

Redes de Comunicação de Dados



PUC -Rio
Departamento de Informática

Luiz Fernando Gomes Soares
lfgs@inf.puc-rio.br



PUC-Rio / DI

TeleMídia

Nível de Transporte



PUC-Rio / DI

TeleMídia

Tipos de Serviço de Transporte

➔ De forma análoga às camadas de enlace e de rede, a camada de transporte oferece ao nível de sessão, dois tipos de serviço:

- serviços não-orientados à conexão.
- serviços orientados à conexão



PUC-Rio / DI

TeleMídia

Serviço Não-Orientado à Conexão

➔ Quando opera no modo não-orientado à conexão, a camada de transporte mapeia o pedido de transmissão de uma SDU (unidades de dados de serviço, ou seja, mensagem do nível superior) de transporte em um pedido de transmissão feito, usualmente, ao serviço não-orientado à conexão fornecido pela camada de rede.

➔ O UDP da Internet é um exemplo de protocolo que implementa o serviço não-orientado a conexão.



PUC-Rio / DI

TeleMídia

Serviço Orientado a Conexão

- ➔ No modo de operação orientado à conexão, o serviço de transporte fornece meios para estabelecer, manter e liberar conexões de transporte entre um par de usuários, possivelmente entidades da camada de sessão, através de pontos de acesso ao serviço de transporte (SAPs). Para o protocolo de transporte, tudo então se passa como se houvesse um circuito ligando cada par de SAPs (denominado circuito virtual).
- ➔ Sobre uma conexão, controle de erro e de fluxo são geralmente aplicados. De fato, os protocolos de transporte lembram, sob vários aspectos, os protocolos do nível de enlace. Todas as técnicas apresentadas para o nível de enlace para controle de erro e fluxo também se aplicam ao nível de transporte, só que agora não mais na conexão de enlace, mas no circuito virtual estabelecido entre os SAPs de transporte.
- ➔ O TCP da Internet é um exemplo de protocolo que implementa o serviço orientado a conexão. Outro exemplo é o padrão TP4 da ISO.



PUC-Rio / DI

TeleMídia

Funções da Camada de Transporte

- ➔ Multiplexação ou *splitting*.
- ➔ Transporte de unidades de dados do serviço de transporte (SDUs ou mensagens).
- ➔ Segmentação e blocagem.
- ➔ Detecção e correção de erros fim a fim.
- ➔ Sequenciação.
- ➔ Controle do fluxo de dados nas conexões de transporte.
- ➔ Transporte de dados expresso.
 - Uma unidade de dados expressa é transferida/processada com prioridade maior que as unidades de dados normais.



PUC-Rio / DI

TeleMídia

Multiplexação

- Custo: Várias concessionárias de serviços de rede fazem a sua tarifação baseada no tempo em que uma conexão de rede está aberta.
 - Em uma aplicação com tráfego em rajada essa conexão pode ficar muito tempo ociosa, tendo o usuário de pagar por esse tempo. Uma solução é a multiplexação de várias conexões de transporte na conexão de rede.
- Banda passante: Quando a conexão de rede oferece uma banda passante muito maior que a utilizada pelas conexões de transporte.



Splitting

- Banda passante: a conexão de rede pode oferecer uma banda passante muito mais baixa do que a necessária pela conexão de transporte. Uma solução nesse caso é realizar a divisão (*splitting*) da conexão de transporte em várias conexões de redes.
- No caso de uma estação possuir vários canais de saída no nível físico, o *splitting* pode ser usado para aumentar ainda mais o desempenho.



Serviços Oferecidos

➡ Com conexão

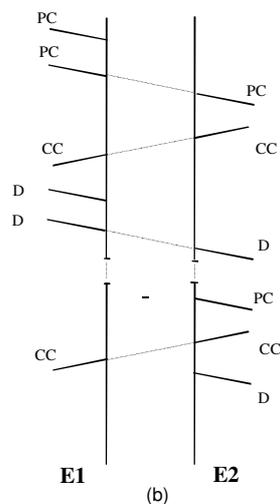
➡ Sem conexão



PUC-Rio / DI

TeleMídia

Estabelecimento de Conexão



PUC-Rio / DI

TeleMídia

Three-way Handshake

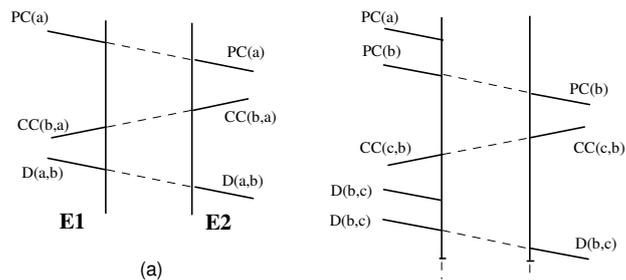
- ➔ Método proposto por Tomlinson [Tomlinson 75]
- ➔ Duas unidades de dados do protocolo de transporte (T-PDU) com numeração idêntica nunca podem estar pendentes ao mesmo tempo.
- ➔ As numerações das T-PDUs não se repetem dentro do período T - tempo de vida de um pacote.



PUC-Rio / DI

TeleMídia

Three-way Handshake



PC(i) = Pedido de conexão (seq=i)
 CC(j,i) = Confirmação de conexão (seq=j, ack=i)
 D(i,j) = Dados (seq=i, ack=j)
 R(j) = Rejeita conexão (ack=j)



PUC-Rio / DI

TeleMídia

Encerramento de Conexões

- Um problema presente é evitar que dados sejam perdidos depois que um dos lados encerrou a conexão.
- Uma entidade de transporte ao pedir uma desconexão deve aguardar por um tempo antes de fechar a conexão, podendo receber dados durante esse período.
- Uma conexão termina apenas depois de decorrido um certo tempo sem que chegue nenhuma T-PDU.
- Dessa forma, se um lado se desconectar, o outro vai notar a falta de atividade e também se desconectar.



Encerramento de Conexões

- Para evitar que uma conexão seja desfeita, os participantes devem assegurar o envio de T-PDUs periódicas informando que estão vivos, quando não têm dados a transmitir.
- Caso muitas T-PDUs se percam durante uma conexão, um dos lados pode fechar a conexão indevidamente.
- Uma combinação de temporizadores com confirmação de desconexão pode ser tentada, sempre minimizando o problema.



Qualidade de Serviço

- O retardo no estabelecimento da conexão.
- O retardo no encerramento da conexão.
- A probabilidade de falha no estabelecimento da conexão.
- A probabilidade de falha na liberação da conexão.
- A vazão em cada sentido da conexão, isto é, a taxa de bits transferidos por segundo.
- O retardo de transferência médio, também em cada sentido.
- O retardo de transferência máximo, também em cada sentido.
- A variação estatística do retardo, expressa, por exemplo, em termos da variância do retardo de transferência.
- A taxa de erro, expressa em porcentagem dos bits transmitidos.
- Probabilidade de queda de uma conexão, isto é, a probabilidade de que a camada de transporte se veja obrigada a fechar uma conexão devido a falhas ou problemas de congestionamento.



PUC-Rio / DI

TeleMídia

Protocolos de Transporte

- Mesmas técnicas do nível de enlace agora aplicadas fim-a-fim
- Controle de Fluxo
 - Sliding Window
- Controle de Erro
 - Selective Repeat
 - Go Back N



PUC-Rio / DI

TeleMídia

Controle de Erro

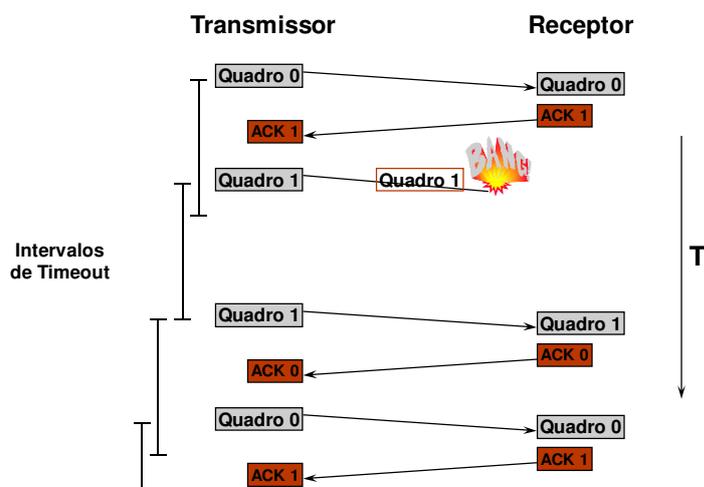
- Detecção de erro
 - Checksum
 - CRC - Cyclic Redundancy Code
- Correção de erro
 - recuperação do quadro original
 - retransmissão do quadro com erro
- Correção de erro
 - Stop-and-Wait ARQ
 - Selective Repeat
 - Go-Back-N



PUC-Rio / DI

TeleMídia

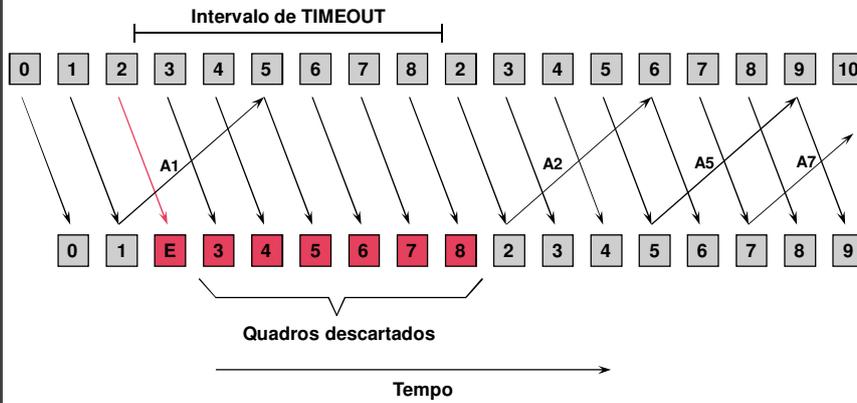
Controle de Erro Protocolo *Stop-and-Wait ARQ*



PUC-Rio / DI

TeleMídia

Controle de Erro Protocolo *GO BACK N*



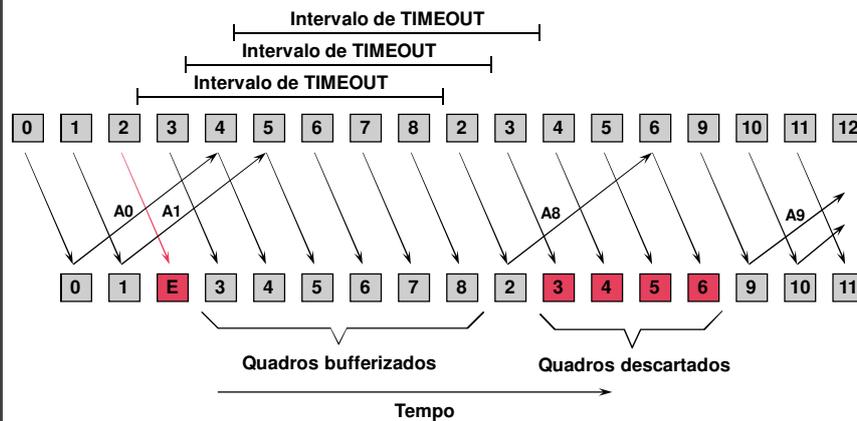
Tamanho da janela de recepção > 1



PUC-Rio / DI

TeleMídia

Controle de erro Protocolo *SELECTIVE REPEAT*



Tamanho da janela de recepção > 1



PUC-Rio / DI

TeleMídia

Serviços de Rede X Serviços de Transporte

Camada de Transporte	Serviço de Datagramas	Serviço de Circuito Virtual	Serviço de Datagramas	Serviço de Circuito Virtual
Camada de Rede	Serviço de Datagramas	Serviço de Datagramas	Serviço de Circuito Virtual	Serviço de Circuito Virtual
				



PUC-Rio / DI

TeleMídia

ISO - TP

➔ *Tipo A*: redes que consideram erro qualquer perda de dados.

- Nesse tipo de rede a quantidade de pacotes duplicados, perdidos, fora de seqüência ou danificados é desprezível.
- O serviço fornecido por esse tipo de rede é orientado à conexão. As redes locais que fornecem o serviço orientado à conexão IEEE 802.2 se aproximam bastante desse perfil.



PUC-Rio / DI

TeleMídia

ISO - TP

➔ *Tipo B*: como as redes tipo A, as redes tipo B consideram erro qualquer perda de dados. Entretanto, nesse tipo de rede os erros são mais freqüentes, sendo então menos confiáveis que as do tipo A.

- Enquadram-se nesse tipo, as redes públicas que fornecem serviço de circuito virtual. O serviço fornecido pelas redes tipo B também é orientado à conexão.



ISO - TP

➔ *Tipo C*: redes que não detectam erros quando pacotes são perdidos, duplicados, entregues fora de ordem ou danificados.

- Essas redes fornecem o serviço datagrama não-confiável (serviço sem conexão e sem confirmação).



TP0

- Fornece mecanismos para estabelecer e encerrar conexões de transporte, utilizando uma conexão de rede para cada conexão de transporte aberta.
- As funções executadas na fase de transferência de dados resumem-se a: transferência de PDUs e, segmentação e remontagem de SDUs.
- Pressupõe a utilização de redes tipo A.



PUC-Rio / DI

TeleMídia

TP1

- O TP1 é semelhante ao TP0, porém foi projetado para recuperar erros causados pela reiniciação (*reset*) de conexões de rede.
- Deve ser utilizado em redes onde ocorrem reiniciações com uma certa frequência.
- As PDUs são numeradas e armazenadas enquanto a entidade de transporte de origem não receber um reconhecimento. Quando a conexão de rede usada por uma conexão de transporte é reiniciada, o fornecedor do serviço de transporte TP1 sabe resincronizar e continuar do ponto em que tinha parado. Nenhum controle de erro ou fluxo é realizado, além daquele fornecido pela própria camada de rede.
- O TP1 foi projetado para ser usado em redes tipo B, porém pode ser usado também em redes tipo A.



PUC-Rio / DI

TeleMídia

TP2

- ➔ O *TP2* implementa a multiplexação do acesso.
- ➔ Redes públicas baseiam, normalmente, sua tarifação na cobrança de uma taxa para cada conexão estabelecida, e de outra taxa para cada pacote transmitido.
- ➔ Adicionalmente, o *TP2* pode executar as funções de controle de fluxo e segmentação/remontagem de SDUs, não realizando qualquer controle de erro ou recuperação em caso de reiniciação de uma conexão.
- ➔ Esse protocolo é, portanto, apropriado para redes tipo A.



PUC-Rio / DI

TeleMídia

TP3

- ➔ O *TP3* combina as funções dos protocolos *TP1* e *TP2*, implementando as funções:
 - multiplexação/demultiplexação de conexões de redes
 - controle de fluxo, segmentação/remontagem e restabelecimento de conexões no caso de reiniciação de conexões de redes.
- ➔ Esse protocolo pode ser usado em redes tipo A ou B.



PUC-Rio / DI

TeleMídia

TP4

- Projetado para redes tipo C (embora possa obviamente ser utilizado em redes de tipo A e B), que fornecem serviço sem conexão e sem confirmação.
- Cabe ao TP4 detectar e recuperar pacotes perdidos, duplicados, danificados e entregues fora de seqüência.



TP4

- Utiliza *three-way handshake*.
- Fornece o serviço de multiplexação de conexões de transporte em uma conexão de rede, e permite também que a carga de uma conexão de transporte seja distribuída (*splitting*) em várias conexões de rede.
- Fornece segmentação e blocagem.
- Utiliza o mecanismo de *checksum* para detectar erros. Usa o *selective repeat* para recuperação de erros.
- Controle de fluxo implementado através da técnica de janela deslizante (*sliding window*) com alocação dinâmica de crédito.
 - Toda vez que envia o reconhecimento das mensagens que recebeu, o receptor envia também o valor do crédito corrente (o valor do crédito indica o número de buffers que o receptor tem disponíveis em um dado momento). Por exemplo: ACK (x, w').
- Oferece um serviço expresso. As PDUs transportando dados expressos têm prioridade em relação às PDUs com dados comuns.



Perguntas

- Em que tipo de rede a comutação rápida de pacotes tem um bom desempenho?
- O que vem a ser comutação rápida por pacotes? Qual a diferença para a comutação de pacotes?
- Por que na comutação rápida de pacotes são necessárias duas pilhas de protocolo?
- Faça passo a passo o procedimento de uma comutação rápida por pacotes em redes com conexão?
- Faça passo a passo o procedimento de uma comutação rápida por pacotes em redes sem conexão?
- Descreva o funcionamento da rede *frame relay*.
- Descreva o funcionamento de uma rede ATM.
- Você poderia usar o mesmo procedimento de comutação rápida de pacotes das redes com conexão em redes sem conexão? Se sim, como? O que é conexão *soft state*?
- Como funciona o padrão MPLS?



PUC-Rio / DI

TeleMídia

Perguntas

- Faça uma avaliação dos serviços de transporte com e sem conexão, sobre serviços de rede com e sem conexão.
- Quais as funções da camada de transporte, obrigatórias e opcionais?
- Quando e por que é necessário a multiplexação no nível de transporte?
- Quando e por que é necessário o *splitting* no nível de transporte?
- Quais os problemas relacionados ao estabelecimento de conexões no nível de transporte? Como podem ser solucionados?
- Quais os problemas relacionados ao encerramento de conexões no nível de transporte? Como podem ser minorados?
- O que vem a ser qualidade de serviço?
- Cite e explique dois mecanismos para controle de erro no nível de transporte.
- Cite e explique dois mecanismos para controle de fluxo no nível de transporte.
- Faça uma avaliação dos diversos tipos de protocolo de transporte ISO, frente aos diversos níveis de rede.



PUC-Rio / DI

TeleMídia

UDP

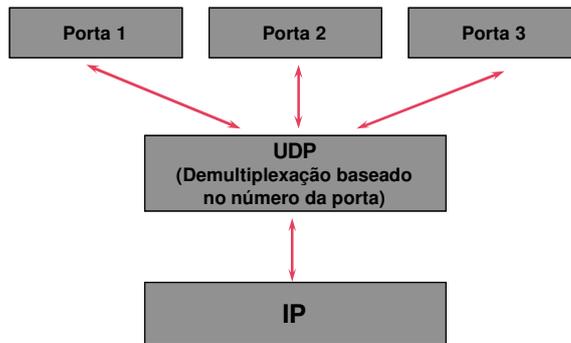


User Datagram Protocol

- ➔ Fornece serviço sem conexão não confiável utilizando o IP
- ➔ Multiplexação de conexões em várias portas



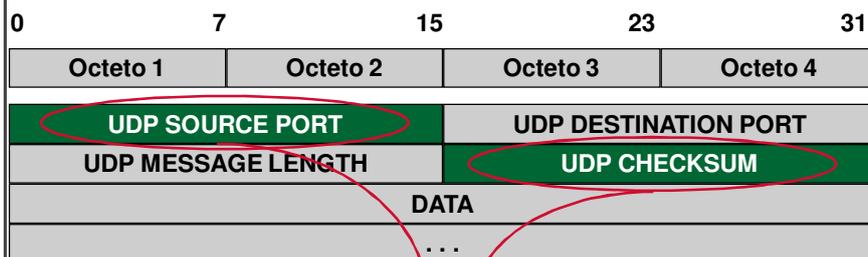
Multiplexação e Demultiplexação UDP



PUC-Rio / DI

TeleMídia

Quadro UDP



Opcionais -- campo = 0

RFC 1123 recomenda uso dos dois campos

Checksum feito somente sobre a mensagem UDP
 não garante que o endereço IP de destino está correto
 ou que os dados do datagrama foram entregues
 ao protocolo correto (checksum do IP pode ter falhado)



PUC-Rio / DI

TeleMídia

TCP



Transmission Control Protocol

- ➔ Serviço confiável orientado a conexão (Circuito Virtual)
 - Sequenciação
 - Detecção e correção de erros fim-a-fim
 - Sliding Window



Portas TCP

- ➔ Cada um dos usuários (processos de aplicação) que o TCP está atendendo em um dado momento é identificado por uma porta diferente.
- ➔ Para obter um endereço que identifique univocamente um usuário TCP, o identificador da porta é concatenado ao endereço IP onde a entidade TCP está sendo executada, definindo um *socket*.
- ➔ Um socket, que é equivalente a um T-SAP, identifica univocamente um usuário TCP em toda a inter-rede.

