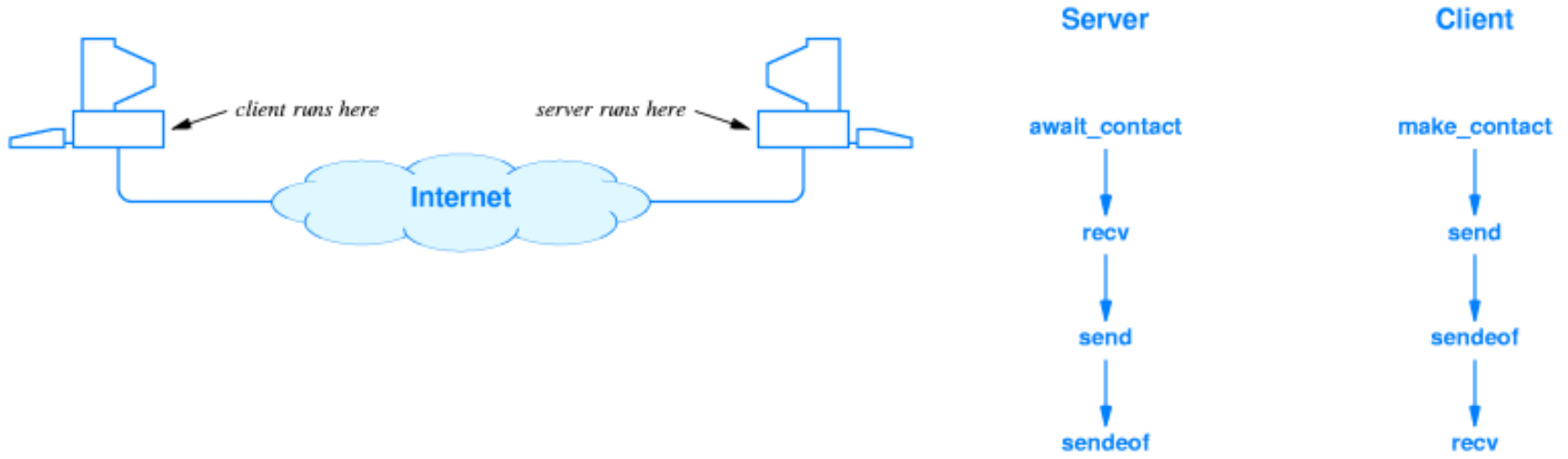
(b) Example

Introdução

○ Modelo de Comunicação simplificado

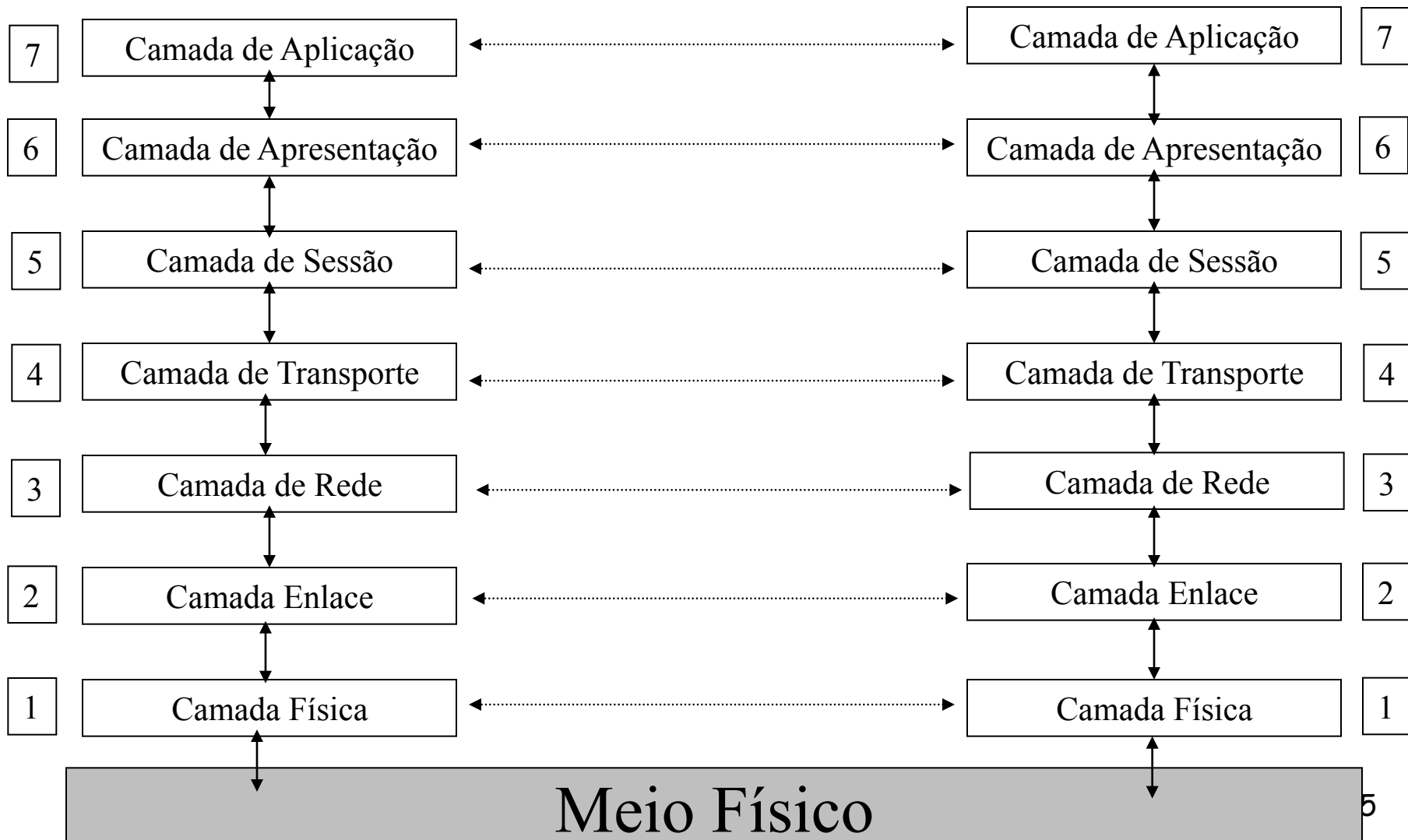


Paradigma Cliente-Servidor

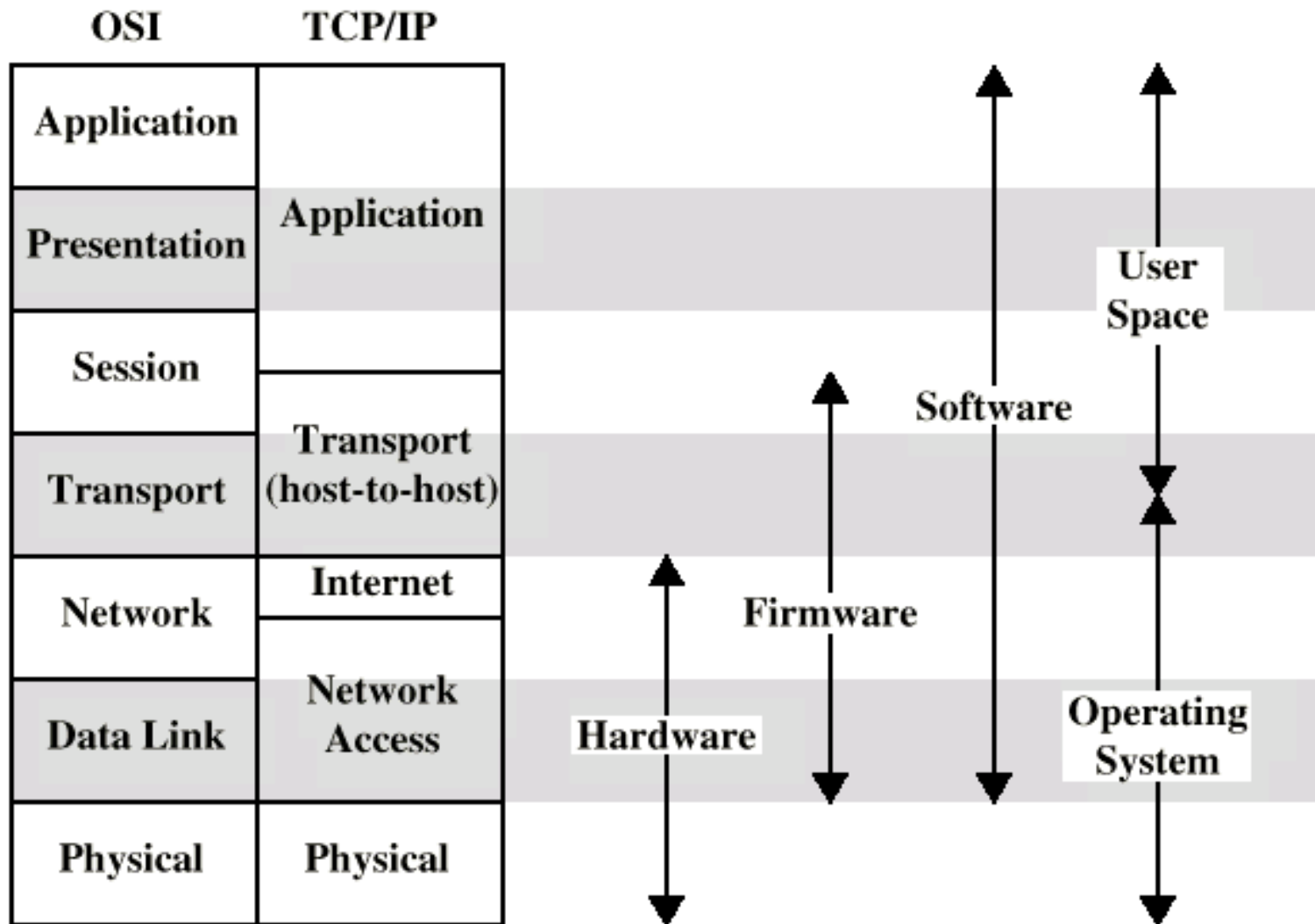


Introdução

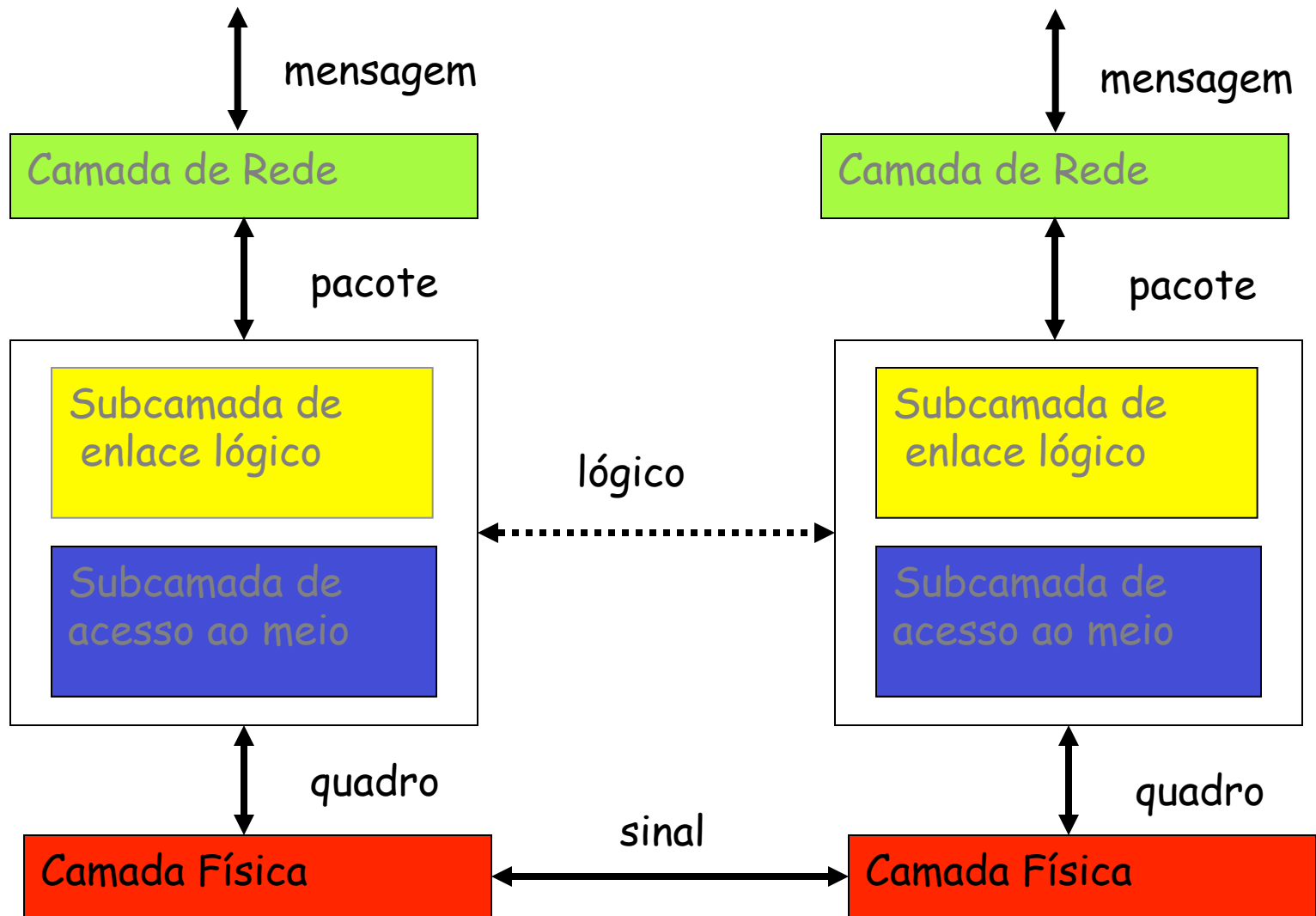
❑ Modelo OSI/ISO de 7 camadas



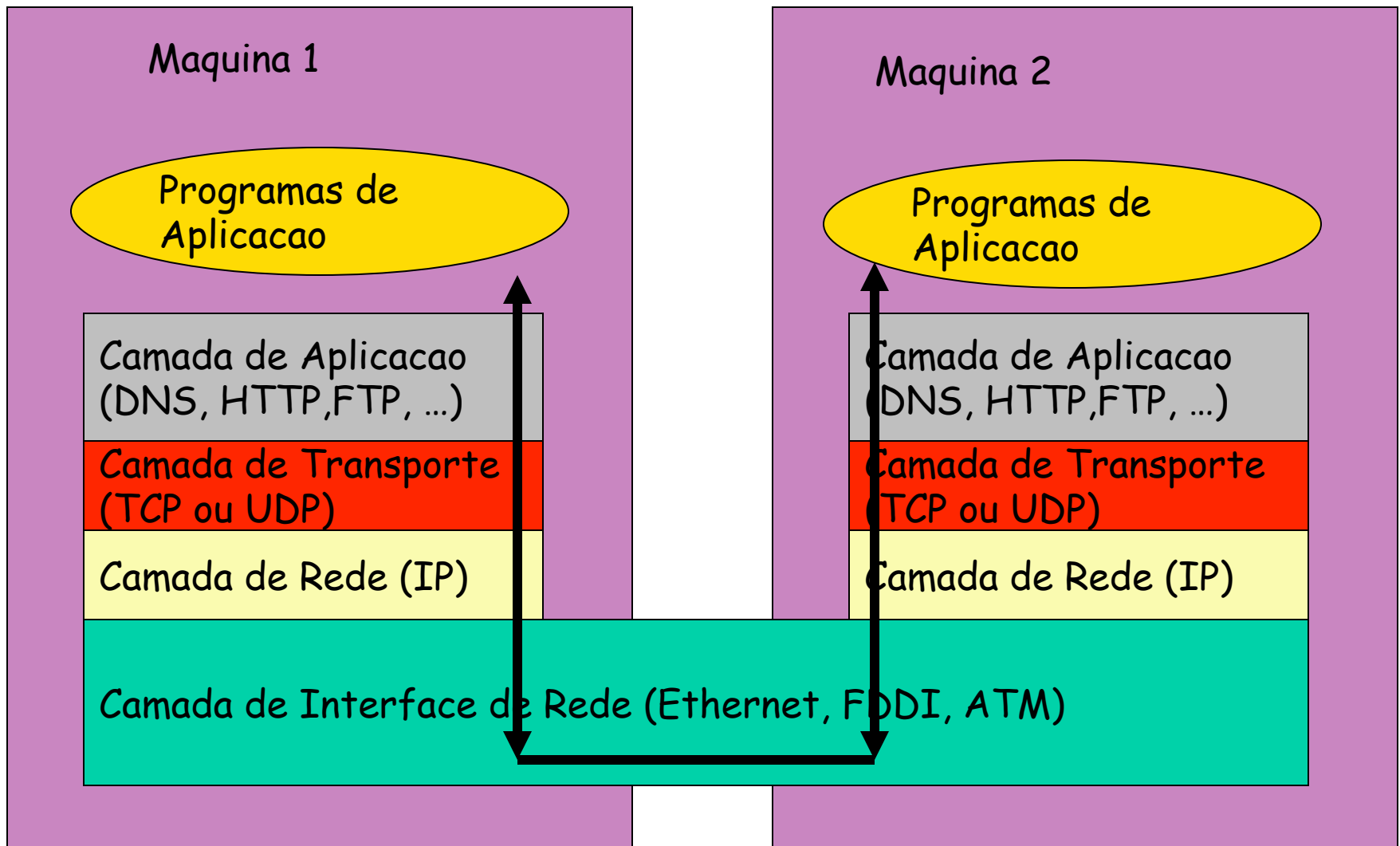
Modelo OSI/ISO e Pilha TCP/IP



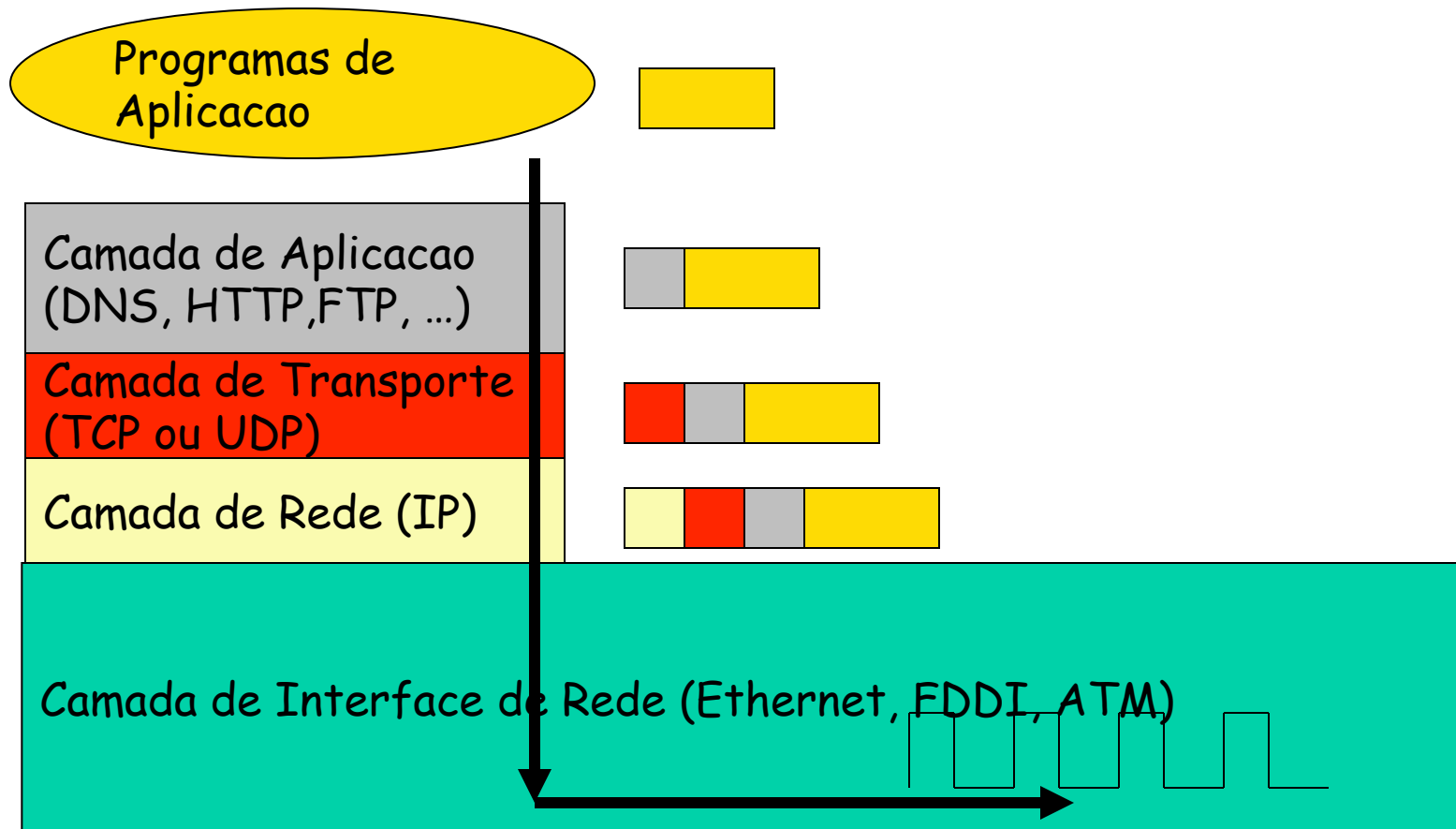
Estrutura do Modelo



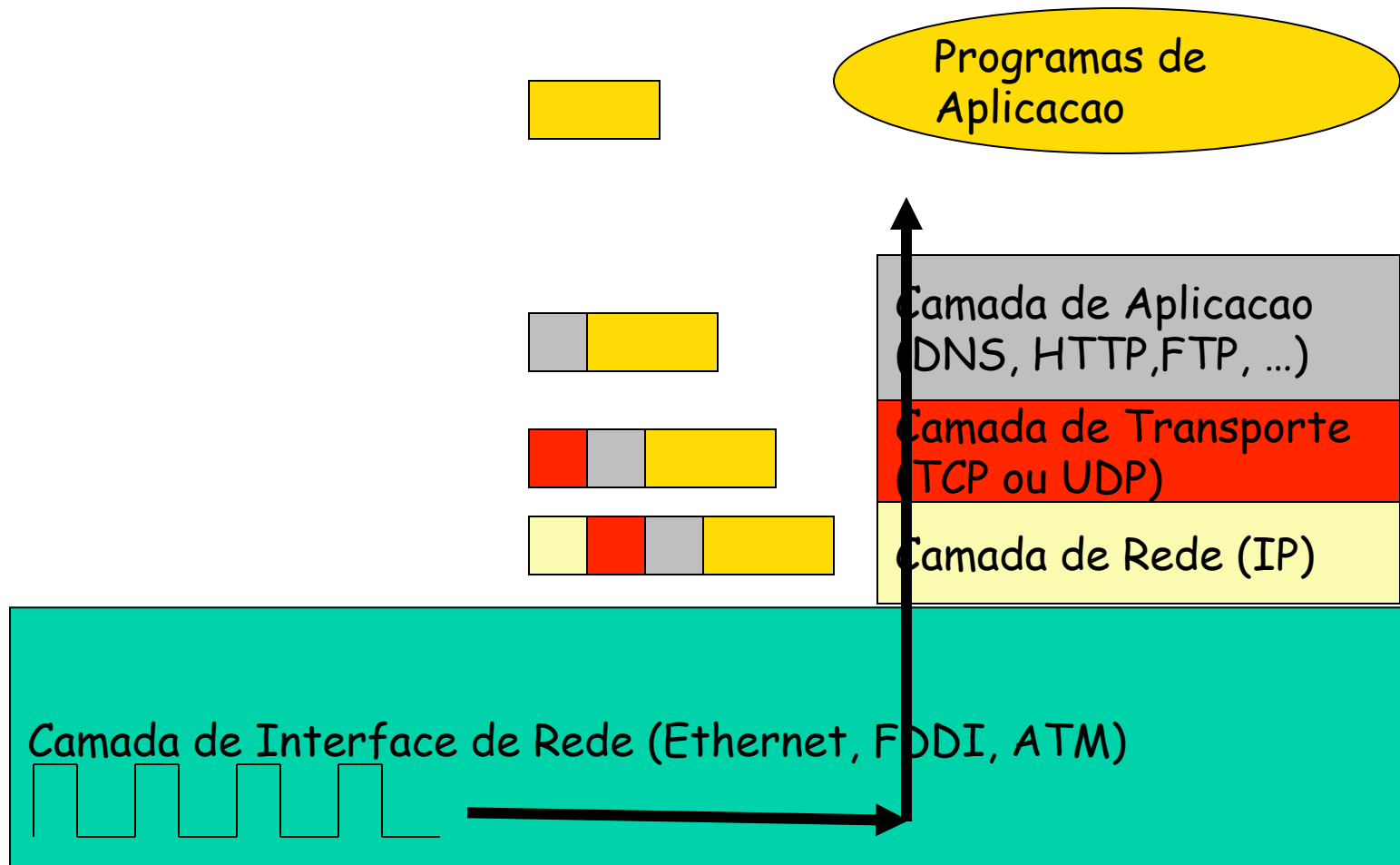
Camadas de uma Redes



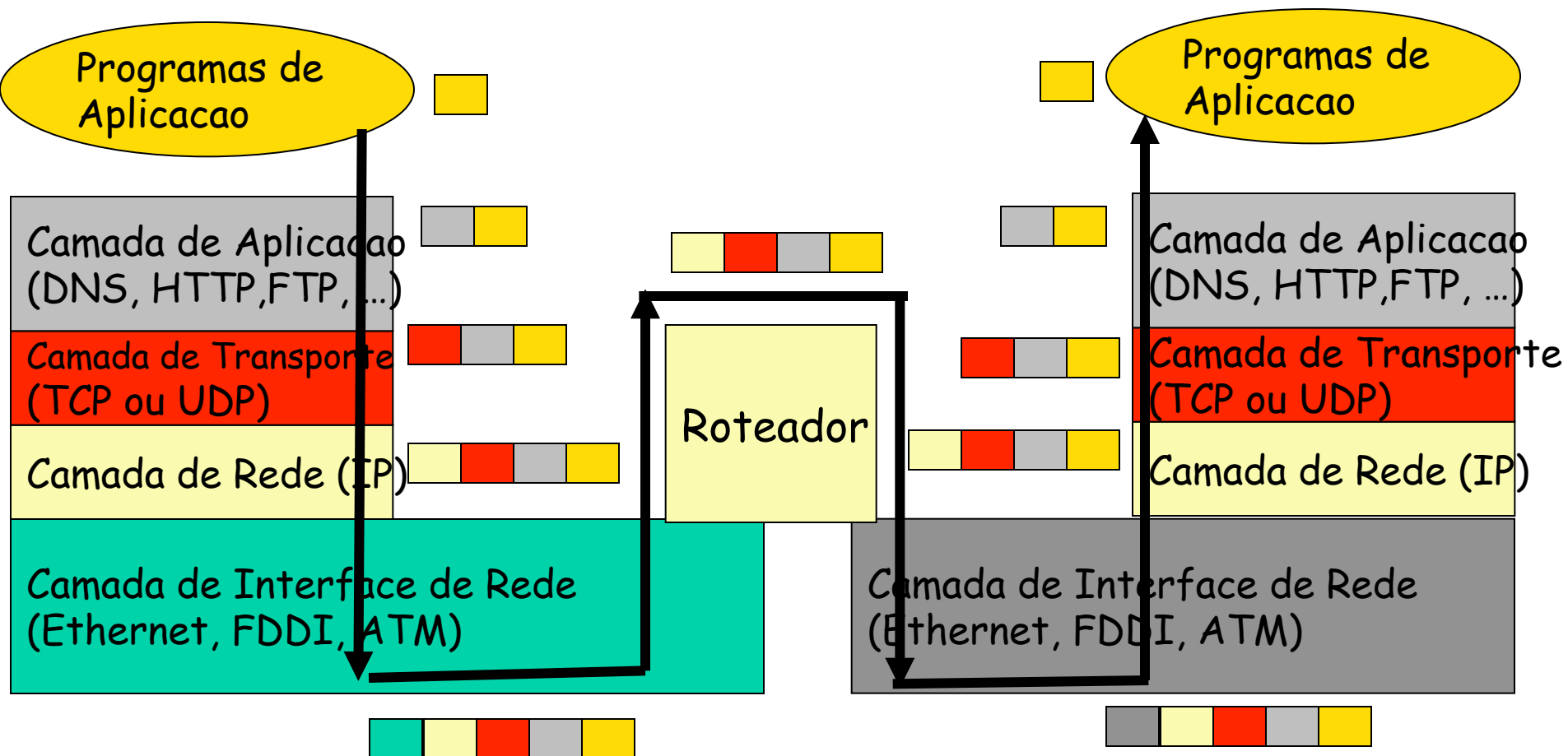
Encapsulação de Dados



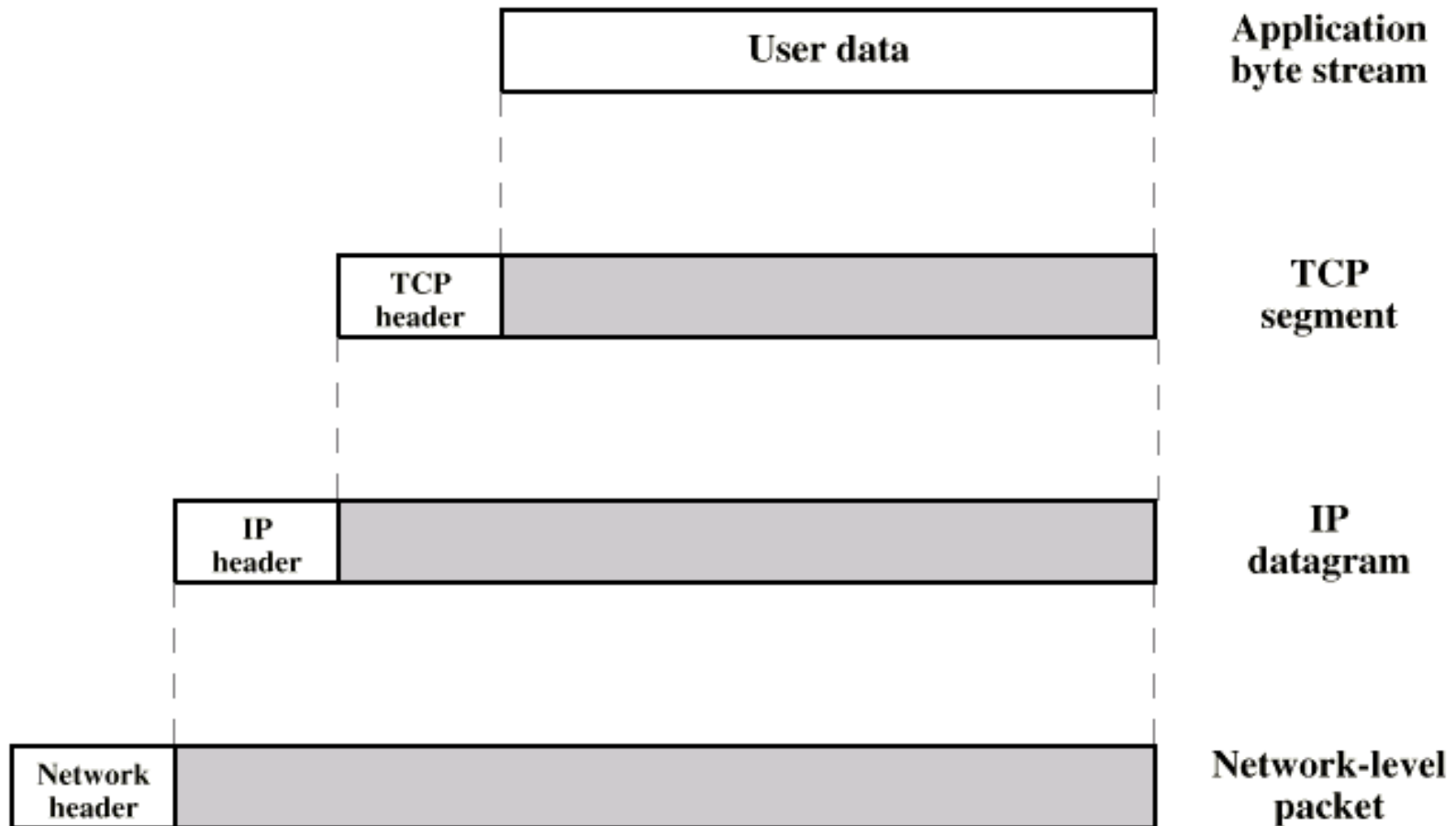
Encapsulação de Dados



Roteamento de Pacotes



PDU da Pilha TCP/IP

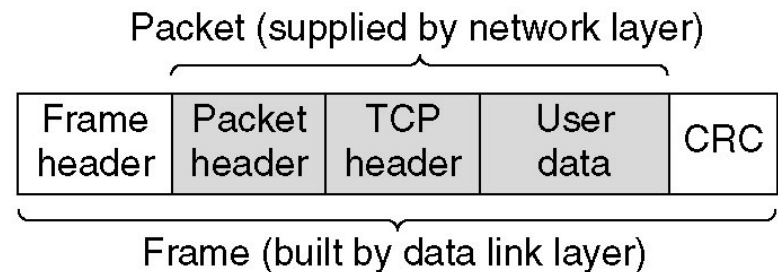


Equipamentos de Rede

❑ Comparação entre os dispositivos de rede.

Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

(a)



(b)

(a) Localização dos dispositivos nas camadas.

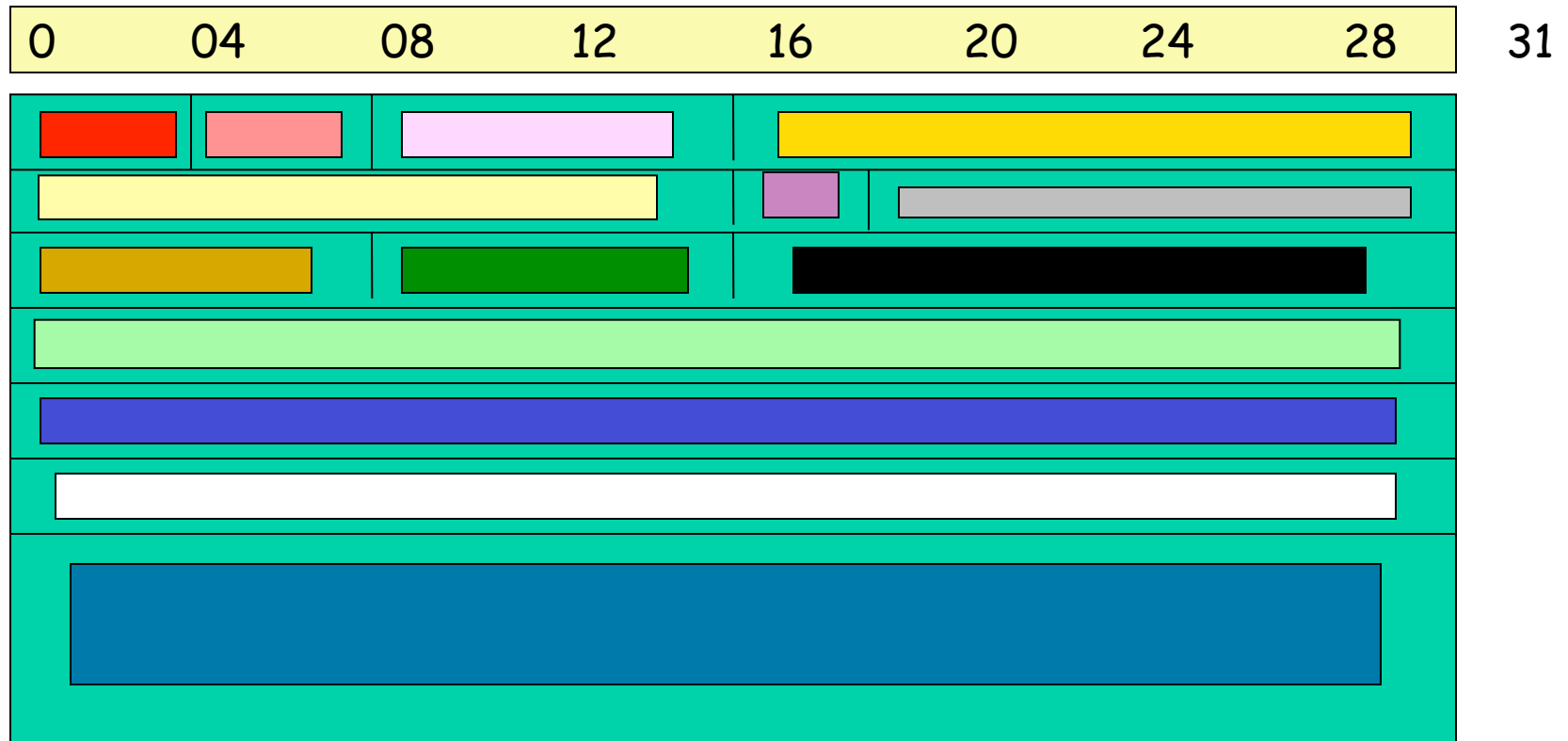
(b) Quadros, pacotes e cabeçalhos.

Camada de Rede

❑ Protocolos IP (Internet Protocol)

