

Camada de Transporte

Conceitos Básicos

Redes de Computadores
A. S. Tanenbaum



Mário Meireles Teixeira, UFMA-DEINF

Protocolos

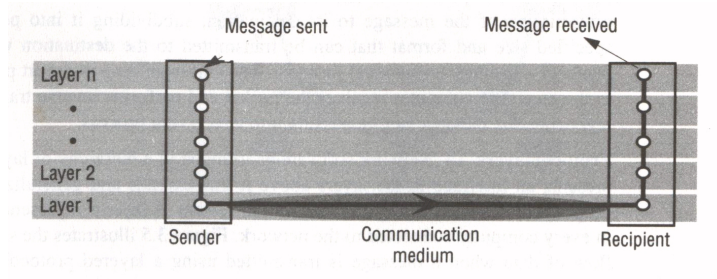


- **Protocolo:**
 - Conjunto de regras e formatos usados para comunicação entre entidades, a fim de permitir a realização de uma tarefa
- **Dois componentes:**
 - Uma especificação da seqüência de mensagens que devem ser trocadas
 - Uma especificação do conteúdo das mensagens
- A existência de protocolos permite que as aplicações de rede sejam desenvolvidas de modo independente

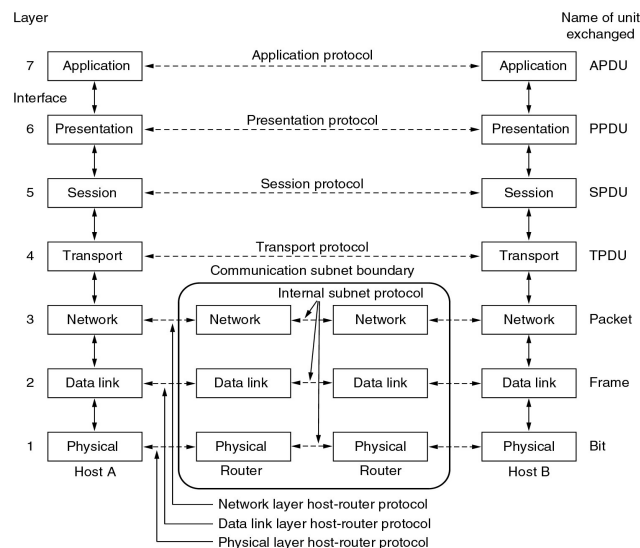


Camadas de Protocolos

- O software de rede geralmente é organizado em um conjunto de camadas
- Cada camada é responsável por certos serviços, oferecendo à camada superior uma interface bem definida de acesso a suas operações



Modelo de Referência OSI





Modelo OSI

- **Camada Física**
 - Definição das características físicas do meio de transmissão: sinalização, cabeamento, etc.
 - Ex. de protocolos: X.21, Ethernet banda básica
- **Camada de Enlace**
 - Transmissão de *frames* de dados livres de erros entre computadores diretamente conectados
 - Ex: CSMA/CD (Ethernet), HDLC
- **Camada de Rede**
 - Responsável pelo roteamento dos *pacotes* entre os nós da rede
 - Ex: IP, X.25, IPX



Modelo OSI

- **Camada de Transporte**
 - Estabelece um enlace entre duas máquinas para transmissão de *mensagens*, não necessariamente na mesma rede
 - Serviço orientado ou não a conexão
 - Ex: TCP, UDP, SPX
- **Camada de Sessão**
 - Estabelece a comunicação entre processos em máquinas diferentes
 - Interações cliente-servidor: RPC, RMI



Modelo OSI

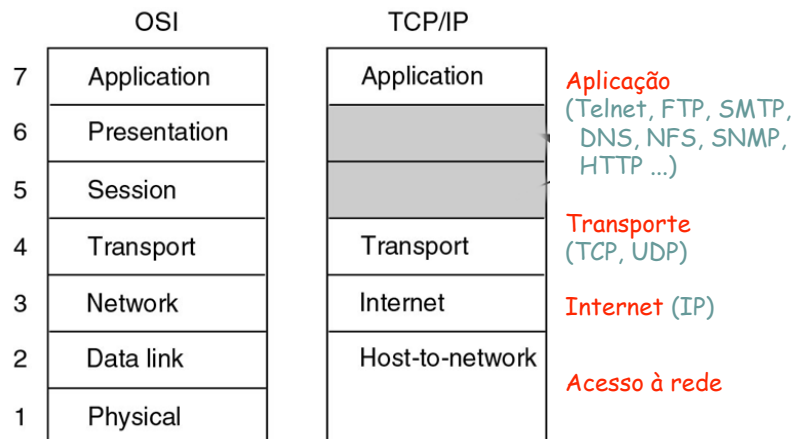
- **Camada de Apresentação**
 - Permite apresentar os dados em um formato independente dos utilizados por cada sistema individualmente
 - Criptografia
 - Ex: XDR, ASN.1, NCP
- **Camada de Aplicação**
 - Nesta camada são executadas as aplicações dos usuários e alguns protocolos utilitários de uso geral
 - Ex: ftp, telnet, ICQ, SMTP, X.400, X.500