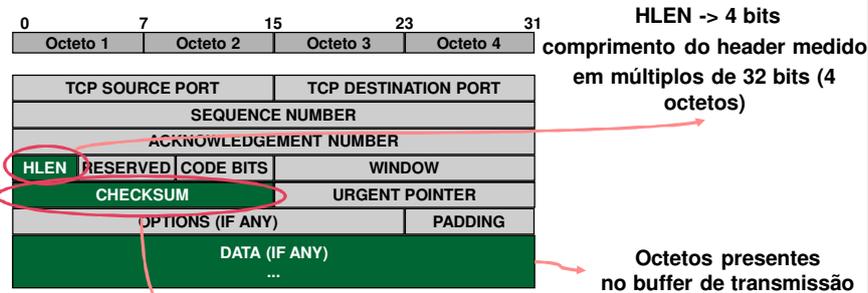


TCP

- ➔ Uma *conexão* é identificada pelo par de sockets de suas extremidades.
- ➔ Um socket local pode participar de várias conexões diferentes com sockets remotos.
- ➔ Uma conexão pode ser usada para transportar dados em ambas as direções simultaneamente, ou seja, as conexões TCP são full-duplex.



Formato do Segmento TCP



CHECKSUM calculado do mesmo modo que no UDP (Pseudo-header, com campo PROTOCOL = 6)

Ao contrário do UDP, SOURCE PORT e CHECKSUM são obrigatórios



PUC-Rio / DI

TeleMídia

TCP

- ➔ O algoritmo de *three-way handshake* é utilizado na abertura de conexões.
- ➔ TCP transfere uma cadeia (stream) contínua de octetos, nas duas direções, entre seus usuários. Normalmente o TCP decide o momento de parar de agrupar os octetos e de, conseqüentemente, transmitir o segmento formado por esse agrupamento. Porém, caso deseje, o usuário do TCP pode fazer uso da função *push* que faz com que o TCP transmita imediatamente os octetos que estão nos seus buffers aguardando transmissão.



PUC-Rio / DI

TeleMídia

TCP

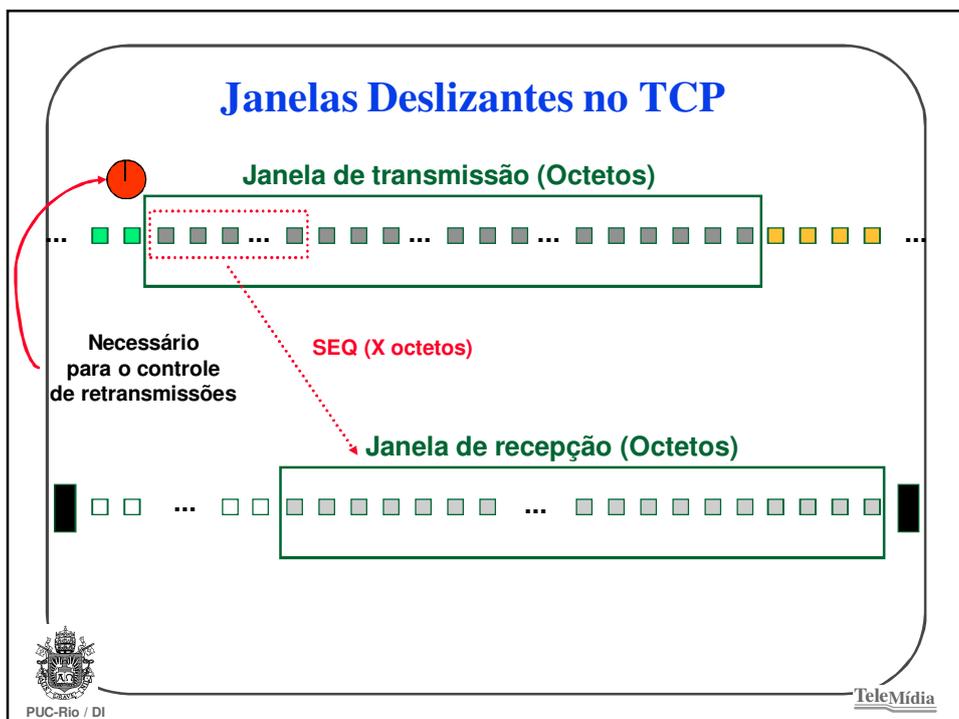
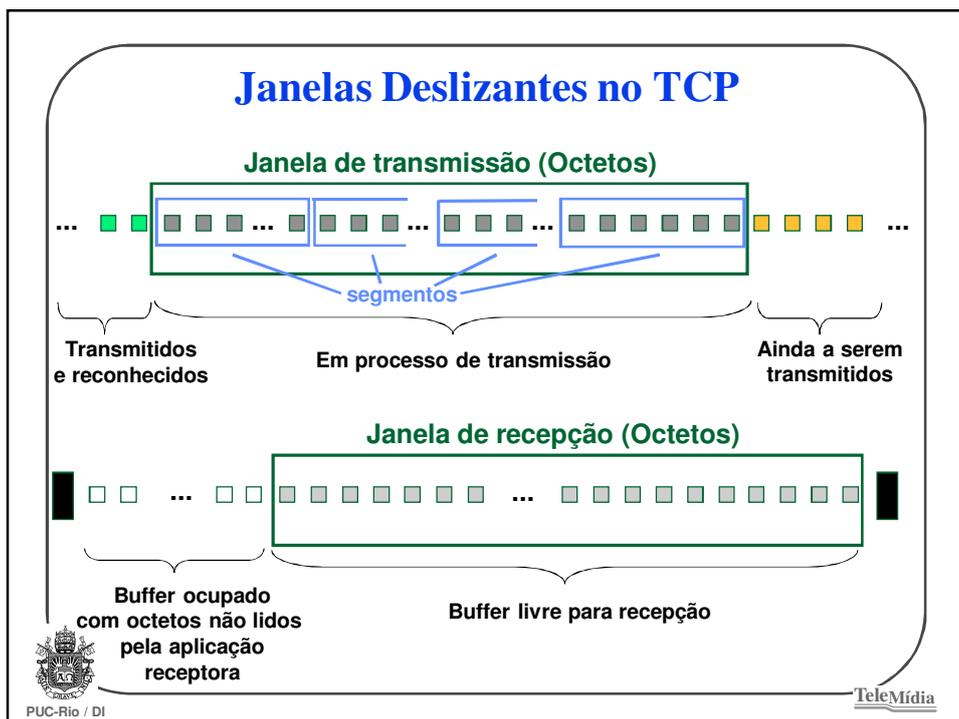
- ➔ Cada octeto transmitido é associado a um número de seqüência. O número de seqüência do primeiro octeto dos dados contidos em um segmento é transmitido junto com o segmento e é denominado *número de seqüência do segmento*.
- ➔ Os segmentos carregam “de carona” (*piggybacking*) um reconhecimento. O reconhecimento constitui-se do número de seqüência do próximo octeto que a entidade TCP transmissora espera receber do TCP receptor na direção oposta da conexão.

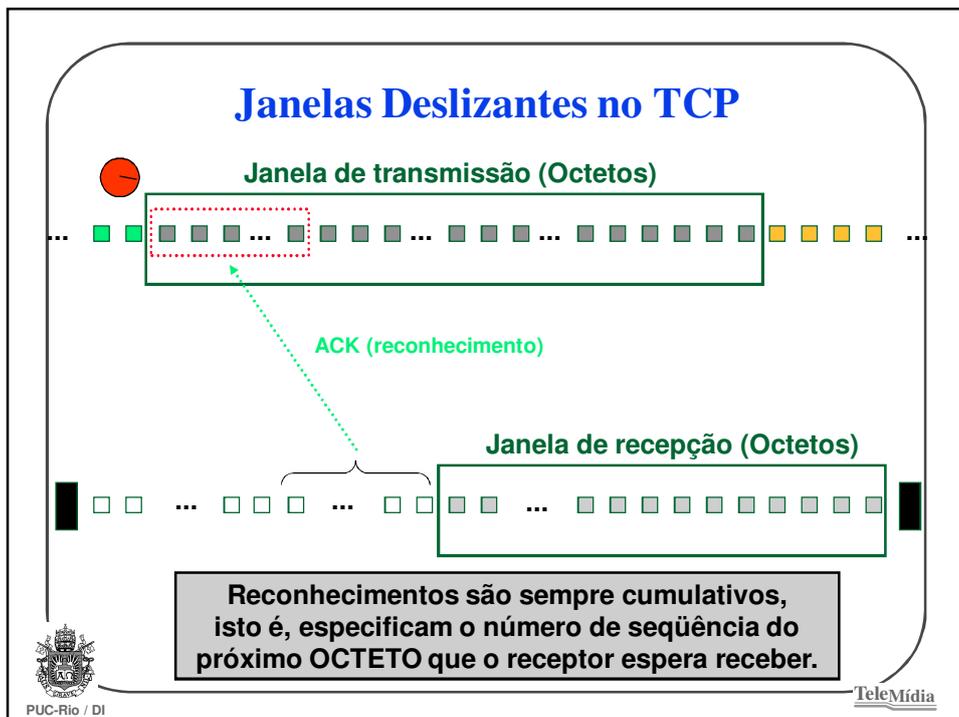
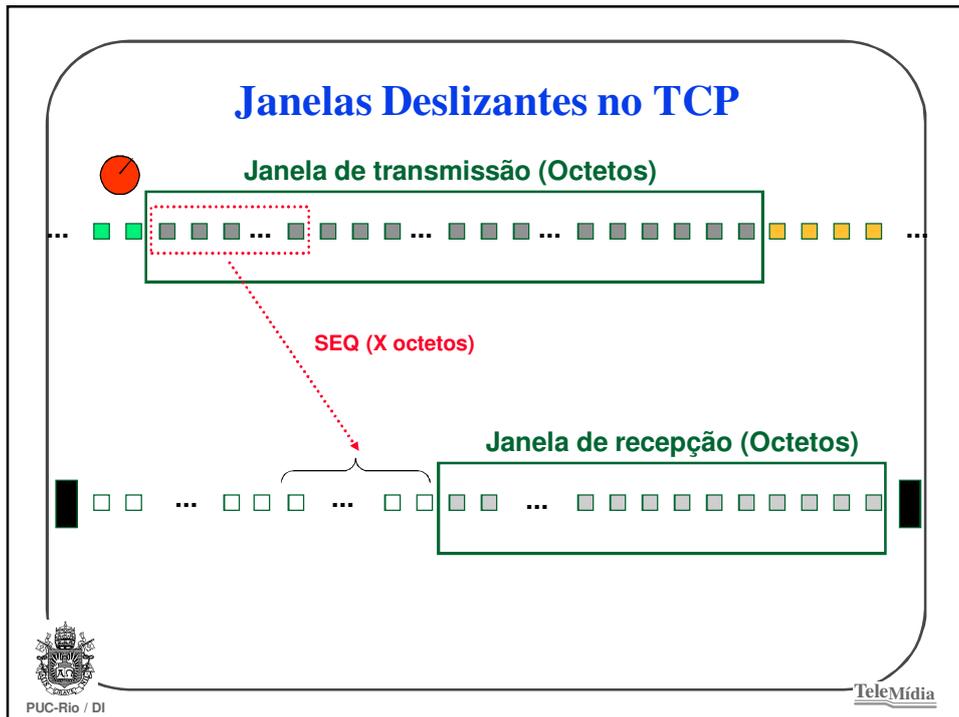


TCP

- ➔ O mecanismo de controle de fluxo baseia-se no envio, junto com o reconhecimento, do número de octetos que o receptor tem condições de receber (tamanho da *janela de recepção*), contados a partir do último octeto da cadeia de dados recebido com sucesso.
- ➔ Com base nessa informação o transmissor atualiza sua *janela de transmissão*, ou seja, calcula o número de octetos que pode enviar antes de receber outra liberação.







Janelas Deslizantes no TCP

Janela de transmissão (Octetos)



Janela de recepção (Octetos)



PUC-Rio / DI

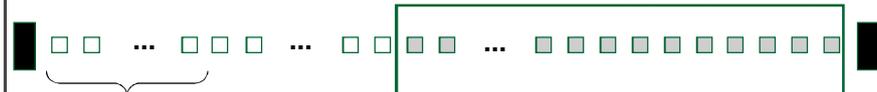
TeleMídia

Janelas Deslizantes no TCP

Janela de transmissão (Octetos)



Janela de recepção (Octetos)



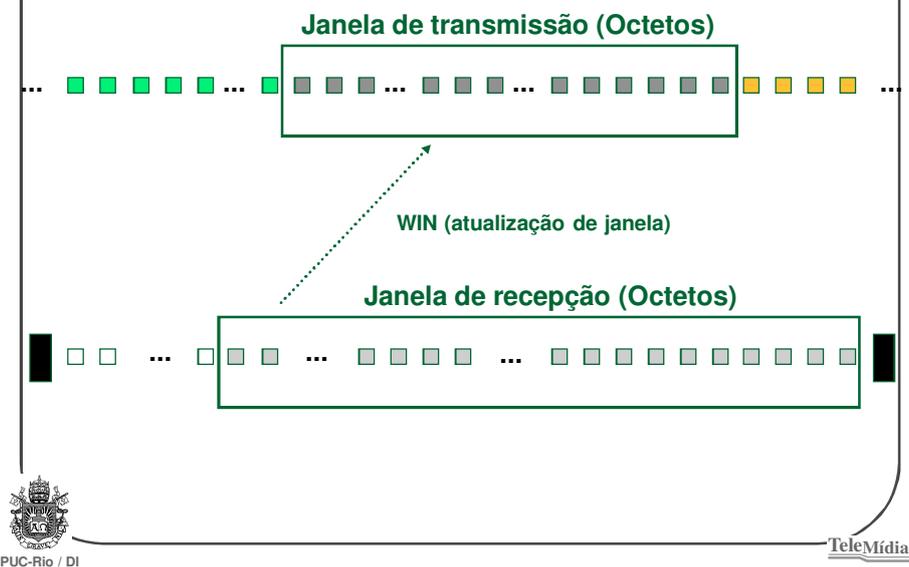
Aplicação
receptora lê
Y octetos



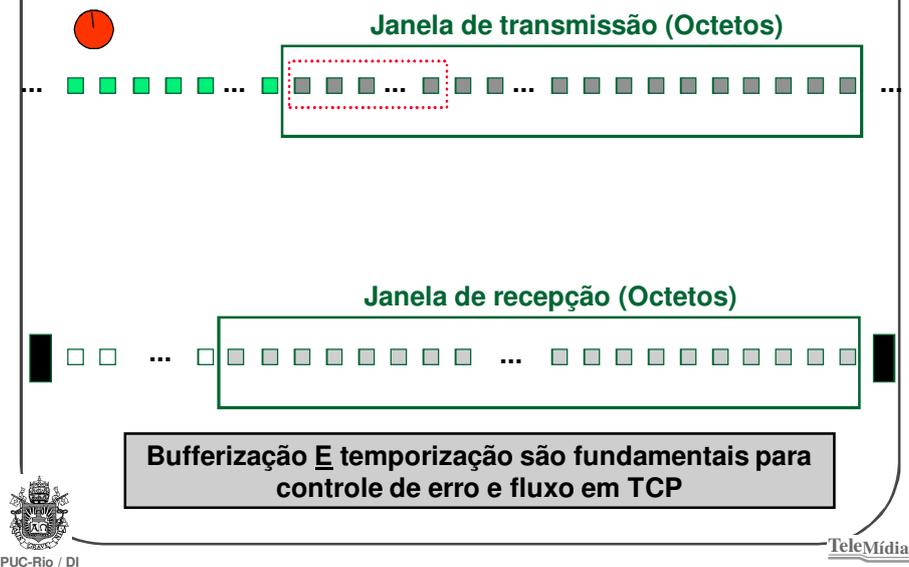
PUC-Rio / DI

TeleMídia

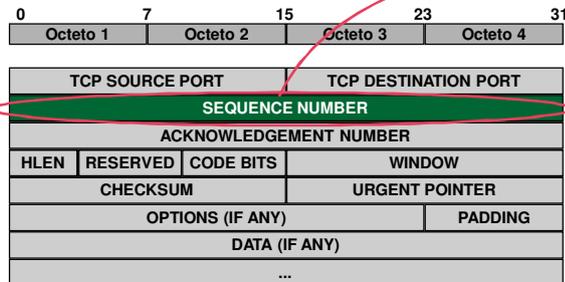
Janelas Deslizantes no TCP



Janelas Deslizantes no TCP



Segmento TCP – Informação da janela de Tx



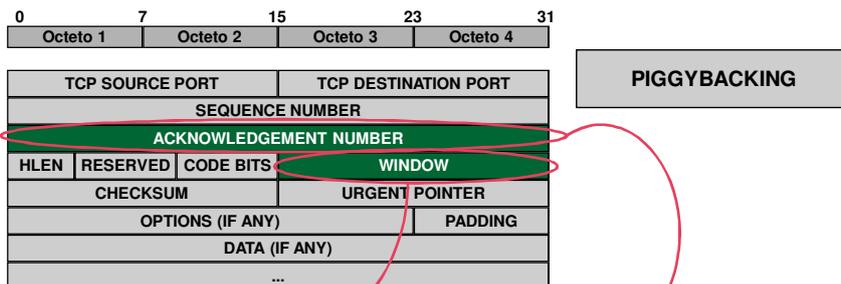
Identifica a posição, relativa ao fluxo de bits da origem deste segmento, dos dados carregados pelo mesmo.
OBS: posição informada em octetos



PUC-Rio / DI

TeleMídia

Segmento TCP – Informação da janela de Rx



Informa ao receptor deste segmento a quantidade máxima de octetos que o transmissor deste segmento pode aceitar no momento

Informa ao receptor deste segmento o próximo número de seqüência que o transmissor deste segmento espera receber no próximo segmento



PUC-Rio / DI

TeleMídia

Protocolo TCP (Transmission Control Protocol)

- Multiplexação do SAP IP por meio de portas
- Serviço orientado a conexão (circuito virtual) e confiável
 - Seqüenciação
 - Controle de fluxo e erro fim-a-fim
- Transmissão em fluxos
 - Unidade de transmissão: octeto
- Desassociação da quantidade de dados enviados pela aplicação
 - Segmentação e blocagem
- Conexão full-duplex
 - Fluxos independentes nas duas direções



PUC-Rio / DI

TeleMídia

Políticas de implementação do TCP

- Transmissão: octetos são enviados (em segmentos) de acordo com implementação
- Recepção: octetos são entregues à aplicação de acordo com implementação
- Aceitação
 - Em ordem
 - Segmentos fora de ordem são descartados
 - Em janela
 - Segmentos fora de ordem são bufferizados
- Retransmissão:
 - Primeiro somente
 - Em lote
 - Um timer por janela
 - Individual
 - Um timer por segmento
- Reconhecimento
 - Imediato
 - Retardado
 - Aguarda piggybacking para enviar reconhecimento



