

O Modelo OSI de Arquitetura

1.1. Introdução

O final da década de 70 apresentava um panorama curioso em termos de comunicação de dados em redes de computadores: por um lado, uma perspectiva de crescimento vertiginoso causados pelo investimento e desenvolvimento que estavam sendo feitos, mas por outro lado uma tendência que poderia acarretar em uma profunda crise no setor, a heterogeneidade de padrões entre os fabricantes, praticamente impossibilitando a interconexão entre sistemas de fabricantes distintos.

Então os fabricantes começaram a perseguir alguns objetivos necessários para a implementação de um sistema aberto. Esses objetivos são:

- interoperabilidade: capacidade que os sistemas abertos possuem de troca de informações entre eles, mesmo que sejam fornecidos por fabricantes diversos;
- interconectividade: é a maneira através da qual se pode conectar computadores de fabricantes distintos;
- portabilidade da aplicação: é a capacidade de um software de rodar em várias plataformas diferentes;
- "scalability": capacidade de um software rodar com uma performance aceitável em computadores de capacidades diversas, desde computadores pessoais até supercomputadores.

Para se atingir estes objetivos, a ISO (International Organization for Standardization) passou a se ocupar em criar um padrão de arquitetura aberta e baseada em camadas. Foi então definido o Modelo de Referência para Interconexão de Sistemas Abertos (Reference Model for Open Systems Interconnection - RM OSI).

A utilização de um ambiente de sistema aberto nos oferece algumas vantagens, como:

- liberdade de escolha entre soluções de diversos fabricantes;

- acesso mais rápido a novas tecnologias e a preços mais acessíveis, já que é mais barato e rápido fabricar produtos baseados em uma plataforma padrão;
- redução de investimentos em novas máquinas, já que os sistemas e os softwares de aplicação são portáteis para os vários tipos de máquinas existentes.

A adoção de um modelo baseado em camadas também não é arbitrária. Considerando que uma rede de computadores tem como objetivo o processamento de tarefas distribuídas pela rede de forma harmônica e cooperativa entre os vários processos de aplicação, o projeto desta deve levar em conta vários fatores, como:

- considerar todos os eventos possíveis de acontecer durante a comunicação;
- conhecer todos os efeitos e causas destes eventos;
- especificar em detalhes todos os aspectos técnico-operacionais dos meios físicos a serem utilizados como suporte à comunicação;
- detalhes das próprias aplicações a serem executadas.

Podemos perceber, então, que o problema é extremamente complexo e abrangente. A fim de se lidar com esta complexidade (facilitando a implementação e manutenção), projeta-se a rede como um conjunto de camadas.

Este conjunto de camadas é hierárquico, ou seja, cada camada baseia-se na camada inferior [Moura 86]. Reduzindo-se o projeto global da rede ao projeto de cada uma das camadas, simplifica-se consideravelmente o trabalho de desenvolvimento e de manutenção. O projeto de uma camada é restrito ao contexto dessa camada e supõe que os problemas fora deste contexto já estejam devidamente resolvidos.

Na realidade existem duas vantagens práticas na utilização de uma arquitetura em camadas. Em primeiro lugar, a complexidade do esforço global de desenvolvimento é reduzida através de abstrações (não interessa para uma determinada camada como as demais implementam o fornecimento de seus serviços, só o que elas oferecem). Na arquitetura hierárquica, a camada (N) sabe apenas que existem a camada (N-1), prestadora de determinados serviços e a camada (N+1), que lhe requisita

os serviços. A camada (N) não toma conhecimento da existência das camadas (N±2), (N±3), etc.

O segundo aspecto é relacionado com a independência entre as camadas. A camada (N) preocupa-se apenas em utilizar os serviços da camada (N-1), independentemente do seu protocolo. É assim que uma camada pode ser alterada sem mudar as demais (facilidade de manutenção) - desde que os serviços que ela presta não sejam modificados. É assim também que novas aplicações podem ser implementadas, na camada apropriada, aproveitando os mesmos serviços já fornecidos pelas outras camadas (redução dos esforços para evoluções).