- Responsável pelo roteamento dos pacotes entre as sub-redes.
- Numeração IP:
 - 04 bytes.
 - 05 Classes de números: A, B, C, D, E.
 - Cada interface de rede tem de ter pelo menos um endereço IP.
 - 127.0.0.1 → endereço de loop-back (localhost)

Formato do Pacote IP



	versão		Identificação		Protocolo		Opções		
	Tamanho cabeçalho		Flags		Checksum/cab.		End. fonte		Dados
	Tipo de serviço		Offset/frag		End. fonte				
	Tamanho do pacote		TTL		End. destino				

Formato do Pacote IP

❑ Campo Versão:

- Atualmente versão quatro: 4
- IPv6 (Versao 6): 6

❑ Campo PROTOCOLO:

- Protocolo TCP: 6
- Protocolo UDP: 17

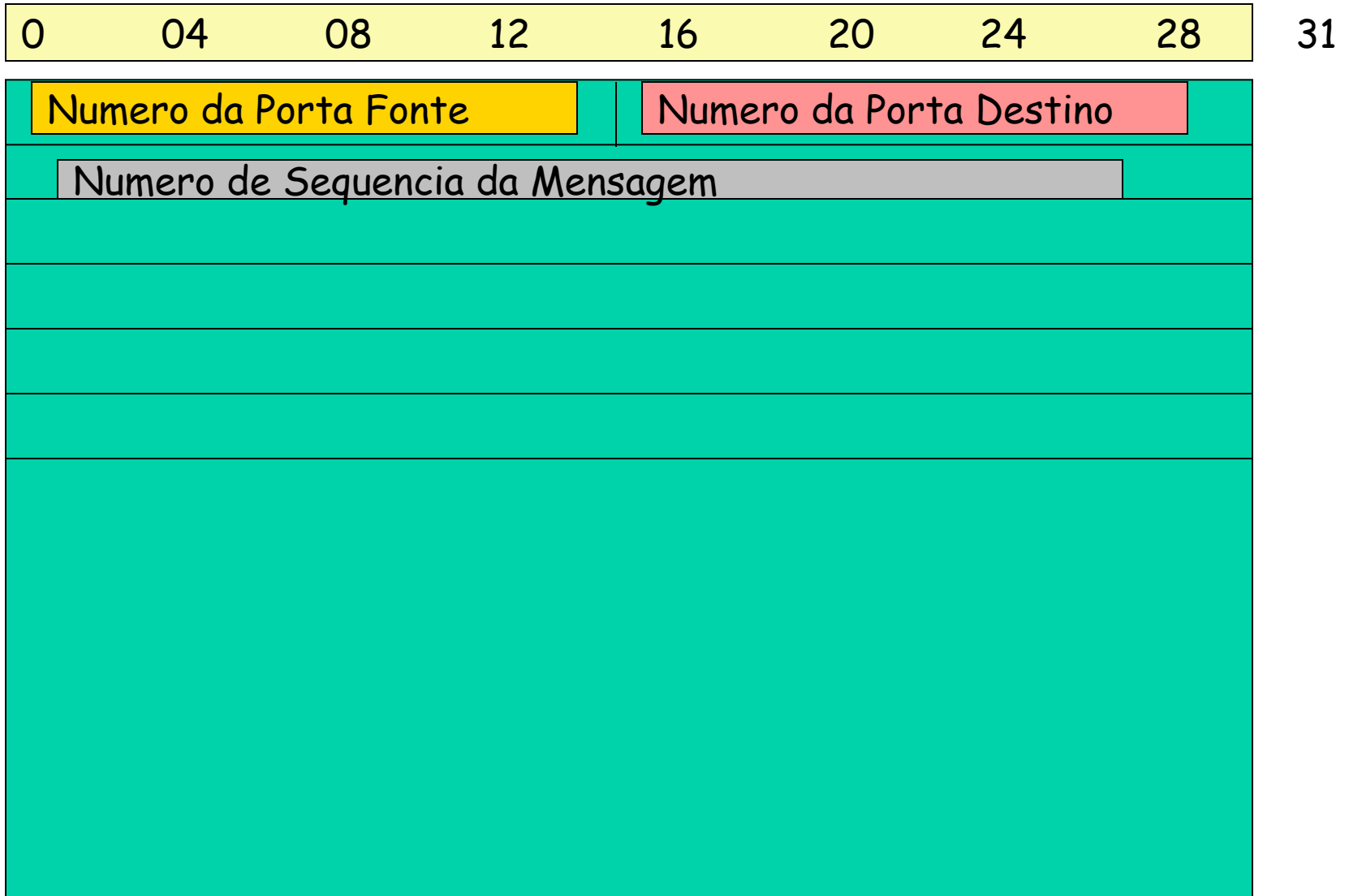
❑ Campos TTL - Time To Live e Offset/Flags:

- Únicos campos que são alterados durante o percurso de roteamento dos pacotes

Camada de Transporte

- ❑ Responsável pela comunicação fim-a-fim.
- ❑ TCP - Transmission Control Protocol
 - Protocolo com orientação de conexão e garantia de entrega de mensagem.
- ❑ UDP - User Datagram Protocol
 - Protocolo sem orientação de conexão e sem garantia de entrega de mensagem.