PUC-Rio / DI

TeleMídia

Considerações a Respeito da Retransmissão

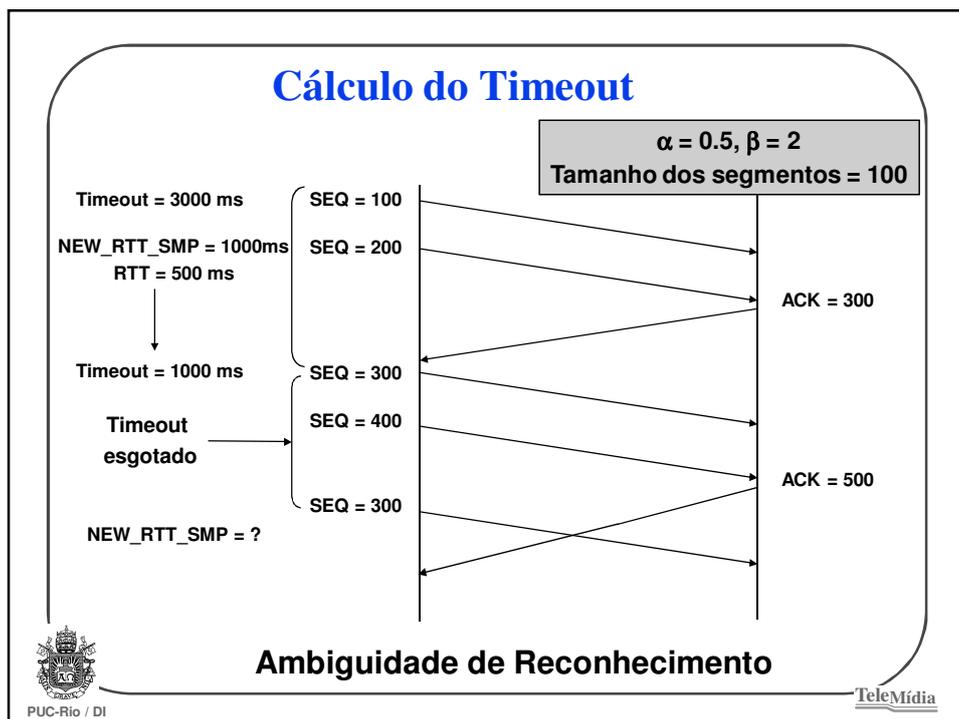
- **Novamente:** reconhecimento especifica o próximo OCTETO esperado pelo receptor
 - TCP não é obrigado a retransmitir o segmento original. Ele pode transmitir um segmento resultante da união de dois ou mais segmentos consecutivos (repacketization), respeitando o tamanho máximo que um segmento pode ter
- Configuração do timeout de retransmissão
 - Timeout pequeno: retransmissões desnecessárias
 - Timeout grande: baixa vazão



Definição do Timeout de Retransmissão

- Variável básica: round-trip time
- Exemplo de fórmula para cálculo adaptativo do RTT
 - $RTT = (\alpha * OLD_RTT) + ((1-\alpha) * NEW_RTT_SAMPLE)$
 - RTT inicial = 0 (RFC 1122)
 - NEW_RTT_SAMPLE é obtida para cada "nova" janela
- Exemplo de fórmula para definição do timeout em função de RTT
 - Timeout = $\beta * RTT$
 - β pode ser estático (=2) ou obtido por outro algoritmo
 - Timeout inicial = 3 segundos
 - Timeout máximo = 240 segundos
- Algoritmo de Jacobson (melhor resposta a altas variações de RTT)
 - β = variância de RTT



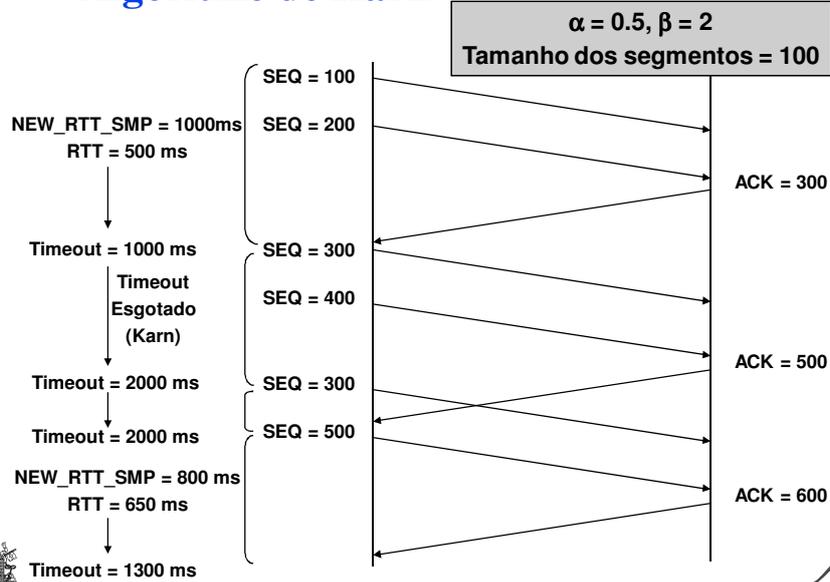


Ambiguidade de Reconhecimento

- Caso o reconhecimento do segmento chegue após a retransmissão do mesmo, que valor atribuo a NEW_RTT_SAMPLE ?
 - Solução: não considerar RTT de segmentos retransmitidos
 - RTT pode permanecer indefinidamente inadequado
- Algoritmo de Karn
 - Timer Backoff: toda vez que um segmento é retransmitido, timeout é multiplicado por N (normalmente 2)
 - Quando é recebido o reconhecimento correspondente a um segmento não retransmitido, RTT e Timeout são recalculados com base nos valores anteriores aos Backoffs

PUC-Rio / DI

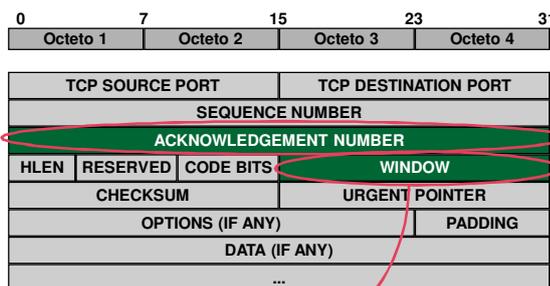
Algoritmo de Karn



PUC-Rio / DI

TeleMídia

Segmento TCP – Informação da janela de Rx



PIGGYBACKING

Informa ao receptor deste segmento a quantidade máxima de octetos que o transmissor deste segmento pode aceitar no momento

Informa ao receptor deste segmento o próximo número de seqüência que o transmissor deste segmento espera receber no próximo segmento



PUC-Rio / DI

TeleMídia

Considerações a Respeito do Controle de Fluxo

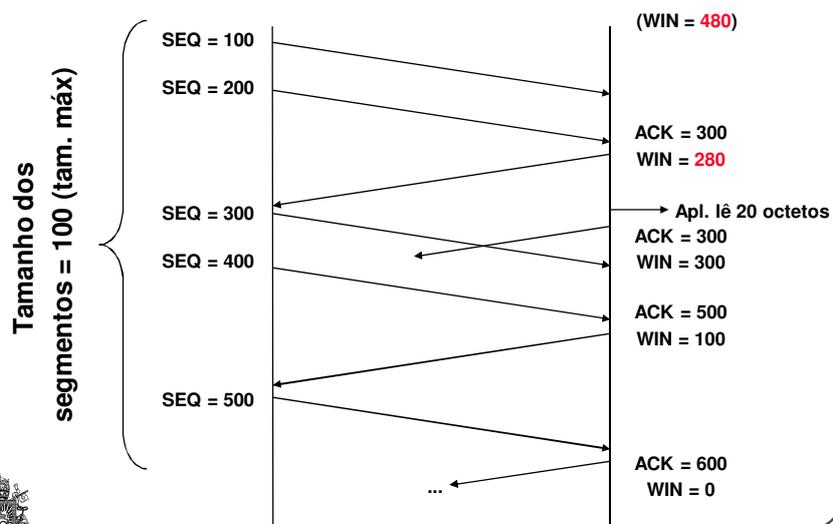
- ➔ Receptor informa WINDOW = 0 quando não consegue mais receber
- ➔ Quando espaço é liberado, receptor envia atualização de janela com WINDOW > 0
 - E se a atualização de janela for perdida ?
 - Transmissor envia segmentos de um único octeto periodicamente (*window probes*) para testar tamanho da janela de recepção do receptor
 - E se a aplicação receptora estiver lendo do buffer blocos de dados muito pequenos, em comparação com a taxa de envio do módulo TCP no transmissor ?



PUC-Rio / DI

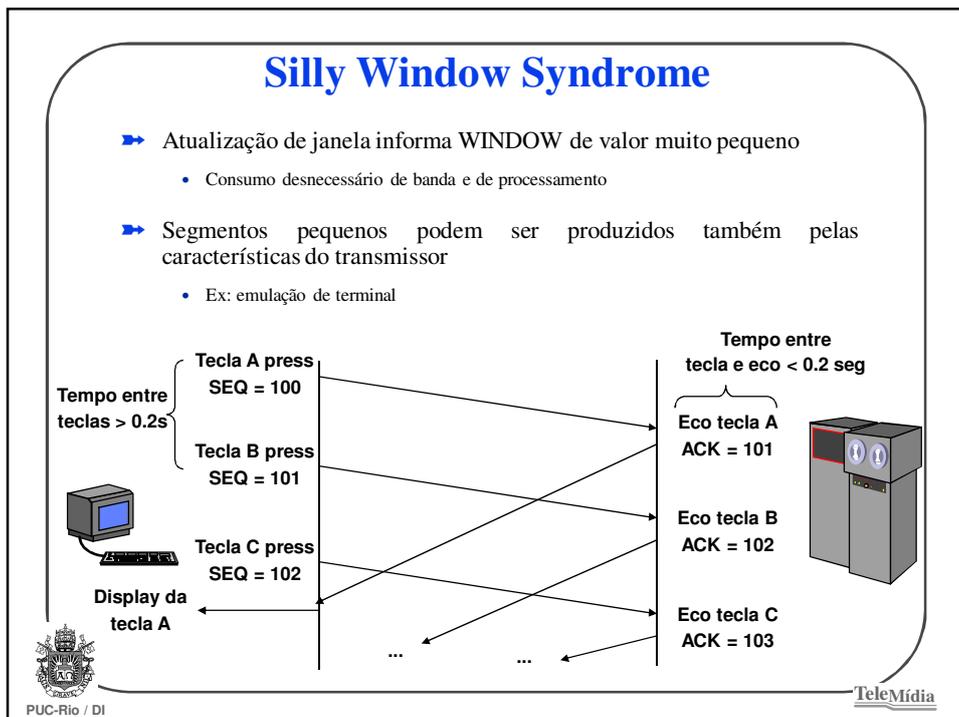
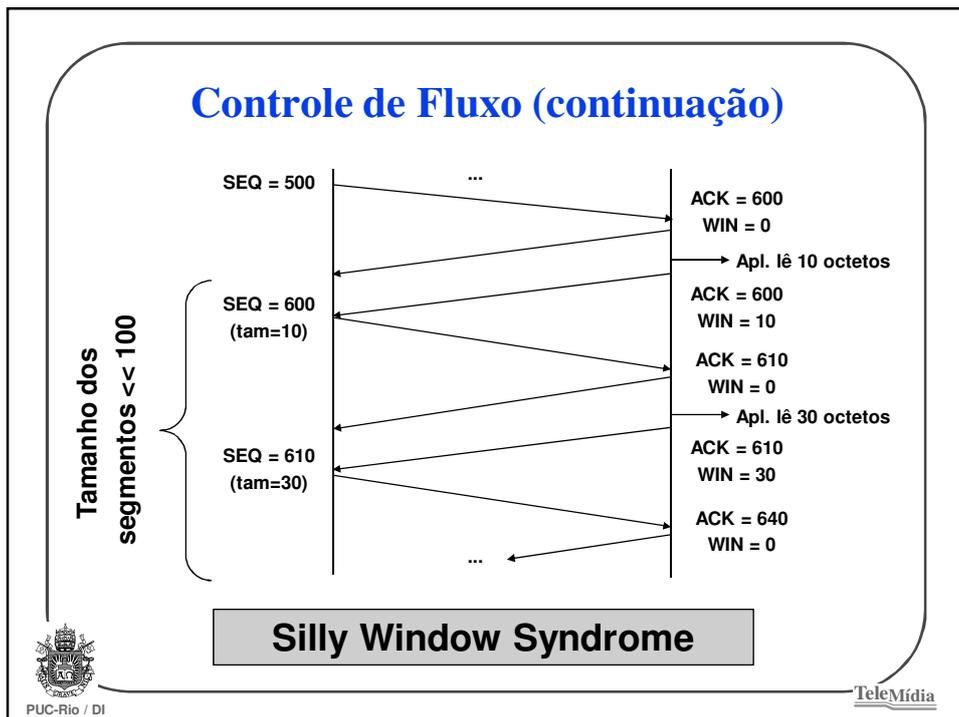
TeleMídia

Controle de Fluxo



PUC-Rio / DI

TeleMídia



Silly Window Syndrome

➔ Solução no transmissor: algoritmo de Nagle

- Quando uma aplicação gera novos dados a serem enviados em uma conexão na qual dados anteriores foram transmitidos, mas não reconhecidos, os novos dados são bufferizados, só sendo enviados quando:
 - for possível completar um segmento de tamanho máximo
- OU
- os dados anteriores forem reconhecidos
- Eficiente em uma inter-rede com retardo alto
- RFC 1122 sugere desabilitar o algoritmo de Nagle em transmissões com interatividade próxima a tempo real



PUC-Rio / DI

TeleMídia

Silly Window Syndrome

➔ Solução no receptor

- Só enviar uma atualização de janela informando $WINDOW > 0$ posteriormente a uma atualização de janela informando $WINDOW = 0$ quando buffer de recepção estiver com espaço livre igual a:
 - pelo menos 50% do buffer
- OU
- espaço correspondente a um segmento de tamanho máximo



PUC-Rio / DI

TeleMídia

Controle de congestionamento

- Controle de fluxo do TCP é fim-a-fim, mas pode ocorrer congestionamento em qualquer roteador
- Sliding Window não garante que o TCP contribua no controle de congestionamento
 - Roteadores na iminência de congestionamento
 - Aumento do RTT
 - Retransmissões devido a reconhecimentos atrasados
 - Aumento da carga nos roteadores congestionados
 - Descarte de datagramas (dados e ACKs) em roteadores congestionados



PUC-Rio / DI

Colapso de congestionamento

TeleMídia

Colapso de congestionamento

- Solução: TCP assume que grande parte das perdas de segmentos se deve a congestionamento
- É associado ao transmissor uma janela de congestionamento, que limita a transmissão de segmentos
 - Janela de congestionamento inicial: 1 segmento
 - Vazão da conexão cresce com o aumento da janela de congestionamento, que em condições normais mantém seu tamanho igual ao da janela de recepção do receptor
- Em caso de retransmissão, a janela de congestionamento é reduzida ao seu tamanho inicial
 - Algoritmo de Karn é utilizado exponencialmente
 - $(N = \text{timeout atual})$



PUC-Rio / DI

TeleMídia

Como aumentar a vazão da conexão ?

➤ Algoritmo de Slow Start

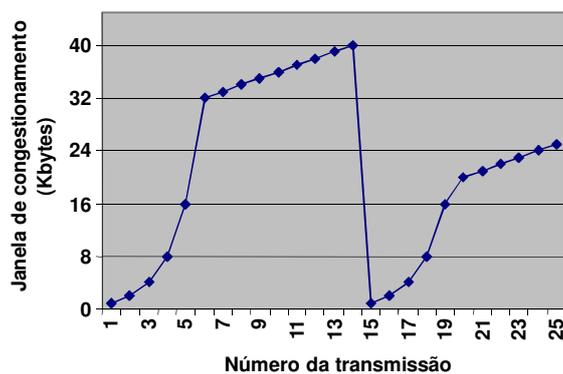
- Para cada reconhecimento recebido (de um segmento não retransmitido), a janela de congestionamento é aumentada de um segmento (de tamanho máximo)
- Em condições “ideais”, o algoritmo de Slow Start pode levar a um crescimento exponencial da janela de congestionamento
 - Sobrecarga na rede pode reiniciar congestionamento
 - Solução: técnica de prevenção contra congestionamento
 - Quando a janela de congestionamento atinge metade de seu tamanho anterior à última retransmissão, o TCP passa a aumentar a janela de congestionamento somente quando TODOS os segmentos internos à janela forem reconhecidos



PUC-Rio / DI

TeleMídia

Slow Start



Tamanho dos segmentos = 1Kb
Janela de Recepção = 64 Kb



PUC-Rio / DI

TeleMídia

Slow Start

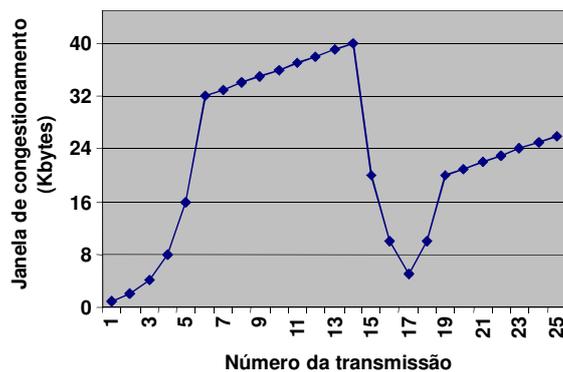
- Algoritmo de Slow Start reduz drasticamente a vazão de uma conexão TCP
- Melhora: técnica de decréscimo multiplicativo
 - Em caso de retransmissão, a janela de congestionamento é reduzida pela metade, até o mínimo de um segmento de tamanho máximo



PUC-Rio / DI

TeleMídia

Slow Start + Decréscimo Multiplicativo



Tamanho dos segmentos = 1Kb
Janela de Recepção = 64 Kb



PUC-Rio / DI

TeleMídia

TCP

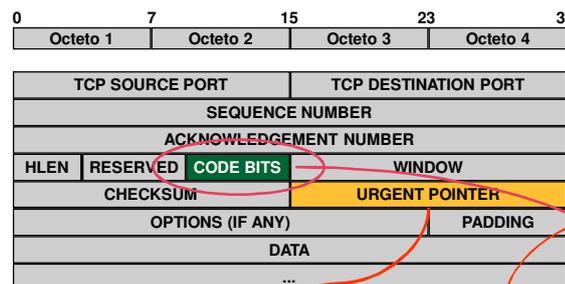
- ➔ O algoritmo de *three-way handshake* é utilizado na abertura de conexões.
- ➔ TCP transfere uma cadeia (stream) contínua de octetos, nas duas direções, entre seus usuários. Normalmente o TCP decide o momento de parar de agrupar os octetos e de, conseqüentemente, transmitir o segmento formado por esse agrupamento. Porém, caso deseje, o usuário do TCP pode fazer uso da função *push* que faz com que o TCP transmita imediatamente os octetos que estão nos seus buffers aguardando transmissão.



