

Capítulo 5

A CAMADA DE TRANSPORTE

A camada de transporte é a primeira camada do modelo OSI a abstrair a topologia e a tecnologia da subrede de comunicação. Esta camada utiliza o serviço de entrega de pacotes da camada de rede para prover comunicação confiável host a host. Se a camada de rede for baseada em datagrama, todo o trabalho para uma comunicação confiável fica a cargo da camada de transporte. Neste caso, a camada de transporte deve gerenciar a perda, duplicação e inversão de ordem de pacotes ocorridas na subrede de comunicação. No caso da camada de rede prover entrega confiável de pacotes, a camada de transporte tem uma estrutura mais simplificada, gerenciando principalmente o fluxo de mensagens (TPDUs) entre os hosts comunicantes e as quebras de conexões de rede (N-RESETs).

5.1 Qualidade do Serviço

A camada de transporte define alguns parâmetros associados à qualidade do transporte de dados. Estes parâmetros são negociados no estabelecimento de conexões de transporte entre os hosts e a subrede de comunicação. Os principais parâmetros são:

- vazão (bytes/s): taxa mínima assegurada de transmissão de mensagens pela conexão de transporte;
- atraso de propagação (milisegundos): tempo máximo de trânsito de uma TPDU entre as camadas de transporte dos hosts comunicantes;
- *jitter* (milisegundos): variação máxima do atraso de propagação;
- tempo de estabelecimento da conexão (milisegundos): máximo intervalo de tempo entre a entidade usuária da camada de transporte solicitar uma conexão e a sua confirmação;
- probabilidade de falha no estabelecimento da conexão: probabilidade da conexão não ser estabelecida no tempo estipulado (devido a falta de buffers nos hosts, congestionamento na subrede de comunicação, etc);
- taxa de erro residual: Taxa de mensagens perdidas ou corrompidas na conexão de transporte (teoricamente zero, mas assumindo um valor muito pequeno na prática);

- tempo de encerramento de conexão (milisegundos): tempo máximo para o término de uma conexão;
- probabilidade de falha no término de conexão: probabilidade do encerramento de uma conexão não se completar no tempo de encerramento estipulado;
- proteção: estipula a privacidade dos dados fluindo pela conexão de transporte quanto a sua interceptação por terceiros;
- prioridade: estipula a prioridade desta conexão em relação às demais (a prioridade da transmissão de TPDU's é função da prioridade das conexões por onde as TPDU's fluem);
- confiabilidade: mede a probabilidade da camada de transporte falhar espontaneamente (devido a bugs, situações não previstas, etc).

Um subconjunto destes parâmetros é passado à camada de transporte no estabelecimento de uma conexão. A conexão não é estabelecida se a qualidade do serviço requisitada não puder ser honrada:

1. pela camada de transporte do host local;
2. pela camada de transporte do host remoto (sendo conectado);
3. pela subrede de comunicação.

5.2 O Serviço de Transporte

O serviço de transporte pode ser orientado a conexão ou a datagrama (sem conexão). Em redes locais, dada a alta confiabilidade da camada de rede, o serviço sem conexão é atrativo para muitas aplicações, dado seu baixo *overhead*. Em redes públicas, mesmo com camada de rede orientada a conexão (como no caso do X.25/camada 3), o serviço de transporte orientado a conexão é o mais seguro para a maioria das aplicações.

O modelo OSI especifica apenas quatro primitivas para transporte orientado a conexão e uma para transporte orientado a datagrama (tabela 5.1). Note a ausência de primitiva do tipo RESET: uma conexão de transporte é muito mais confiável que uma conexão de rede (camada 3).

As primitivas T-CONNECT, T-DISCONNECT e T-DATA estabelecem, terminam e transmitem TPDU's por uma conexão. São equivalentes àquelas da camada de rede.

A primitiva T-EXPEDITED-DATA transmite *dados expressos*: TPDU's com alta prioridade que são entregues à aplicação fora de sequência. São empregados para transmitir informações urgentes como um caracter de controle (BREAK, DELETE, etc) teclado pelo usuário numa sessão de *login* remoto ¹.

¹Imagine um programa em loop que supostamente deveria ler caracteres já digitados pelo usuário. O que aconteceria se o caracter BREAK, teclado pelo usuário para abortar o programa, for entregue em sequência?