Formato da Mensagem TCP

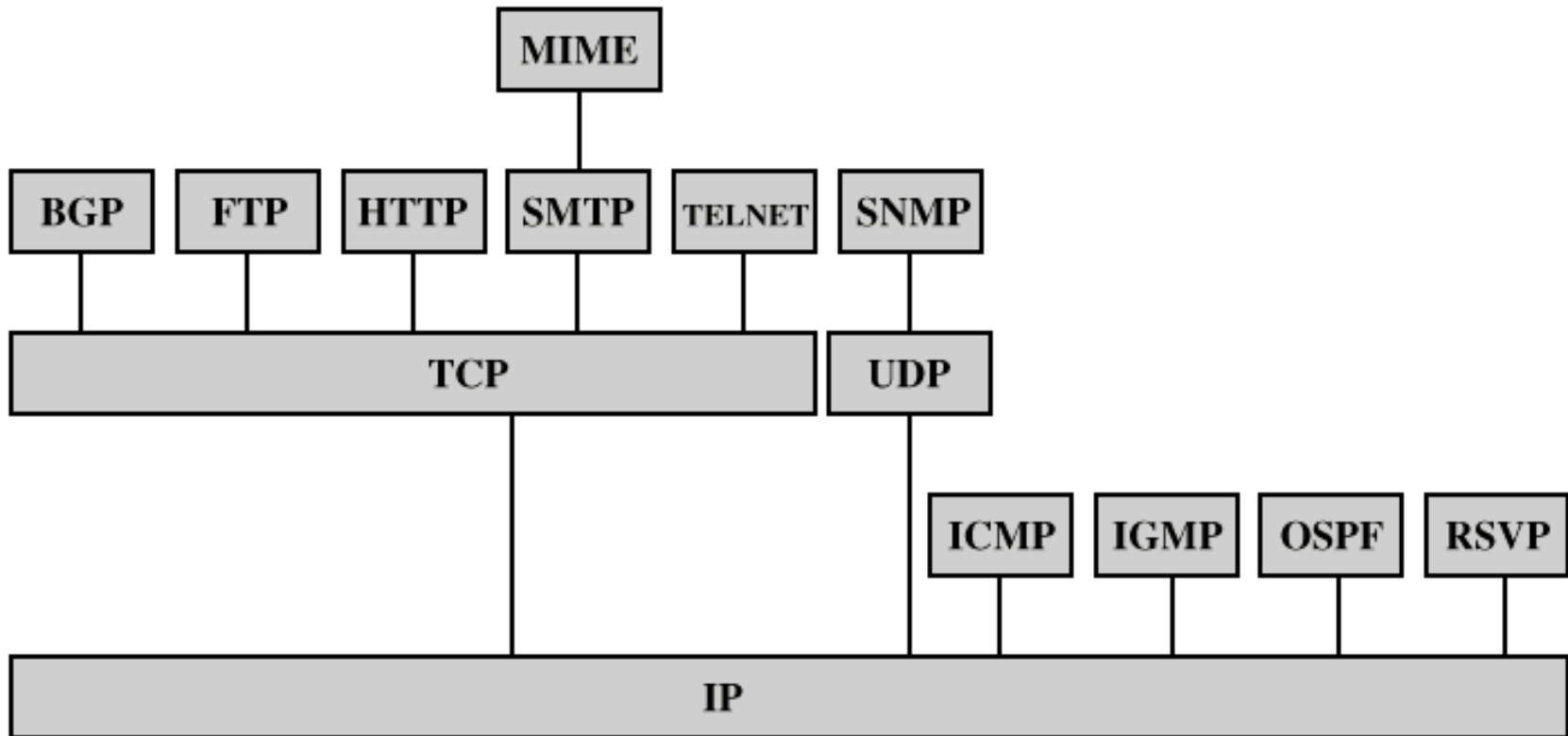


Camada de Transporte

❑ Conceito de serviço

- Serviços estão associados com portas.
- Portas são números inteiros representados por 02 bytes:
 - 65.536 portas possíveis
 - As portas de 1 a 1023 são destinadas a serviços específicos.

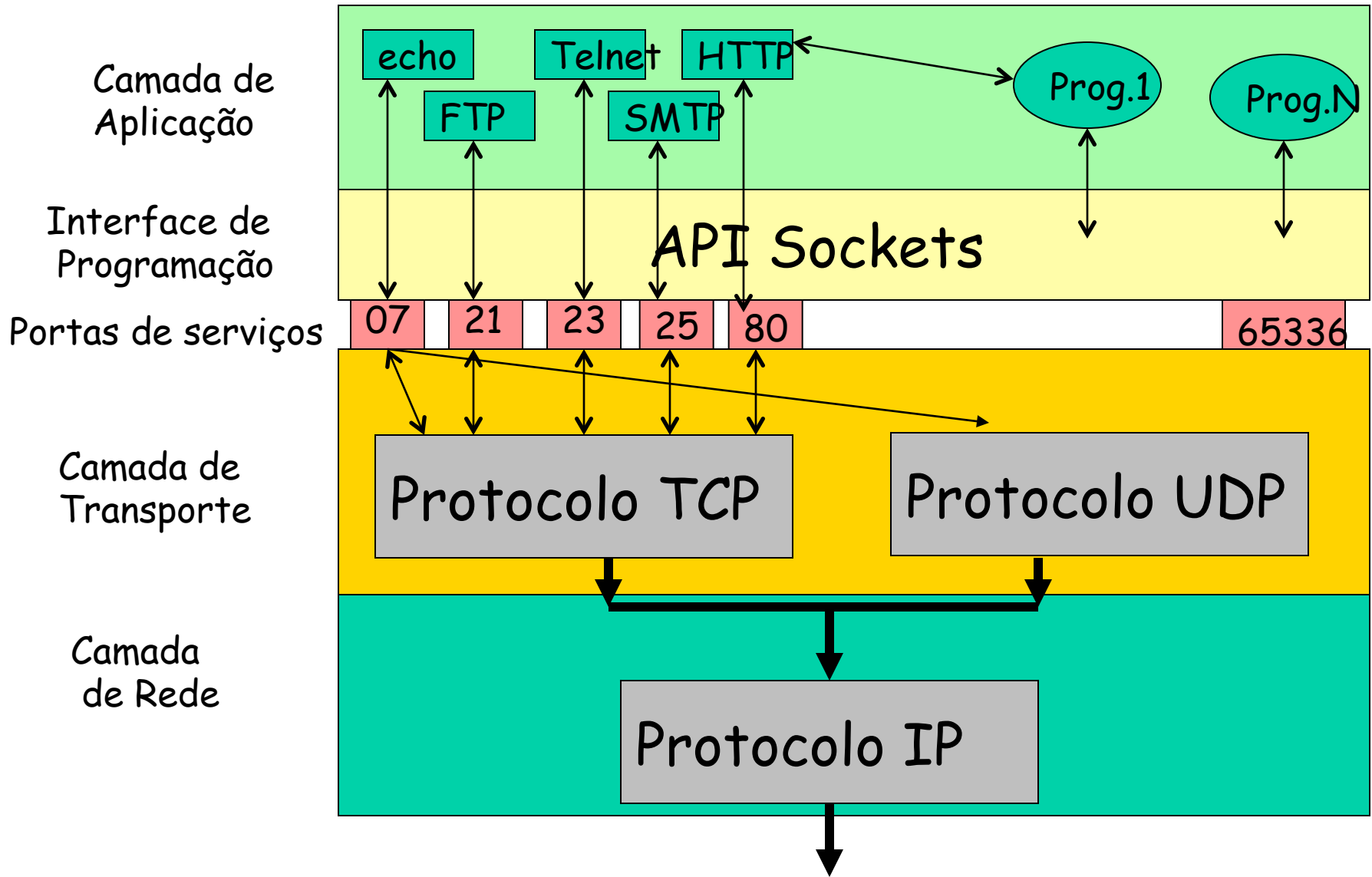
Exemplos de Protocolos a Pilha TCP/IP



BGP = Border Gateway Protocol
FTP = File Transfer Protocol
HTTP = Hypertext Transfer Protocol
ICMP = Internet Control Message Protocol
IGMP = Internet Group Management Protocol
IP = Internet Protocol
MIME = Multi-Purpose Internet Mail Extension

OSPF = Open Shortest Path First
RSVP = Resource ReSerVation Protocol
SMTP = Simple Mail Transfer Protocol
SNMP = Simple Network Management Protocol
TCP = Transmission Control Protocol
UDP = User Datagram Protocol

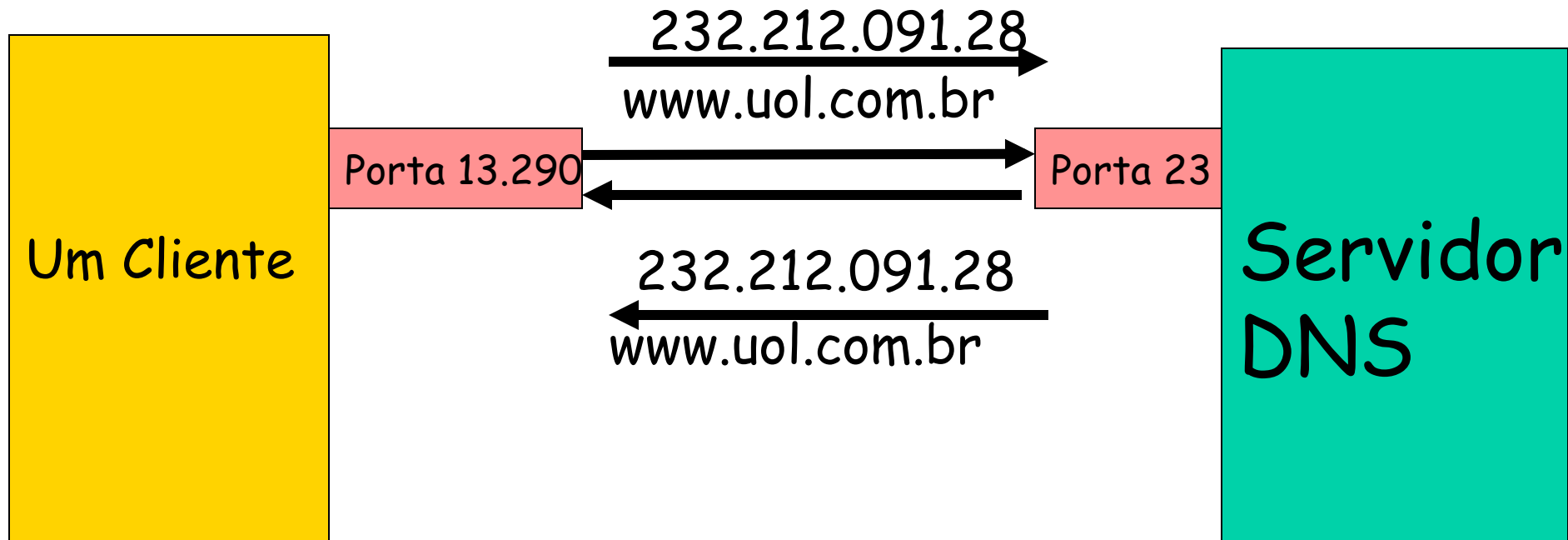
Comunicação num Host



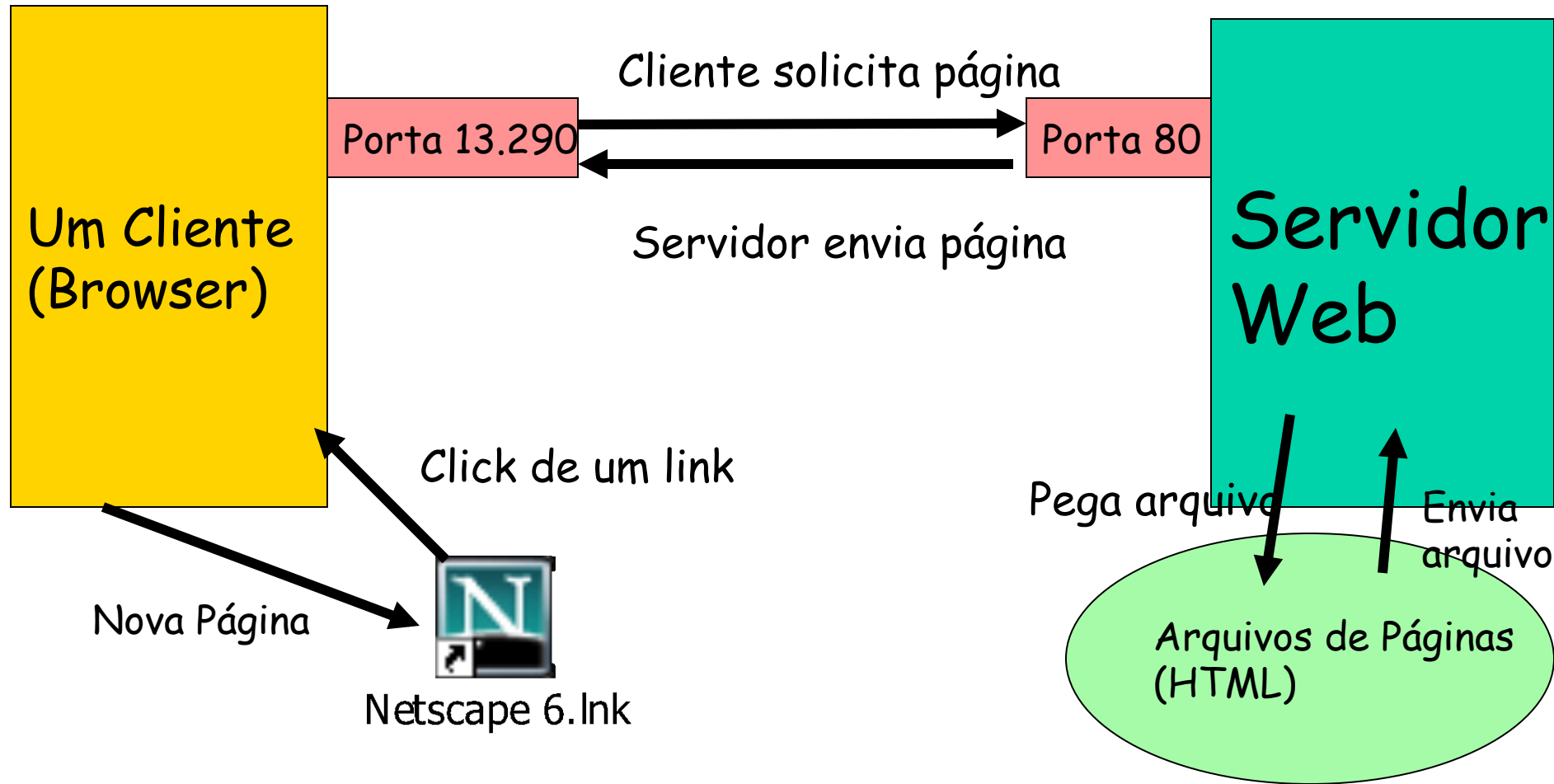
Protocolo da Camada de Aplicação

Protocolo	Porta	Protocolo de Transporte	Proposito
FTP	20 e 21	TCP	Transferência de arquivos
HTTP	80	TCP	Transferência de hipertextos
SMTP	25	TCP	Envio de e-mail para servidor de e-mail
POP3	110	TCP	Transferência de e-mails acumulados no servidor para um cliente
DNS	23 (?)	TCP ou UDP	Converter nome de máquina em número e vice-versa

Exemplo de Comunicação com um Servidor DNS



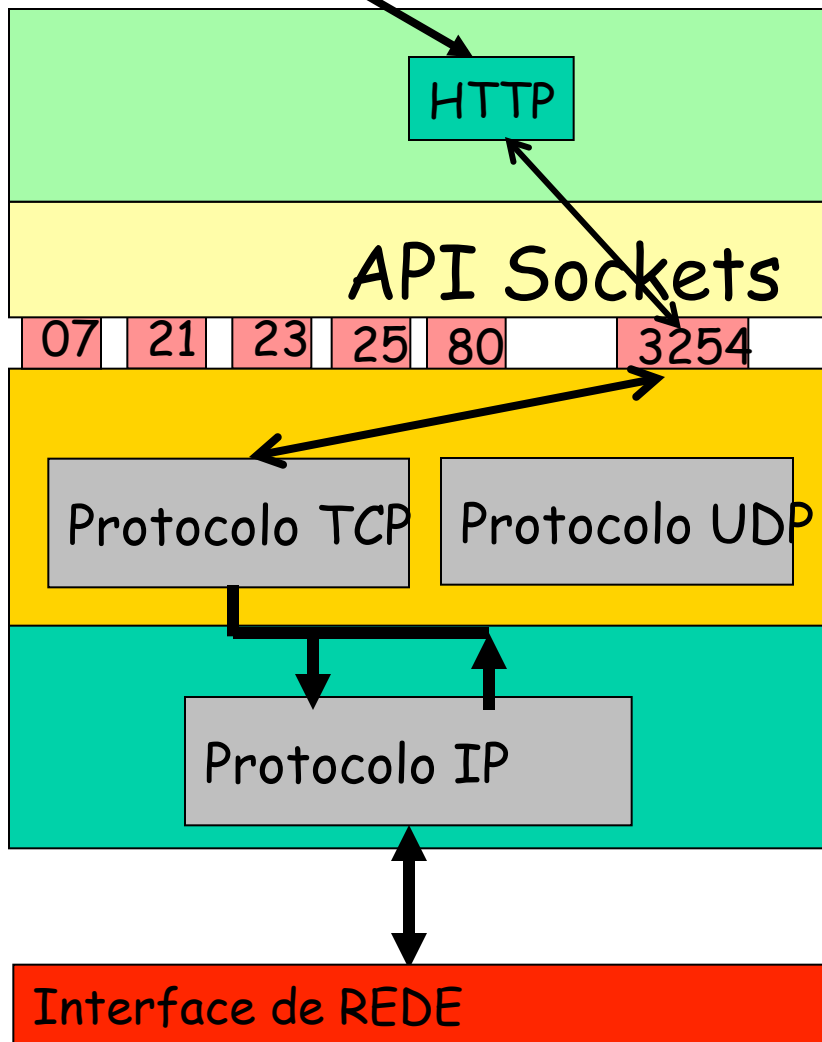
Exemplo de Comunicação com um Servidor WEB