PUC-Rio / DI

TeleMídia

Segmento TCP



Uma vez que uma conexão tenha sido estabelecida, este flag permanece sempre setado

CODE BITS (6 bits)

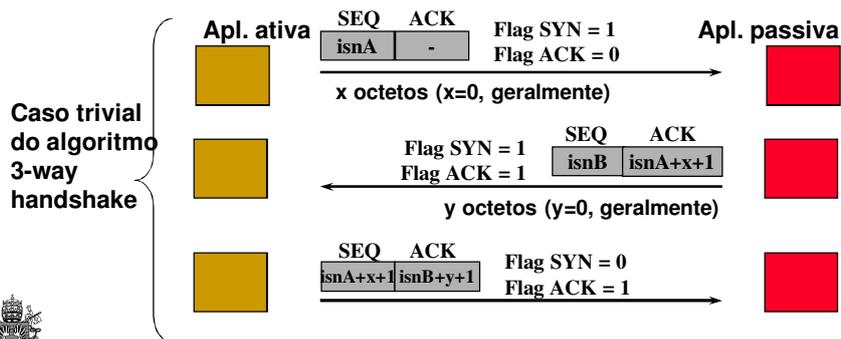


PUC-Rio / DI

TeleMídia

Estabelecimento de conexões

- ➔ SEQ carrega número de seqüência inicial (ISN) que cada módulo TCP vai utilizar para sequenciar segmentos TCP
 - ISN incrementado a cada $4 \mu s$
- ➔ ACK permite que dados do usuário sejam enviados junto ao pedido de estabelecimento (reconhecimento na confirmação da conexão)



PUC-Rio / DI

TeleMídia

Timeout de Estabelecimento

- ➔ Timeout para segmentos SYN é igual ao timeout inicial (3 seg)
- ➔ Maioria das implementações permite mais de uma tentativa
 - Algoritmo de Karn
 - Windows NT: máximo de 5 tentativas

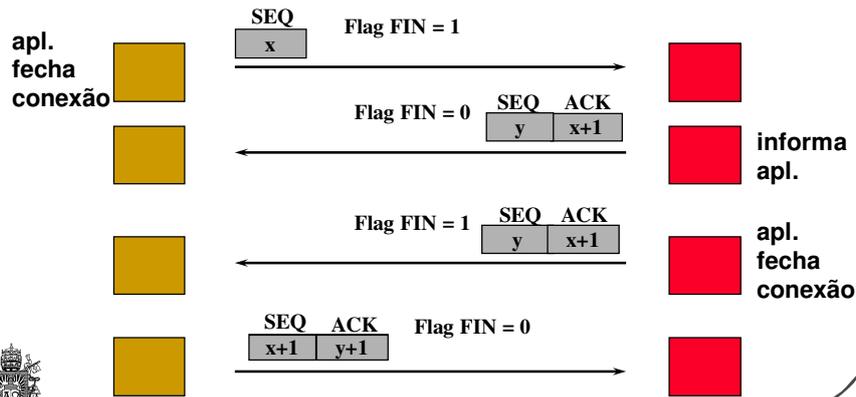


PUC-Rio / DI

TeleMídia

Encerramento de conexões

- Conexão full-duplex: fluxos são encerrados de modo independente
- Na prática, o encerramento dos fluxos costuma ocorrer consecutivamente

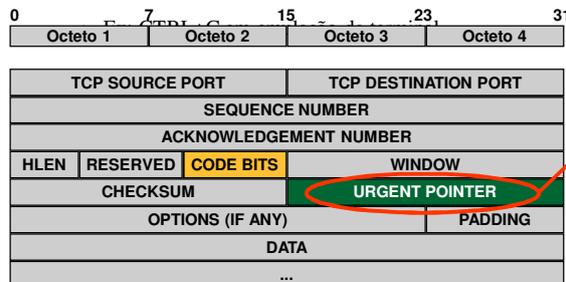


PUC-Rio / DI

TeleMídia

Processamento de dados urgentes

- Flag URG do campo CODE BITS setado
- Dados urgentes são, em geral, informações de controle transmitidas junto com os dados propriamente ditos (sinalização in-band).



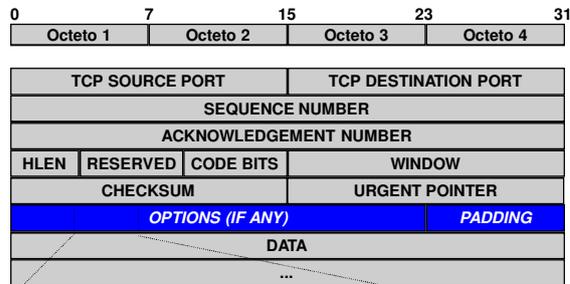
Delimita dados urgentes no segmento



PUC-Rio / DI

TeleMídia

Campo TCP Options



PUC-Rio / DI

TeleMídia

Opções TCP Principais

Código	Comp.	Descrição
0	-	Fim da lista de opções.
2	4	Maximum Segment Size (válido apenas em segmento com flag SYN = 1)
3	3	Window Scale Factor (aumenta as janelas de transmissão de 16 p/ 32 bits. Válido apenas em segmento com flag SYN =1)
4	2	Selective ACK Permitted (permite ao transmissor retransmitir SOMENTE segmentos perdidos. Válido apenas em segmento com flag SYN=1)
5	Variable	Selective ACK (identifica segmentos a serem retransmitidos seletivamente)
8	10	Timestamp (permite obter vários NEW_RTT_SAMPLE por janela, mesmo utilizando retransmissão em lote).



PUC-Rio / DI

TeleMídia

Perguntas

- Quais as funções realizadas pelo protocolo UDP e como são realizadas?
- O que é o UDP Pseudo-Header? Para que serve?
- O que é um socket? Qual sua relação com o T-SAP?
- Como é identificada uma conexão TCP? Um socket pode participar de mais de uma conexão?
- Por que o TCP é dito orientado a “stream”? O que vem a ser o “push”?
- Como são numerados os pacotes TCP?
- Como é realizado o controle de erro no TCP?
- Como é realizado o controle de fluxo no TCP? O que é a janela de recepção e a janela de transmissão?
- Existe uma única implementação TCP? Discuta as diversas políticas de implementação?



PUC-Rio / DI

TeleMídia

Perguntas

- É necessário que o TCP retransmita exatamente o mesmo pacote no caso de timeout?
- Quais os problemas que podem acontecer com a escolha de um timeout pequeno? E de um timeout grande?
- Explique o algoritmo de cálculo de time-out através de vários cenários de exemplos. Por que é necessário um valor mínimo inicial e um valor máximo?
- Explique o problema de ambiguidade de reconhecimento. Como pode ser resolvido?
- Explique o algoritmo de Karn.
- Quais os problemas que podem acontecer com relação ao controle de fluxo? Cite pelo menos 3 e explique.
- Como resolver o problema da perda da informação da janela?
- Como resolver o problema denominado “silly window”, tanto do receptor quanto do transmissor?



PUC-Rio / DI

TeleMídia

Perguntas

- ▶ Por que o TCP não funciona bem em redes com alta taxa de erro?
- ▶ O que é janela de congestionamento? Qual seu limite máximo?
- ▶ Explique o algoritmo de “slow start”. Por que ele pode reiniciar uma condição de congestionamento? Como pode ser reduzido o problema?
- ▶ O que vem a ser decréscimo multiplicativo? Por que ele é necessário?
- ▶ Para que serve o campo “code bits” do TCP? Explique o “three-way-handshake” utilizando esse campo. Explique também o encerramento de conexões.



PUC-Rio / DI

TeleMídia

API de Transporte



PUC-Rio / DI

TeleMídia

Servidores de Processo e de Nome

- ➔ Existem várias formas de um processo descobrir o T-SAP do processo de destino com quem deseja se comunicar.
- ➔ O servidor de processos e o servidor de nomes são duas dessas formas.



PUC-Rio / DI

TeleMídia

Servidor de Processo

- ➔ Servidor de processos (ou logger) especial, conectado a um T-SAP bem conhecido, através do qual todos os serviços devem ser solicitados.
- ➔ Quando um processo cliente deseja algum serviço, ele deve começar pelo estabelecimento da conexão com o servidor de processos (com o T-SAP do servidor de processos).
- ➔ Uma vez que a conexão tenha sido estabelecida, o cliente manda ao servidor de processos o nome do serviço que deseja.
- ➔ Servidor de processo escolhe um T-SAP livre na estação e gera o processo do serviço requisitado dizendo a esse processo para escutar o T-SAP escolhido.
- ➔ Servidor de processos manda uma mensagem ao cliente remoto contendo o T-SAP escolhido e encerra a conexão.



PUC-Rio / DI

TeleMídia

Servidor de Nomes

- Nesse modelo existe um processo especial chamado servidor de nomes.
- Na realidade o servidor de nomes pode ser implementado de forma distribuída, com um processo em cada estação.
- Usuário cliente estabelece uma conexão com o servidor de nomes, que está vinculado a um T-SAP conhecido.
- Cliente envia uma mensagem solicitando o endereço do serviço, recebendo como resposta o T-SAP requisitado.
- Cliente encerra a conexão com o servidor de nomes e abre uma conexão com o T-SAP do serviço desejado.
- O modelo servidor de nomes exige que cada serviço na rede se cadastre no servidor de nomes, indicando seu T-SAP.



PUC-Rio / DI

TeleMídia

API de Transporte

- Uma interface para programa de aplicação (*Application Program Interface* — API) é uma interface disponível para programadores.
- A disponibilidade de uma API depende tanto do sistema operacional usado, quanto da linguagem de programação.



PUC-Rio / DI

TeleMídia

APIs de Transporte

- ➔ As duas APIs de comunicação mais conhecidas para sistemas Unix são os *Berkeley Sockets* e a TLI (*System V Transport Layer Interface*). Depois estendidas também para outros sistemas operacionais, como o WinSocket.
 - Interfaces desenvolvidas para linguagem C.
- ➔ NetBIOS (Network Basic Input Output System), depois NetBeui, desenvolvida para DOS, Windows, etc.



PUC-Rio / DI

TeleMídia

Berkeley Sockets

- ➔ Uma *associação* (association) é definida como uma quintupla {protocolo, endereço local, processo local, endereço remoto, processo remoto}.
- ➔ A *socket System Call* especifica apenas o protocolo. A chamada tem como argumentos de entrada uma família (no nosso caso específico definindo o uso dos protocolos TCP/IP) e um tipo (no nosso caso específico indicando se é um socket para TCP, ou UDP, ou IP). A chamada tem como retorno um valor inteiro, referido como descritor do socket: *sockfd*.



PUC-Rio / DI

TeleMídia

Berkeley Sockets

- ➔ Uma vez criado um socket, podemos associar um endereço (TCP, UDP ou IP) a ele, através da chamada *bind*, que tem como argumentos o *sockfd* e um ponteiro para o endereço, que como vimos, no ambiente TCP/IP se constitui de: <endereço de porta, endereço de rede, endereço de estação>.
- ➔ Quando na chamada *bind* o endereço de porta é especificado como zero, cabe ao sistema escolher um T-SAP ocioso, isto é, um endereço ocioso, e associá-lo ao socket. A chamada *bind* completa o processo local e o endereço local de uma associação.



Berkeley Sockets

- ➔ A chamada *getsockname* retorna o endereço local e o processo local associado ao *sockfd*, passado como argumento de entrada.
- ➔ A chamada *listen* é usada para alocação de um espaço para o armazenamento de pedidos de conexão. A chamada tem como argumentos de entrada o *sockfd* e o número de pedidos de conexão que devem ser aceitos.



Berkeley Sockets

- ➔ A chamada *getpeername* é semelhante à chamada *getsockname*. A diferença é que ela retorna o <endereço remoto, processo remoto> associado ao *sockfd*, passado como argumento de entrada.
- ➔ A chamada *connect* serve para estabelecer uma conexão entre dois sockets.
- ➔ A chamada tem como argumentos de entrada o *sockfd*, o processo remoto e o endereço remoto.
- ➔ A interface não exige que tenha sido realizada antes a chamada *bind*. Se não foi, a própria chamada *connect* atribuirá um endereço local e porta local ao *sockfd*, caso tenha-se escolhido na chamada *socket* um serviço com conexão. Nesse caso, após a chamada *connect* toda a quintupla da associação, {protocolo, endereço local, processo local, endereço remoto, processo remoto}, estará formada.



Berkeley Sockets

- ➔ Um serviço sem conexão também pode utilizar a chamada *connect*.
- ➔ Nesse caso, a única tarefa realizada pela chamada é o armazenamento da associação formada, para que o sistema possa saber no futuro a quem encaminhar os dados escritos no *sockfd*, e de quem receber os datagramas futuros enviados ao *sockfd*.



Berkeley Sockets

- ➔ A chamada *accept* deve ser realizada sempre depois de uma chamada *listen*, que é capaz apenas de armazenar pedidos de conexão vindos de outros processos.
- ➔ *Accept* toma a primeira requisição de conexão da lista e cria um novo socket, com as mesmas propriedades do *sockfd*. Dessa forma, uma única porta pode ser usada para o estabelecimento de múltiplas conexões.
- ➔ Após a realização da conexão, a chamada retorna o novo *sockfd* criado, o endereço remoto e o processo remoto. Caso a conexão não tenha sucesso, é retornada uma indicação de erro, junto com o endereço remoto e o processo remoto.



Berkeley Sockets

- ➔ As chamadas *send*, *sendto*, *recv* e *recvfrom* são usadas para envio e recebimento de dados. Todas passam como argumento o *sockfd*.
- ➔ A chamada *sendto* tem também como argumento o endereço e processo de destino dos dados.
- ➔ A chamada *recvfrom* tem também como argumento o endereço e processo de origem dos dados. Todas essas chamadas podem ser usadas nos protocolos com e sem conexão.

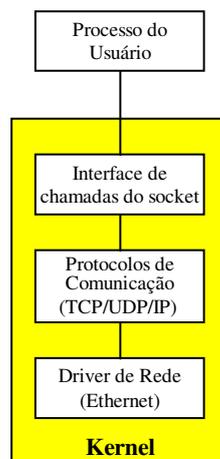


Berkeley Sockets

- ➔ A chamada *close* é usada para eliminar um socket. Se existir uma conexão associada ao socket ela é encerrada.
- ➔ No sistema 4.3BSD, sockets são implementados dentro do *kernel* do Unix. Todas as chamadas do sistema que discutimos são pontos de entrada no kernel.



Sockets



Protocolos de Aplicação TCP/IP

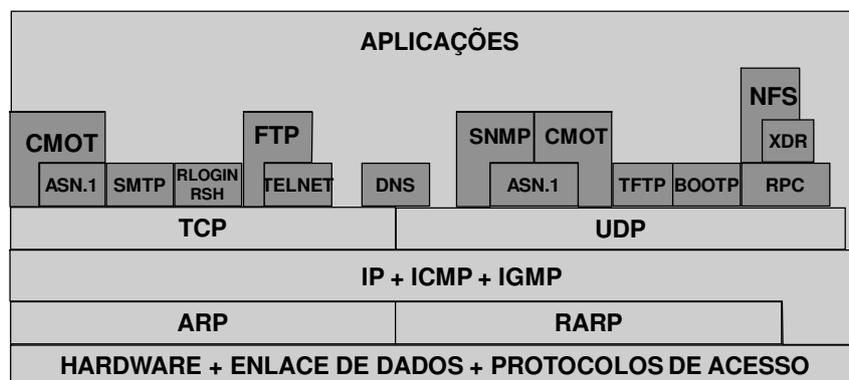
- FTP Transferência de Arquivos
- TELNET Emulação de Terminal
- NFS Sistema de Arquivos Distribuído
- SMTP Correo Eletrônico
- SNMP Gerenciamento
- DNS Serviço de Diretório
- RPC Chamada Remota de Procedimentos



PUC-Rio / DI

TeleMídia

Protocolos e Inter-Dependência



PUC-Rio / DI

TeleMídia

NetBIOS - Network Basic Input Output System

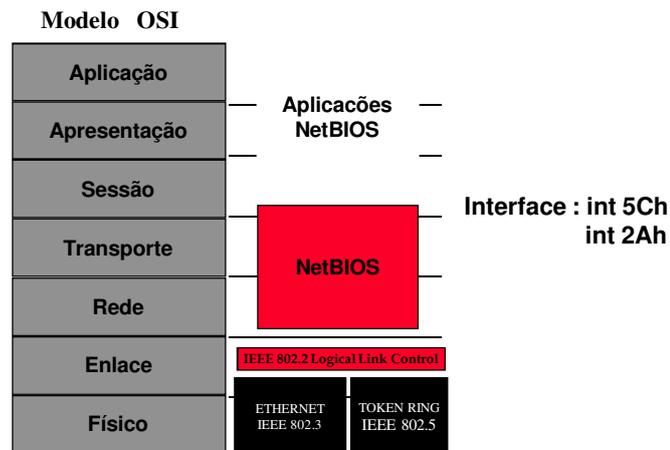
- Desenvolvido pela IBM
- Extensão da BIOS para serviços de comunicação através de uma rede local
- Oferece uma interface de baixo nível para programação de aplicações distribuídas
- Conjunto padronizado de serviços acessados através das interrupções de software 5Ch e 2Ah
- Toda troca de dados entre aplicação e NetBIOS é realizada através de uma estrutura de dados padrão chamada NCB (Network Control Block)



PUC-Rio / DI

TeleMídia

NetBIOS



PUC-Rio / DI

TeleMídia

Comandos NetBIOS

Comando	Cód	Descrição
ADAPTER STATUS	33h	Obtém informações sobre uma interface de rede local ou remota
CANCEL	35h	Cancela comandos pendentes
RESET	32h	Limpa tabelas de nomes e sessões

Comandos Gerais

Comando	Cód	Descrição
ADD GROUP NAME	36h	Adiciona um nome de grupo na tabela de nomes local
ADD NAME	30h	Adiciona um nome na tabela de nomes local
DELETE NAME	31h	Remove um nome da tabela de nomes local
FIND NAME	F8h	Determina a localização de um nome na rede

Comandos de suporte a Nomes Lógicos



PUC-Rio / DI

TeleMídia

Comandos NetBIOS (cont.)

Comandos de suporte ao Serviço de Sessões

Comando	Cód	Descrição
CALL	10h	Cria uma conexão lógica com uma aplicação local ou remota
CHAIN SEND	17h	Envia dois buffers de dados através de uma sessão estabelecida
HANG UP	12h	Finaliza uma conexão lógica
LISTEN	11h	Aceita pedidos de conexões lógicas
RECEIVE	15h	Recebe dados através de uma conexão lógica específica
RECEIVE ANY	16h	Recebe dados de qualquer conexão lógica existente
SEND	14h	Envia dados através de uma conexão lógica
SESSION STATUS	34h	Recebe informações de estado de todas conexões lógicas ativas



PUC-Rio / DI

TeleMídia

Comandos NetBIOS (cont.)