T-CONNECT.request(TSAP_origem, TSAP_destino, exp_req, QoS, dados)
T-CONNECT.indication(TSAP_origem, TSAP_destino, exp_req, QoS, dados)
T-CONNECT.response(QoS, TSAP_destino, exp_req, dados)
T-CONNECT.confirm(QoS, TSAP_destino, exp_req, dados)
T-DISCONNECT.request(dados)
T-DISCONNECT.indication(justificativa, dados)
T-DATA.request(dados)
T-DATA.indication(dados)
T-EXPEDITED-DATA.request(dados)
T-EXPEDITED-DATA.indication(dados)
T-UNITDATA.request(TSAP_origem, TSAP_destino, QoS, dados)
T-UNITDATA.indication(TSAP_origem, TSAP_destino, QoS, dados)

Tabela 5.1: Serviços oferecidos pela camada de transporte: as primitivas T-CONNECT, T-DISCONNECT e T-DATA são empregadas em serviços com conexão; T-UNITDATA em serviços sem conexão. *exp_req* é um flag que indica se dados expressos serão enviados pela conexão ou não. QoS estabelece a qualidade de serviço proposta ou aceita.

A primitiva T-UNIDATA transmite uma TPDU sem o estabelecimento de conexão e sem garantia de entrega, sequência ou ausência de duplicação.

A figura 5.1 mostra sequências típicas de ocorrência destas primitivas.

A camada de transporte provê serviços nos TSAPs (Transport Service Access Points). Na camada de rede um NSAP identificava um host. Na camada de transporte, um TSAP identifica um *port* de comunicação mantido por um processo (em host conhecido). A entidade usuária da camada de transporte normalmente indica o host e o port aos quais a mensagem se destina. A camada de transporte manipula o port, passando o host (no formato adequado de NSAPs) à camada de rede.

Via de regra, um TSAP é uma estrutura muito mais simples que um NSAP: tipicamente um inteiro ocupando 2 bytes que identifica o port. Um TSAP pode ser entendido como um identificador de uma fila para onde mensagens são enviadas (ou de onde mensagens são recebidas). A forma como um processo solicita este recurso depende da interface de acesso aos serviços de transporte provida pelo sistema operacional.

O estado em que um TSAP pode se encontrar (e o que causa as transições de estado) é apresentado no diagrama da figura 5.2.

5.3 Classes de Protocolos de Transporte

Os protocolos de transporte são classificados em cinco categorias em função da camada de rede da qual se utilizam e de certas funcionalidades que provêm.

As subredes de comunicação se classificam em três tipos:

Tipo A : Livres de erros e de N-RESETs. Apenas as LANs se aproximam desta classe.

