Capítulo 6

A CAMADA DE SESSÃO

A camada de sessão tem por finalidade organizar a troca de informação (diálogo) entre dois hosts comunicantes. Esta camada provê mecanismos para:

- estabelecer e terminar sessões de diálogo;
- trocar informação em uma sessão de diálogo;
- gerenciar o diálogo;
- sincronizar o diálogo;
- definir e gerenciar unidades de diálogo (atividades);
- reportar erros detectados no decorrer do diálogo.

Vamos examinar cada uma das funcionalidades acima.

6.1 Estabelecimento e Término do Diálogo

Dependendo do protocolo de transporte, uma conexão de transporte pode se encerrar abruptamente por uma das partes, causando, eventualmente, perda de informação. A figura 6.1(a) ilustra esta situação onde uma conexão foi terminada antes da outra parte encerrar o envio de dados. Isto de deve ao fato das primitivas OSI de transporte prever o encerramento da conexão como um serviço sem confirmação (primitivas do tipo *request* e *indication* apenas, ver figura 5.1).

A camada de sessão provê encerramento de uma conexão de sessão como um serviço confirmado (primitivas *request, indication, response* e *confirm*). Neste caso, uma aplicação envolvida em uma sessão de diálogo, ao receber uma indicação de desconexão, emite a resposta afirmativa apenas caso não tenha mais informação para transmitir. Isto garante que a conexão se encerra sem a perda de informação, mesmo que a conexão de transporte da qual a conexão de sessão se utilizava se encerre abruptamente. A figura 6.1(b) ilustra o encerramento ordenado de uma sessão de diálogo

Uma conexão de sessão pode se encerrar abruptamente caso a própria camada de sessão tome a iniciativa do encerramento. Um exemplo desta situação é a quebra da conexão de transporte da qual a conexão de sessão fazia uso.

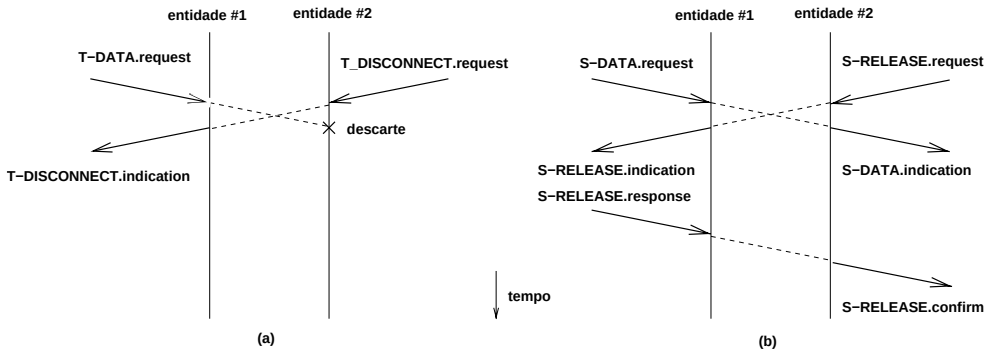

Figura 6.1: Encerramento de conexões: (a) de transporte; (b) de sessão (RELEASE tem o significado de DISCONNECT).

6.2 Troca de Informação

Sessões de diálogo podem conduzir quatro tipos de dados:

1. dados regulares: informação que diz respeito apenas às entidades comunicantes da aplicação;
2. dados expressos: informação urgente (gerada pela aplicação) entregue fora de sequência;
3. dados tipados: informação enviada sem que o transmissor detenha o *token* de envio (veja seção 6.3);
4. dados de capacidade: empregados exclusivamente para controle interno da camada de sessão, permitem alterar parâmetros do diálogo.

As primitivas para envio dos três primeiros tipos de dados de sessão são sem confirmação. O envio de dados de capacidade é um serviço confirmado, pois a alteração de parâmetros do diálogo dependem do aceite da outra parte envolvida no diálogo.

6.3 Gerenciamento do Diálogo

Conexões de transporte usualmente são do tipo full-duplex, onde a informação flui nos dois sentidos simultaneamente. Entretanto, muitas aplicações se comunicam em modo half-duplex, onde a informação flui em um sentido de cada vez. Comunicação half-duplex pode ser implementada em conexão full-duplex pela introdução de um *token*: apenas quem detiver o *token* está habilitado a enviar informação pela conexão.