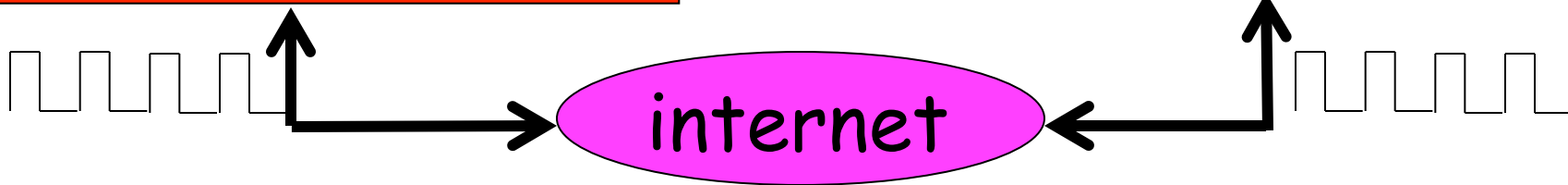
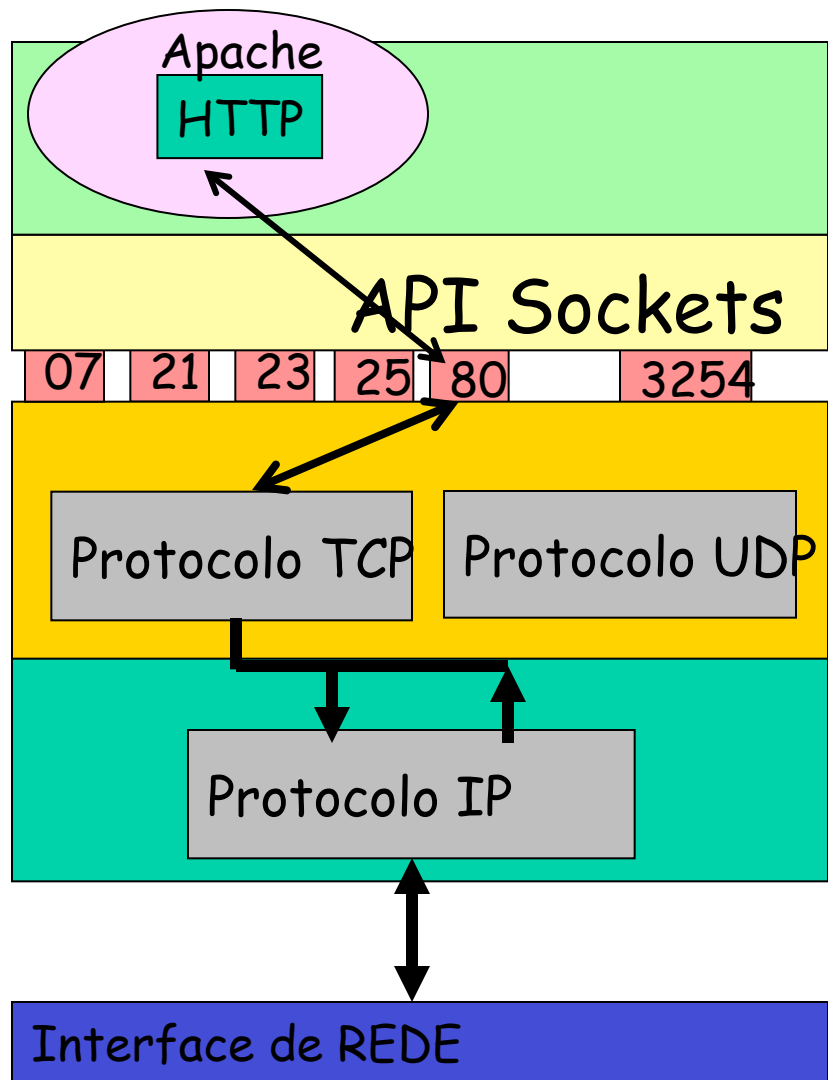


LADO CLIENTE

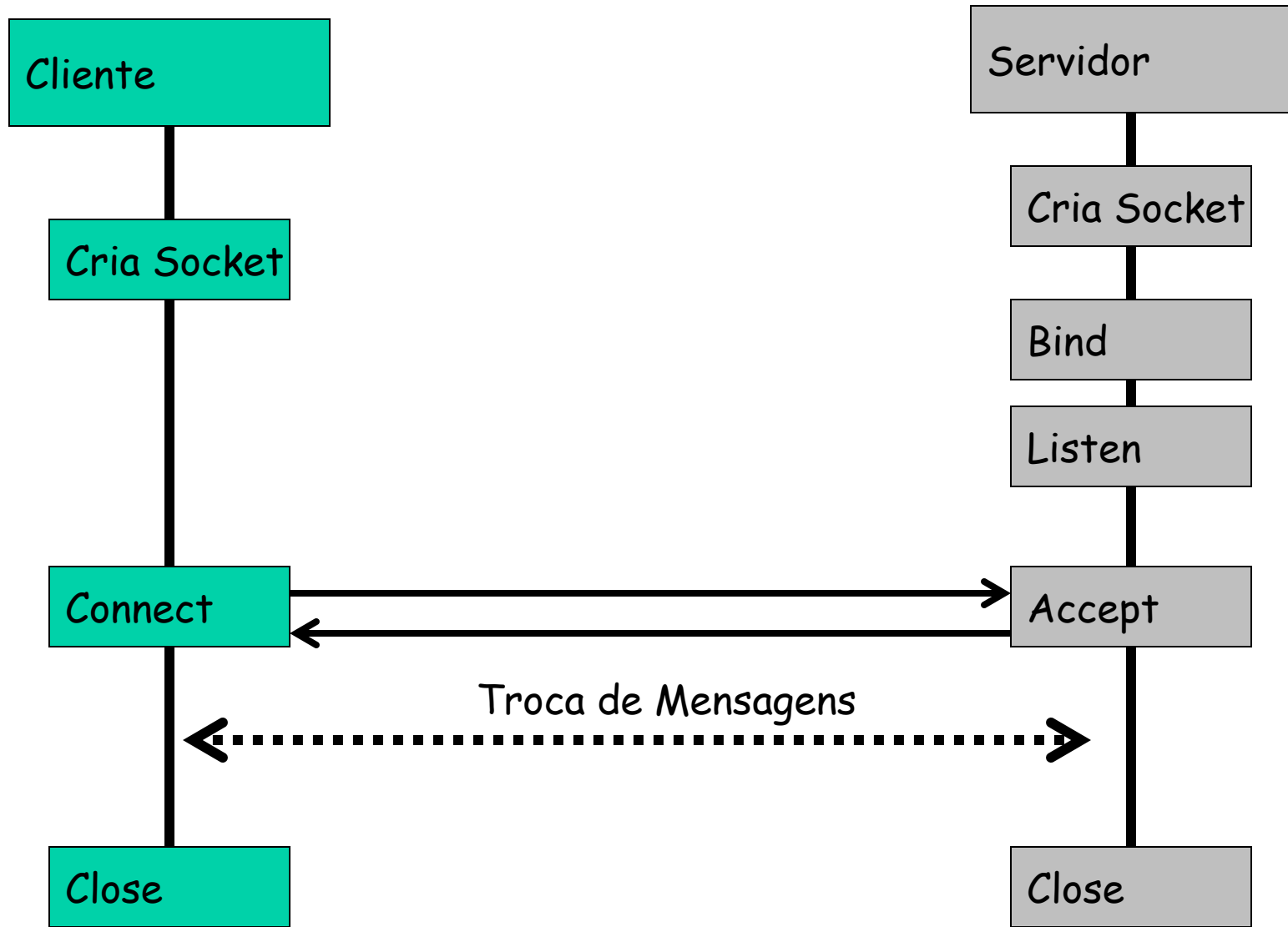
Netscape 6.lnk



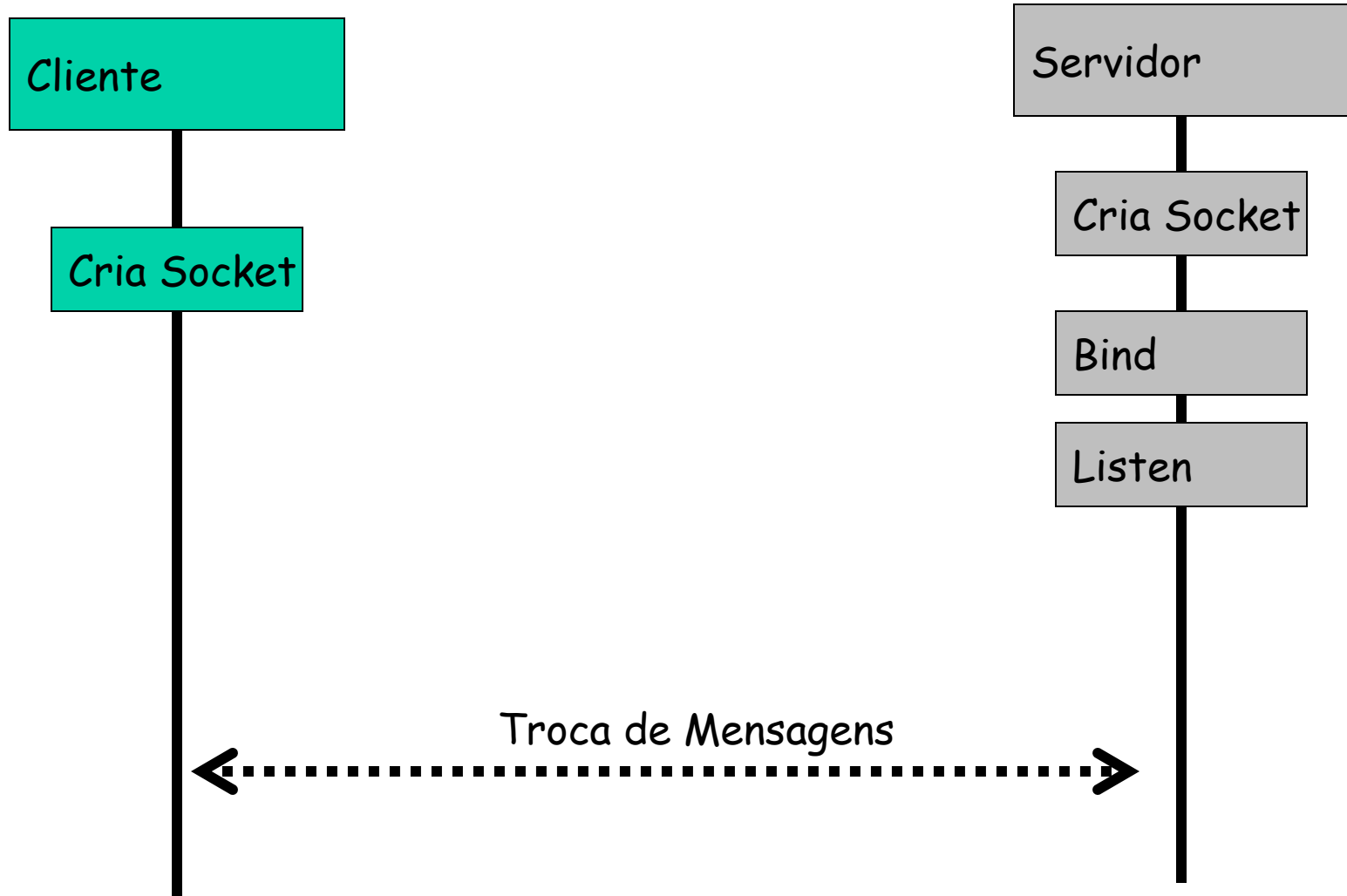
LADO SERVIDOR



Socket TCP



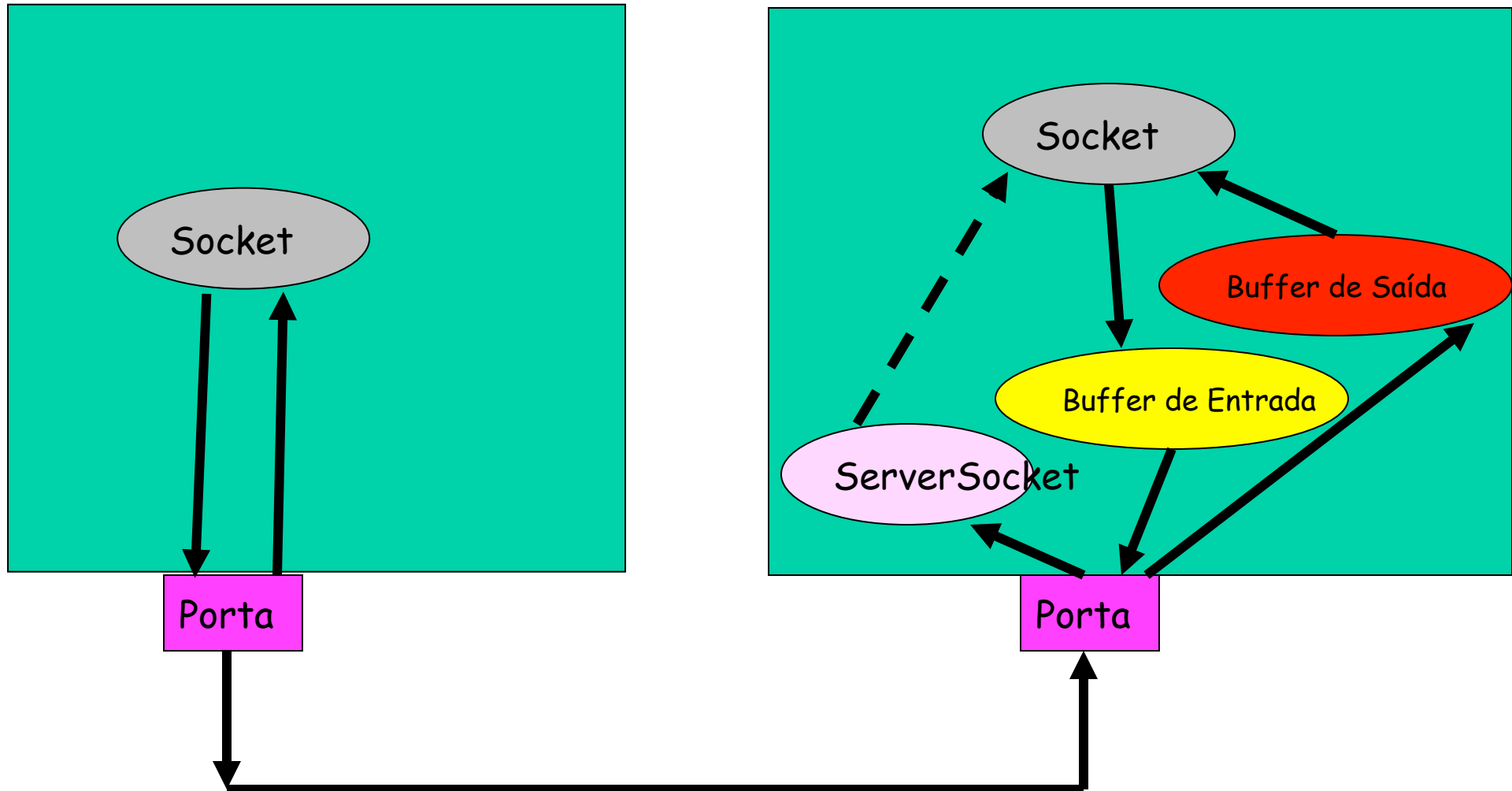
Socket UDP



Estrutura de Comunicação via Sockets

Cliente

Servidor



Ciclo de Vida de um Servidor

1. Um novo `ServerSocket` é criado.
2. Ele fica escutando uma porta: `accept()`.
3. Após aceitar uma conexão, `accept()` retorna um objeto do tipo `Socket`.
4. O objeto tipo `Socket` utiliza as funções `send()` e `receive()` para receber e enviar dados, respectivamente.
5. Cliente e Servidor fecham a conexão.
6. O Servidor (`ServerSocket`) retorna ao passo 1.

Comunicação Multicast

- ❑ Tipos de comunicação:
 - Unicast
 - Multicast
 - Broadcast

- ❑ Por que utilizar comunicação multicast?

Roteamento de Grupo

❑ Protocolo IGMP

❑ TTL - Time To Live

- Máximo número de roteadores que o datagrama poderá ser roteado
- $TTL = 0 \rightarrow$ Localhost
- $TTL = 1 \rightarrow$ Rede Local

❑ Necessidade de ser transparente para o programador da aplicação

- A aplicação deve enviar pacotes para o endereço multicast.
- TCP ou UDP ???

❑ Conceito de GRUPO:

- conjunto de hosts que compartilham um mesmo endereço multicast.
- todo dado enviado para um endereço multicast é recebido por todo os membros do grupo.
- membros se associam-se e desligam-se do grupo de forma autônoma.

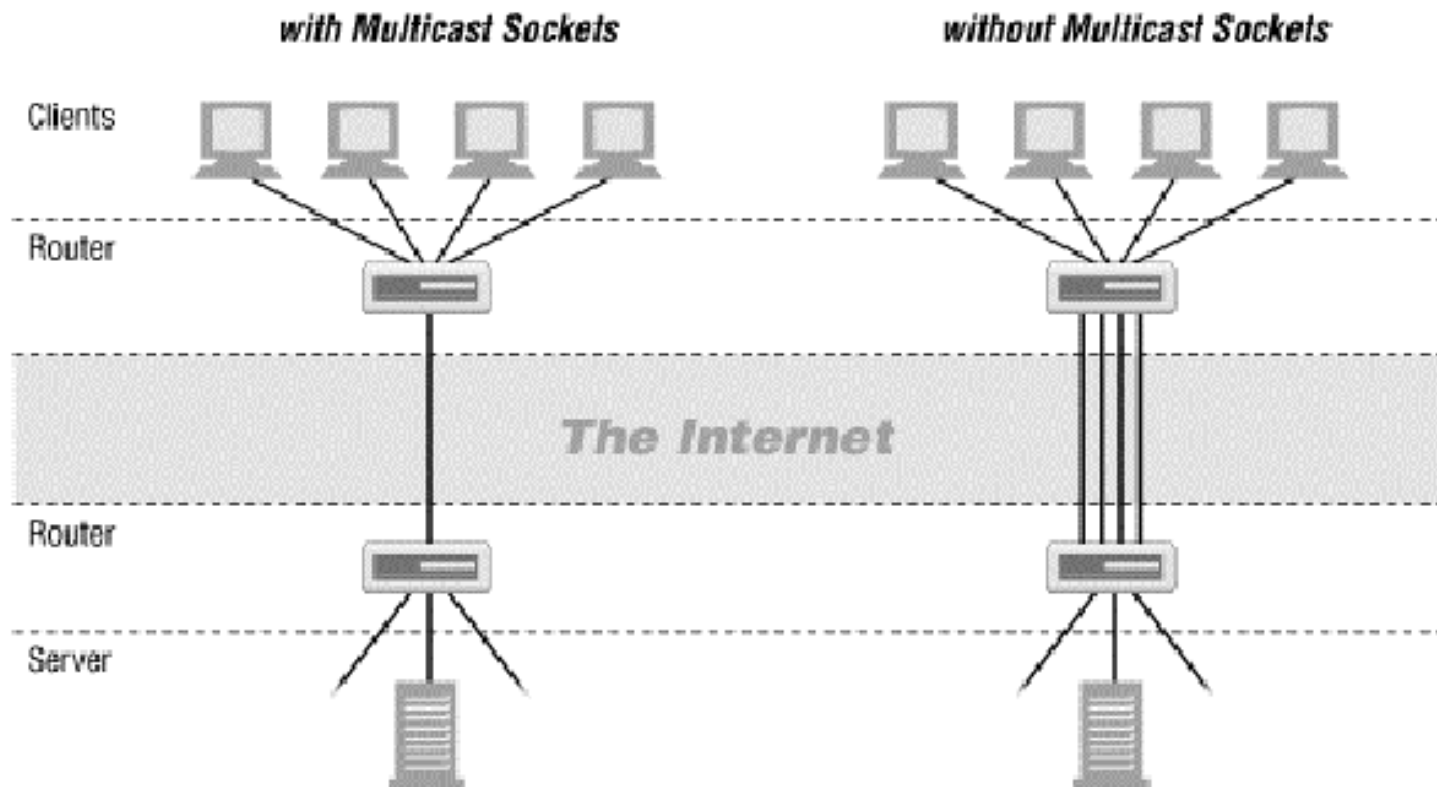
❑ Endereçamento Multicast

- Endereço do grupo
- Classe D: 224.0.0.0 - 239.255.255.255

Endereços Multicast

- ❑ 224.0.0.0 - Endereço reservado.
- ❑ 224.0.0.1 - Grupo de todos os sistemas que suportam multicast na rede local.
- ❑ 224.0.1.1 - Network Time Protocol.
- ❑ 224.0.1.32 - Versão multicast do traceroute.
- ❑ 224.2.0.0 a 224.2.255.255 - MBONE (Multicast Backbone on the Internet)