Porém a elaboração de um sistema aberto passa por algumas etapas obrigatórias que podemos observar claramente na definição do modelo OSI, da ISO:

- definição do modelo do sistema aberto (padrão para arquitetura do sistema aberto);
- definição dos padrões dos componentes que fazem parte do modelo (padrões de interoperabilidade e portabilidade), não só os relacionados à comunicação, mas também alguns não relacionados, como estrutura de armazenamento de dados, etc;
- seleção dos perfis funcionais.

Podemos observar que o modelo OSI da ISO corresponde exatamente ao primeiro item citado acima. O modelo OSI é um modelo de referência e define apenas a arquitetura do sistema. O padrão criado para o modelo OSI, então, define exatamente o que cada camada deve fazer, mas não define como isto será feito, ou seja, define os serviços que cada camada deve prestar, mas não o protocolo que o realizará. Este primeiro passo já está bem definido pela ISO.

A definição dos protocolos de cada camada, então, fica por conta do segundo passo. Esta parte também está definida pela ISO, mas é realizado por grupos de estudo diversos. Este passo é uma tarefa muito dinâmica, pois novas tecnologias de transmissão surgem a todo instante. Portanto por um lado temos alguns padrões bem documentados, mas por outro, temos tecnologias emergentes que precisam ser adaptadas às condições do modelo OSI e ainda estão em processo de definição.

Já a terceira etapa não é uma fase de responsabilidade da ISO. Esta etapa de definição de perfis funcionais é realizada por cada país, que escolhe os padrões que lhe cabem baseados em condições tecnológicas, base instalada, visão futura, etc. Por exemplo, no Brasil temos o Perfil Funcional do Governo Brasileiro. A escolha do Perfil Funcional é uma etapa importante, pois apesar de dois sistemas seguirem o Modelo OSI, se eles adotarem perfis diferentes, eles nunca vão conseguir interoperar.

A arquitetura OSI foi desenvolvida a partir de três elementos básicos [Moura 86]:

- os processos de aplicação existentes no ambiente OSI;
- as conexões que ligam os processos de aplicação e que lhes permitem trocar informações;
- os sistemas.

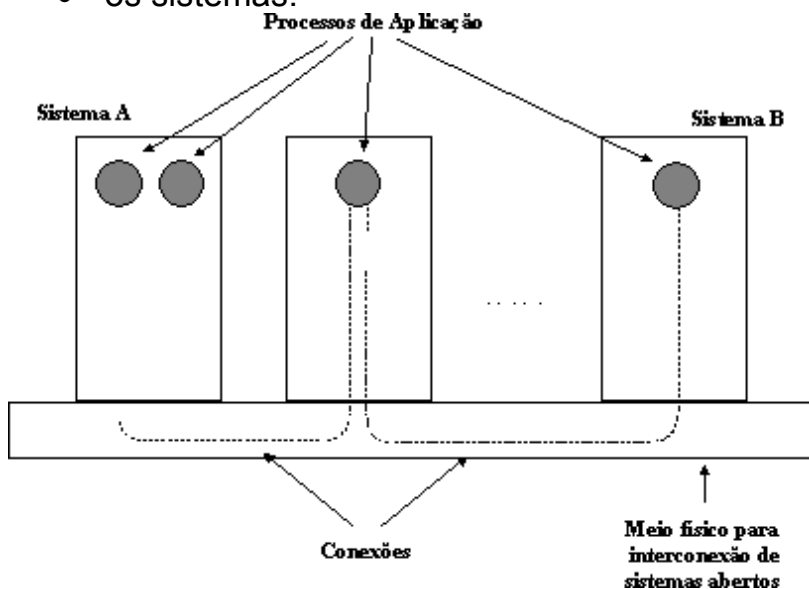


Figura 2.1: Processos de aplicação, conexões e sistemas

O desenho abaixo nos dá uma idéia da arquitetura de uma máquina pertencente a um sistema de comunicação:

