

# Redes de Computadores

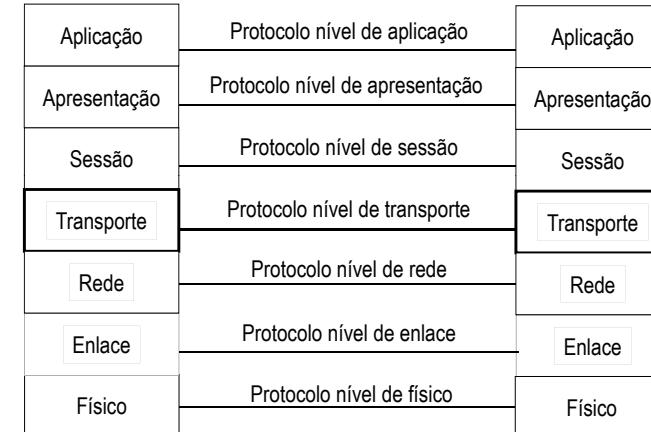
## Nível de Transporte



Trabalho sob a Licença Atribuição-SemDerivações-SemDerivados 3.0 Brasil Creative Commons.  
Para visualizar uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-nd/3.0/br/>

### Aula 20

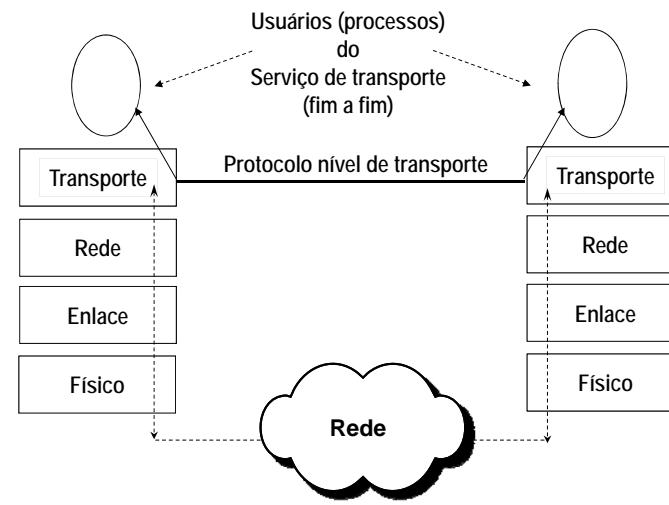
## Nível de transporte



Redes de Computadores

2

## Contexto dos protocolos de transporte na Internet



Redes de Computadores

4

## Introdução

- Motivação
  - Complexidade das redes (diversidade e quantidade de dispositivos)
    - Heterogeneidade dispositivos, enlaces e tecnologias
- Objetivo:
  - Tornar complexidade transparente aos processos de aplicação
    - Esconder detalhes e fornecer uma visão simplificada
  - Fornecer comunicação lógica entre processos de aplicação
    - Camada de rede oferece comunicação lógica entre hospedeiros



Redes de Computadores

3

## Principais pontos

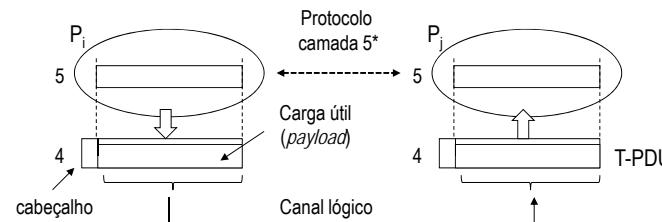
- Serviços são oferecidos pela camada de transporte
  - Orientados a conexão
  - Não orientado a conexão
- Construídos sobre recursos abstratos (virtuais)
  - Entidade de transporte (TSAP)
  - Canal lógico (noções de conexão ou de ponto de comunicação)
- Serviços da entidade de transporte são
  - Executados nos sistemas finais
  - Disponibilizados às camadas superiores através de chamadas de sistema
    - Ex.: no TCP/IP corresponde a chamadas de funções da interface de *sockets*

## Principais Serviços (modelo OSI)

- Encapsulamento e desencapsulamento
- Multiplexação e demultiplexação
- Controle de fluxo
- Controle de erro
- Controle de congestionamento
- Orientados a conexão e não orientados a conexão
- Segurança
- Qualidade de serviço