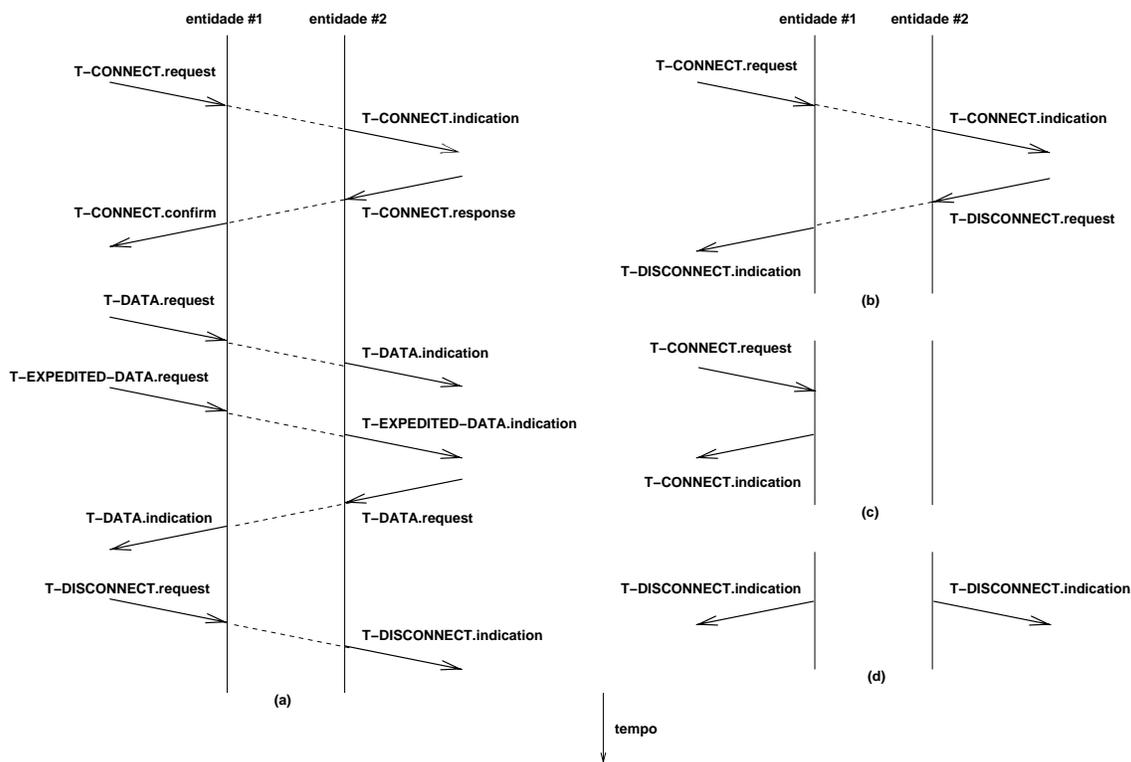


Figura 5.1: Algumas sequências de ocorrência das primitivas OSI: (a) sequência normal de transferência de dados; (b) conexão recusada pelo host remoto; (c) conexão recusada pela camada de transporte; (d) término de conexão requisitado pela camada de transporte.

Tipo B : Transporte confiável de pacotes, mas com a ocorrência de N-RESETs (quebras de conexões). Redes públicas X.25 pertencem a esta classe.

Tipo C : Transporte de pacotes não confiável. Redes com protocolo IP (Internet Protocol) como a Internet, por exemplo, pertencem a esta classe.

As cinco categorias de protocolos de transporte são descritas a seguir.

Classe 0 : Operam em subredes tipo A. São os protocolos mais simples pois toda a confiabilidade do transporte de dados recai sobre a subrede de comunicação.

Classe 1 : Para subredes tipo B, provêem recuperação básica de erros (principalmente N-RESETs). No mais, deixam a cargo da subrede de comunicação a confiabilidade do transporte de dados.

Classe 2 : Para redes tipo A, são idênticos aos de Classe 0, permitindo apenas a multiplexação de várias conexões de transporte numa única conexão de rede.

Classe 3 : Para redes tipo B, provêem as funcionalidades das classes 1 e 2.

Classe 4 : Os únicos a operarem em subredes tipo C, provêem detecção e recuperação de erros não tratados pela subrede de comunicação (usualmente baseadas em datagrama). São os protocolos de transporte mais complexos.

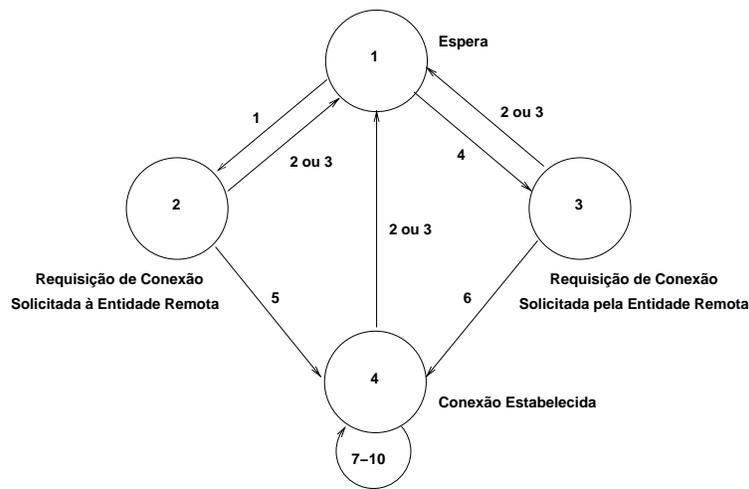


Figura 5.2: Estados de um TSAP. Os eventos que causam transição de estado são: (1) T-CONNECT.request gerado pela camada usuária; (2) T-DISCONNECT.indication recebido da camada de transporte; (3) T-DISCONNECT.request gerado pela camada usuária; (4) T-CONNECT.indication recebido da camada de transporte; (5) T-CONNECT.confirm recebido da camada de transporte; (6) T-CONNECT.response gerado pela camada usuária; (7) T-DATA.request gerado pela camada usuária; (8) T-DATA.indication recebido da camada de transporte; (9) T-EXPEDITED-DATA.request gerado pela camada usuária; (10) T-EXPEDITED-DATA.indication recebida da camada de transporte.