Comandos de suporte ao serviço de Datagramas

Comando	Cód	Descrição
RECEIVE BROADCAST DATAGRAM	23h	Recebe um datagrama de qualquer estação da rede
RECEIVE DATAGRAM	21h	Recebe um datagrama de uma estação específica da rede
SEND BROADCAST DATAGRAM	22h	Envia um datagrama para todas estações da rede
SEND DATAGRAM DATAGRAM	20h	Envia um datagrama para uma estação específica da rede



PUC-Rio / DI

TeleMídia

Network Control Block - NCB

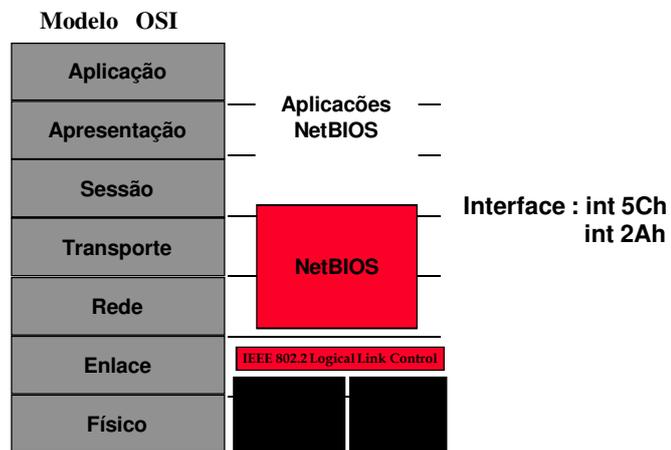
Campo	Tam	Descrição
NCB_COMMAND	1	Código do comando NetBIOS desejado
NCB_RETCODE	1	Código de retorno do comando
NCB_LSN	1	Número da sessão local (retornado por CALL ou LISTEN)
NCB_NUM	1	Identifica um nome local (retornado por ADD [GROUP] NAME)
NCB_BUFFER	4	Ponteiro para um buffer de dados (segment:offset)
NCB_LENGTH	2	Tamanho do buffer de dados
NCB_CALLNAME	16	Identifica o nome do destinatário (aplicação remota)
NCB_NAME	16	Identifica o nome do remetente (aplicação local)
NCB_RTO	1	Valor do timeout para recepção
NCB_STO	1	Valor do timeout para transmissão
NCB_POST	4	Rotina callback executada ao final de um comando assíncrono
NCB_LANA_NUM	1	LAN Adapter Number - identificação da placa adaptadora
NCB_CMD_CPLT	1	Status do comando (FFh - Pendente, 00h - Sucesso, outros - Erro)
NCB_RESERVE	14	Área reservada



PUC-Rio / DI

TeleMídia

NetBIOS



Nível de Sessão, Apresentação e Aplicação



RM-OSI - Protocolos de Alto Nível

- ➔ O RM-OSI ainda define mais três camadas em sua arquitetura. No entanto, na prática, poucas arquiteturas de redes atuais seguem essa estruturação, que passa assim a ganhar apenas o aspecto didático de apresentar quais funções são ainda necessárias.
- ➔ Como o nível de transporte, essas camadas só existem fim a fim.



Nível de Sessão



Nível de Sessão

- O objetivo do nível de sessão é fornecer os meios necessários para que seus usuários (normalmente entidades do nível de apresentação) possam organizar e sincronizar seus diálogos e gerenciar suas trocas de dados.
- Operando no modo orientado à conexão, o nível de sessão provê os serviços necessários para que seus usuários possam estabelecer conexões, transmitir e receber dados, e encerrar conexões de forma ordenada.
- No modo não orientado à conexão o nível de sessão mapeia endereços de transporte em endereços de sessão, repassando os dados que recebe dos seus usuários para o nível de transporte (também operando no modo não-orientado à conexão), e vice-versa. Em outras palavras, suas funções são quase nulas.



PUC-Rio / DI

TeleMídia

Conexão de Sessão

- Uma conexão de sessão é mapeada em uma única conexão de transporte em um dado instante.
- A duração das conexões de sessão e transporte podem ser diferentes.
 - Uma conexão de transporte pode suportar várias conexões de sessão consecutivas, ou uma conexão de sessão pode estender-se sobre várias conexões de transporte.
 - A possibilidade de uma conexão de sessão estender-se sobre mais de uma conexão de transporte permite que o serviço de sessão recupere erros na rede que provoquem o encerramento de conexões de transporte.

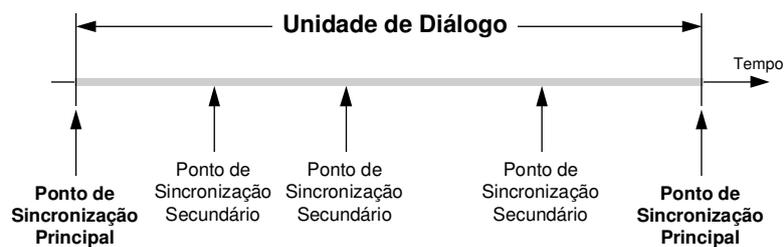


PUC-Rio / DI

TeleMídia

Ponto de Sincronização Principal

- ➔ O *ponto de sincronização principal* é usado para estruturar a troca de dados em unidades de diálogo. Dois pontos de sincronização principais adjacentes definem uma *unidade de diálogo*.



PUC-Rio / DI

TeleMídia

Ponto de Sincronização Principal

- ➔ Ao definir um ponto de sincronização principal, o usuário suspende a transmissão de dados até que o ponto de sincronização seja reconhecido pelo seu parceiro no diálogo.
- ➔ O ponto de sincronização principal marca o limite para recuperação de dados, não sendo possível recuperar dados enviados antes do último ponto de sincronização principal transmitido e reconhecido.



PUC-Rio / DI

TeleMídia

Ponto de Sincronização Secundário

- Os pontos de sincronização secundários são usados para estruturar a troca de dados dentro de uma unidade de diálogo.
- Eles aumentam a flexibilidade em termos de recuperação de falhas.
- Após transmitir um ponto de sincronização secundário, o transmissor não precisa suspender a transmissão de novos dados para esperar pelo reconhecimento do ponto de sincronização secundário.
- A qualquer momento, é possível resincronizar o diálogo em qualquer um dos pontos de sincronização secundários dentro da unidade de diálogo corrente, obviamente também é possível retomar o diálogo a partir do seu início, que é definido por um ponto de sincronização principal.
- Fica a cargo da aplicação decidir entre a sobrecarga envolvida na emissão e confirmação de pontos de sincronização e a diminuição do tempo de recuperação caso seja necessária uma resincronização do diálogo.



PUC-Rio / DI

TeleMídia

Atividade

- Cada atividade consiste em uma ou mais unidades de diálogo.



PUC-Rio / DI

TeleMídia

Atividade

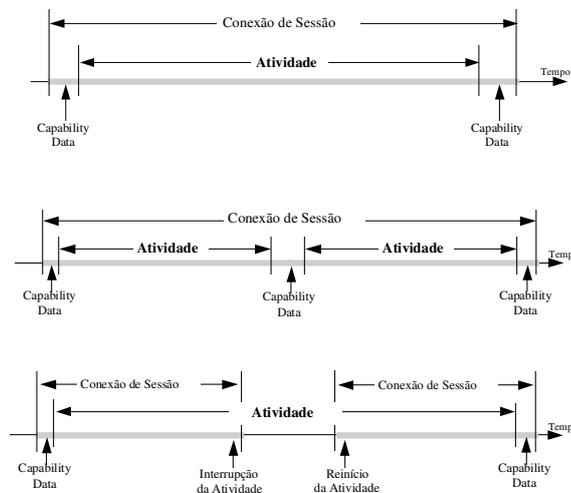
- Atividades podem ser interrompidas e posteriormente retomadas na mesma conexão ou em conexões de sessão subsequentes.
- Uma atividade pode ser usada, por exemplo, para suspender uma transferência em curso, de um arquivo muito longo (atividade A), para que seja transferido um arquivo de maior prioridade (atividade B).
- Esse conceito também pode ser usado quando é necessário suspender uma atividade muito longa, por exemplo, no final do expediente de trabalho, para retomá-la no início do próximo expediente.
- Cabe salientar que o serviço de sessão simplesmente guarda o último ponto de sincronização (principal ou secundário) confirmado, permitindo que a atividade seja retomada a partir desse ponto. Cabe ao usuário do serviço de sessão armazenar qualquer outro tipo de informação necessária ao reinício da atividade suspensa.



PUC-Rio / DI

TeleMídia

Atividades e Sessões



PUC-Rio / DI

TeleMídia

Token

- ▶ Um *token* é um recurso de uma conexão de sessão que é dinamicamente atribuído a um usuário em um dado instante do tempo, concedendo a esse usuário o direito exclusivo de executar determinados serviços.
- ▶ Quatro tokens são definidos:
 - *Dados*: usado para controlar o direito de transmitir dados.
 - *Sincronização secundária*: define o usuário que tem direito de emitir pontos de sincronização secundários.
 - *Sincronização principal/Atividade*: é usado para gerenciar a emissão de pontos de sincronização principal e a estrutura das atividades.
 - *Encerramento de conexão*: define quem tem o direito de encerrar a conexão de sessão.



Nível de Apresentação



Nível de Apresentação

- ➔ A função do nível de apresentação é compatibilizar a forma como são representadas as informações intercambiadas pelos usuários, realizando transformações adequadas nos dados, antes de seu envio ao nível de sessão.
- ➔ Transformações típicas dizem respeito à compressão de textos, criptografia, conversão de padrões de terminais e arquivos para padrões de rede e vice-versa.



PUC-Rio / DI

TeleMídia

Sintaxes

- ➔ Face à complexidade das estruturas intercambiáveis por diversas aplicações, surgem duas questões.
 - Como podem ser definidas as estruturas de dados a serem intercambiadas?
 - Como pode uma instância de uma estrutura (p.ex., uma transferência bancária específica) ser intercambiada e compreendida por diferentes sistemas, independentemente da plataforma de hardware ou software utilizadas?
- ➔ A descrição das estruturas de dados dos objetos a serem intercambiados.
- ➔ Uma codificação não ambígua para instâncias dessas estruturas.



PUC-Rio / DI

TeleMídia

Sintaxe Abstrata

- ▶ A *sintaxe abstrata* para uma estrutura de dados é uma especificação da organização dessa estrutura de forma independente da codificação interna utilizada por máquinas para representá-la.
- ▶ Uma sintaxe abstrata torna possível a definição de tipos de dados e permite também que sejam atribuídos valores aos tipos de dados.
- ▶ Ao invés de se utilizar de uma linguagem de programação, a ISO definiu uma linguagem própria para a especificação de estruturas de dados de forma abstrata, linguagem esta padronizada e denominada *Abstract Syntax Notation One* (ASN.1).



ASN.1

```
RegFunc ::= [APPLICATION 0] IMPLICIT SET {  
    Nome,  
    Idade    [1]  IMPLICIT INTEGER,  
    Profissão [2] VisibleString }  
  
Nome ::= [APPLICATION 1] IMPLICIT SEQUENCE {  
    UltimoNome [1] VisibleString,  
    Complemento [2] VisibleString OPTIONAL,  
    PrimeiroNome [3] VisibleString }
```



Sintaxe Concreta

- ➔ Definida uma representação abstrata para uma estrutura de dados, várias *representações concretas* são possíveis.
- ➔ Denomina-se *sintaxe concreta* a especificação de um formato para a codificação de instâncias de estruturas de dados a partir de uma sintaxe abstrata.



PUC-Rio / DI

TeleMídia

Representações Concretas do Número Inteiro 534

Esquema de Codificação	Endereço Menor	Endereço Maior
Representação binária usada pelo processador Motorola 68.000	0000 0010	0001 0110
Representação binária usada pelo processador Intel 8086	0001 0110	0000 0010
Representação binária codificada em decimal (BCD) usada no IBM 370	0000 0101	0011 0100
Representação ASCII usada em microcomputadores	0011 0101	0011 0011 0011 0100
Representação EBCDIC usada em computadores de grande porte	1111 0101	1111 0011 1111 0100



PUC-Rio / DI

TeleMídia

Sintaxe Concreta

- ➔ Definida uma representação abstrata para uma estrutura de dados, várias *representações concretas* são possíveis.
- ➔ Denomina-se *sintaxe concreta* a especificação de um formato para a codificação de instâncias de estruturas de dados a partir de uma sintaxe abstrata.
- ➔ Para a definição de uma sintaxe concreta para o padrão ASN.1, a ISO definiu o padrão ASN.1 BER (ASN.1 Basic Encoding Rules) [ISO 90d].



ASN.1 BER

```
60 APPLICATION 0, RegFunc tag (IMPLICIT SET constructed)
36 Length of RegFunc fields (54 decimal)
61 APPLICATION 1, Nome tag (IMPLICIT SEQUENCE constructed)
26 Length of Nome fields (38 decimal)
A1 UltimoNome tag, (constructed)
08 Length
1A VisibleString
06 Length
Xavier UltimoNome Value
A2 Complemento tag, (constructed)
0F Length
1A VisibleString
0D Length
Jose da Silva Complemento Value
A3 PrimeiroNome tag, (constructed)
09 Length
1A VisibleString
07 Length
Joaquim PrimeiroNome Value
81 Idade tag (IMPLICIT INTEGER)
Length
1F Value (decimal 31)
A2 Profissão
0A Length
1A VisibleString
07 Length
Alferes Value
```



Sintaxe de Transferência

- ➔ A sintaxe concreta utilizada para o intercâmbio é denominada *sintaxe de transferência*.
- ➔ Uma possível sintaxe de transferência é a obtida através das regras estabelecidas pelo padrão ASN.1 BER.
- ➔ A principal função do nível de apresentação é realizar o mapeamento e a conversão entre uma representação utilizada para intercâmbio e a representação interna de aplicações.

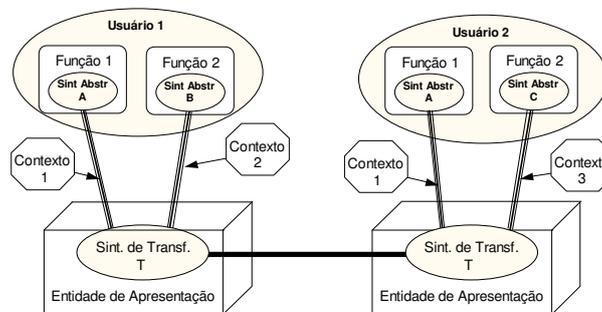


PUC-Rio / DI

TeleMídia

Contexto de Apresentação

- ➔ Em geral, podem ser utilizadas mais de uma combinação de sintaxes abstratas e sintaxes de transferência.
- ➔ Cada combinação de sintaxe abstrata e sintaxe de transferência é denominada *contexto de apresentação*.



PUC-Rio / DI

TeleMídia

Camada de Apresentação

- O nível de apresentação amplia o serviço do nível de sessão, acrescentando basicamente a negociação das sintaxes de transferência utilizadas na comunicação e a transformação das sintaxes usadas pelas aplicações na sintaxes de transferência negociadas.
- Os serviços do nível de sessão (por exemplo, estabelecimento de conexão, transferências de dados, gerenciamento de token, encerramento de conexão etc.) são fornecidos às entidades de aplicação na forma de serviços de apresentação. Quando for esse o caso, a camada de apresentação atua simplesmente repassando as primitivas que dizem respeito aos serviços de sessão entre as entidades de aplicação e de sessão.
- Como não existe multiplexação ou splitting no nível de apresentação, uma conexão de apresentação é mapeada em uma única conexão de sessão.



PUC-Rio / DI

TeleMídia

Nível de Apresentação

- A coleção dos contextos negociados em uma conexão define o conjunto dos contextos da conexão (Defined Context Set DCS).
- Depois do estabelecimento da conexão e, portanto, da definição de um DCS inicial, é possível opcionalmente adicionar-se ou remover-se contextos.
- Dados definidos em termos de uma sintaxe abstrata são representados no ambiente local do sistema por uma sintaxe concreta. Uma transformação entre essa sintaxe concreta e a sintaxe de transferência é necessária para que os dados sejam transferidos.
- Conceitualmente, essa conversão se dá em duas fases: inicialmente a representação concreta local é mapeada na representação abstrata, e depois, a representação abstrata é transformada na representação concreta de transferência.



PUC-Rio / DI

TeleMídia

Nível de Apresentação

- ▶ Quando uma representação abstrata local é transformada na representação concreta de transferência, a entidade de apresentação pode aplicar técnicas de compressão ou compactação, para diminuir o volume dos dados transferidos.
- ▶ A entidade de apresentação pode também utilizar algoritmos de criptografia para evitar que os dados transferidos sejam interpretados por usuários não autorizados.



Nível de Aplicação



Nível de Aplicação

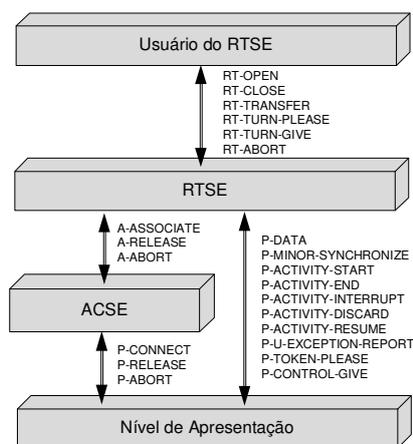
- ➔ O nível de aplicação oferece aos processos de aplicação os meios para que estes utilizem o ambiente de comunicação OSI.
- ➔ Nesse nível são definidas funções de gerenciamento e mecanismos genéricos que servem de suporte à construção de aplicações distribuídas.



PUC-Rio / DI

TeleMídia

Nível de Aplicação



PUC-Rio / DI

TeleMídia

Perguntas

- Como funciona um servidor de processos? É um servidor de nomes?
- O que vem a ser um API de transporte? Dê exemplos.
- O que representa cada parâmetro de uma *associação* (association) {protocolo, endereço local, processo local, endereço remoto, processo remoto}?
- Como pode ser especificado um SAP de transporte específico para uma associação? Como pode der passado para o protocolo de transporte esta responsabilidade? Quando é útil cada uma das opções?
- Detalhe o processo de estabelecimento de conexões usando sockets, explicando cada chamada de sistema realizada e como elas realizam suas funções.
- Para que serve a chamada *connect* em um serviço sem conexão?
- O que é NetBIOS, uma API, um protocolo? Como pode ser implementado em uma rede local isolada (não ligada a outras redes)? Como pode ser implementado em uma rede local ligada a Internet?
- Compare o NetBIOS com a API Socket.



PUC-Rio / DI

TeleMídia

Perguntas

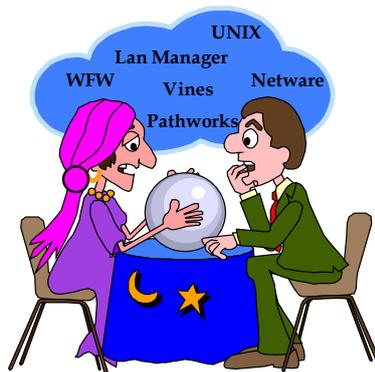
- Como é realizada a chamada às funções do NetBIOS?
- Como é realizado o suporte a nomes lógicos? Como isto é realizado pela API Socket?
- Quais as funções do nível de Sessão?
- Como é realizado o sincronismo em um diálogo? Para que serve? Quando é útil?
- Relacione a duração de conexões do nível de Sessão com suas respectivas conexões de Transporte? Por que elas podem ser diferentes? Quando isto é útil?
- Para que servem os pontos de sincronização secundários? Eles são necessários ou pode se operar só com pontos de sincronização principais?
- Quais as funções do nível de Apresentação?
- O que vem a ser sintaxe abstrata, sintaxe concreta e sintaxe de transferência?
- O que é um contexto de apresentação?
- Baseado nos contextos de apresentação, explica as funções do nível de apresentação.