## Encapsulamento e desencapsulamento

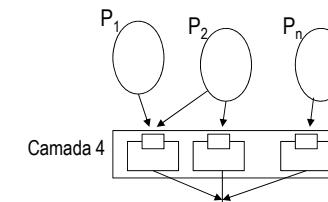
- Inserção/retirada de informações de controle da camada de transporte
  - Ex.: número de sequência, controle de fluxo e de erro
- Considera fragmentação e remontagem



\*Lembrando que no modelo TCP/IP a camada superior ao transporte é a camada de aplicação (7)

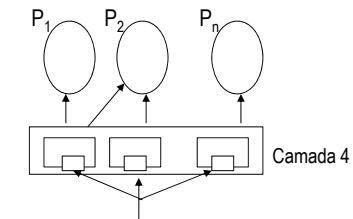
## Multiplexação e demultiplexação

**Multiplexação**  
Uma única entidade recebe dados de várias origens (muitos para um)



As entidades de transporte são identificadas por seu TSAP (Transport Service Access Point)

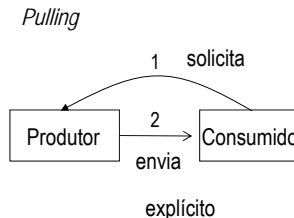
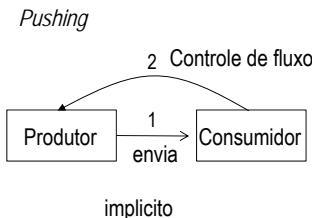
**Demultiplexação**  
Uma única entidade encaminha dados para mais de um destino (um para muitos)



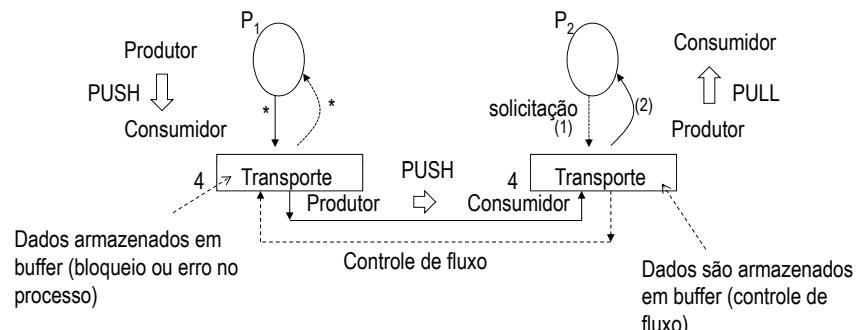
TCP (Proto 06), UDP (Proto 17) → Para IP  
TCP (STREAM), UDP (DGRAM) → Para aplicação

## Controle de fluxo

- Objetivo é evitar a perda de dados do lado consumidor (destino)
- Comunicação é uma relação produtor-consumidor
  - Emissor produz dados que são consumidos pelo destino
  - Problema: taxa de produção maior que a taxa de consumo
- Formas de entrega:



## Na prática....



## Controle de erro

- O controle de erro necessita identificar
  - Quais T-PDUs devem ser reenviadas (erro ou perda), estão duplicados ou estão fora de ordem
- Responsável por
  - Detectar e descartar T-PDUs com erro
  - Identificar T-PDUs faltantes e solicitar seu reenvio (ou reenvio por *timeout*)
  - Reconhecer T-PDUs duplicadas e descartá-las
  - Armazenar T-PDUs de forma a garantir a entrega na ordem, sem erros, sem duplicação para o destino final (processo de aplicação)
- Baseado em número de sequência e em confirmações

## Números de sequência

- Identificam T-PDUs
  - TPDU-s são numerados sequencialmente na sua origem
  - Possibilidade de identificador "lacunas" nos números de sequência devido a perdas e chegada fora de ordem
  - Possibilidade de detectar duplicação ao receber duas T-PDUs com o mesmo número

Questão associada: quantos bits se usa para o número de sequência?  
Evitar confusão entre um "novo zero" e a retransmissão de um "velho zero"

## Confirmação

---

- Controle de erros é feito através de confirmações
  - Positivas: quando uma T-PDU foi recebida corretamente
  - Negativas: quando uma T-PDU não foi recebida corretamente
- Confirmações são associadas a números de sequência
- Ações sobre T-PDUs
  - Recebidas sem erros: se tudo OK, confirma sua recepção correta
    - Se duplicada: descarta e confirma sua recepção
    - Se fora de ordem: ordena e confirma a recepção das T-PDUs ordenadas
    - Se faltando: armazena e espera chegar a que falta
  - Recebidas com erro: são descartadas no destino e reenviadas por pedidos explícitos ou por *timeout*