No estabelecimento de uma conexão de sessão é especificado o tipo da conexão half- ou full-duplex. No caso de ser escolhida conexão half-duplex, deve-se especificar também a qual das partes será atribuído o *token* primeiro.

Terminado o envio de informação, o detentor do *token* passa-o à outra parte. O *token* pode ser requisitado a qualquer momento pela entidade que não o detém. Recebida a requisição, a entidade detentora do *token* pode, a seu critério, passá-lo a quem o solicitou.

A figura 6.2 ilustra as o emprego do *token* e dados tipados em comunicação half-duplex.

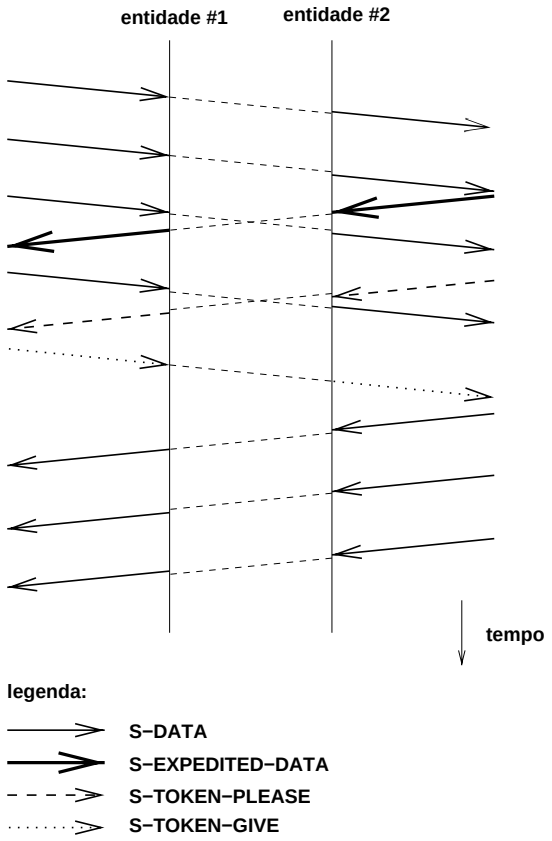

Figura 6.2: Comunicação típica no nível de sessão.

6.4 Sincronização do Diálogo

Em uma sessão de diálogo pode-se definir *pontos de sincronismo*. Pontos de sincronismo estabelecem marcas no diálogo para fins de retomada da comunicação a partir destas marcas. Pontos de sincronismo são definidos pelas entidades comunicantes, não pela camada de sessão (esta os adiciona unicamente sob requisição da aplicação). A introdução de pontos de sincronismo necessita o reconhecimento da outra entidade comunicante, sendo portanto um serviço confirmado.

Seja o exemplo de um a rede se serviços integrados que oferece transmissão por fax. Cada página do texto é transmitida como uma série de mensagens de dados. Antes do envio de cada página, o transmissor adiciona um ponto de sincronismo. Suponha que o receptor roteie os dados das SPDUs diretamente para uma impressora, sem armazená-los localmente. Se ocorrer uma falha na impressão (por exemplo, travamento do papel), o receptor solicita ao transmissor que volte ao ponto de sincronismo referente à página afetada pelo problema, passando a receber todas as mensagens a partir daí. O que ocorreu neste caso foi uma re-sincronização do diálogo a partir de um ponto do passado.

Caso pontos de sincronismo sejam empregados, o transmissor deve armazenar as mensagens na eventualidade do receptor re-sincronizar o diálogo. Obviamente, a quantidade de mensagens armazenadas deve ser limitada. Para tal, define-se dois tipos de pontos de

sincronismo: menor (como os do exemplo acima) e maior.

Pontos de sincronismo maior (PSM) definem barreiras cujo retrocesso não pode ser ultrapassado, limitando, portanto, a quantidade de mensagens armazenadas para fins de resincronização. Retomando o exemplo anterior, suponha que o transmissor adicione um PSM a cada cinco pontos de sincronismo menor (que delimitam as páginas). Agora, ao reconhecer um PSM, o receptor deve ter certeza que todas as páginas transmitidas até então foram corretamente impressas, pois as mensagens transmitidas antes de um PSM são descartadas pelo transmissor.

A quantidade de informação transmitida entre dois PSMs é denominada *unidade de diálogo* ou *atividade* (figura 6.3).