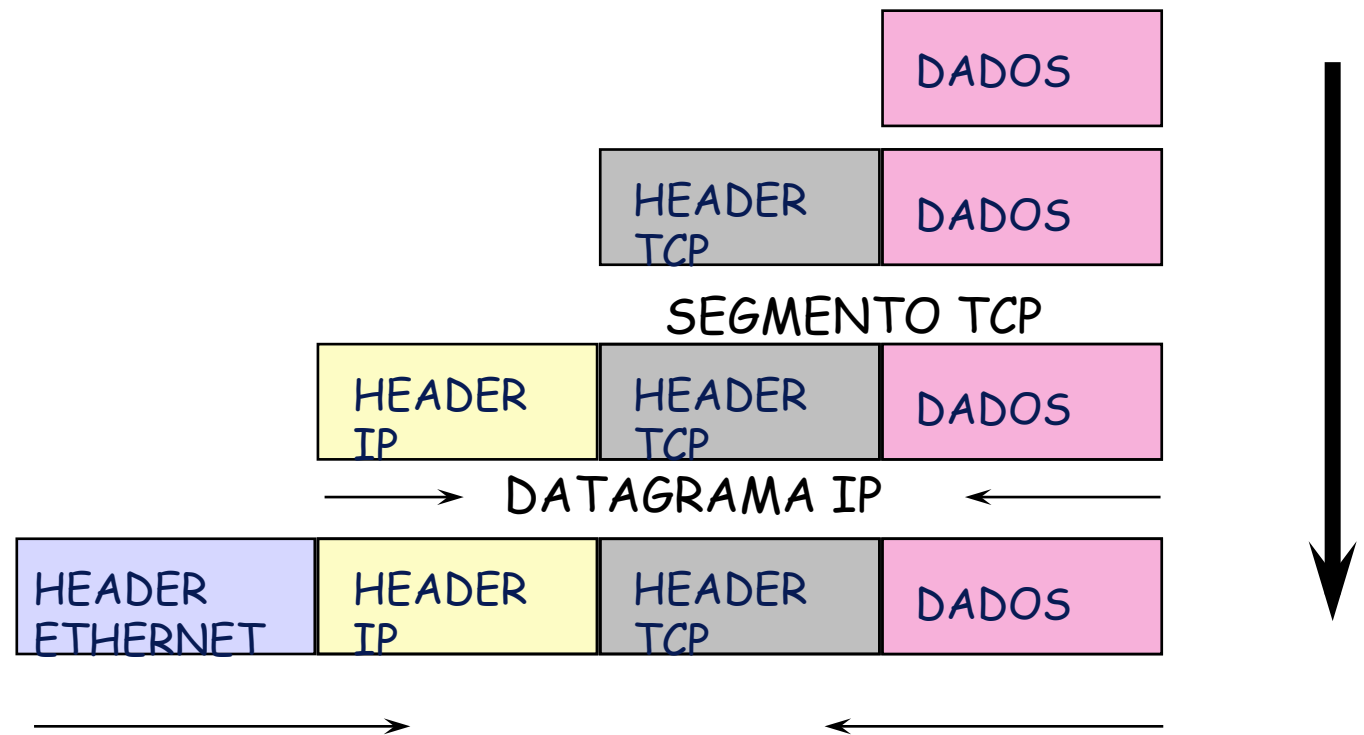
Sockets Multicast



Estrutura de uma comunicação multicast

1. Criação do Socket
2. Associação a um grupo multicast (se for receber dados)
3. Envio/Recebimento de dados para/de membros do grupo
4. Desligamento do grupomulticast
5. Fechamento do Socket

Overhead de comunicação

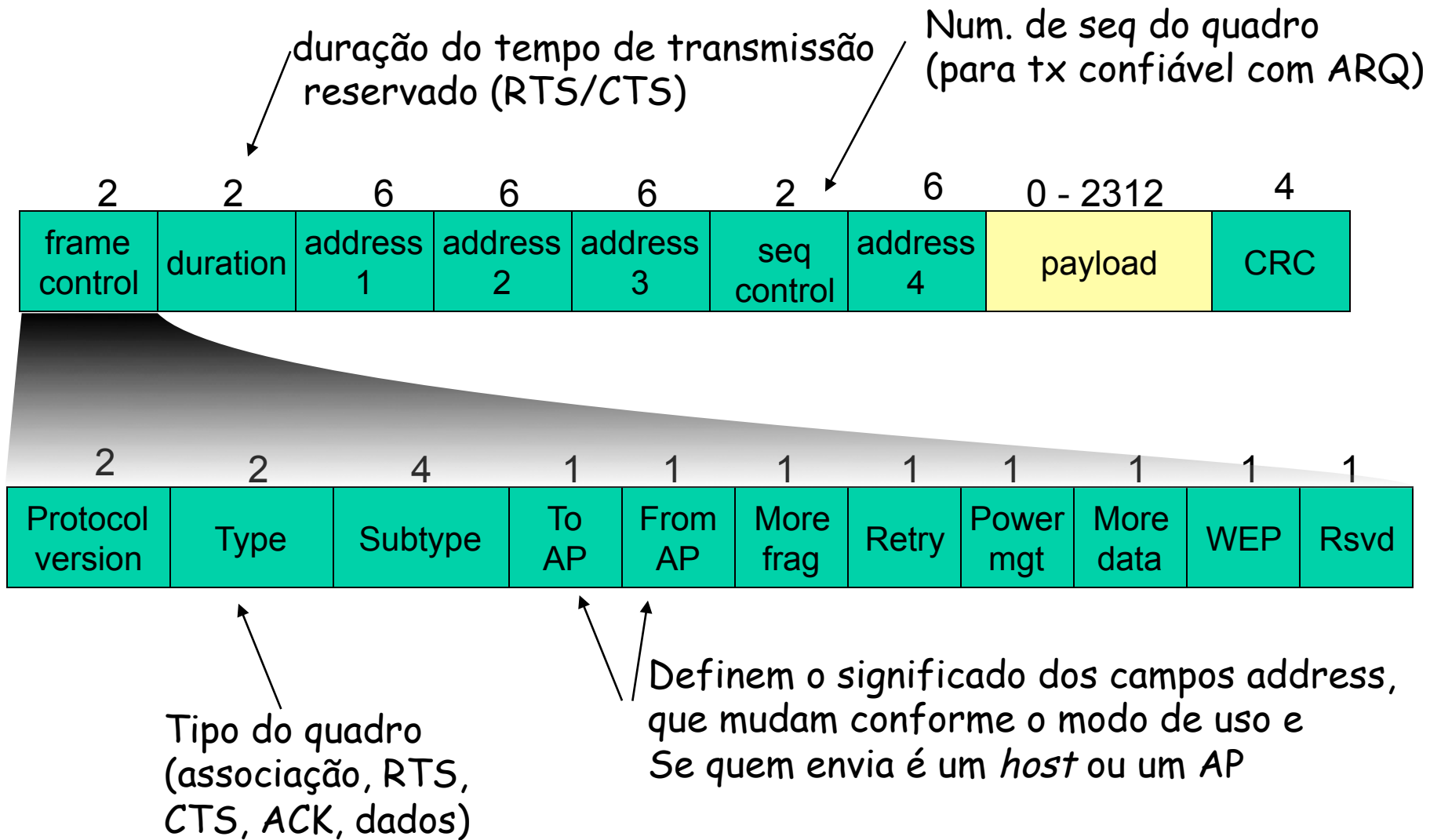


Quadro Ethernet

Preambolo	End. destino	End.Origem	Tipo de Quadro	Dados	CRC
8	6	6	2	46-1500	4

46

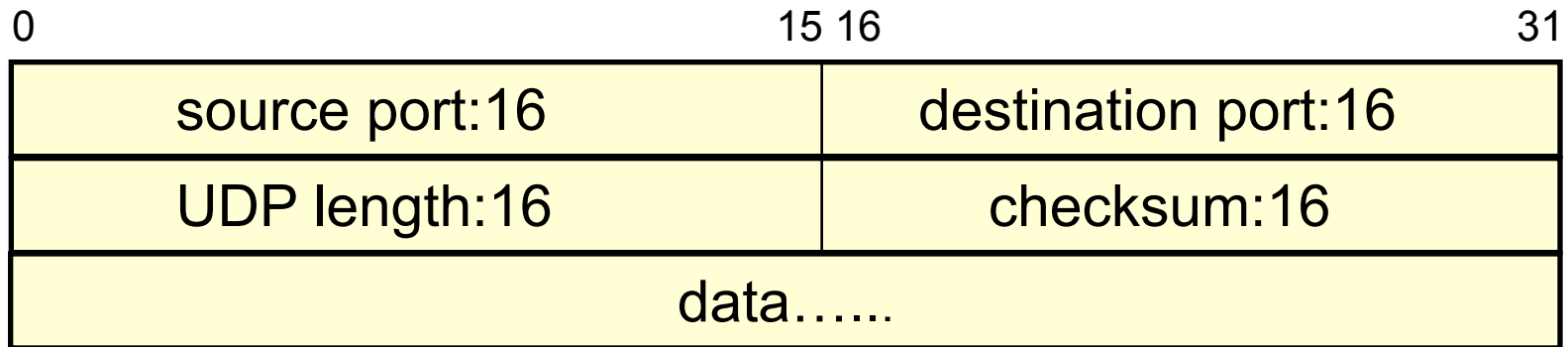
Quadro 802.11 (Wifi)



Overhead - Cabeçalho IP

0	4	8	16	19	24	31
VERS		HLEN	TYPE OF SERV.		TOTAL LENGTH	
IDENT			FLAGS		FRAGMENT OFF.	
TIME		PROTO		HEADER CHECKSUM		
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
OPTIONS					PADDING	
DATA						
...						

Overhead - Cabeçalho UDP



Overhead - Cabeçalho TCP

0	15 16			31
source port:16			destination port:16	
sequence number:32				
acknowledgment number:32				
data offs:4	resv:6	flag:6	window size:16	
checksum :16			urgent pointer:16	
options and padding				

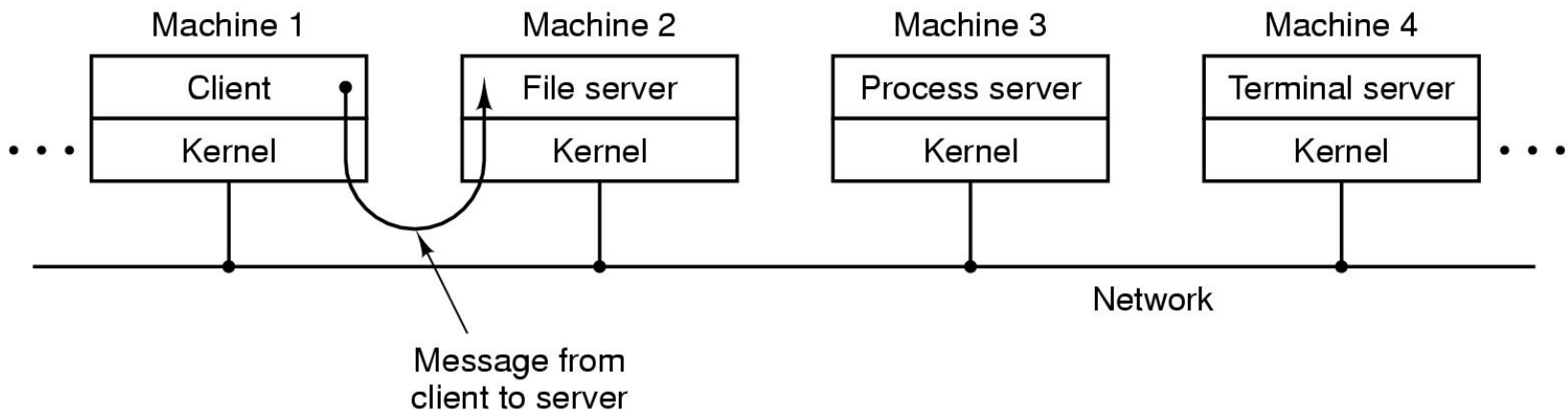
Real-Time Protocol (RTP-RTCP)

- ❑ Opera sobre o UDP.
- ❑ Possui baixo overhead de cabeçalho
 - 16 bytes de cabeçalho.

Sistemas Operacionais de Rede

❑ Paradigma Cliente-servidor

- Distribuir os serviços do SO entre diversos computadores
- Não há transparência de localização



Comunicação de tempo-Real

❑ Onde pode haver atraso e erros?

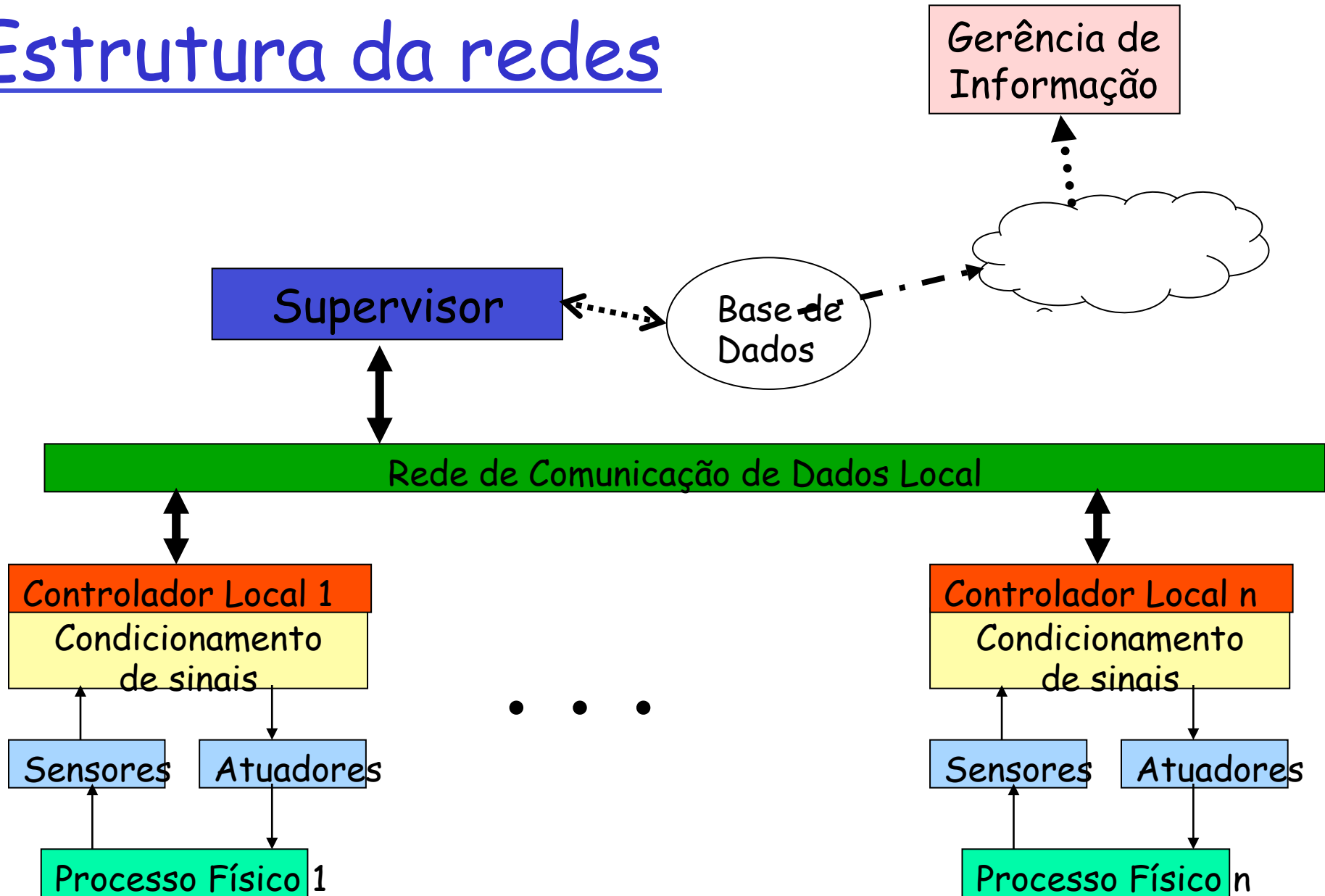
○ No nó

- Hardware, SO, Aplicação.
- Sensores e atuadores

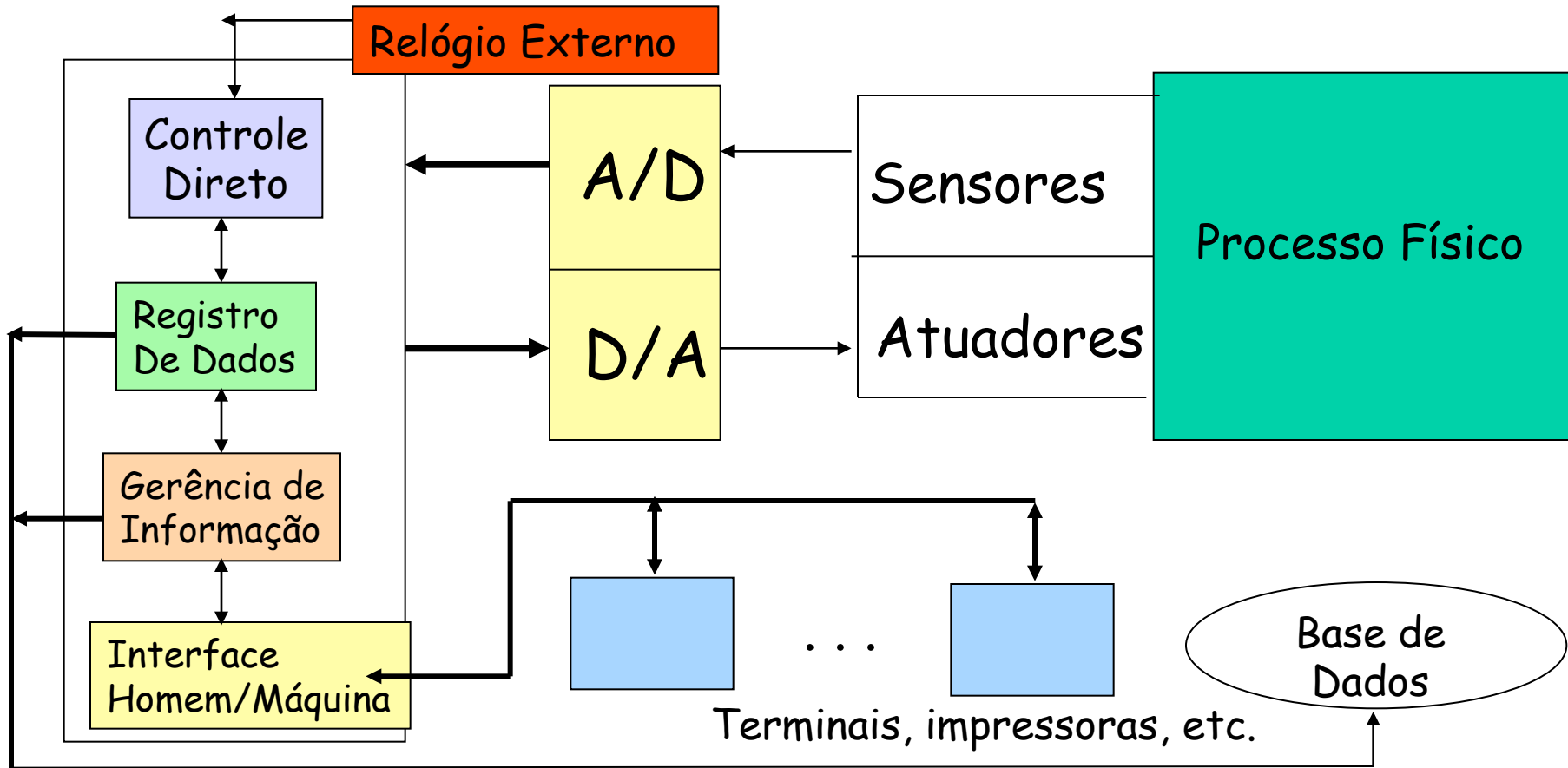
○ Na comunicação:

- Meio físico
- Roteadores
- Interfaces de redes

Estrutura da redes



O Problema de Controle



Mecanismos de Comunicação

- ❑ Origem-Destino
 - Mestre-escravo
 - Ponto-a-ponto
 - Cliente-servidor
- ❑ Produtor-Consumidor
 - Publisher-Subscriber