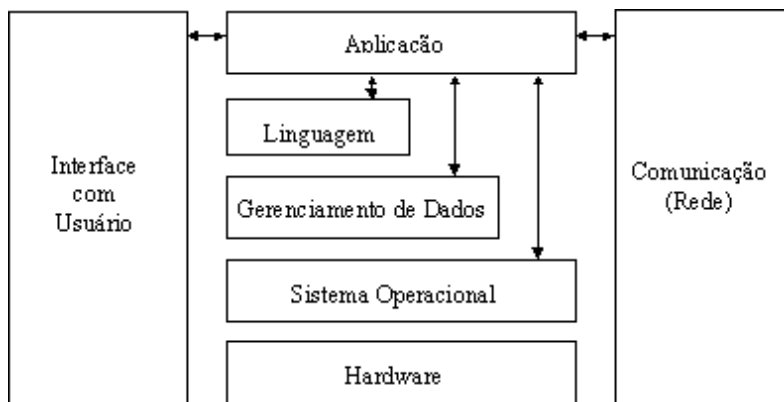


Figura 2.2: Arquitetura de uma máquina do sistema

Hardware: provê a infra-estrutura necessária (no nível mais baixo) para o processamento da aplicação, como a manipulação de bits, acesso a disco, etc. · Sistema operacional: provê os serviços básicos de acesso a hardware, etc. · Gerenciamento de dados: cuida de tarefas como o acesso, manipulação e troca de vários tipos de dados. Uma consistência nesta tarefa é um grande passo rumo à portabilidade de aplicações. Existem várias formas de implementação de acesso a bancos de dados, mas a mais comum e aceita pela indústria é a SQL (Structure Query Language). ·

Linguagem: tem sido feitos esforços em relação à criação de uma linguagem com independência da plataforma, de forma a prover a portabilidade de código. · Interface com o usuário: um dos principais fatores de portabilidade, já que provê a interface com o usuário da aplicação. Cada vez mais estão sendo desenvolvidas interfaces gráficas e orientadas a objetos baseadas em janelas, ícones e menus.

Os principais padrões para desenvolvimento de interfaces gráficas são X Window e Motif. · Comunicação: a parte de comunicação é o objeto principal do nosso estudo. Ela vai prover a comunicação e interoperação entre máquinas e sistemas diferentes, cuidando de características como padrões de interoperação, endereçamento, etc.

O modelo OSI, então, se encaixa na figura 2.2 como um conjunto de funções que possibilitam que máquinas distintas possam se comunicar e trocar informações. Ele possui sete camadas (figura 2.3), onde cada camada é responsável por uma determinada função específica. Os princípios utilizados para se chegar a estas camadas são [Tanenbaum 94]:

- uma camada deve ser criada onde é necessário um nível de abstração diferente;
- cada camada deve desempenhar uma função bem definida;
- a função de cada camada deve ser definida tendo em vista a definição de protocolos padrões internacionais;
- as fronteiras entre as camadas devem ser escolhidas de forma a minimizar o fluxo de informações através das interfaces;
- o número de camadas deve ser grande o suficiente para que não seja preciso agrupar funções em uma mesma camada por necessidade, e pequeno o suficiente para que a arquitetura fique manejável.

Cada camada é usuária dos serviços prestados pela camada imediatamente inferior e presta serviços para a camada imediatamente superior. Esta troca de informações entre as camadas adjacentes ocorre por meio da troca de primitivas de serviços nas interfaces entre as camadas.

Apesar do modelo OSI estar dividido em sete níveis, pode-se considerar genericamente que as três camadas mais baixas cuidam dos aspectos relacionados à transmissão propriamente dita e a camada de transporte lida com a comunicação fim-a-fim, enquanto que as três camadas superiores se preocupam com os aspectos relacionados à aplicação, já em nível de usuário.

A comunicação entre sistemas ocorre em nível de camadas, ou seja, a camada de aplicação do sistema A se comunica com a camada de aplicação do sistema B e assim por diante até o nível físico, onde ocorre a comunicação física entre os sistemas.