6.5 Gerenciamento de Atividades

Atividades são unidades delimitadas de uma sessão de diálogo. Geralmente a delimitação das atividades está relacionada com fronteiras naturais reconhecidas pelas entidades comunicantes. Por exemplo, em uma sessão de transferência de arquivos, a transmissão de cada arquivo estaria associada uma atividade (figura 6.3).

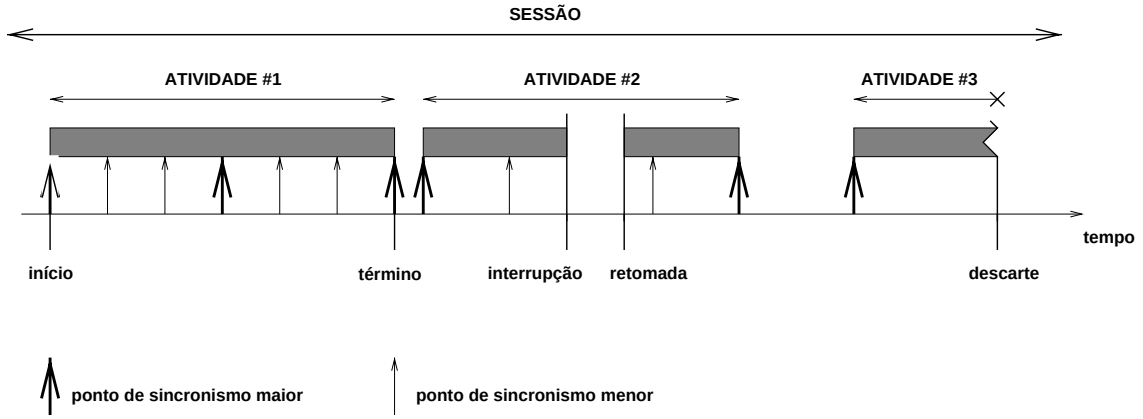

Figura 6.3: Partição de uma sessão de diálogo em atividades.

Atividades em uma mesma sessão de diálogo estão delimitadas por pontos de sincronismo maior. Como consequência, durante o processamento de uma atividade não podemos resincronizar o diálogo a um ponto pertencente à atividade anterior.

As operações relacionadas com atividades são:

- início: marca o início de nova atividade inserindo um ponto de sincronismo maior (exemplo: inicio da transmissão de arquivo);
- término: informa que a atividade corrente terminou de forma normal (exemplo: fim da transmissão de arquivo);
- descarte (término anormal): abandona a atividade corrente (exemplo: descarte da parte já transmitida do arquivo);

- interrupção: suspende temporariamente uma atividade (exemplo: suspensão de transmissão de arquivo);
- retomada: retoma uma atividade a partir do ponto em que foi interrompida (exemplo: retoma a transmissão de arquivo).

As consequências de se iniciar, terminar, descartar, suspender e retomar atividades dizem respeito apenas às aplicações usuárias da camada de sessão. A camada de sessão apenas provê as primitivas para tal, gerando as respectivas SPDUs de controle que causam as respectivas indicações na camada de sessão oposta. As primitivas para iniciar e retomar atividade interrompida são serviços sem confirmação; as demais são serviços confirmados.

Finalmente, nada impede ao usuário inserir pontos de sincronismo maior menor no interior de uma atividade.

6.6 Detecção de Erros

A camada de sessão provê apenas duas primitivas para o informe de situações anormais. A primeira é utilizada pela aplicação. Por exemplo, uma aplicação remota solicita à local a transferência de um arquivo inexistente. Esta primeira primitiva define um serviço sem confirmação.

A segunda primitiva é utilizada pela própria camada de sessão quando esta detecta alguma situação anormal. As aplicações conectadas em uma sessão de diálogo recebem apenas uma indicação do erro, podendo proceder como desejarem. Exemplo de erro neste nível é a perda do sequenciamento na transmissão das SPDUs que compõem uma mensagem.

6.7 Primitivas OSI de Sessão

A tabela 6.1 apresenta as primitivas OSI para a camada de sessão. As primitivas fornecem os serviços (orientados à conexão) descritos nas seções precedentes. Primitivas iniciando com S-P possuem o modo *indication* apenas e se prestam para o informe de eventos detectados pela entidade provedora do serviço (neste caso, os protocolos da camada de sessão). Uma única primitiva, S-UNITDATA, provê transmissão de informação sem o estabelecimento de conexão.