Modelos de Comunicação em Redes

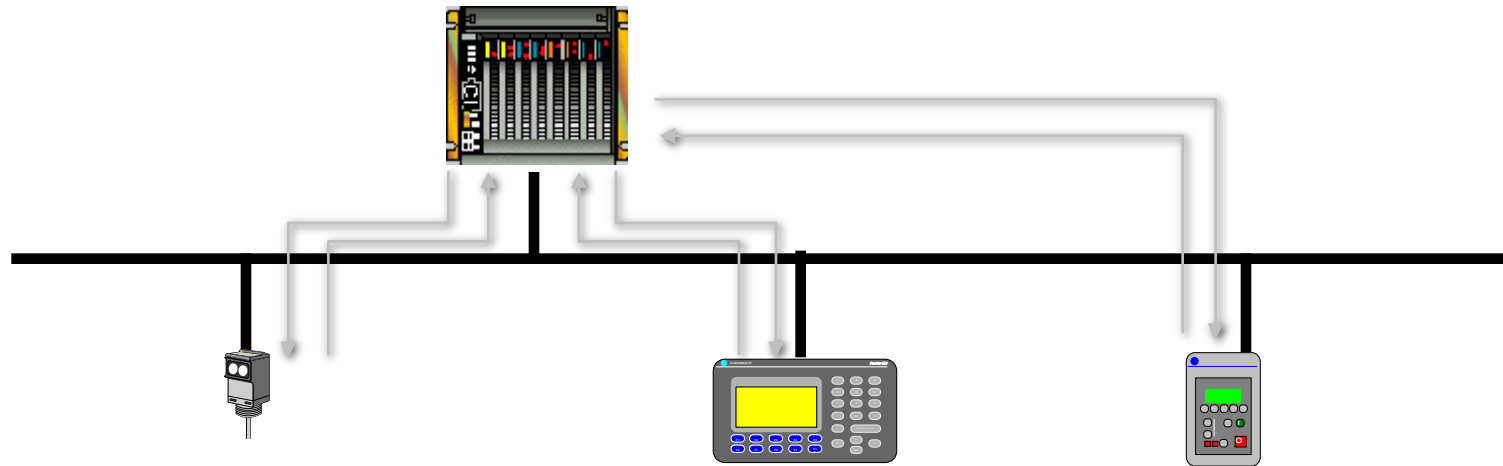


Origem/Destino (ponto a ponto)



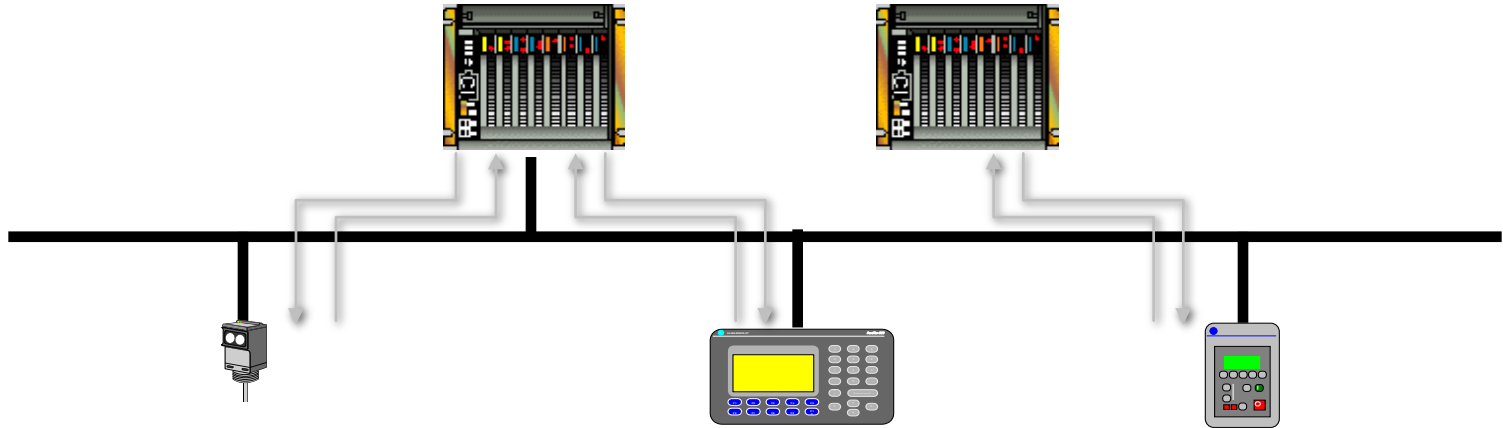
Produtor/Consumidor
(comunicação multicast)

Mestre/Escravo



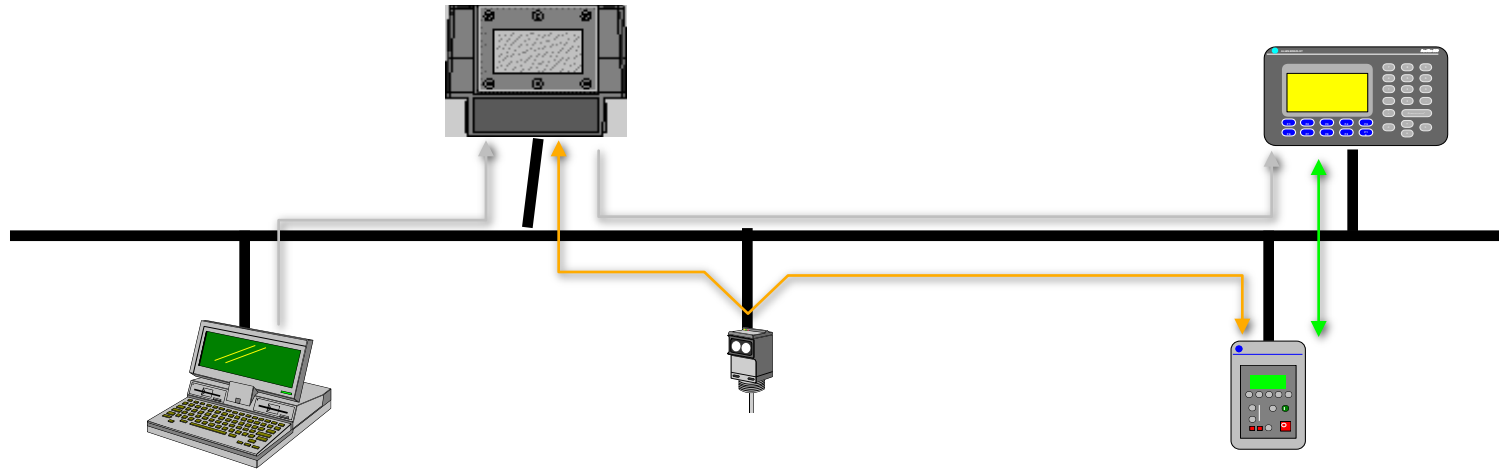
- ❑ Um Mestre, múltiplos escravos
- ❑ Dispositivos escravos trocam dados apenas com o Mestre
- ❑ Dados de E/S (Mensagens Implícitas) são predominantes neste tipo de comunicação

Multimestre



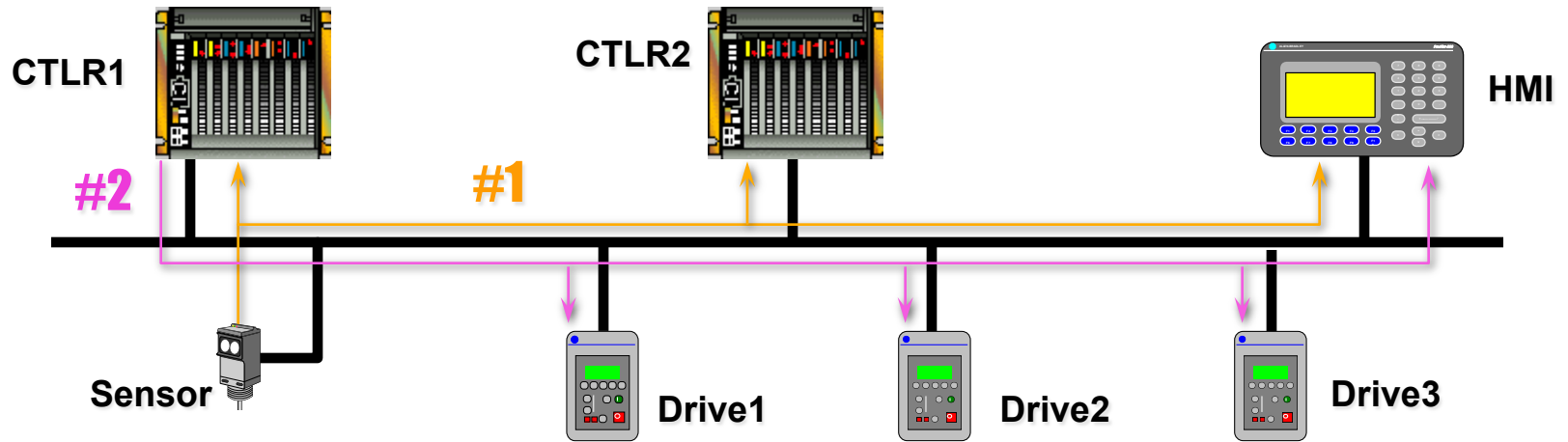
- ❑ Mais de um mestre
- ❑ Cada mestre tem seu próprio conjunto de escravos
- ❑ Dispositivos escravos apenas trocam dados com seus mestres
- ❑ Dados de E/S (Mensagens Implícitas) também predominam neste tipo de comunicação

"Ponto a Ponto"



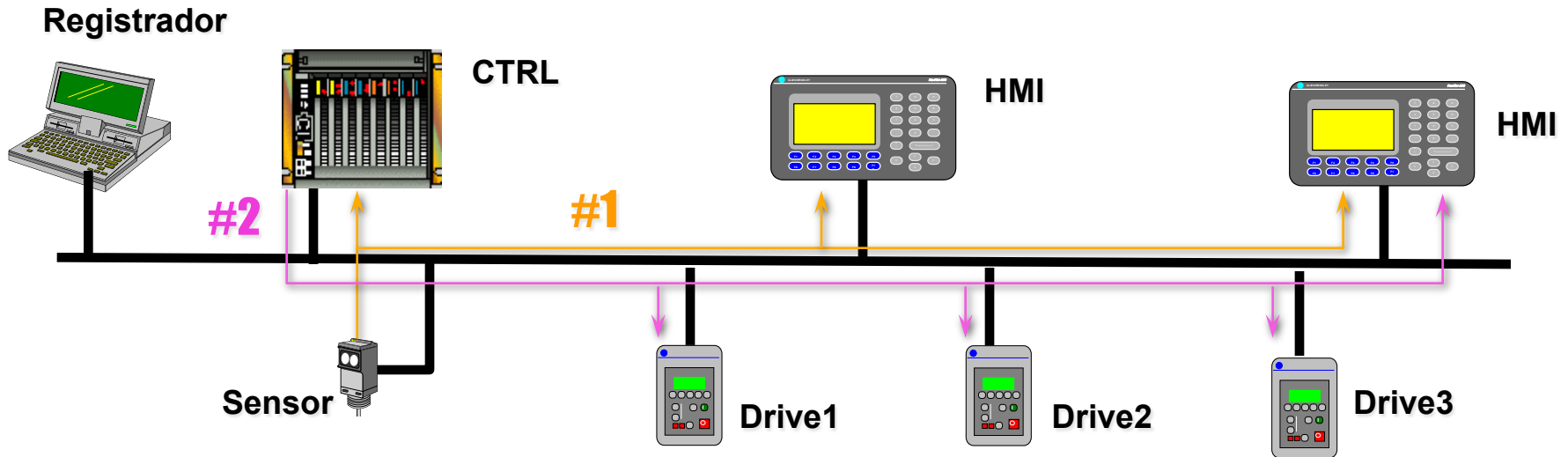
- ❑ Dispositivos enquadrados numa mesma categoria livres para tomar iniciativa de comunicação
- ❑ Dispositivos podem trocar dados com mais de um dispositivo ou múltiplas trocas com um mesmo dispositivo
- ❑ Mensagens Explícitas predominam neste tipo de comunicação

Produtor/Consumidor



- **Mensagem #1**
 - referência de posição do sensor transmitida em multicast aos CTRL1, 2 e IHM
- **Mensagem #2**
 - comando de velocidade do CTRL1 transmitido simultaneamente aos 3 drives e IHM

Publisher/Subscriber



- **Variação do produtor-consumidor**

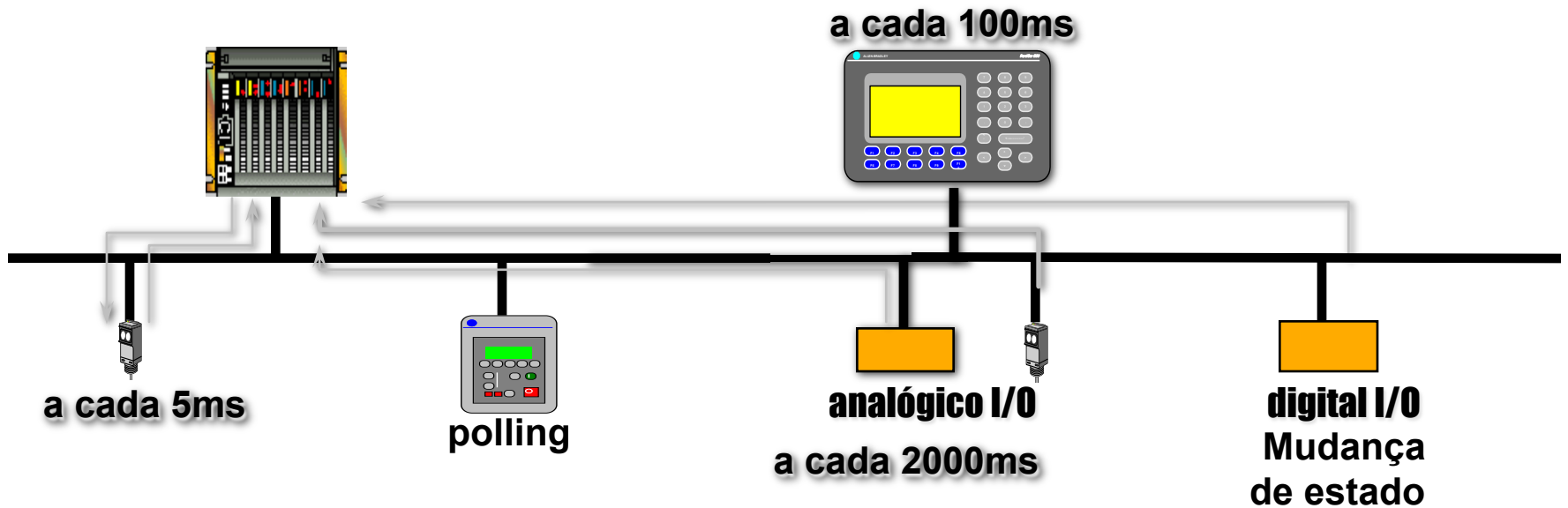
- Há um entidade que armazena os tipos de mensagens que são publicadas.
- Para se quer publicar um tipo de mensagem, há a necessidade de registrá-la nessa entidade.
- Quando se quer receber um tipo de mensagem, há a necessidade de consultar/registrar-se nessa entidade.
- A implementação pode ser centralizada (refletor) ou distribuída.

Métodos para troca de dados

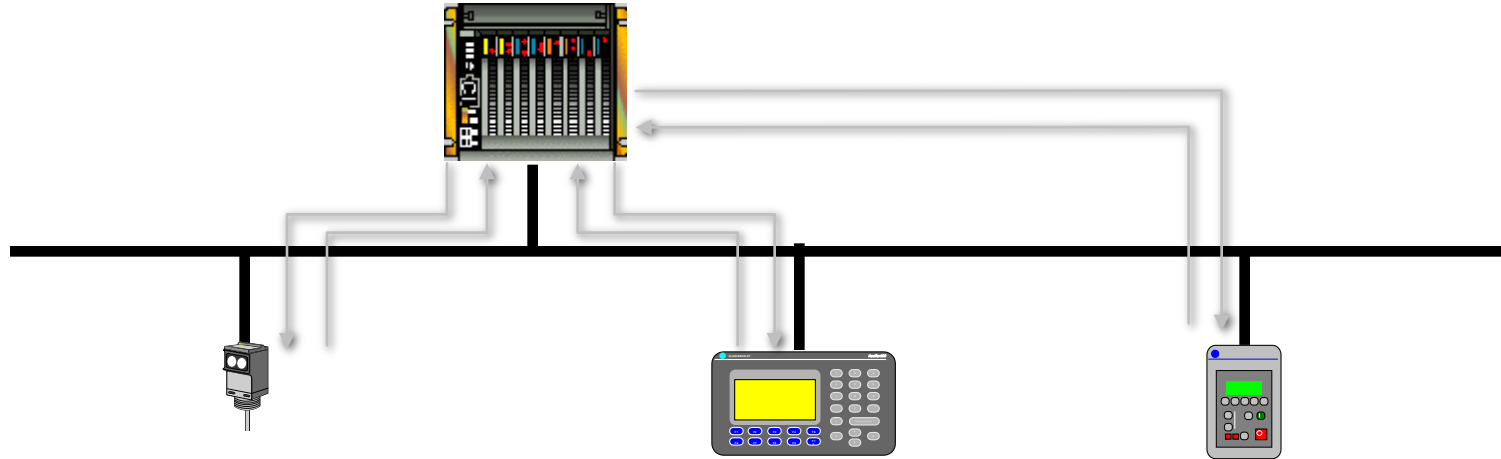
Polling

Cíclico

Mudança de Estado

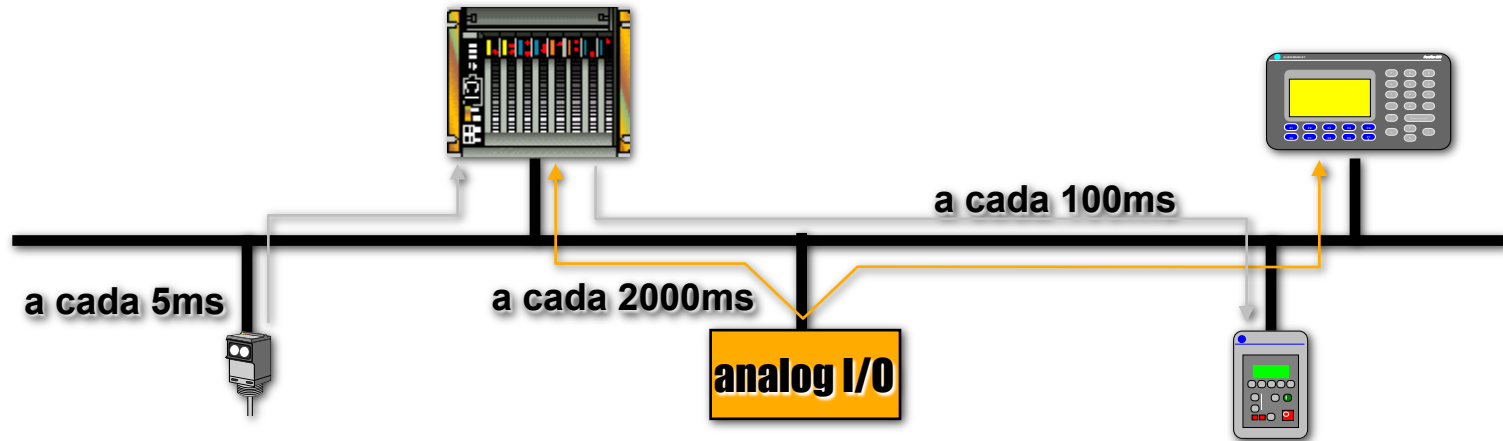


Métodos de troca de dados: “Polling”



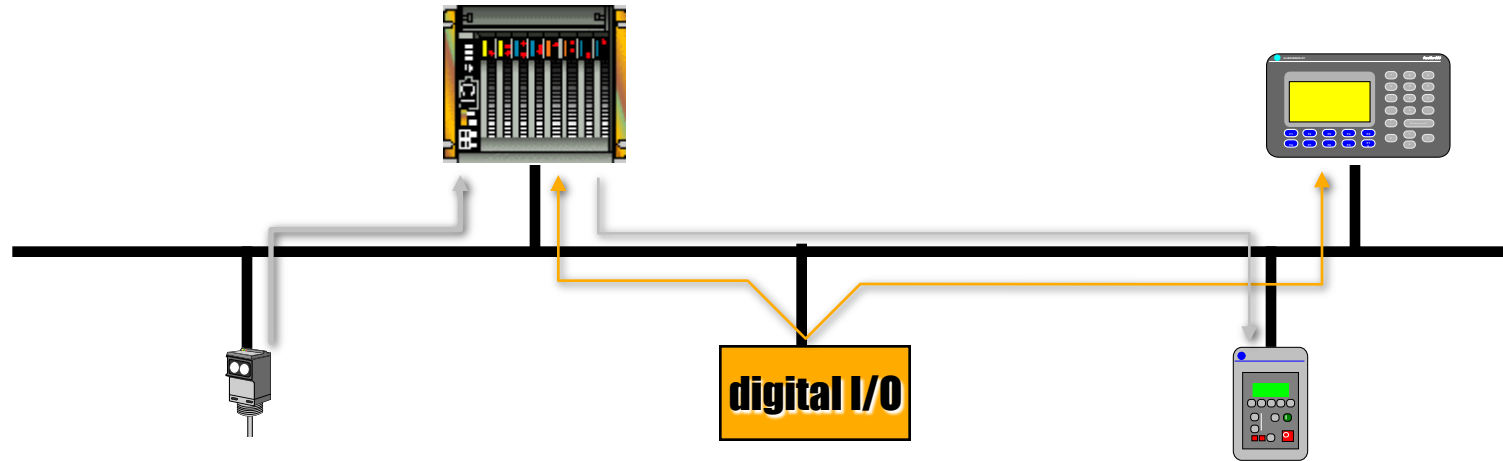
- ❑ Quando os dispositivos recebem dados, imediatamente os enviam
- ❑ Compatível com sistemas Mestre/Escravo & Multimestre
 - Normalmente não é utilizado com “ponto a ponto”
- ❑ Desenvolvido sobre origem/destino, mestre/escravo
- ❑ Inerentemente ponto a ponto, não há multicast

Métodos de troca de dados: Cíclica



- ❑ Dispositivos produzem dados a uma taxa configurada pelo usuário
- ❑ Transferência cíclica é eficiente porque:
 - os dados são transferidos numa taxa adequada ao dispositivo/aplicação
 - recursos podem ser preservados p/ dispositivos com alta variação
 - melhor determinismo
- ❑ Compatível com Mestre/Escravo, Multimestre, "ponto-a-ponto" e Multicast

Métodos de troca de dados: Mudança de estado



- ❑ Dispositivos produzem dados apenas quando têm seu estado alterado
 - Sinal em segundo plano transmitido ciclicamente para confirmar que o dispositivo está ok.
- ❑ Mudança de estado é eficiente porque:
 - reduz significativamente o tráfego da rede
 - recursos não são desperdiçados processando-se dados antigos

Redes CAN

- ❑ O barramento CAN (*Controller Area Network*) foi desenvolvido pela empresa Alemã BOSCH e disponibilizado em meados dos anos 80.
- ❑ Usa mecanismo de Produtor/Consumidor → orientada a mensagens.
- ❑ **Há necessidade de sincronização entre os seus nós.**
- ❑ Desenvolvido inicialmente para área automotiva.
- ❑ Devido à sua comprovada confiabilidade e robustez, também está sendo adotada em outras aplicações de tempo real.
 - É possível implementar mecanismos de priorização de mensagens

Áreas de Aplicação do CAN

- ❑ **Veículos (marítimo, aéreo, terrestre)** - carros de passeio, off-road, trens, sistema de semáforo (trens e carros), eletrônica marítima, máquinas agrícolas, helicópteros, transporte público.
- ❑ **Sistema de Controle Industrial** - controle de planta industriais de pequeno e médio porte, de maquinário, robôs, sistema de supervisão.
- ❑ **Automação Predial** - controle de elevadores, ar condicionado, iluminação.
- ❑ **Aplicações Específicas** - sistemas médicos, telescópios, simuladores de vôo, satélites artificiais, entre outros.