## Segurança

---

- Mecanismos de segurança
  - Autenticação
  - Controle de acesso (autorização)
  - Confidencialidade
- Os serviços/protocolos são classificados em seguros e inseguros

## Qualidade de serviço (QoS)

---

- Serviços que permitem definição de parâmetros de funcionamento
  - Vazão, latência, variação do atraso (jitter), taxa de perda etc
- Normalmente são parâmetros das camadas inferiores que são mapeados na camada de transporte
  - Se não são ofertados pela camada N-1, não há como uma camada N fazer garantias para uma camada superior N+1

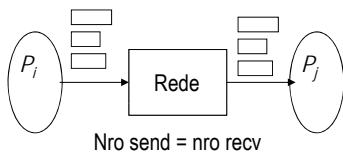
## Estudo de casos: protocolos Internet

---

- Protocolo UDP
  - Não possui controle de fluxo, nem faz controle de erro
    - Há apenas detecção de erro em um datagrama individual
- Protocolo TCP
  - Realiza controle de fluxo (esquema de créditos)
  - Realiza controle de erro via confirmações positivas e transmissões por *timeout*
  - Mecanismos baseados em números de sequência
- Tanto o TCP quanto o UDP
  - são protocolos inseguros
  - não oferecem mecanismos de QoS

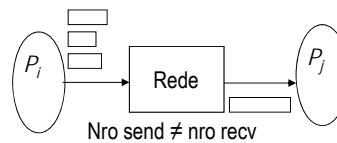
## Modelo de T-PDUs

- Orientado a mensagem
  - Os dados são delimitados em T-PDUs independentes
  - Modelo usado na camada de transporte da Internet (UDP)
    - T-PDU é denominada de datagrama



Redes de Computadores

- Sequência contínua (*byte stream*)
  - Não há delimitação de T-PDUs
  - T-PDUs são inter-relacionadas
  - Modelo empregado na camada de transporte da Internet (TCP)
    - T-PDU é denominada de segmento



17

## Multiplexação e demultiplexação: O conceito de porta

Porta	Protocolo	Aplicação
20	TCP	FTP-data
21	TCP	FTP-control
25	TCP	SMTP
53	TCP/UDP	DNS
80	TCP	HTTP
110	TCP	POP3
161	UDP	SNMP

TCP	UDP
IP	Internet Protocol
Interface de rede	

Instituto de Informática - UFRGS  
(c) A. Carissimi - 15.06.17

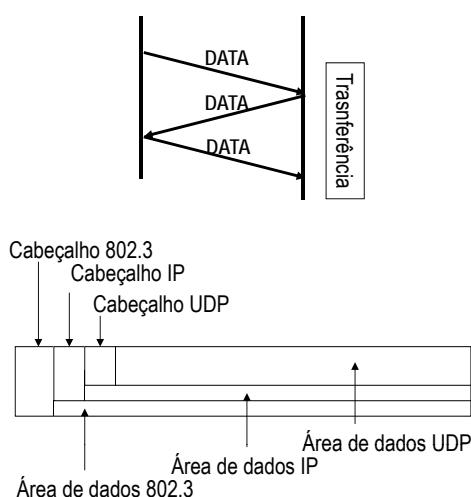
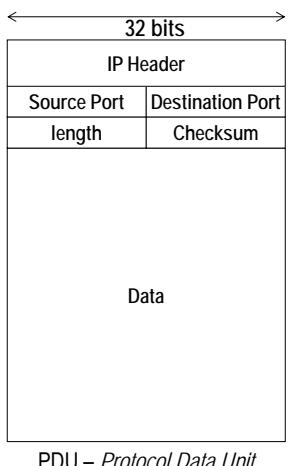
Redes de Computadores

- É um número de 16 bits utilizado como identificador
- Existem três tipos de portas [RFC 6335]
  - Bem conhecidas (*well known ports or system ports*): 1 a 1023
  - *Registered ports or user ports*: 1024 a 49151
    - Listadas para coordenar o uso de serviços não padrão
    - *Dynamics and/or private ports*: 49152 a 65535
- Portas podem estar em um de três estados
  - *Assigned, reserved ou unassigned* (dynamic ports)
  - <http://www.iana.org/assignments/port-numbers>
- Portas TCP são independentes de Portas UDP
  - Porta 100 (TCP) ≠ Porta 100 (UDP), mas se convenceu “alocar” as duas simultaneamente para um mesmo protocolo

Conceito de portas é como o TCP e o UDP multiplexam e demultiplexam seu uso pela camada de aplicação.