Protocolos TCP/IP

- Surgiram com a Internet, para a interligação dos seus computadores
- Padrão de fato, em LANs e WANs
- Funcionam sobre diversas tecnologias de rede (cabos coaxiais, fibras óticas, linhas telefônicas, redes sem fio...)
- **Características especiais:**
 - Uso de padrões abertos
 - Independência de tecnologia de rede
 - Protocolos padronizados - RFCs (rede, transporte e aplicação)
 - Interconexão total - identificação única por dispositivo (endereço IP)
 - Confirmações (ACK) fim-a-fim

Arquitetura TCP/IP



Camada de Transporte



- É o núcleo de toda a hierarquia de protocolos
- Seu objetivo é realizar a transferência de dados, de forma confiável, independente da rede, entre a origem e o destino, de fim a fim
- A camada de transporte corrige as “imperfeições” da camada de rede e fornece uma interface de alto nível às camadas superiores
- Seus usuários são os processos da camada de aplicação
- Utiliza diversos serviços da camada de rede a fim de cumprir com a sua função



Serviços de Transporte

- **Orientado à conexão**
 - É preciso estabelecer um canal lógico entre dois processos para que eles se comuniquem
 - Transmite-se um fluxo de dados (*stream*) entre o emissor e o receptor
 - Operação mais complexa:
 - *Estabelecimento da conexão;*
 - *Transferência de dados;*
 - *Encerramento.*
 - Controle de erros mais rígido (serviço “confiável”)
 - Analogia: serviço telefônico
 - Aplicação: login remoto, e-mail, FTP, HTTP
 - Protocolo: TCP



Serviços de Transporte

- **Sem conexões**
 - A mensagem é dividida em datagramas, roteados individualmente até o processo receptor. Não é necessário estabelecer uma conexão
 - Operação e gerenciamento mais simples: menos sobrecarga
 - Serviço “não-confiável” de entrega de mensagens: cabe às aplicações realizar o controle de fluxo e detectar perda de datagramas e erros de seqüência
 - Analogia: telegrama
 - Aplicação: comunicação C/S em redes locais, sistemas P2P, DNS, NFS, SNMP, mídia contínua (streaming)
 - Protocolo: UDP

Aspectos de Protocolos de Transporte



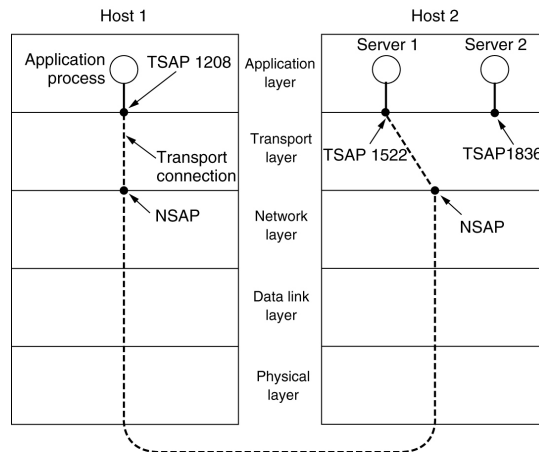
- Endereçamento
- Estabelecimento de conexões
- Encerramento de conexões
- Controle de fluxo e uso de buffers
- Multiplexação
- Recuperação de falhas

Endereçamento



- Estabelecimento de uma conexão → processo precisa especificar com “quem” deseja se conectar
- Envio de mensagens → para “quem” enviar a mensagem
- Solução: **Endereços de Transporte**
 - TCP/IP: portas
 - ATM: AAL-SAPs
 - Genericamente: TSAP (Transport Service Access Point)
 - Camada de rede: NSAP (Network SAP)
- Processos precisam se associar a um TSAP para estabelecer conexão e enviar/receber mensagens
- Vários TSAPs podem compartilhar o mesmo NSAP