5.4 Protocolos de Transporte Classe 4

Protocolos classe 4 operam em subredes tipo C que oferecem serviços de datagramas sem garantia de entrega, ordem ou ausência de duplicação. Portanto, cabe ao serviço de transporte adicionar a confiabilidade não oferecida pela camada de rede. Protocolos classe 4 são empregados também em subredes classe B, onde a camada de transporte pode se valer de serviços orientados à conexão oferecidos pela camada de rede. Neste caso, as diferenças entre um protocolo classe 3 e 4 são desprezíveis em termos de desempenho, já que ambos terão que gerenciar apenas as quebras de conexão no nível da camada de rede.

Um protocolo de transporte classe 4 deve gerenciar a perda, duplicação e inversão de ordem dos pacotes em três situações:

1. durante o estabelecimento de conexões de transporte;
2. durante a transferência de TPDU's de dados;
3. durante o fechamento de conexões.

Vamos examinar as soluções comumente empregadas nos protocolos classe 4 para estas três situações.

5.4.1 Estabelecimento de Conexões

Dois cenários devem ser gerenciados por um protocolo de transporte classe 4: duplicação de uma TPDU que carrega uma requisição de conexão (gerado pela chamada *T-CONNECT.request*); e perda de uma TPDU que carrega a confirmação do estabelecimento da conexão (que iria gerar um *T-CONNECT.confirm*).

No primeiro caso, ao receber uma TPDU duplicada, o destinatário irá supor que uma segunda conexão foi solicitada. No segundo cenário, o host que solicitou a conexão supõe que sua requisição se perdeu e a envia novamente. Uma segunda conexão será estabelecida pelo receptor, como no caso anterior.

Os dois cenários descritos acima apresentam uma característica em comum: uma conexão estabelecida apenas num host, e que portanto jamais será utilizada. Os recursos alocados a tais conexões são totalmente inúteis e permanecem alocados até esta situação ser detectada e corrigida.

Uma solução comumente empregada é o estabelecimento de conexões em três fases. Esta técnica evita conexões inúteis geradas por duplicação ou perda de TPDU.

Ao enviar um pedido de conexão, o host solicitante informa um número inicial de sequência na numeração das TPDU de dados que ele enviar pela conexão: x (primeira fase). Ao receber a solicitação, o destinatário reconhece x e informa que seu número inicial de sequência será y (segunda fase). Ao receber a confirmação da conexão, o host que a iniciou reconhece y numa TPDU de reconhecimento ou na primeira TPDU de dados que enviar (terceira fase). Este método de estabelecimento de conexões (figura 5.3) é imune a duplicação ou perda de TPDU empregadas no estabelecimento de conexões.

No primeiro cenário, o host que recebeu a segunda confirmação de conexão, responderá com um REJECT na terceira fase, causando a desativação imediata da conexão gerada pela TPDU duplicada. No segundo cenário, a primeira conexão será desativada por falta do recebimento de confirmação na terceira fase (após *timeout*).

5.4.2 Transferência de TPDU de Dados

Perdas, duplicações e inversões de ordem são tratadas de forma adequada por protocolos de controle de fluxo com reconhecimento e *piggybacking*. Um problema adicional nos protocolos de transporte classe 4 reside na alocação de buffers. TPDU transmitidas devem permanecer armazenadas no emissor até a chegada de reconhecimento por parte do receptor. O receptor necessita também armazenar TPDU até que se complete uma janela (no caso de reconhecimento contínuo) ou uma mensagem (composta de várias TPDU).

Via de regra, os esquemas de *bufferização* empregam um *pool* de buffers compartilhados por todas as conexões ou um buffer de grandes dimensões para cada conexão. Neste último caso, o buffer único abriga várias TPDU e sua manutenção lembra gerenciamento de memória em sistemas operacionais.