18

## Formato do datagrama UDP

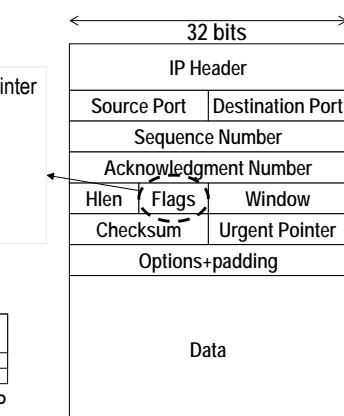
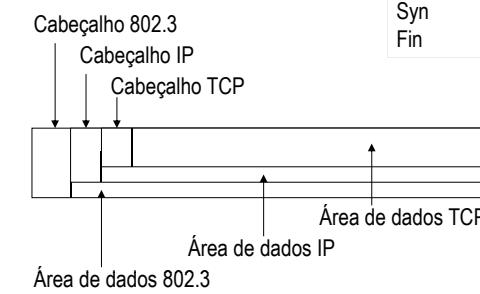


Redes de Computadores

19

## Formato do segmento TCP

- Controle de erro e fluxo
- Garantia de entrega e de ordem
- Gerenciamento da conexão



PDU – Protocol Data Unit

Instituto de Informática - UFRGS  
(c) A. Carissimi - 15.06.17

Redes de Computadores

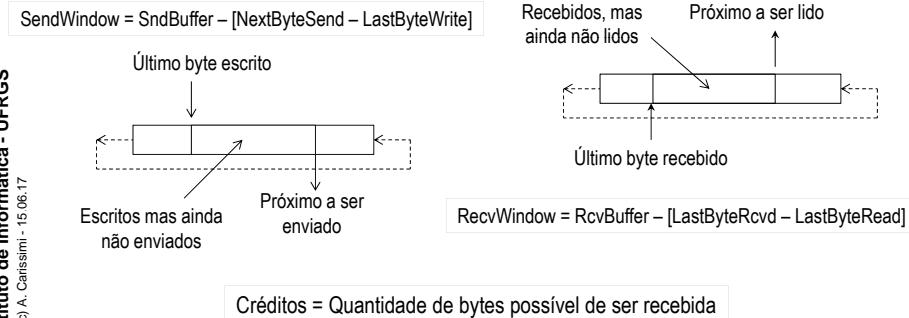
20

## Controle de fluxo do TCP

- Baseado em um esquema de crédito (genérico)
  - Variação de janela deslizante onde os buffers de transmissão e recepção são de tamanho variável (blocos múltiplos de bytes = segmentos TCP)
  - Emissor tem crédito para enviar até  $n$  bytes ao destino
    - Segmento ao ser aceito (processado), o destino renova a quantidade de créditos por um valor  $c$  ( $0 < c \leq n$ )
    - Créditos podem ser renegociados
  - Tamanho da janela é negociado no estabelecimento da conexão (*Maximum Segment Size – MSS*). (valor *default* é 536 bytes para Ethernet)
  - Confirmações (ACKs) servem para renovar os créditos

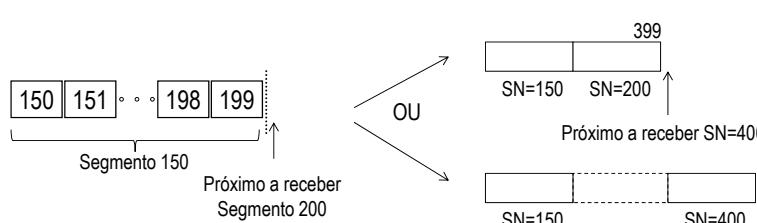
## Janela de transmissão e recepção no TCP

- Sistema de créditos
  - Capacidade de buffer no receptor
  - Janelas de transmissão e recepção