Características do CAN

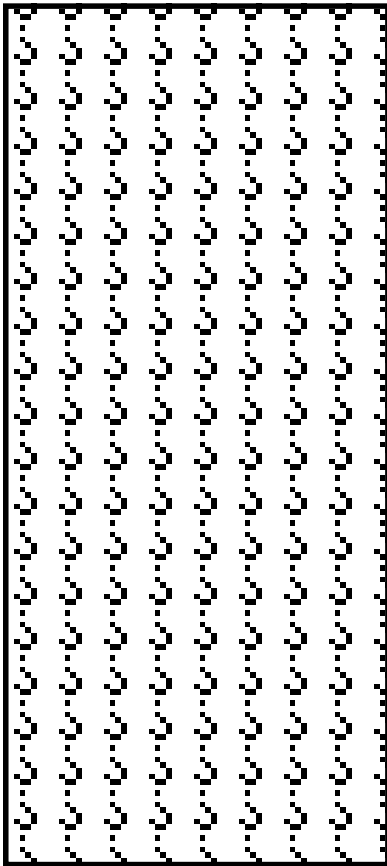
- ❑ Protocolo Digital e Comunicação Serial Síncrono
- ❑ Conceitos baseados na técnica **CSMA/CR** (*Carrier Sense Multiple Access /Collision Resolution*)
- ❑ Priorização de Mensagens
- ❑ Grande Flexibilidade de Configuração
- ❑ Recepção Multicast
- ❑ Garantia de Consistência dos Dados
- ❑ Detecção/Sinalização de erros
- ❑ Retransmissão Automática de Mensagens Corrompidas

Camadas do CAN

- ❑ O CAN foi dividido em duas camadas, obedecendo o modelo OSI/ISO:
 - *Data Link Layer*
 - *Logical Link Control (LLC)*
 - *Medium Access Control (MAC)*
 - *Physical Layer*

Camadas do Protocolo CAN

Modelo OSI/ISO

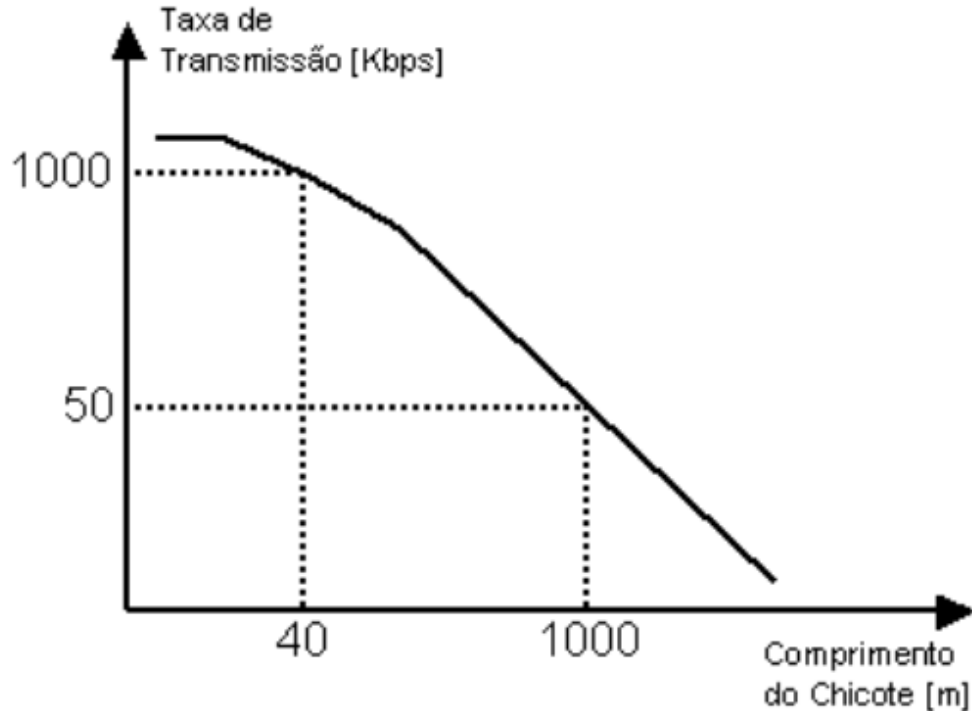


- ❑ **Camada Física - Physical Layer**
 - Codificação / Decodificação dos Bits
 - Temporização dos Bits
 - Sincronização
- ❑ **Camada de Enlace**
 - Controle de acesso ao meio via CSMA-CR
 - Controle de erro (CRC)
- ❑ **Camadas Superiores**
 - Não implementada
 - CAN Open

Camadas do Protocolo CAN

Camada Física - Physical Layer

Velocidade de Transmissão - Até 1Mbps



Taxa Kbit/s	Distância máxima (m)
1000	40
500	130
250	270
125	530
100	620
50	1300
20	3300
10	6700
5	10000

Taxa de transmissão X distância para o barramento CAN

Camadas do Protocolo CAN

Camada Física - Physical Layer

❑ ISO11898

- Alta Velocidade de transmissão de dados - 125 Kbps a 1 Mbps

❑ ISO11519-2

- Baixa Velocidade de Transmissão de dados - 10 Kbps a 125 Kbps

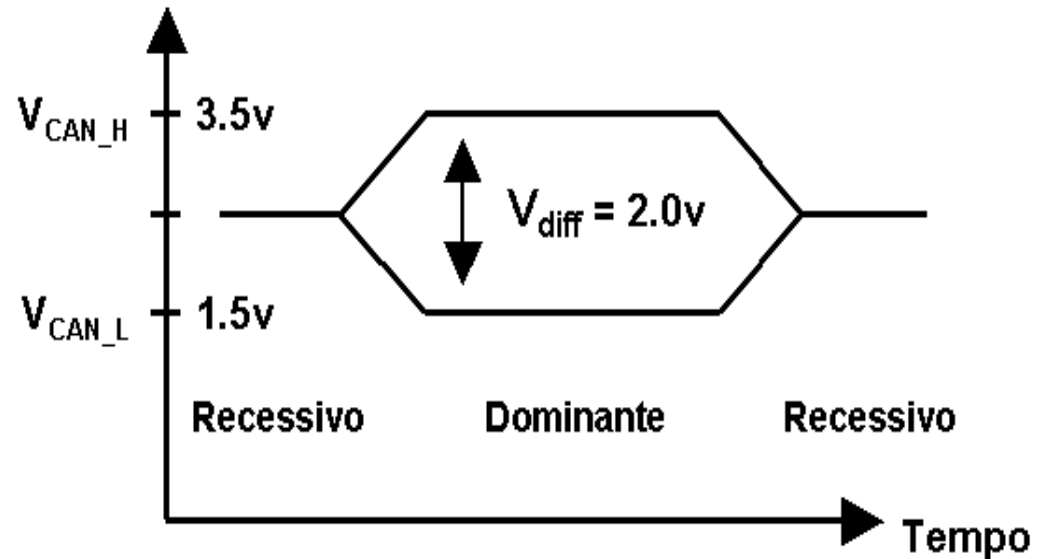
Camadas do Protocolo CAN

Camada Física - Physical Layer

Meio de Transmissão Fios Elétricos

Formas de Constituição de um barramento CAN:

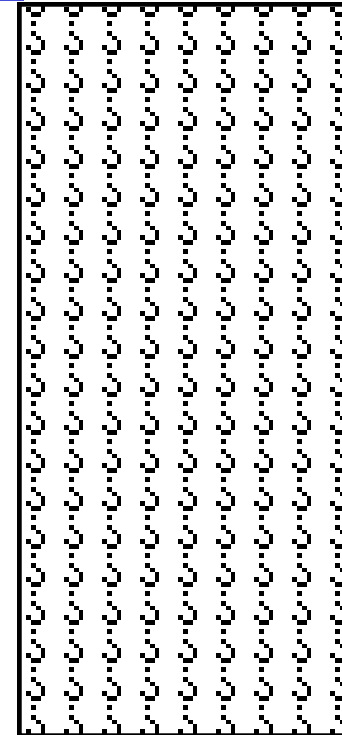
- ❑ 1 Fio - Fio de dados
- ❑ 2 Fios - CAN_H, CAN_L
- ❑ 4 Fios
 - CAN_H
 - CAN_L
 - Vcc
 - GND



Camadas do Protocolo CAN

Modelo OSI/ISO

- **Camada de Enlace - Data Link Layer**
 - **Logical Link Control (LLC)**
 - Recepção
 - Filtragem
 - Notificação de Overload
 - Gerenciamento de Recuperação
 - **Medium Access Control (MAC)**
 - Encapsulamento/ Desencapsulamento dos dados
 - Codificação dos Quadros
 - Gerenciamento de Acesso ao meio
 - Detecção e sinalização de erros
 - Reconhecimento
 - Serialização / Deserialização



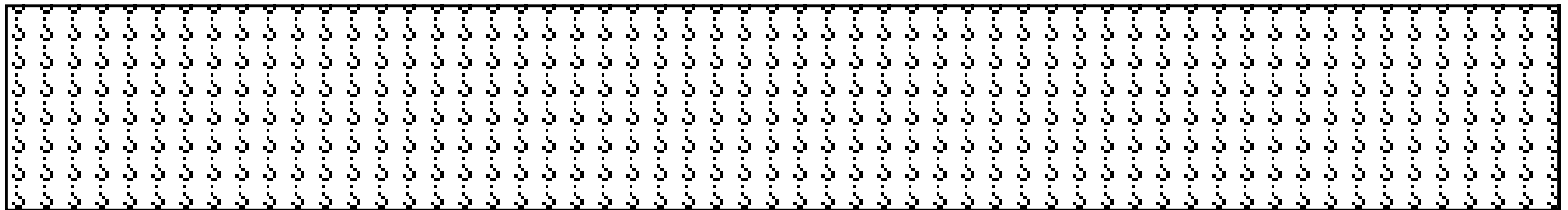
Quadros CAN

- ❑ O barramento CAN utiliza 4 tipos de quadros (frames) para controlar a transferência de mensagens
 - Quadro de Dados (Data Frame)
 - Quadro Remoto (Remote Frame)
 - Quadro de Erro (Error Frame)
 - Quadro de Sobrecarga (Overload Frame)