Tanto no emprego de um *pool* de buffers quanto no emprego de um buffer único fica a questão: qual o tamanho dos buffers? Esta questão é importante pois o tamanho das TPDU varia de poucos bytes como numa sessão de *login* remoto, à milhares de bytes como em transferência de arquivos.

contador sem a chegada de confirmação, o pedido de encerramento de conexão é retransmitido. O processo se repete por determinado número de vezes quando a conexão é encerrada unilateralmente pelo lado do emissor.

O receber uma TPDU solicitando o encerramento de conexão, o receptor também dispara um contador de tempo (com intervalo bem superior ao do emissor), enviando prontamente a confirmação ao solicitante. Se este tempo se expirar sem o recebimento da "confirmação da confirmação", o host encerra unilateralmente a conexão. A figura 5.4 ilustra este procedimento.

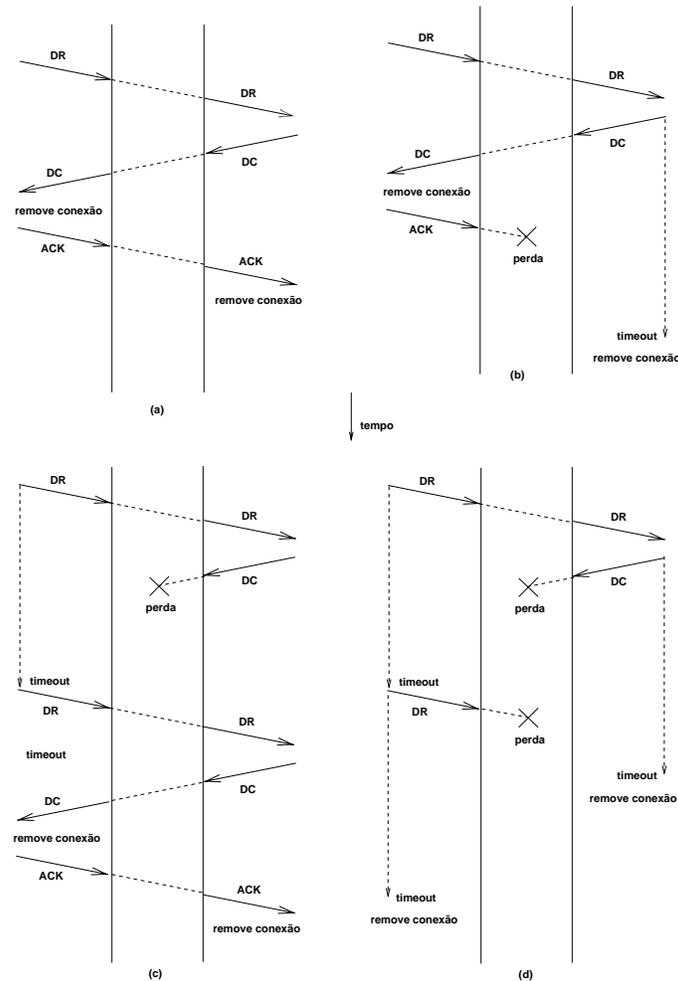


Figura 5.4: Encerramento de conexão em 3 fases: (a) situação Normal; (b) falha na terceira fase; (c) falha na segunda fase; (d) falhas na segunda e terceira fase.

Este procedimento garante também que TPDU's "atrasadas" não serão descartadas. Simplesmente, um host não aceita o encerramento da conexão (emitindo um REJECT na terceira fase) se uma ou mais TPDU de dados ainda não teve seu recebimento confirmado.

Encerramento de conexão em três fases não garante que uma conexão será encerrada em cem por cento dos casos (por exemplo, quando nenhuma das TPDU's solicitando o encerramento da conexão atingem o destino). Uma segurança adicional é encerrar uma conexão

caso permaneça inativa por um determinado período de tempo. TPDU's de conteúdo nulo podem ser empregados para manter a conexão viva, fazendo com que sua recepção zere o contador de tempo que monitora e inatividade da conexão.

5.5 Multiplexação

Multiplexação no nível de camada de transporte é a capacidade de:

1. operar múltiplas conexões de transporte através de uma única conexão de rede (multiplexação *upward*); ou
2. uma única conexão de transporte utilizar múltiplas conexões de rede (multiplexação *downward*).