6.8 Protocolo OSI de Sessão

O protocolo OSI de sessão (ISO 8327) é inspirado em um protocolo antigo da ITU-T (ex-CCITT) para teletexto. A forma geral das SPDUs é dada na figura 6.4(a). A campo SI (Session Identifier) de um byte contém o tipo da SPDU. O campo LI (Length Identifier), também de um byte, indica o tamanho (em bytes) do campo de parâmetros. LI varia entre 0 e 254: um valor de 255 indica a existência de 2 bytes adicionais para este campo.

Após os campos SI e LI, vêm os parametros da SPDU. Parâmetros têm o formato da figura 6.4(b). PI (Parameter Identifier) informa o tipo do parâmetro; LI seu tamanho; PV (Parameter Value) seu valor.

primitiva	finalidade
S-CONNECT	Estabelecimento de uma conexão de sessão
S-RELEASE	Término de uma conexão de sessão
S-U-ABORT	Termina adruptamente uma conexão
S-P-ABORT	Informa o término adrupto de uma conexão
S-DATA	Transfere de dados
S-EXPEDITED-DATA	Transfere de dados expressos
S-TYPED-DATA	Transfere de dados fora de faixa (out-of-band)
S-CAPABILITY-DATA	Transfere de informação de controle
S-TOKEN-GIVE	Passa o *token* à outra entidade
S-TOKEN-PLEASE	Solicita o *token* de outra entidade
S-CONTROL-GIVE	Passa todos os *tokens* à outra entidade
S-SYNC-MAJOR	Insere um ponto de sincronização maior
S-SYNC-MINOR	Insere um ponto de sincronização menor
S-RESYNCHRONIZE	Retorna a um ponto de sincronização
S-ATIVITY-START	Inicia uma atividade
S-ACTIVITY-END	Termina uma atividade
S-ACTIVITY-DISCARD	Descarta uma atividade
S-ACTIVITY-INTERRUPT	Interrompe uma atividade em curso
S-ATIVITY-RESUME	Retoma uma atividade interrompida
S-U-EXCEPTION-REPORT	O usuário informa sobre uma exceção
S-P-EXCEPTION-REPORT	A entidade provedora informa sobre uma exceção
S-UNITDATA	Transfere dados sem o estabelecimento de conexão

Tabela 6.1: Primitivas OSI para a camada de sessão. As primitivas em negrito são serviços confirmados.

Certas SPDUs provêem campo de dados do usuário para que os usuários da camada de sessão enviem informação adicional àquelas passadas nas primitivas (a camada de sessão não interpreta estes dados).

Na maioria dos casos, a chamada de cada primitiva de sessão gera uma SPDU do tipo da primitiva. Primitivas que implementam serviços confirmados definem também suas correspondentes SPDUs de reconhecimento.

6.9 Problemas

1. Quais as funcionalidades que a camada de sessão provê?

2. Quais as diferenças entre uma conexão de transporte e uma conexão de sessão?

3. Como a camada de sessão implementa comunicação half-duplex?

4. Como a camada de sessão classifica os dados por ela transmitidos?

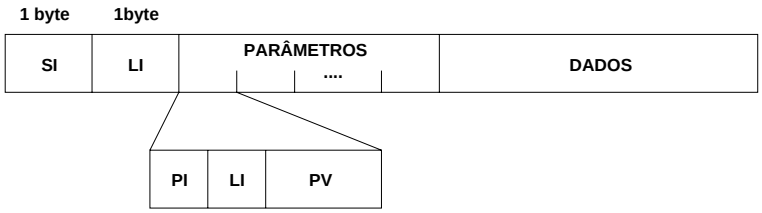

Figura 6.4: Formato geral de um SPDU (para compatibilidade com o protocolo ITU-T para teletexto).

5. Como funciona o mecanismo de sincronização da camada de sessão?
6. Quais as vantangens de se organizar o diálogo em atividades?
7. Quais as operações de gerenciamento de atividades?
8. Faça um diagrama temporal mostrando uma re-sincronização do diálogo. Idem para o início, suspensão, retomada e término de uma atividade.
9. Cite as primitivas OSI de sessão.
10. Descreva os campos de uma SPDU do protocolo OSI de sessão.