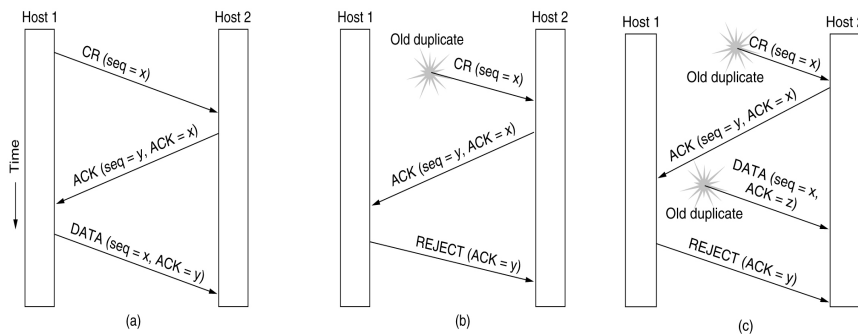
Endereçamento



Endereçamento

- Como descobrir o endereço de transporte da entidade remota?
 - Well-known addresses (UNIX /etc/services)
 - (21: **FTP**; 23: **Telnet**; 25: **SMTP**; 80: **HTTP**)
 - Servidor de processos (um proxy: inet.d)
 - *Clientes contactam servidor*
 - *Servidor gera a conexão para o serviço solicitado*
 - *Cliente interage com o novo serviço*
 - Binder, Port Mapper (RPC) / Serviço de Nomes (CORBA) / UDDI (Web Services)
 - *Cliente pergunta ao binder pela existência de um serviço*
 - *Binder responde com o endereço do serviço*
 - *Novos serviços devem se registrar no binder*

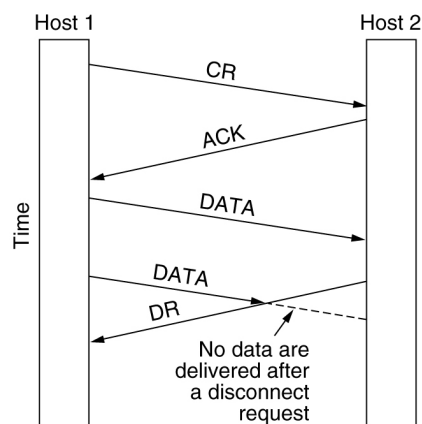
Estabelecimento de Conexões



- **Handshake de Três Vias (Three-way handshaking)**
 - (a) Sucesso
 - (b) Duplicata de CR surge repentinamente
 - (c) Duplicatas de CR e ACK

Encerramento de Conexões

- **Encerramento Assimétrico**
 - Qualquer dos usuários pode encerrar a conexão, de forma unilateral
 - Pode levar à perda de dados

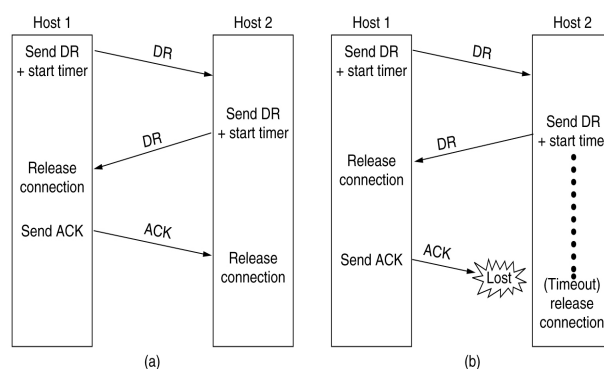




Encerramento de Conexões

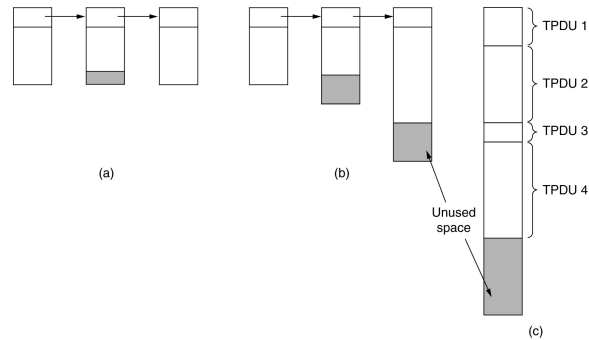
- **Encerramento Simétrico**
 - A conexão é tratada como duas conexões unidirecionais independentes
 - Cada direção tem que ser encerrada separadamente
 - Quando um das partes se desconecta, significa que não tem mais dados a enviar, mas ainda pode receber dados
 - Solução: Handshake de três vias
 - A parte que propôs a desconexão deve confirmá-la