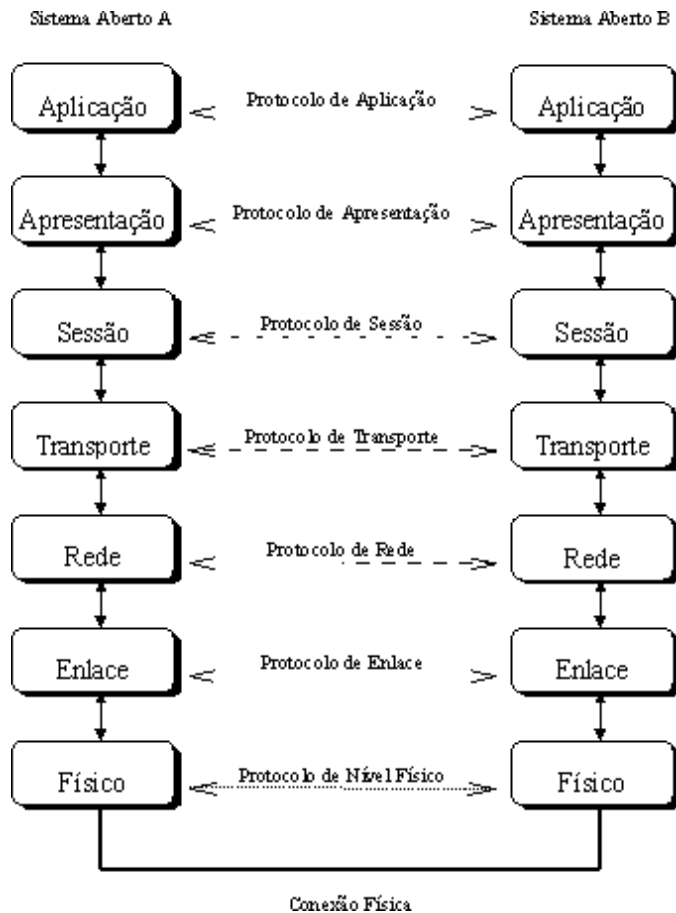


Figura 2.3: Modelo OSI

Uma maneira bastante fácil e simplista de se enxergar a funcionalidade de um modelo em camadas, como o modelo OSI, é imaginar que cada camada tem como função adicionar um cabeçalho aos dados do usuário a serem transmitidos para outro sistema (figura 2.4). Deste modo a função de cada camada do outro sistema é exatamente a inversa, ou seja, retirar os cabeçalhos dos dados que chegam e entregá-los ao usuário em sua forma original.

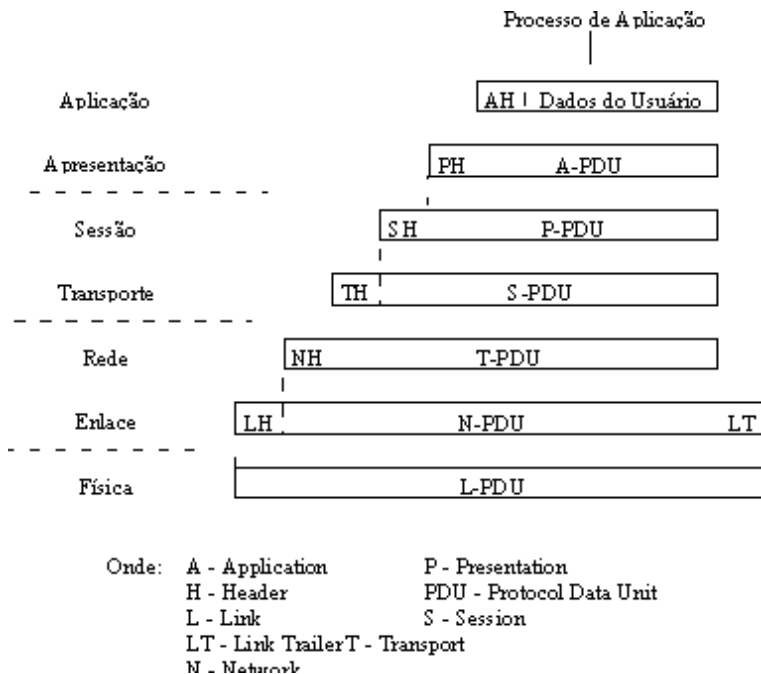


Figura 2.4: Transferência de Dados entre Camadas

Os dados entregues pelo usuário à camada de aplicação do sistema recebem a denominação de SDU (Service Data Unit). A camada de aplicação, então, junta à SDU (no caso, os dados do usuário) um cabeçalho chamado PCI (Protocol Control Information). O objeto resultante desta junção é chamado de PDU (Protocol Data Unit), que corresponde à unidade de dados especificada de um certo protocolo da camada em questão.