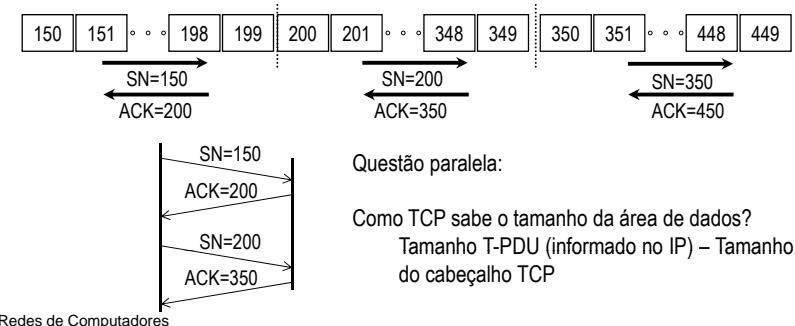
## Controle de sequência no TCP

- Garante a recepção dos dados na ordem da emissão e sem duplicação
- Cada byte enviado possui um número de sequência associado
- O segmento TCP é identificado pelo número de sequência do seu primeiro byte



## Confirmação positiva (ACK)

- Baseado no valor no campo *acknowledge number* (ACK)
  - Indica o número de sequência dos bytes já processados pela aplicação
  - Informa ao transmissor o próximo byte a ser recebido
  - TCP não faz confirmação negativa



## Controle de erros

- Objetivo: garantir o recebimento correto dos segmentos
  - Sem erros, na ordem, sem duplicação
- Baseado em:
  - Confirmação positiva (ACK), similar ao *go-back N*, ou seletiva (SACK)
    - RFC 2018, permite o reconhecimento seletivo (similar ao selective repeat)
  - Retransmissão por *time-out*
- Ações nos segmentos TCP com problemas
  - Recebidos com erro ou duplicados são descartados
  - Perdidos são retransmitidos por *time-out*
  - Fora de ordem são armazenados em buffer até ser possível reordenar

## Retransmissão

- O protocolo TCP emprega *timeout* por segmentos
  - Processo origem dispara um *timeout* para cada segmento enviado
  - Segmento é retransmitido quando a origem não recebe a confirmação antes da expiração do temporizador
- Tratamento da duplicação é feito pelo *sequence number*
  - Destino espera segmento com número  $x$ , qualquer segmento com número inferior é considerado duplicado e é descartado

## Controle de fluxo e congestionamento no TCP

- O controle de fluxo evita a perda de dados no receptor final, porém
  - pode haver sobrecarga (*overflow*) em buffers dos sistemas intermediários
  - controle de fluxo é fim a fim (como o TCP), o congestionamento seria um problema do IP (por tratar de intermediários)
    - Como o IP não faz controle de congestionamento, o TCP faz no seu lugar
- Estratégia de detecção
  - Ocorrência de dois eventos: expiração de *time-out* e quatro ACKs iguais
    - Time-out: considera que perda ocorreu devido a congestionamento
    - ACKs iguais: implica que segmento está faltando, porém três segmentos posteriores foram recebidos

## Controle de congestionamento no TCP

- Baseado em uma nova janela (congestionamento)
  - Tamanho da janela = MIN (janela de recepção; janela de congestionamento)
- Janela de congestionamento (*cwnd* – *congestion window*)
  - Inicia com  $cwnd = 1 * MSS$  (*Maximum Segment Size*)
  - Para cada ACK,  $cwnd = cwnd + 1 * MSS$
  - Algoritmo denominado de partida lenta (*slow start*)
- Há ainda algoritmos
  - Prevenção de congestionamento (*congestion avoidance*)
  - Recuperação rápida (*fast recovery*)

Detalhes desse funcionamento na disciplina de protocolos!!

## Leituras complementares

---

- Stallings, W. Data and Computer Communications (6<sup>th</sup> edition), Prentice Hall 1999.
  - Capítulo 15, seção 15.3, 15.4
- Tanenbaum, A. Redes de Computadores (4<sup>a</sup> edição), Campus, 2000.
  - Capítulo 6, seção 6.1, 6.2 e 6.3
- Carissimi, A.; Rochol, J; Granville, L.Z; Redes de Computadores. Série Livros Didáticos. Bookman 2009.
  - Capítulo 6, seções 6.1 a 6.3