Mensagens do CAN

Formato das Mensagens - Tipos de Quadros Camada de Enlace

- ❑ Quadro de Dados
 - Composto por 7 (sete) diferentes campos de bits



Mensagens do CAN

Formato das Mensagens - Tipos de Quadros Camada de Enlace

❑ Quadro Padrão - CAN 2.0A

- Identificador de 11 bits

- É possível ter até 2048 mensagens em uma rede

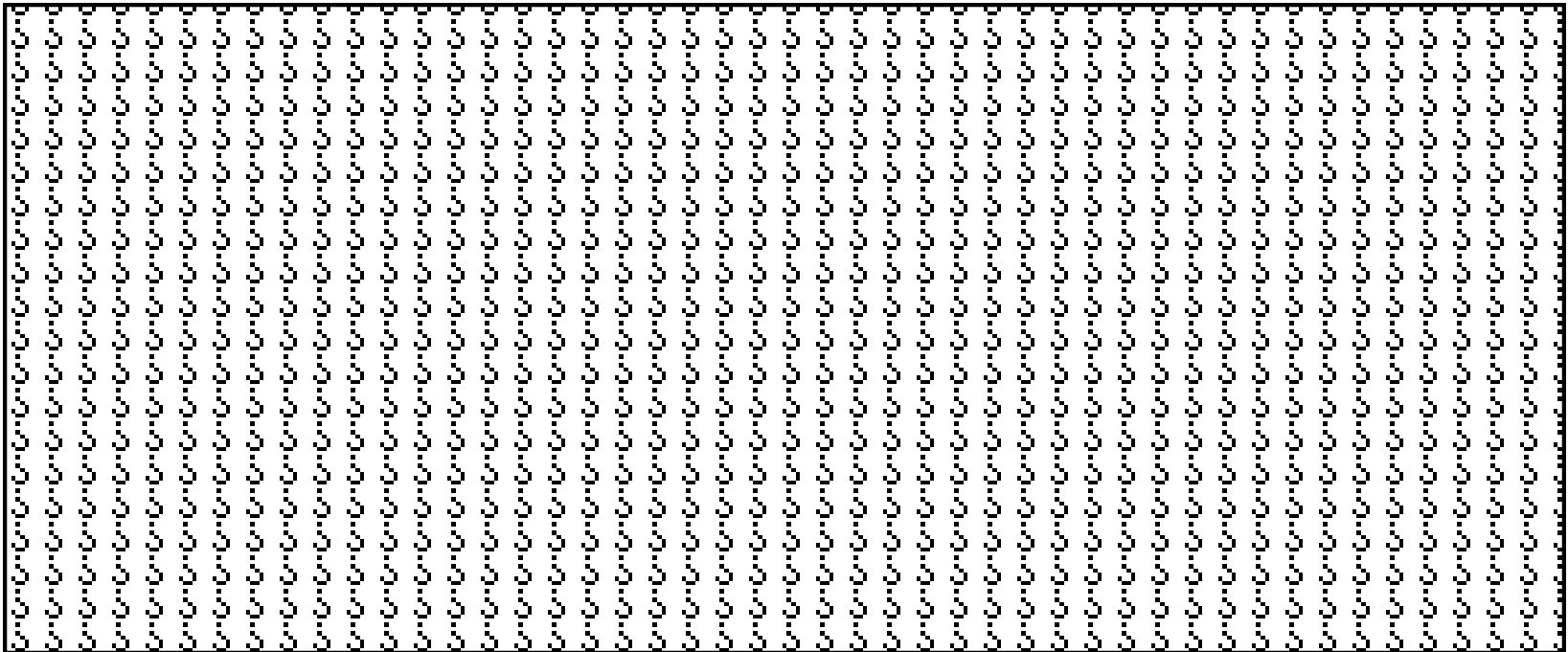
❑ Quadro Estendido - CAN 2.0B

- Identificador de 29 bits

- É possível ter até 537 milhões de mensagens em uma rede

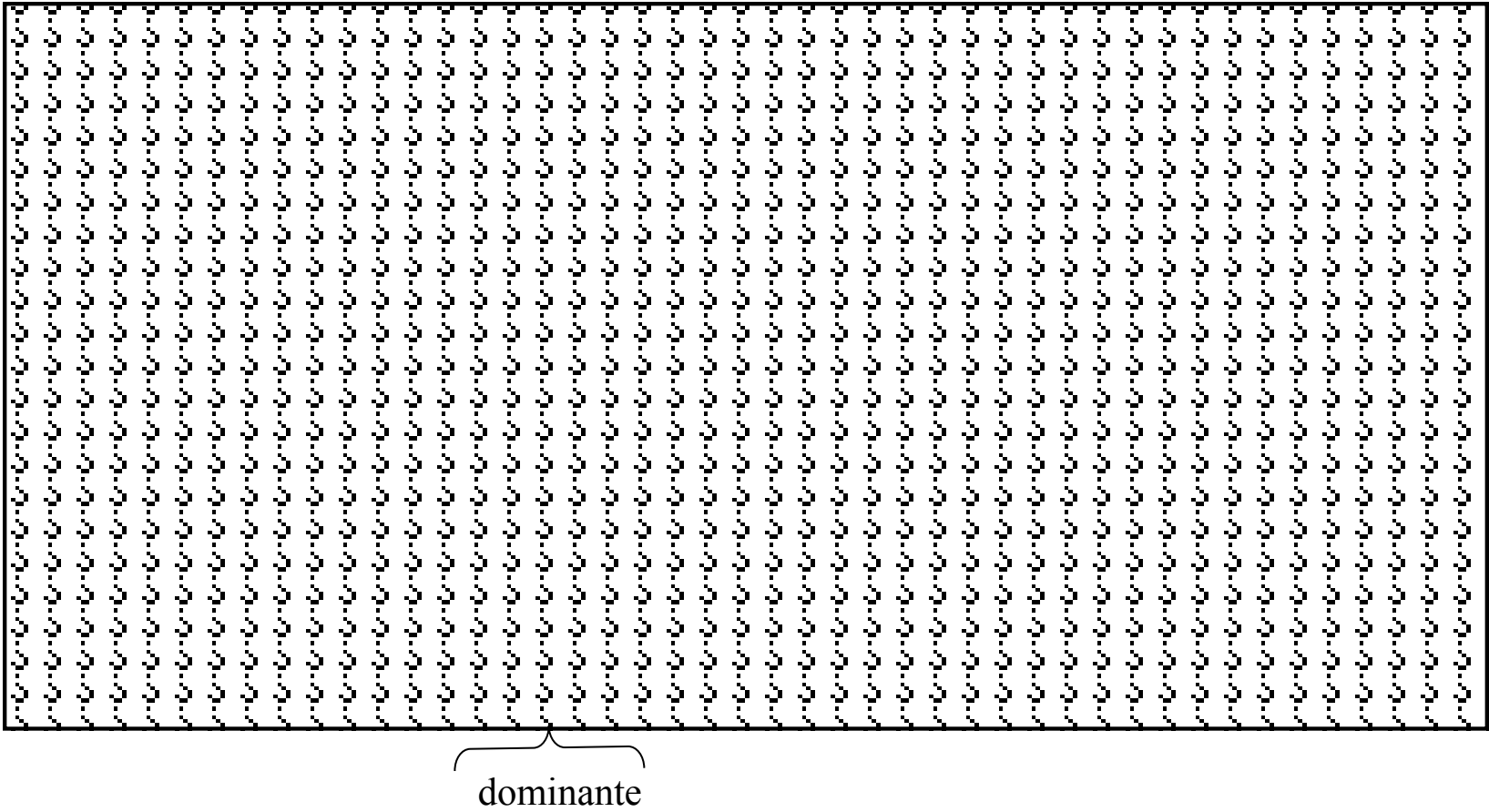
Mensagens do CAN

Formato das Mensagens - Campo de Arbitragem

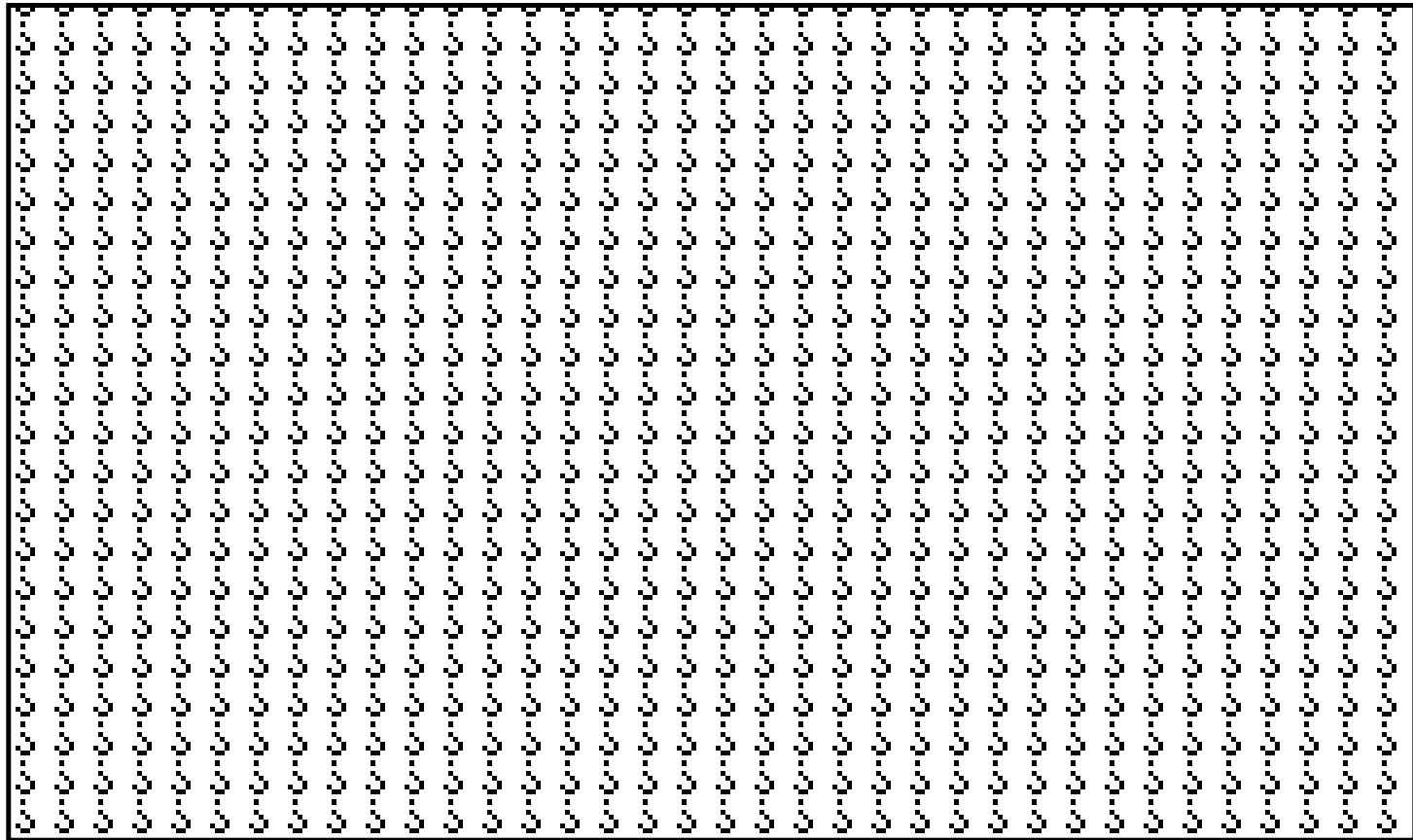


Mensagens do CAN

Formato das Mensagens - Campo de Controle



Codificação de Tamanho para o Campo de Dados CAN

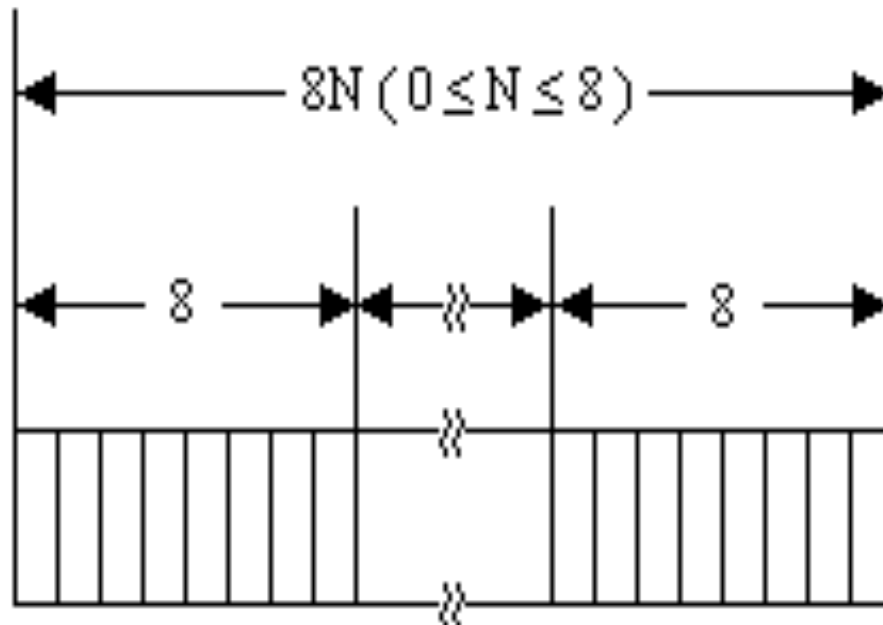


Mensagens do CAN

Formato das Mensagens

❑ Campo de Dados

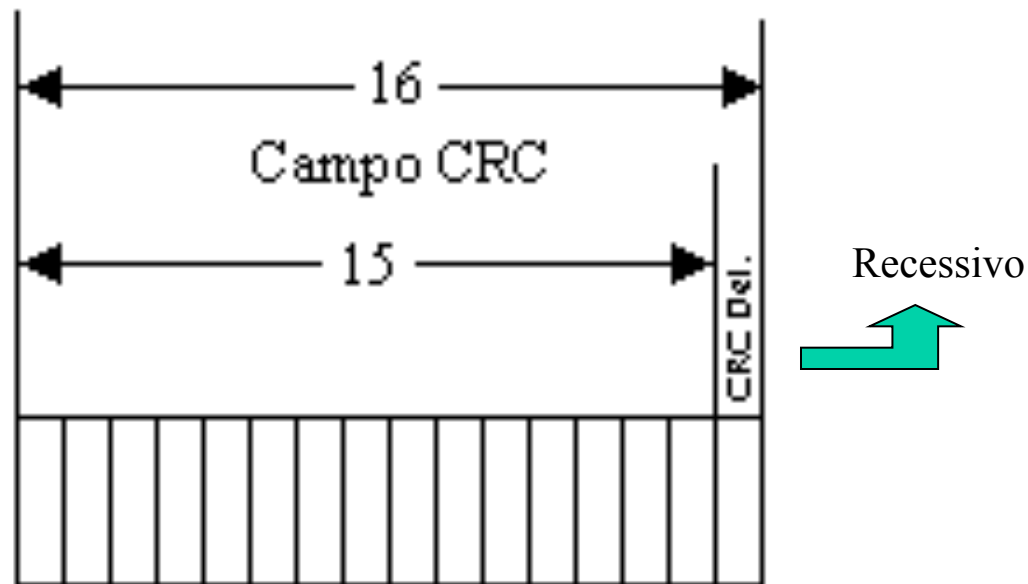
- Pode comportar de 0 (zero) a 8 (oito) bytes, de 8 bits cada.



Mensagens do CAN

Formato das Mensagens

- ❑ Campo CRC (*Cyclic Redundancy Check*)
 - Composto por 15 (quinze) bits
 - CRC delimitador



Mensagens do CAN

Formato das Mensagens

- ❑ Campo de Reconhecimento (Ack Field)
 - Composto por 2 (dois) bits
 - ACK Slot
 - ACK delimiter

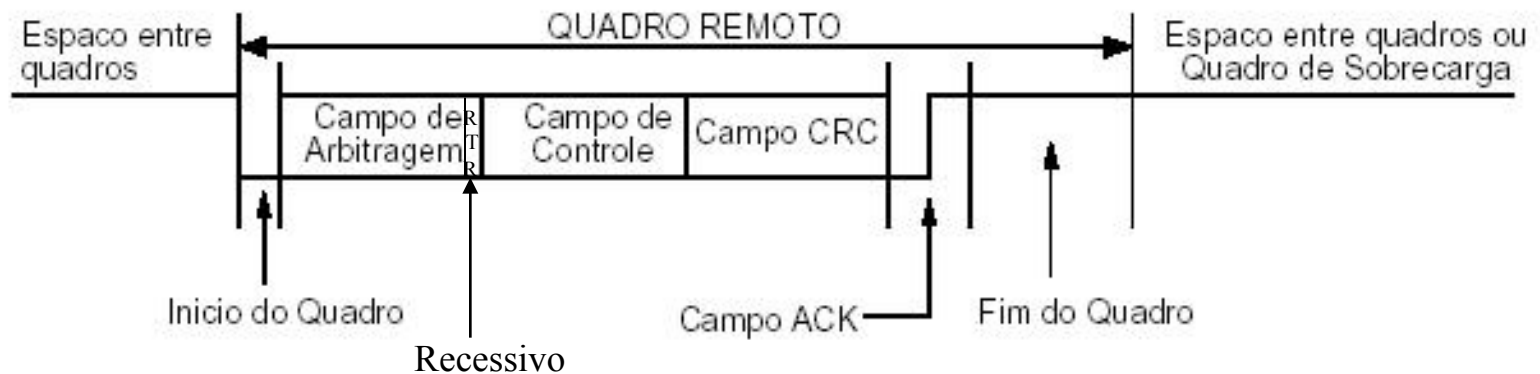
- ❑ Fim de Quadro (End of Frame)
 - Composto por sete bits recessivos

Mensagens do CAN

Formato das Mensagens - Tipos de Quadros

❑ Quadro Remoto - Remote Frame

- É enviado toda vez que um determinado nó, atuando como receptor, necessita receber uma mensagem.
- Mesma formação do Quadro de Dados, entretanto não possui o Campo de Dados.
- O Bit RTR nesse quadro é recessivo

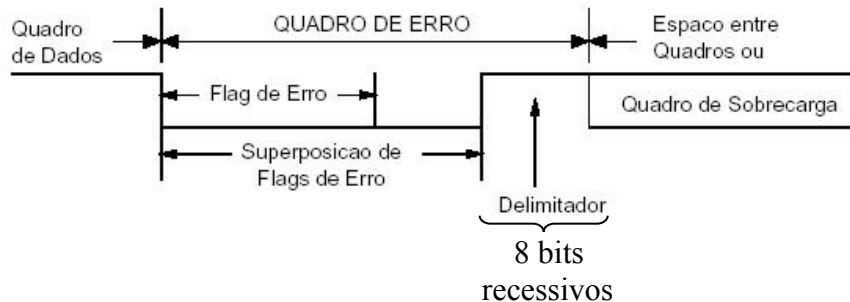


Mensagens do CAN

Formato das Mensagens - Tipos de Quadros

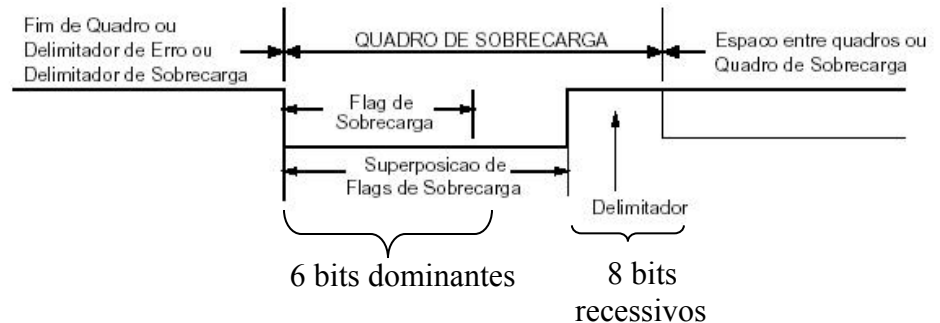
■ Quadro de Erros (Error Frame)

- Composto por dois campos:
 - Flag de erro
 - Error Active
 - Error Passive
 - Delimitador de quadro



■ Quadro de Sobrecarga (Overload Frame)

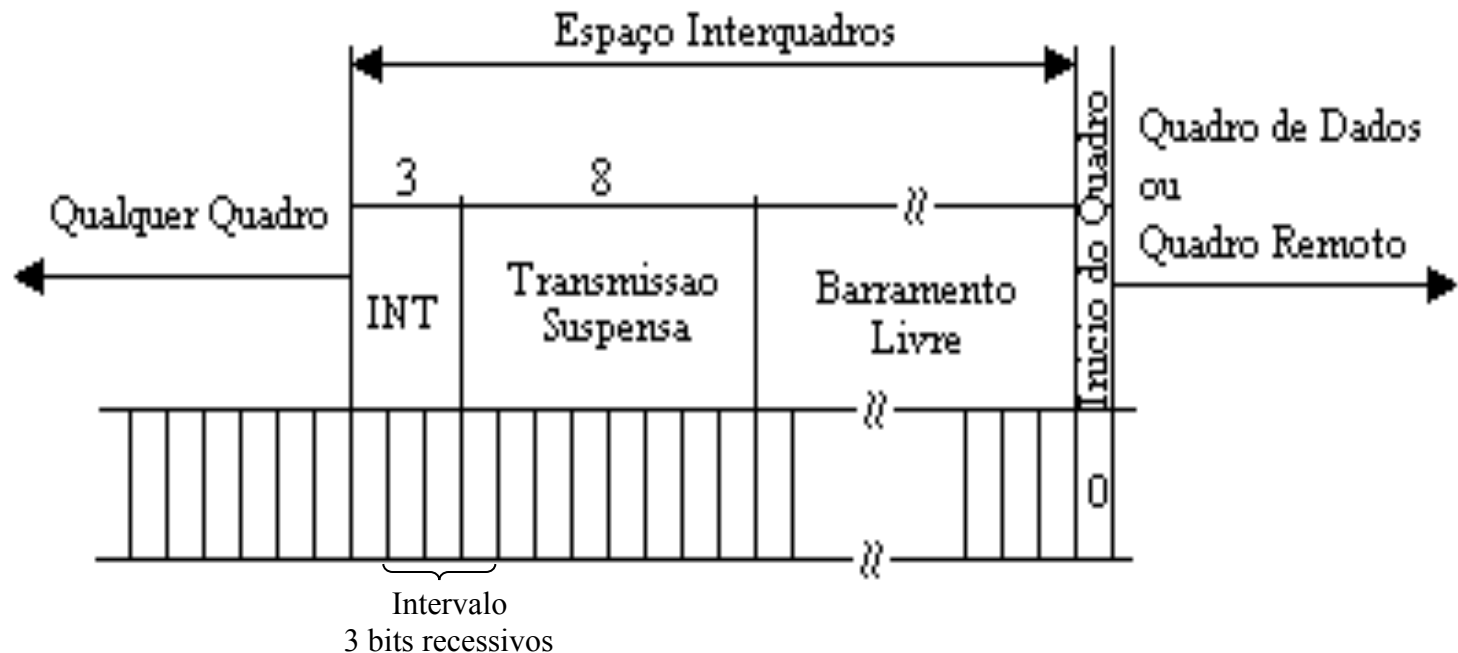
- Composto por dois campos:
 - Flag de Sobrecarga
 - Delimitador de quadro



Mensagens do CAN

Formato das Mensagens - Tipos de Quadros

□ Espaço Interquadros - *Interframe Space*



Codificação CAN

Campos Codificados

Bit Stuffing

Quadro de Dados / Remoto

- Início de Quadro
- Campo de Arbitragem
- Campo de Controle
- Campo de Dados
- CRC

Campos Não Codificados

(Formato Fixo)

Quadro de Dados / Remoto

- Delimitador de CRC
- Campo de Reconhecimento
- Fim de Quadro

Quadros de Erros

Quadro de Overload

Tratamento de Erros

Deteccção e Sinalização

Deteccção

Nível de Bit

- ❑ *Bit monitoring*
- ❑ *Bit Stuffing*

Nível de Mensagem

- ❑ *CRC ou Cyclic Redundancy Check*
- ❑ *Frame Check*
- ❑ *Acknowledgment Error Check*

Sinalização

*Erro de CRC - O bit flag de erro é enviado após o bit **Ack Delimiter***

Outros tipos de erros - O bit flag de erro é enviado após a detecção

Tratamento de Erros

Mecanismo de Falhas - Fault Confinement

Dois contadores - Nó do CAN

- ❑ Erros de Transmissão - 8 pontos
- ❑ Erros de Recepção - 1 ponto

Estado dos Nós

- ❑ 1 a 127 pontos - Error Active
- ❑ 128 a 255 - Error Passive
- ❑ Acima de 255 - Bus Off

Filtragem e Validação das Mensagens

❑ Filtragem Através de Máscaras e Códigos

Ex: Para um Quadro Padrão de 11 bits

1100000000

1010000000

❑ Validação das Mensagens

○ Transmissor

- Após Último bit do Fim de Quadro

○ Receptor

- Após Penúltimo bit do Fim de Quadro

Protocolos de Alto Nível


High Layer Protocols - Modelo OSI

Algumas Tarefas Desenvolvidas

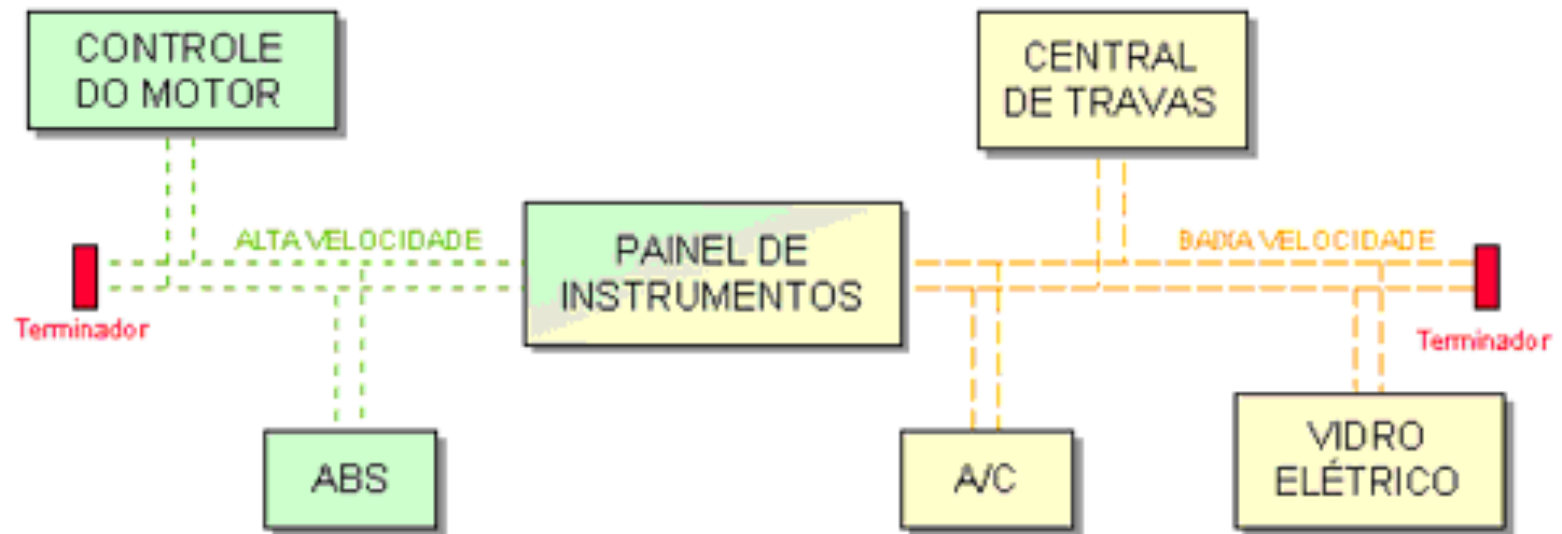
- ❑ Inicialização dos diversos componentes do sistema
- ❑ Distribuição dos identificadores de mensagem
- ❑ Interpretação do conteúdo do Quadro de Dados
- ❑ Gerenciamento do status do sistema

Exemplo de Protocolos de Alto Nível - HLP

- CAN OPEN
- DEVICENET
- CAN Kingdom

 The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Exemplo de Aplicação com Rede CAN



Aspectos Positivos do CAN

- Flexibilidade do Sistema;
- Roteamento de Mensagens;
- Multicast e Multi-mestre;
- Consistência dos Dados;
- Bastante utilizado em aplicações embarcadas.