PUC-Rio / DI

TeleMídia

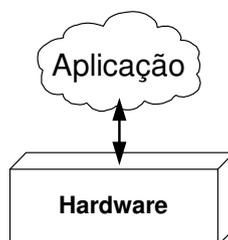
Sistemas Operacionais de Rede



PUC-Rio / DI

TeleMídia

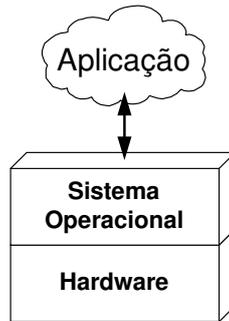
Sistemas Operacionais



PUC-Rio / DI

TeleMídia

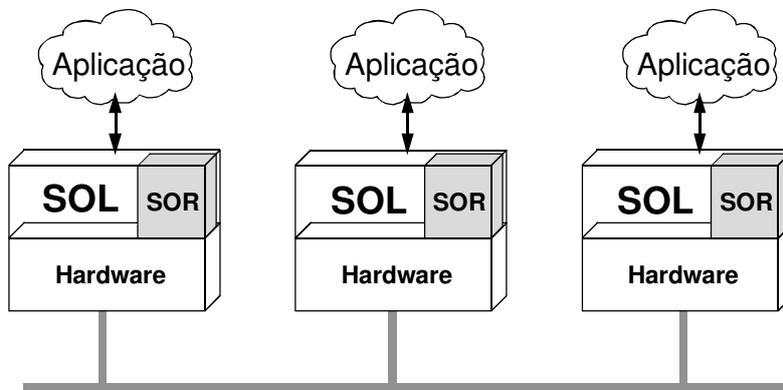
Sistemas Operacionais



PUC-Rio / DI

TeleMídia

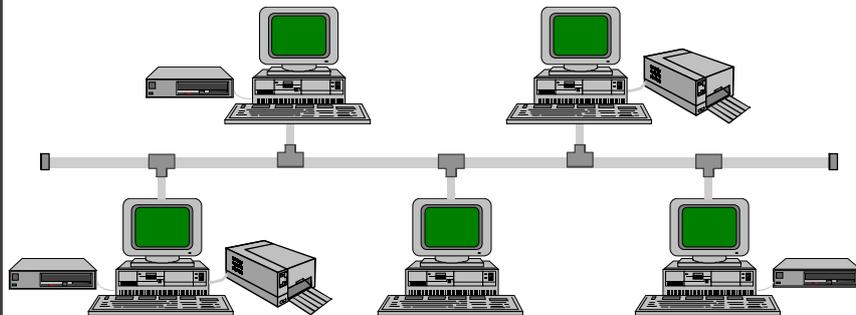
Sistemas Operacionais



PUC-Rio / DI

TeleMídia

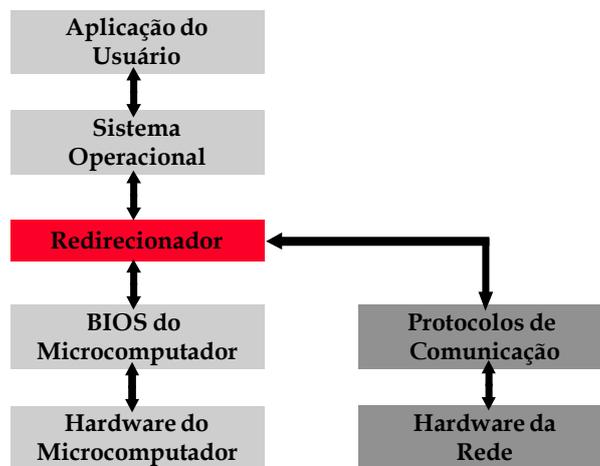
Compartilhamento de Recursos



PUC-Rio / DI

TeleMídia

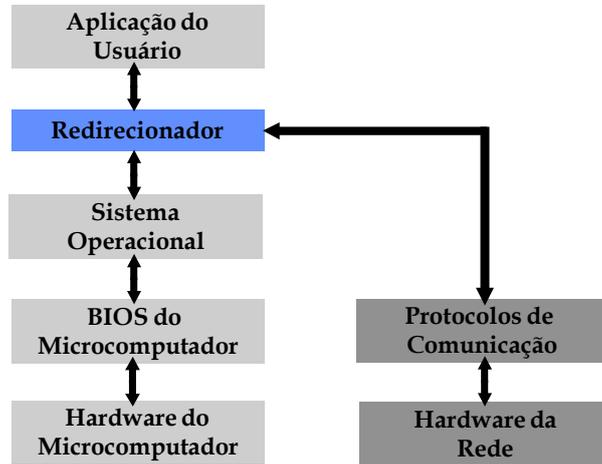
Redirecionamento no Nível de BIOS



PUC-Rio / DI

TeleMídia

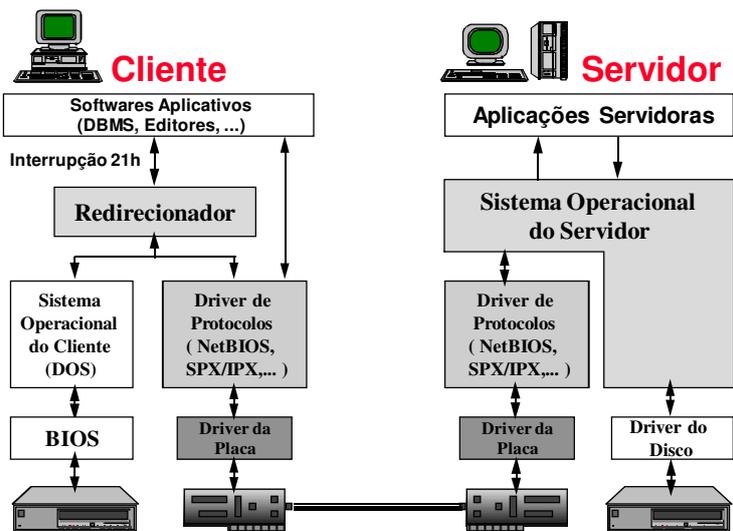
Redirecionamento no Nível de Sistema Operacional



PUC-Rio / DI

TeleMídia

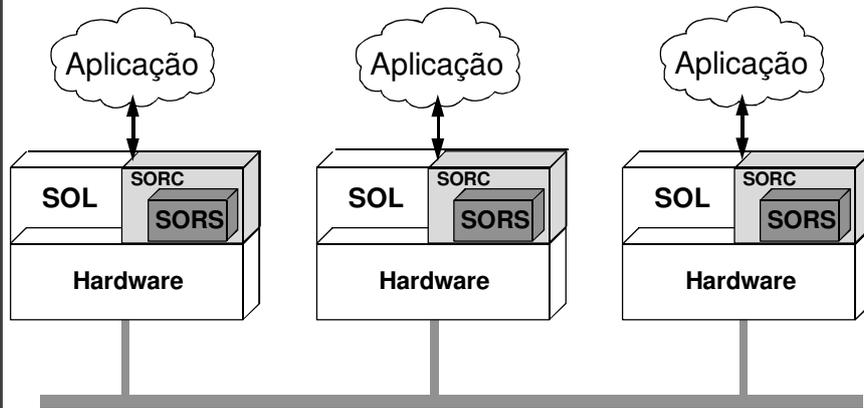
Paradigma Cliente-Servidor



PUC-Rio / DI

TeleMídia

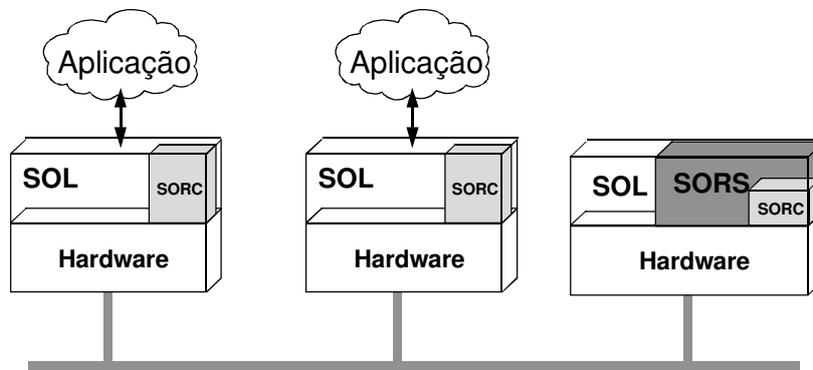
Arquitetura Peer-to-Peer



PUC-Rio / DI

TeleMídia

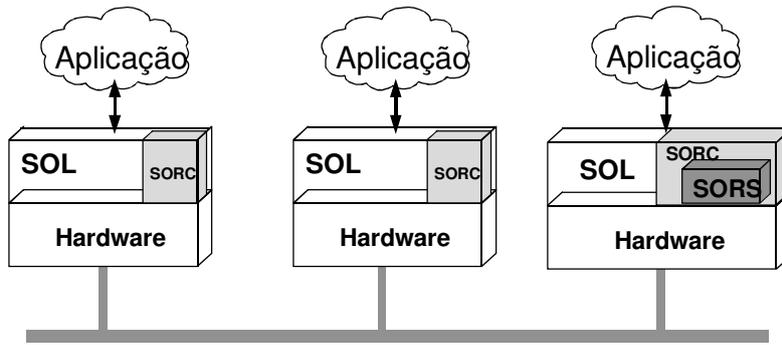
Arquitetura Cliente-Servidor Servidor Dedicado



PUC-Rio / DI

TeleMídia

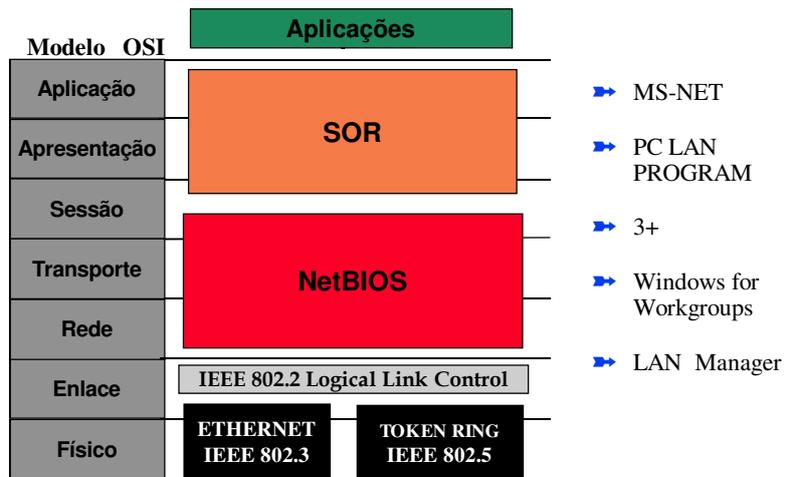
Arquitetura Cliente-Servidor Servidor Não Dedicado



PUC-Rio / DI

TeleMídia

Sistemas Operacionais de Redes



PUC-Rio / DI

TeleMídia

PC LAN PROGRAM

Main Menu - Task Selection

1. Message tasks
2. Printer tasks
3. Disk or directory tasks
4. Print queue tasks
5. Network status tasks
6. Pause and continue tasks
7. Save or cancel the network setup

Choice


PUC-Rio / DI
TeleMídia

PC LAN PROGRAM

Start sharing your Disk or directory

DOS name for disk or directory
C:\word

Network name for your disk or directory
public

Password for disk or directory
abracadabra

1. Read only	4. Write Create Delete
2. Read Write	5. Read Write Create Delete
3. Write only	

Choice **1**

Tab - Cursor to next field

Enter - Continue

Crtl-Home - Return to Mein Menu

F1 - Help

Esc - Previous menu


PUC-Rio / DI
TeleMídia

Exemplos de Comandos Net

```
A> net share c:\word public abracadabra r  
A>
```



PUC-Rio / DI

TeleMídia

PC LAN PROGRAM

Start using a network Disk or directory

DOS name for disk or directory

F:

Network name for your disk or directory

public

Password for disk or directory

abracadabra



PUC-Rio / DI

TeleMídia

Exemplos de Comandos Net

```
A> net use f: public abracadabra  
A>
```



PUC-Rio / DI

TeleMídia

Comandos Net

Comando	Descrição
NET	Ativa o programa menu
NET CONTINUE	Reativa uma função que foi suspensa
NET ERROR	Mostra um log com os erros que ocorreram na rede
NET FILE	Mostra o nome do usuário e os bloqueios correntes de um arquivo
NET FORWARD	Estabelece temporariamente um novo endereço para um usuário
NET LOG	Encerra ou inicia o "logging" de mensagens para um arquivo/disp.
NET NAME	Permite a criação de nomes adicionais para recepção de mensagens
NET PAUSE	Suspende temporariamente a execução de uma função
NET PRINT	Imprime em uma impressora compartilhada
NET SEND	Envia mensagens para outros computadores
NET SEPARATOR	Insere um separador entre "jobs" em uma impressora compartilhada
NET SHARE	Torna os recursos da rede disponíveis para os usuários
NET START	Inicia a operação de um servidor da rede
NET USE	Conecta um computador a um recurso compartilhado



PUC-Rio / DI

TeleMídia

PC LAN PROGRAM

- ➔ Arquivo “.bat” com comandos net
- ➔ Senhas associadas aos recursos compartilhados
- ➔ Servidores não dedicados



PUC-Rio / DI

TeleMídia

Como proteger as senhas e associar os recursos a um usuário específico ?

- ➔ Arquivo com comandos net só executável armazenado em um diretório alocado ao usuário



PUC-Rio / DI

TeleMídia

Modelo de Segurança *Share-Level*

- ➔ Cada recurso compartilhado possui uma ou mais senhas de acesso.
- ➔ O dono do recurso é responsável por administrar suas senhas.
- ➔ Qualquer pessoa que conheça a senha possui acesso ao recurso.
- ➔ Exemplo: PC LAN Program, Windows for Workgroups, Windows 95, Windows98 , e Windows2000.



PUC-Rio / DI

TeleMídia

Modelo de Segurança *User-Level*

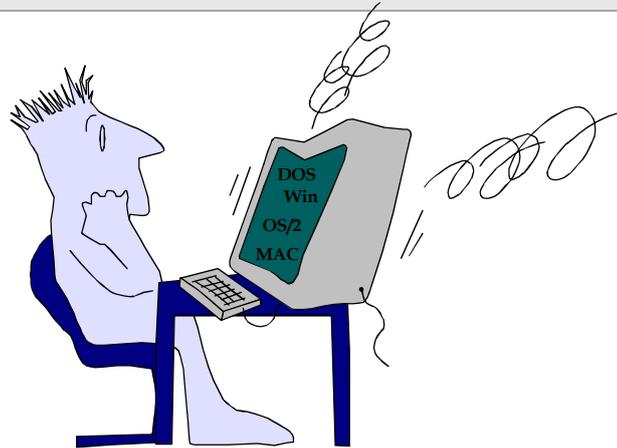
- ➔ Cada usuário possui uma senha pessoal que autoriza seu acesso ao sistema
- ➔ Cada recurso possui uma lista de permissões de acesso
- ➔ Somente os usuários que pertencem à lista de permissões está autorizado a acessar o recurso
- ➔ Cabe ao administrador do sistema gerenciar o cadastro de usuários e as listas de permissões dos recursos compartilhados
- ➔ Exemplo: Netware 3.11, Windows NT



PUC-Rio / DI

TeleMídia

Módulo Cliente



PUC-Rio / DI

TeleMídia

Características

- ➔ Máquinas já existentes que devem ser incorporadas ao novo ambiente de rede
- ➔ Normalmente exigem recursos limitados de hardware:
 - CPU
 - Memória
 - Disco



PUC-Rio / DI

TeleMídia

Serviços acessados através do Módulo Cliente

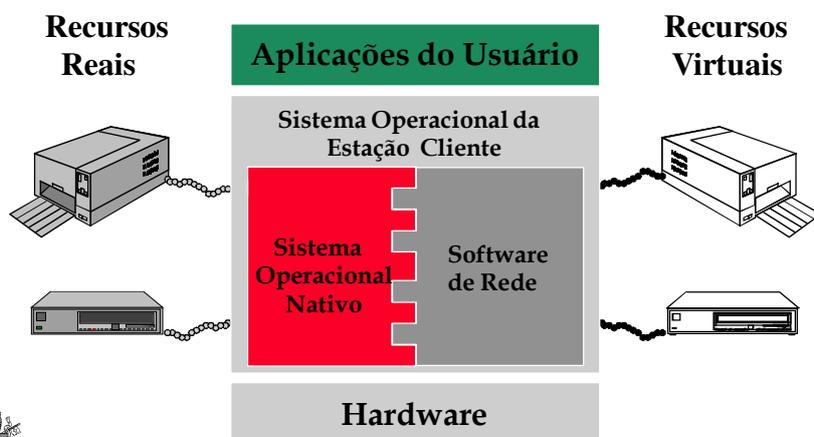
- ▶ Acesso transparente aos recursos remotos e compartilhados:
 - arquivos
 - impressoras
 - outros...
- ▶ Emulação de Terminal
- ▶ Correio Eletrônico
- ▶ Comunicação Interativa
- ▶ Transferência de Arquivos
- ▶ outros ...



PUC-Rio / DI

TeleMídia

Diagrama Geral de um Módulo Cliente



PUC-Rio / DI

TeleMídia

Requisitos do Software de Rede

- ▶ Baixa utilização dos recursos de hardware da estação
- ▶ Acesso aos recursos remotos, através de um software Redirecionador
- ▶ Facilidade de utilização
- ▶ Alta confiabilidade
- ▶ Segurança de acesso e utilização



PUC-Rio / DI

TeleMídia

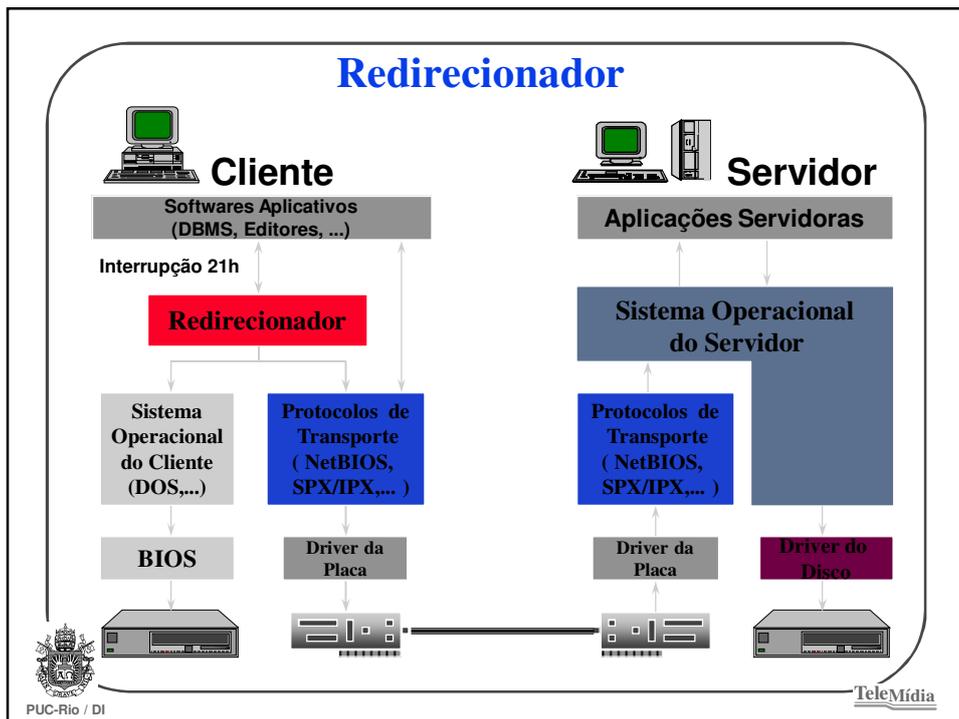
Redirecionador

- ▶ Parte fundamental do Software de Rede.
- ▶ Oferece serviços de acesso aos recursos remotos.
- ▶ Permite que aplicações sejam desenvolvidas sem nenhuma preocupação adicional com a localização física de arquivos ou impressoras.
- ▶ Intercepta os pedidos de serviço solicitados pelas aplicações ao sistema operacional nativo, redirecionando-os para o software de rede, no caso de acesso a recursos remotos



PUC-Rio / DI

TeleMídia



Características

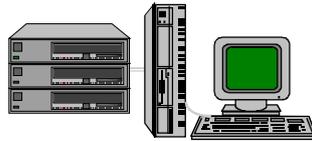
- Máquinas caras
- Hardware específico para cada tipo de serviço oferecido (Ex.: Servidores de Impressão)
- Em alguns casos exige grande poder de processamento (Ex.: Servidores SQL)
- Devem ser bastante confiáveis, pois representam ponto central de falhas



PUC-Rio / DI

TeleMídia

Servidores de Arquivos



- Oferecem serviços de armazenamento e recuperação de arquivos
- Controle de acesso simultâneo aos arquivos compartilhados
- Controle de Backups
- Segurança de acesso
- Técnicas de tolerância a falhas