O primeiro caso é comum em redes públicas onde a taxação é função do número de conexões requisitadas à subrede de comunicação. Obviamente a qualidade do serviço é pobre em conexões de transporte multiplexadas numa mesma conexão de rede. Protocolos de transporte classe 2 em diante são dotados de capacidade de multiplexação *upward*.

O segundo caso (multiplexação *downward*) ocorre quando uma conexão de transporte necessita determinada vazão não assegurada pela conexão de rede. Neste caso, distribui-se as TPDU's por várias conexões de rede, aumentando a vazão da conexão de transporte.

5.6 Protocolos de Transporte OSI Orientados a Conexão

Os protocolos de transporte OSI é um padrão ISO (8073) tipicamente em conjunção com o X.25. Este protocolo define 10 classes de TPDU's. Os formatos são os mesmos para as 5 classes de protocolos, exceto que certas classes não necessitam de todos as TPDU's.

Os três primeiros campos das TPDU's são comuns. O primeiro, LI (Length Indicator), ocupa 1 byte e contém o tamanho do cabeçalho (partes fixa e variável). Segue o segundo campo, também de 1 byte, contendo nos primeiros 4 bits o tipo de TPDU e nos 4 bits restantes o tamanho da janela (Cdt: crédito), quando for o caso. O terceiro campo ocupa 2 bytes e contém um identificador da conexão. Uma conexão é identificada por dois inteiros denominados referência destino (estabelecido por quem aceita a conexão) e referência fonte (estabelecido por quem inicia a conexão).

Os demais campos dependem do tipo de TPDU. Os 10 tipos são descritos a seguir e apresentados na figura 5.5.

CR (Connection Request): inicia uma conexão de transporte. O campo CLASSE propõe uma classe de protocolo para a conversação (geralmente a classe 4 é empregada). A parte variável consiste de:

1. os TSAPs de origem e destino;
2. o tamanho máximo das TPDU's (de 128 a 8192, em potências de 2);
3. a versão do protocolo;

4. uma chave de proteção (para fins de criptografia, direito de acesso ao serviço, etc);
5. um indicador de uso de *checksum*;
6. parâmetros de qualidade do serviço.

Esta TPDU pode carregar até 32 bytes de dados do usuário (entregue à entidade que mantém o TSAP destino).

CC (Connection Confirm): Provê confirmação positiva ou negativa de uma requisição de conexão. Possui formato idêntico ao CC, carregando na parte variável a contraproposta referente aos parâmetros de qualidade de serviço contidos na requisição.

DR (Disconnect Request): solicita o encerramento da conexão. O campo MOTIVO provê um indicativo do motivo de tal requisição. A parte variável, quando for o caso, carrega informações complementares ao campo MOTIVO.

DC (Disconnect Confirm): confirma o término da conexão. A parte variável carrega apenas o *checksum*, se sua utilização foi decidida no estabelecimento da conexão.

DT (Data): transporta dados. O campo EOT (End of Text) ocupa 1 bit e indica se esta TPDU é a última de uma mensagem ou não (em sendo o último, a mensagem pode ser entregue à entidade receptora). Completando o byte, o campo TPDU-N, de 7 bits, provê o número de sequência da TPDU em módulo 7.

ED (Expedited Data): carrega dados expressos. A parte variável contém o *checksum*, quando for o caso.

AK (Acknowledgement): reconhece TPDU's de dados (DT ou ED). O campo Cdt pode alterar a janela previamente em uso para fins de controle de fluxo. A parte variável contém o *checksum*, quando for o caso.

RJ (Reject): utilizado nos protocolos de classe 1 e 3, solicita a transmissão de todas as TPDU's a partir e inclusive do número de sequência constante no campo TPDU-ESPERADA.

ER (Error Report): informa a ocorrência de erros tais como parâmetros desconhecidos, tipo de TPDU inválido, etc. O campo MOTIVO informa o erro detectado. Caso necessário, maiores informações são supridas na parte variável.

5.7 Protocolos de Transporte OSI sem Conexão

O protocolo de transporte OSI sem conexão define um único tipo de TPDU utilizada no transporte de dados (figura 5.5). A parte variável do cabeçalho contém os TSAPs de origem e destino, e, opcionalmente, o *checksum*.

Cada TPDU é transmitida num pacote separado, sem confirmação de recebimento. Não há garantia de entrega, ordem, ou ausência de duplicação. Protocolos de transporte sem conexão é utilizado principalmente em redes classes A (locais).