1.2. Primitivas de Serviços

As primitivas de serviços são informações trocadas entre duas camadas adjacentes de forma a realizar um serviço. No modelo OSI são definidas quatro tipos de primitivas:

- Pedido (Request): utilizada para solicitar ou ativar um determinado serviço;
- Indicação (Indication): informa a ocorrência de um determinado evento;
- Resposta (Response): utilizada para responder a um determinado evento;

- Confirmação (Confirmation): utilizada para confirmar a execução de um serviço solicitado.

As primitivas possuem parâmetros de entrada e saída. Por exemplo, em um pedido de conexão, os parâmetros podem especificar a máquina à qual se conectar, o tipo de serviço desejado e o tamanho máximo de mensagem a ser utilizada e os parâmetros em uma indicação de conexão podem conter a identidade do solicitante, o tipo de serviço e o tamanho máximo de mensagem proposto. Quem cuida dos detalhes desta negociação é o protocolo. Por exemplo, caso duas propostas para o tamanho máximo das mensagens trocadas seja conflitante, o protocolo deve decidir qual das duas será aceita.

Os serviços prestados podem ser basicamente de dois tipos: confirmado e não confirmado. No serviço confirmado, há um pedido, uma indicação, uma resposta e uma confirmação. Já no serviço não confirmado, há apenas um pedido e uma indicação. Um exemplo de um serviço confirmado é o estabelecimento de uma conexão, enquanto que a desconexão é um serviço não confirmado. Vejamos o exemplo de um serviço de conexão na figura 2.5.

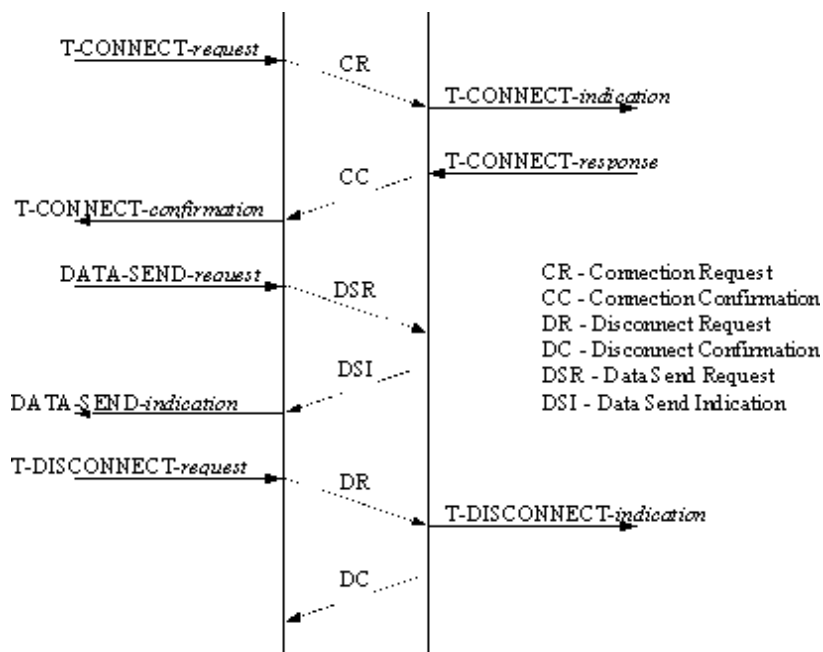


Figura 2.5: Diagrama de Tempo de Estabelecimento de Conexão

Este serviço pode ser descrito da seguinte forma:

- request.CONEXÃO - solicita o estabelecimento de uma conexão;

- indication.CONEXÃO - informa à parte chamada;
- response.CONEXÃO - entidade chamada aceita ou rejeita chamadas;
- confirmation.CONEXÃO - indica ao solicitante se a chamada foi aceita;
- request.DADOS - solicita a transmissão de dados;
- indication.DADOS - avisa sobre a chegada de dados;
- request.DESCONEXÃO - solicita que a conexão seja liberada;
- indication.DESCONEXÃO - informa ao parceiro sobre o pedido.