PUC-Rio / DI

TeleMídia

Subsistemas Funcionais dos Servidores de Arquivos

- Subsistema de Gerência de Arquivos
 - Controlam todo acesso físico aos meios de armazenamento
 - Alguns servidores implementam o seu próprio Subsistema de Gerência de Arquivos a fim de obter melhor eficiência e mais segurança
 - Alguns servidores utilizam os serviços do sistema operacional nativo
- Subsistema de Cache de Disco
 - Aumentam a eficiência do servidor acessando o disco em blocos grandes de dados, mantendo-os em memória RAM, tentando assim diminuir o número de acessos físicos às unidades de armazenamento
- Subsistema de Acesso Compartilhado e Segurança
 - Controlam o acesso simultâneo aos arquivos
 - Permitem o compartilhamento e a utilização concorrente dos arquivos



PUC-Rio / DI

TeleMídia

Segurança de Acesso

- Parte fundamental do Servidor de Arquivos
- Controle de acesso:
 - ao sistema a partir das estações com módulos clientes
 - direto da estação servidora
 - aos recursos compartilhados
 - aos recursos não compartilhados
- Acesso restrito a horas específicas
- Acesso restrito a partir de módulos clientes específicos
- Modelos de segurança:
 - *SHARE-LEVEL*
 - *USER-LEVEL*



PUC-Rio / DI

TeleMídia

Técnicas para melhorar a eficiência do Servidor de Arquivos

- ➔ *Cache de Disco*
- ➔ *Cache de Diretório*
- ➔ *Elevator Seeking*
- ➔ *Directory Hashing*
- ➔ *Turbo FAT*



PUC-Rio / DI

TeleMídia

Elevator Seeking

- ➔ A cabeça leitora varre toda a superfície do disco em uma mesma direção atendendo os pedidos nesta ordem, quando chega ao final do disco inverte a direção e recomeça o processo
- ➔ Todo acesso ao disco é realizado na melhor sequência possível, evitando tempos desnecessários de reposicionamento da cabeça leitora
- ➔ O acesso é feito respeitando não a ordem de chegada dos pedidos, mas sim a posição atual da cabeça leitora



PUC-Rio / DI

TeleMídia

Directory Hashing - Turbo FAT

- ➔ Indexação da árvore de diretórios permitindo uma localização mais eficiente dos diretórios e seus componentes
- ➔ Tabela de Alocação de Arquivos (FAT) modificada, permitindo localização eficiente dos blocos de disco que compõem um arquivo
- ➔ Muito importante para o servidor de arquivos, pois uma varredura sequencial na tabela de alocação é muito demorada



PUC-Rio / DI

TeleMídia

Técnicas de Tolerância a Falhas

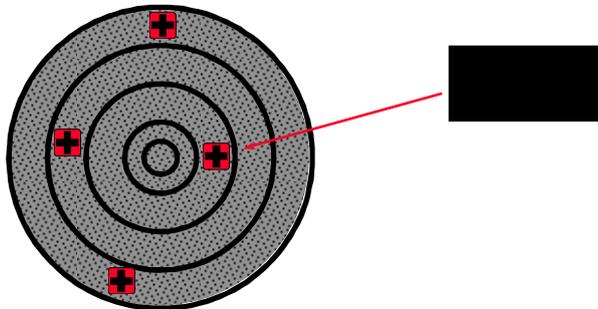
- ➔ *HotFix* - Revetoração Automática de Blocos Defeituosos
- ➔ *Disk Mirroring* - Espelhamento de Disco
- ➔ *Disk Duplexing* - Duplicação de Disco



PUC-Rio / DI

TeleMídia

HotFix



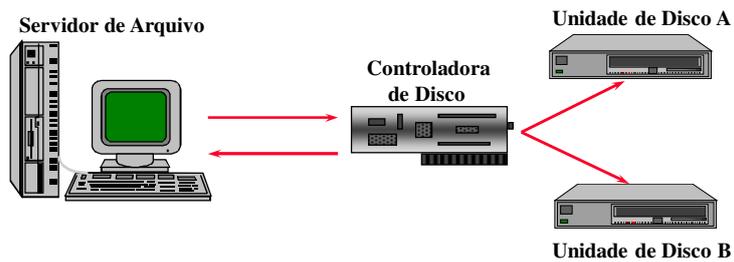
- Detecção e isolamento de blocos defeituosos na mídia de armazenamento
- Detecção de erros através de leitura após escrita



PUC-Rio / DI

TeleMídia

Disk Mirroring



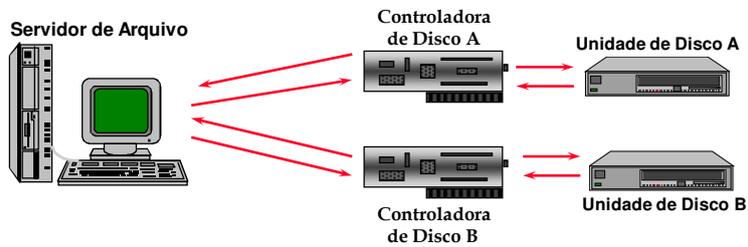
- Utilização de duas unidades de disco de maneira redundante, ou seja, toda gravação em uma é espelhada na outra
- Em caso de problemas com alguma unidade de disco, a outra é utilizada sem interrupção dos serviços



PUC-Rio / DI

TeleMídia

Disk Duplexing



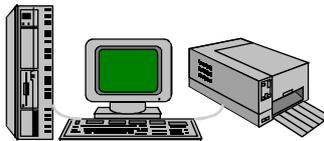
- Idêntico à técnica de *Disk Mirroring*, porém cada unidade de disco possui sua própria controladora, aumentando o nível de tolerância a falhas
- Aumenta a eficiência do servidor de arquivos, pois as duas unidades de disco são operadas em paralelo



PUC-Rio / DI

TeleMídia

Servidores de Impressão



- ➔ Oferecem serviços de compartilhamento de impressoras e impressão remota
- ➔ Controle da fila de impressão associada à impressora, permitindo:
 - ver a fila de impressão
 - cancelar impressão
 - manipular a ordem de impressão



PUC-Rio / DI

TeleMídia

Servidores de Impressão

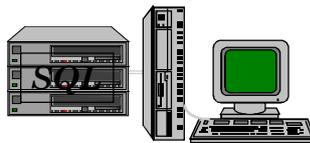
- Normalmente servidores não dedicados
- Quando gerenciam várias impressoras com alta carga de utilização, torna-se necessário uma máquina dedicada para esta função
- Spool de Impressão
 - Área de armazenamento temporário dos arquivos a serem impressos
 - Normalmente localizado em um servidor de arquivos
- Servidores de Impressão permitem o posicionamento estratégico das impressoras em locais de fácil acesso



PUC-Rio / DI

TeleMídia

Servidores de Banco de Dados



- Oferecem serviços de banco de dados
- SQL
- Controle de Transações
- Segurança
- Controle de Integridade



PUC-Rio / DI

TeleMídia

Evolução dos Sistemas Gerenciadores de Banco de Dados (SGBD)

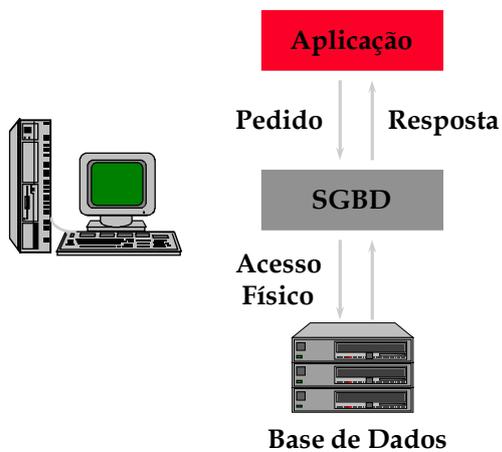
- ➔ SGBD Tradicionais
- ➔ SGBD Tradicionais utilizando um Servidor de Arquivos
- ➔ SGBD Cliente-Servidores
- ➔ SGBD Distribuídos



PUC-Rio / DI

TeleMídia

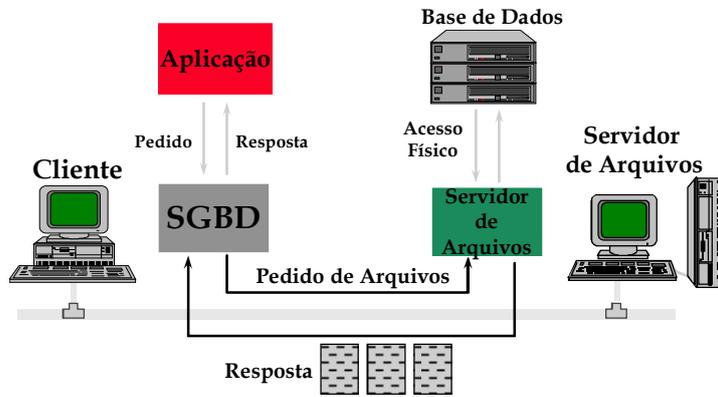
SGBD Tradicionais



PUC-Rio / DI

TeleMídia

SGBD Tradicionais utilizando um Servidor de Arquivos



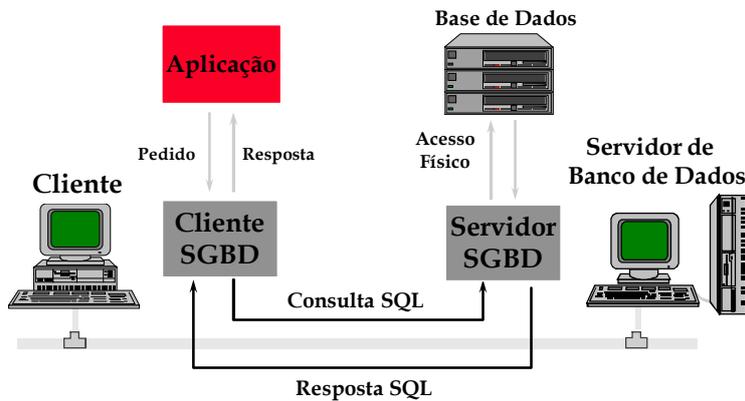
Todo Processamento da pesquisa é executado localmente pelo SGBD



PUC-Rio / DI

TeleMídia

SGBD Cliente-Servidor



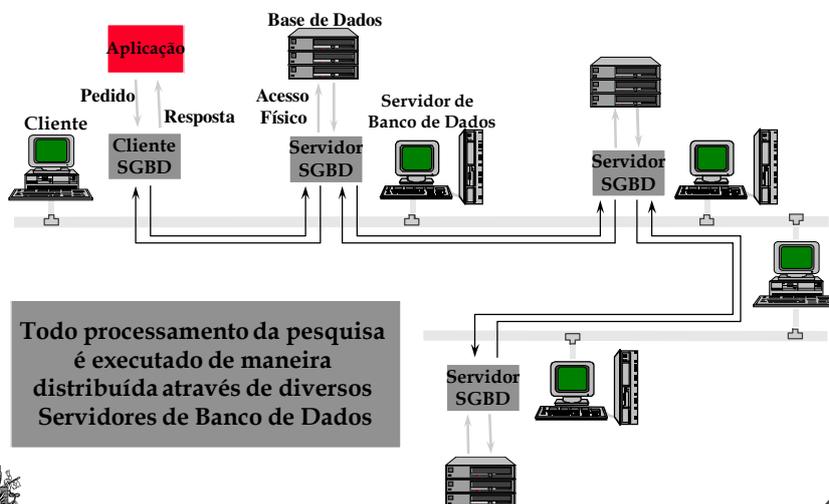
Todo Processamento da pesquisa é executado remotamente pelo Servidor de Banco de Dados



PUC-Rio / DI

TeleMídia

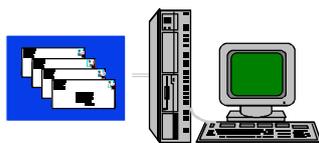
SGBD Distribuídos



PUC-Rio / DI

TeleMídia

Servidores de Correio Eletrônico



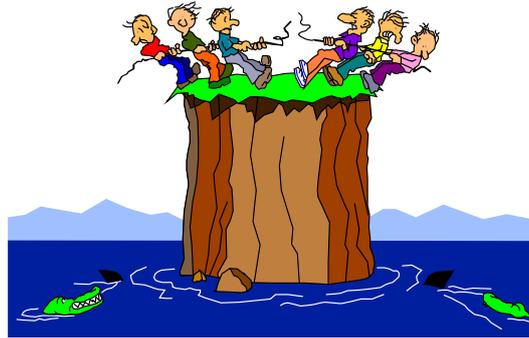
- Gerenciam recursos de correio eletrônico e as caixas postais dos usuários
- Permitem troca de correspondências eletrônicas entre os usuários da rede



PUC-Rio / DI

TeleMídia

Implementações Comerciais de Sistemas Operacionais de Rede



PUC-Rio / DI

TeleMídia

Netware

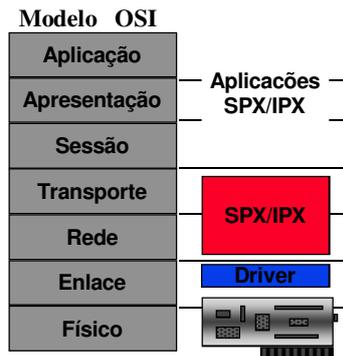


PUC-Rio / DI

TeleMídia

SPX/IPX

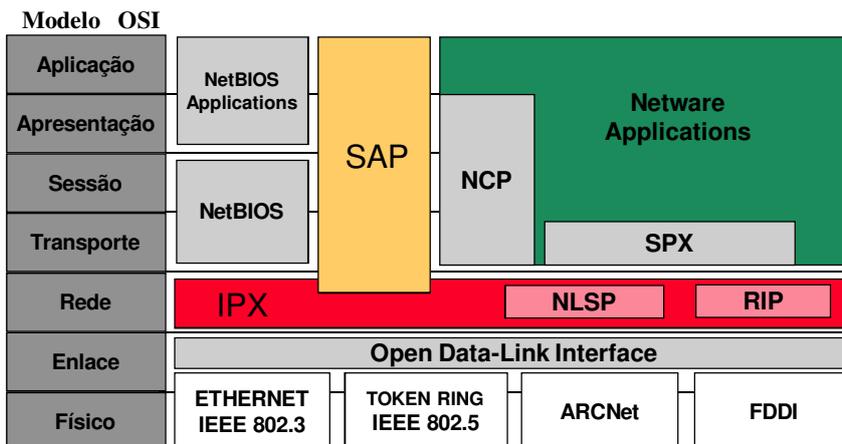
- Desenvolvido pela Novell
- Baseado nos protocolos da Xerox : XNS
- Protocolos de transporte do Netware



PUC-Rio / DI

TeleMídia

Pilha de Protocolos Netware



PUC-Rio / DI

TeleMídia

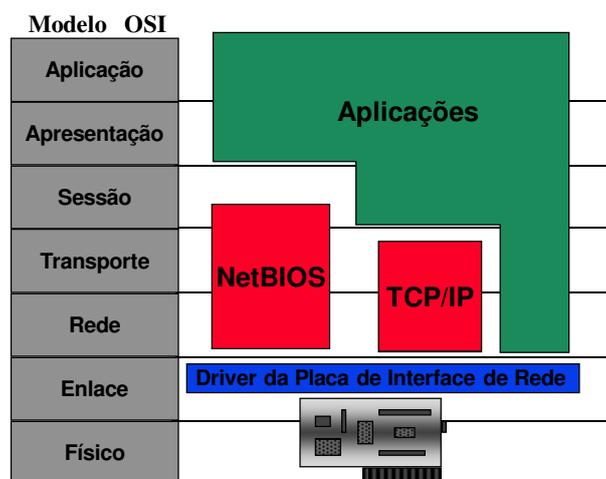
Windows 95, 98, 2000



PUC-Rio / DI

TeleMídia

Protocolos e Aplicações



PUC-Rio / DI

TeleMídia

Windows 95

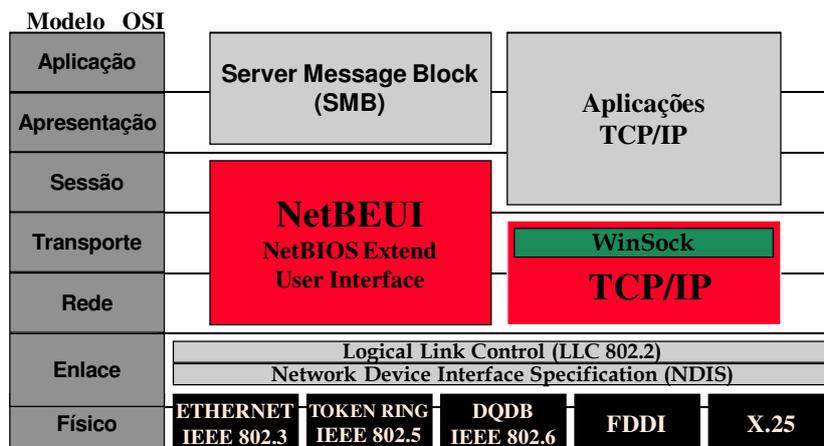
- ➔ Desenvolvido pela Microsoft
- ➔ Arquitetura Peer-to-Peer
- ➔ Solução para integração de pequenos grupos de trabalho (Workgroups)
- ➔ Conceito de Grupo de Trabalho
 - pessoas inter-relacionadas
 - computadores pessoais
 - compartilhamento usual de recursos
 - troca frequente de mensagens



PUC-Rio / DI

TeleMídia

Pilha de Protocolos Windows For Workgroups



PUC-Rio / DI

TeleMídia

Windows NT



PUC-Rio / DI

TeleMídia

Windows NT

- Sistema operacional de rede da Microsoft
- Recursos de rede integrados ao sistema
- Clientes:
 - Windows NT Workstations;
 - Windows 95; 98, 2000
 - MSDOS;
 - Lan Manager server ou client;
 - Netware;
 - Macintosh.
- Integração Windows, Netware e Lan Manager
- Arquitetura cliente-servidor permitindo compartilhamento *peer-to-peer*



PUC-Rio / DI

TeleMídia

Windows NT

- Modelo de segurança *user level*
- Rede organizada em domínios:
“agrupamento lógico de servidores e clientes que compartilham informações comuns de segurança e contas de usuário”
- vantagens:
 - login único por usuário;
 - administração centralizada;
 - acesso universal a recursos;
 - organização de usuários em grupos de trabalho.