Encerramento de Conexões



- (a) Sucesso
- (b) ACK final perdida

Tamanho dos Buffers



- Opção entre tamanho e organização depende do tipo de tráfego
- (a) Buffers de tamanho fixo em uma lista encadeada
- (b) Buffers de tamanho variável em uma lista encadeada
- (c) Buffers de tamanho variável em uma lista circular

Alocação Dinâmica de Buffers

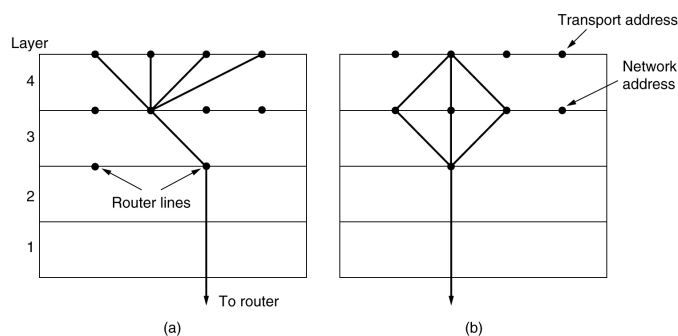
- À medida que o padrão de tráfego se altera, transmissor e receptor precisam ajustar dinamicamente a quantidade de buffers alocados às conexões
- Solução: usar uma **janela de tamanho variável**
 - Inicialmente, transmissor solicita um certo número de buffers
 - Receptor informa o número de buffers disponíveis
 - Ao enviar uma msg, transmissor decrementa numbuf, parando quando numbuf = 0
 - Ao retornar as confirmações (ACK), o receptor atualiza a alocação de buffers (*piggybacking*)

Alocação Dinâmica de Buffers

A	Message	B	Comments
1 →	< request 8 buffers>	→	A wants 8 buffers
2 ←	<ack = 15, buf = 4>	←	B grants messages 0-3 only
3 →	<seq = 0, data = m0>	→	A has 3 buffers left now
4 →	<seq = 1, data = m1>	→	A has 2 buffers left now
5 →	<seq = 2, data = m2>	...	Message lost but A thinks it has 1 left
6 ←	<ack = 1, buf = 3>	←	B acknowledges 0 and 1, permits 2-4
7 →	<seq = 3, data = m3>	→	A has 1 buffer left
8 →	<seq = 4, data = m4>	→	A has 0 buffers left, and must stop
9 →	<seq = 2, data = m2>	→	A times out and retransmits
10 ←	<ack = 4, buf = 0>	←	Everything acknowledged, but A still blocked
11 ←	<ack = 4, buf = 1>	←	A may now send 5
12 ←	<ack = 4, buf = 2>	←	B found a new buffer somewhere
13 →	<seq = 5, data = m5>	→	A has 1 buffer left
14 →	<seq = 6, data = m6>	→	A is now blocked again
15 ←	<ack = 6, buf = 0>	←	A is still blocked
16 ...	<ack = 6, buf = 4>	←	Potential deadlock

(...) indicam perda de uma mensagem

Multiplexação



- Várias conexões de transporte podem utilizar o mesmo ponto de acesso à rede (NSAP, p.ex. endereço IP)
- (a) Multiplexação ascendente
- (b) Multiplexação descendente



Recuperação de Falhas

- Podem ocorrer falhas na rede ou nos hosts
- Se a camada de rede oferecer um serviço de **datagramas**:
 - as entidades de transporte saberão como tratar mensagens perdidas
- Se a camada de rede oferecer um serviço **orientado a conexões**:
 - uma falha será contornada estabelecendo-se um novo circuito virtual
- Como tratar falhas nos hosts?
 - Repassar a mensagem à aplicação de destino e enviar um ACK são dois eventos distintos e indivisíveis, que não podem acontecer simultaneamente



Recuperação de Falhas

Strategy used by sending host	Strategy used by receiving host					
	First ACK, then write			First write, then ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

OK = Protocol functions correctly
 DUP = Protocol generates a duplicate message
 LOST = Protocol loses a message

- S1/S0 : confirmação pendente/não pendente
- A = ACK; C = Crash; W = Write
- **Conclusão**: a recuperação de uma falha na camada N só pode ser feita pela camada $N + 1$, desde que haja informações suficientes