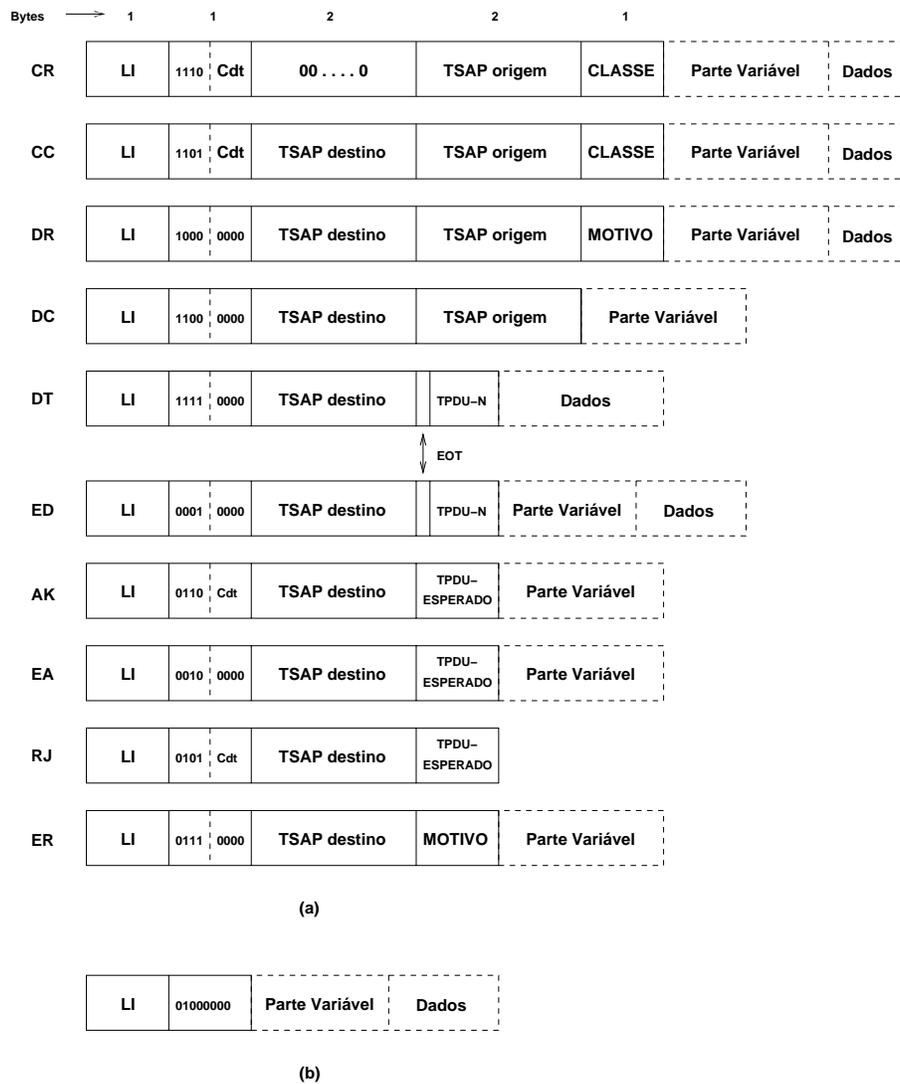


Figura 5.5: PDUs de transporte definidos pela ISO: (a) transporte orientado a conexão; (b) transporte orientado a datagrama.

5.8 Problemas

1. Quais as funções precípua da camada de transporte?
2. O que é "qualidade de serviço" e sob que parâmetros é determinada?
3. Qual a finalidade de se definir classes de protocolos de transporte?
4. Descreva como conexões são estabelecidas em protocolos de transporte classe 4.
5. Como se dá o controle de fluxo em protocolos de transporte classe 4?
6. Quais as diferenças no procedimento de três fases para se abrir e fechar conexões?

7. Defina multiplexação *upward* e *downward*.
8. Mapeie as primitivas OSI nos PDUs de transporte definidos pela ISO.
9. Defina um algoritmo a ser utilizado num protocolo de transporte para tratar os N-RESETs produzidos pela camada de rede.
10. É razoável se empregar um protocolo de transporte sem conexão em subredes de comunicação orientadas à conexão (X.25, por exemplo)? Justifique.