PUC-Rio / DI

TeleMídia

Controladores de Domínio

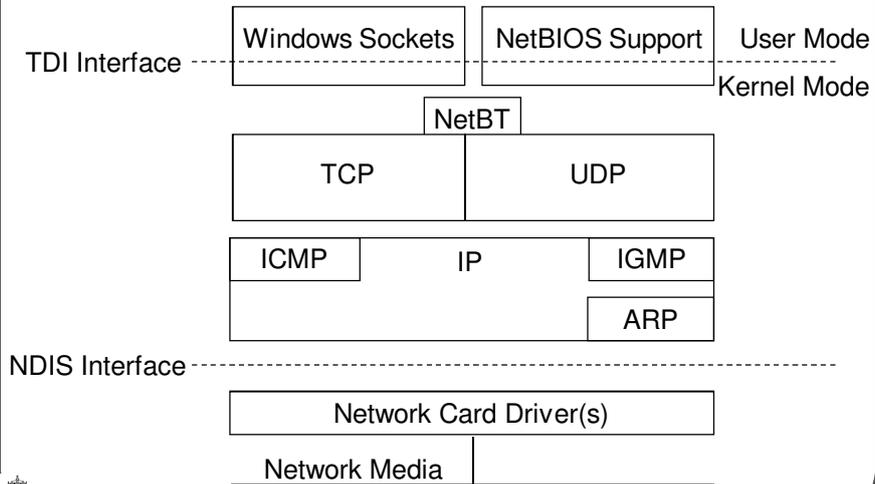
- Servidores que compartilham informações do domínio são controladores de domínio:
 - *Primary Domain Controller (PDC)*;
 - *Backup Domain Controller (BDC)*.
- um domínio tem apenas 1 PDC e pelo menos 1 BDC que armazenam o banco de dados de diretório com nomes, senhas e direitos dos usuários e grupos



PUC-Rio / DI

TeleMídia

Arquitetura TCP/IP



PUC-Rio / DI

TeleMídia

Outros

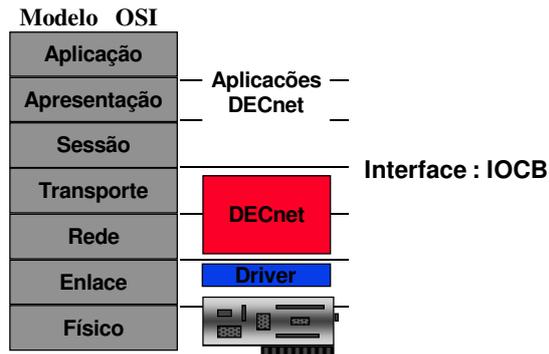


PUC-Rio / DI

TeleMídia

DECnet

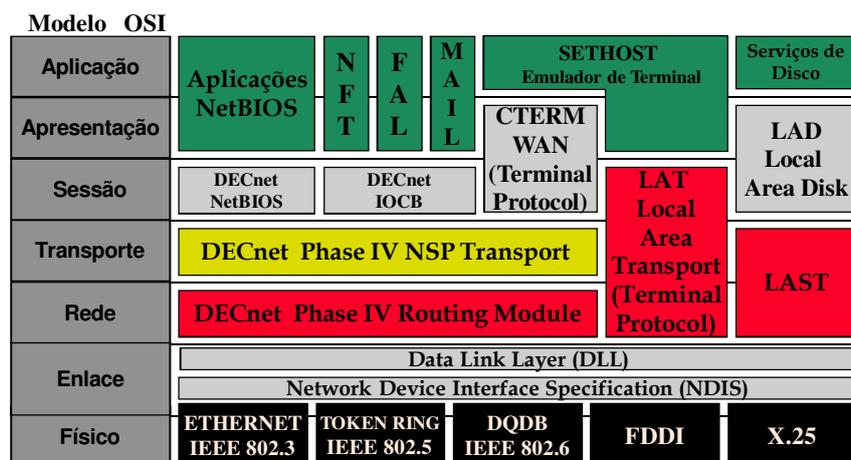
- Desenvolvido pela Digital
- DNA - Digital Network Architecture
- Protocolos de transporte do Pathworks



PUC-Rio / DI

TeleMídia

Pilha de Protocolos PathWorks

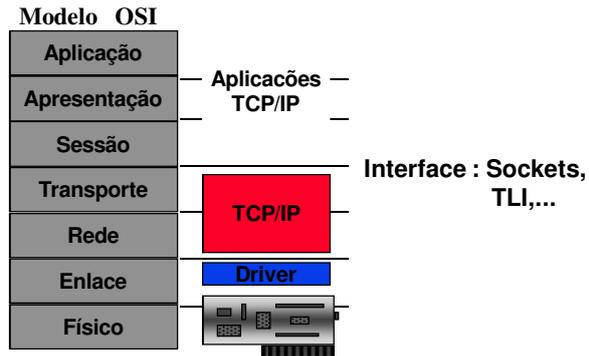


PUC-Rio / DI

TeleMídia

S.O. Baseados em TCP/IP

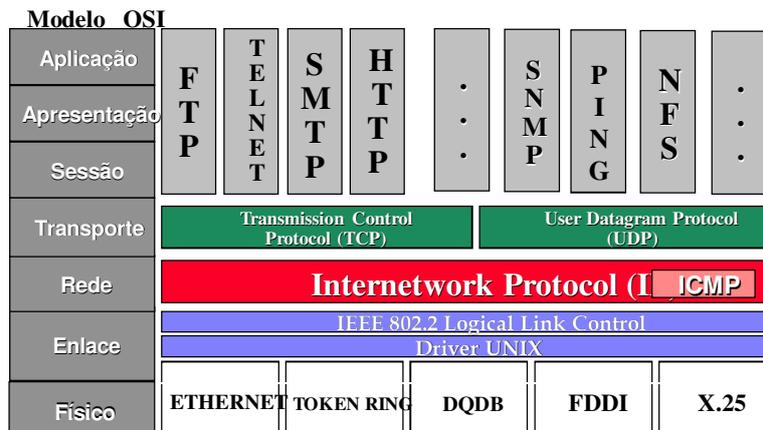
- Desenvolvido pelo Departamento de Defesa Americano (DoD)
- Amplamente difundido
- Protocolos de transporte do UNIX



PUC-Rio / DI

TeleMídia

Pilhas de Protocolos UNIX



PUC-Rio / DI

TeleMídia

NFS

- ➔ Desenvolvido pela Sun Microsystems, permite que um sistema tenha acesso a arquivos localizados remotamente de um modo integrado e transparente.
- ➔ O NFS fornece a ilusão que os discos, impressoras, ou outros dispositivos, fisicamente localizados em um sistema remoto, estão diretamente conectados ao sistema local.
- ➔ O NFS, ao contrário das outras aplicações Internet, não é implementado por um único utilitário, sendo dividido em três componentes:
 - o protocolo NFS propriamente dito, uma ferramenta genérica que controla a execução de chamadas a procedimentos remotos (Remote Procedure Call RPC), e uma terceira ferramenta que compatibiliza formatos diferentes de representação de dados denominada XDR (External Data Representation). A divisão foi realizada para permitir que o RPC e o XDR pudessem ser utilizados em outras aplicações.



RPC

- ➔ No módulo cliente, o programador declara alguns dos procedimentos como sendo *remotos*, forçando o compilador a incorporar à chamada de procedimento, o código que repassa os argumentos da chamada ao cliente RPC.
- ➔ No módulo servidor, são implementados os procedimentos a serem compartilhados, que são declarados como procedimentos servidores.
- ➔ Quando o programa cliente faz uma chamada a um procedimento remoto, os argumentos da chamada são entregues ao cliente RPC, que compõe uma mensagem e a envia ao servidor, passando então a aguardar o retorno do resultado.
- ➔ O servidor RPC, ao receber uma mensagem invocando a execução de um procedimento, dispara sua execução e, ao seu término, envia os resultados obtidos de volta para o cliente RPC.
- ➔ O cliente RPC, ao receber de volta a mensagem com o resultado da execução do procedimento remoto, entrega esses resultados ao processo de aplicação, colocando-os nos respectivos argumentos da chamada de procedimento.



XDR

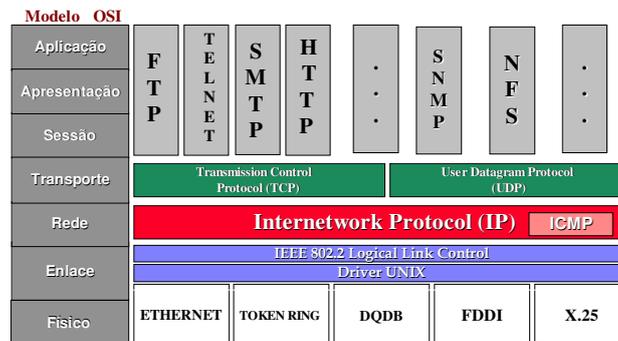
- ▶ Quando deseja transmitir dados, um programa de aplicação chama os procedimentos XDR que convertem a representação local para a representação de transferência e, em seguida, transmite os dados convertidos.
- ▶ O processo que recebe os dados chama os procedimentos XDR, que convertem a representação de transferência para a representação local específica.



Nível de Aplicação TCP/IP



Exemplos de Aplicações Internet TCP/IP



DNS

- ➔ O DNS é um esquema de gerenciamento de nomes, hierárquico e distribuído.
- ➔ O DNS define:
 - a sintaxe dos nomes usados na Internet
 - regras para delegação de autoridade na definição de nomes
 - um banco de dados distribuído que associa nomes a atributos (entre eles o endereço IP)
 - um algoritmo distribuído para mapear nomes em endereços.



Resolução DNS *Copacabana.Inf.Puc-Rio.Br*

- Inicialmente deve ser consultado um servidor central, denominado servidor raiz, para descobrir onde está o servidor Br.
- O servidor Br é o responsável pela gerência dos nomes das instituições brasileiras ligadas à Internet. O servidor raiz informa como resultado da consulta o endereço IP de vários servidores de nome para o nível Br (pode existir mais de um servidor de nomes em cada nível, para garantir a continuidade da operação quando um deles pára de funcionar).
- Um servidor do nível Br pode então ser consultado, devolvendo o endereço IP do servidor Puc-Rio.
- De posse do endereço de um servidor Puc-Rio é possível solicitar que ele informe o endereço de um servidor Inf, quando, finalmente, pode-se consultar o servidor Inf sobre o endereço da máquina Copacabana.



PUC-Rio / DI

TeleMídia

DNS

- O DNS não se limita a manter e gerenciar endereços Internet.
- Cada nome de domínio é um nó em um banco de dados, que pode conter informações definindo várias propriedades.
 - Por exemplo, o tipo da máquina e a lista de serviços fornecidos por ela.
 - O DNS permite que seja definido um *alias* (nome alternativo) para o nó.
 - Também é possível utilizar o DNS para armazenar informações sobre usuários, listas de distribuição ou outros objetos.



PUC-Rio / DI

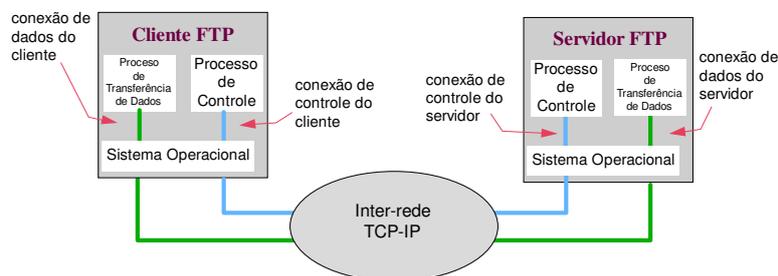
TeleMídia

FTP

- ➔ O FTP permite que um usuário em um computador transfira, renomeie ou remova arquivos remotos; ou crie, remova e modifique diretórios remotos.
- ➔ O FTP só permite a transferência de arquivos completos.
- ➔ Antes de executar qualquer operação o usuário solicitante (cliente) envia seu nome (*login name*) e sua senha para o servidor, que impede a execução de qualquer operação, caso o usuário não tenha sido registrado.



FTP



- ➔ A conexão de controle permanece aberta enquanto durar a sessão FTP. Durante uma sessão podem ser transferidos vários arquivos, cada um deles em uma conexão de transferência de dados estabelecida especificamente para tal.



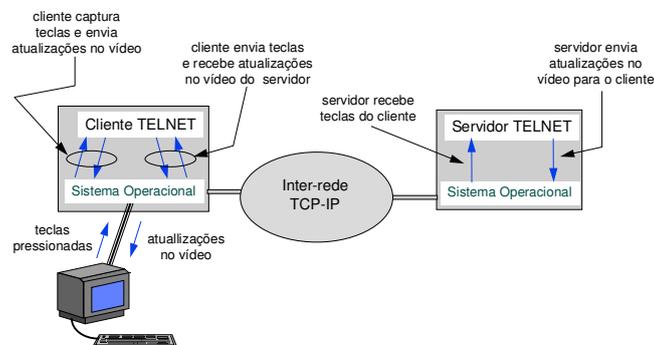
HTTP

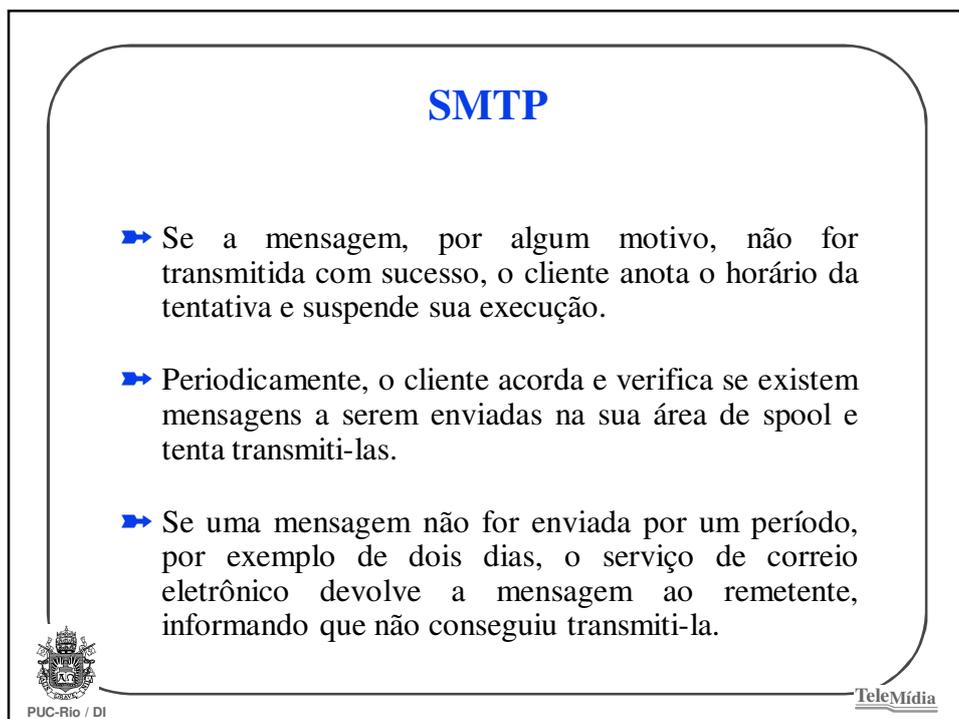
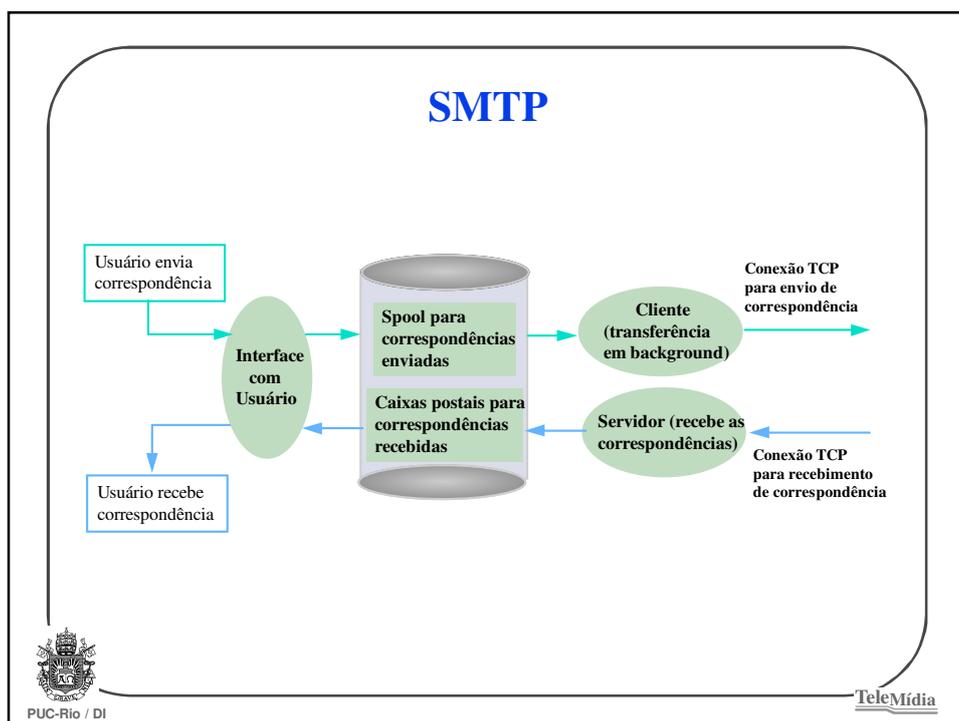
- ➔ O HTTP (Hypertext Transfer Protocol) é um protocolo usado para transferência de informações no WWW.
- ➔ Os dados transferidos pelo HTTP podem ser texto não estruturado, hipertextos, imagens, ou qualquer outro tipo de dados.



Telnet

- ➔ Permite que um usuário utilizando uma máquina A estabeleça uma sessão interativa com uma máquina (B) na rede.



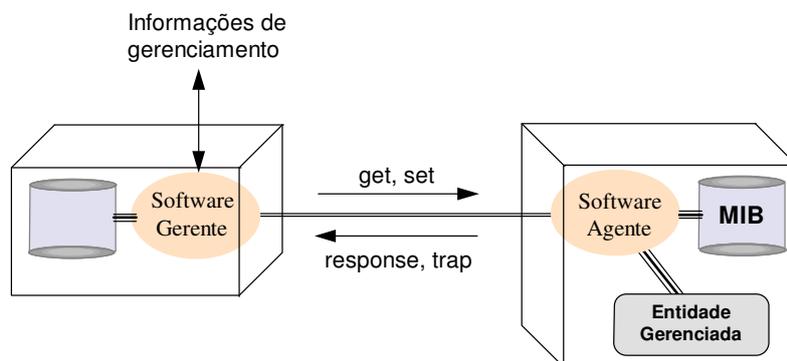


SNMP

- O sistema de gerenciamento de redes da arquitetura Internet TCP/IP opera na camada de aplicação e baseia-se no protocolo SNMP (Simple Network Management Protocol).
- Como no esquema de gerenciamento OSI, os processos que implementam as funções de gerenciamento Internet atuam como agentes ou gerentes.
 - Os agentes coletam junto aos objetos gerenciados as informações relevantes para o gerenciamento da rede.
 - O gerente processa as informações recolhidas pelos clientes, com o objetivo de detectar a presença de falhas no funcionamento dos componentes da rede (hosts, gateways, processos executando os protocolos de comunicação etc.), para que possam ser tomadas providências no sentido de contornar os problemas que ocorrem como consequência das falhas.



SNMP



Perguntas

- Qual a diferença de um redirecionador no nível de BIOS de um no nível de sistema operacional?
- Qual a diferença entre paradigma cliente-servidor e arquitetura cliente-servidor, com relação a sistemas operacionais de rede?
- Qual a diferença, em termos de funções de um módulo cliente para um módulo servidor de um sistema operacional de rede?
- Descreva as diferenças de uma arquitetura peer-to-peer, servidor dedicado e servidor não dedicado? Por que todas usam o paradigma cliente-servidor?
- O que vem a ser modelo de segurança “share level” e “user level”?
- Como poder-se-ia emular um modelo “user level” a partir de um “share level”?



PUC-Rio / DI

TeleMídia

Perguntas

- Comente algumas técnicas para se melhorar a eficiência de servidores de arquivos.
- Comente alguma técnicas de tolerância a falhas usadas em servidores de arquivo.
- Comente sobre a evolução dos servidores de banco de dados, desde a utilização de servidores de arquivos até os servidores de banco de dados distribuídos. Discuta mencionando o impacto das técnicas na eficiência do sistema.
- O que é o DNS? Como é realizada a resolução de DNS na Internet?
- Como é realizado o protocolo FTP? Quais os tipos de conexões usadas e para que servem? Como é possível a transferência simultânea de vários arquivos?
- Explique o funcionamento do protocolo SMTP.
- Explique o funcionamento do protocolo SNMP.



PUC-Rio / DI

TeleMídia