Um exemplo muito didático é a analogia com o sistema telefônico. Por exemplo, você liga para uma pessoa e a convida para sair:

- request.CONEXÃO - você disca o telefone da pessoa;
- indication.CONEXÃO - o telefone dela toca;
- response.CONEXÃO - ela atende ao telefone;
- confirmation.CONEXÃO - você ouve o sinal de chamada parar de tocar;
- request.DADOS - você convida a pessoa para sair;
- indication.DADOS - ela ouve seu convite;
- request.DADOS - ela responde que sim;
- indication.DADOS - você ouve a aceitação dela;
- request.DESCONEXÃO - você desliga o telefone;
- indication.DESCONEXÃO - ela ouve e desliga também.

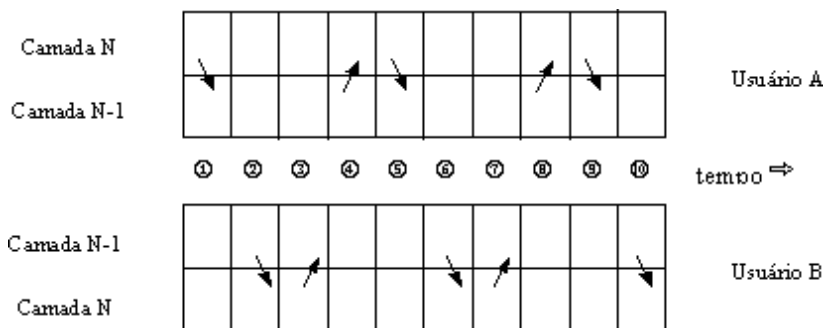


Figura 2.6: Diagrama de Tempo da Conversação Telefônica

A camada N é o usuário, ou seja, você e a pessoa com quem está falando. A camada N-1 é a operadora do serviço. De um modo bastante simplificado, este exemplo nos mostra a troca de primitivas em uma conversação genérica, mas que pode ser perfeitamente aplicada a situações mais complexas, como o modelo OSI.

1.3. Serviços e Protocolos

Faz-se necessário neste ponto deixar bem clara a distinção entre serviços e protocolos. Um serviço é um conjunto de primitivas que uma camada oferece à camada superior adjacente, ou seja, é uma interface entre duas camadas onde a inferior se comporta como provedora do serviço e a superior a usuária do serviço. O serviço define as operações que a camada está preparada para realizar em nome de seus usuários, mas não diz nada a respeito do modo como isso deve ser implementado.

Já um protocolo é um conjunto de regras que governa o formato e significado dos quadros, pacotes ou mensagens trocados entre entidades parceiras dentro de uma mesma camada. Os protocolos são utilizados para implementar os serviços, não sendo diretamente visíveis aos usuários, ou seja, o protocolo utilizado pode ser modificado, desde que o serviço oferecido ao usuário permaneça o mesmo.

Devemos sempre lembrar que ao se falar em serviços, estamos falando em camadas adjacentes (níveis diferentes, no mesmo sistema), e protocolo falamos de entidades pares (no mesmo nível, em sistemas diferentes).

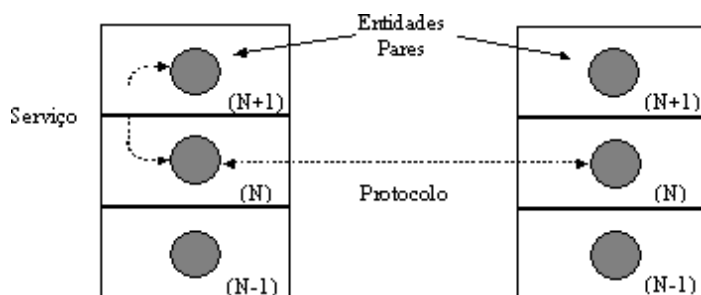


Figura 2.7: Serviços e protocolos no modelo OSI

Os serviços providos pela camada (N) são disponíveis para a entidade (N+1) através dos SAP's (Service Access Point). Os SAP's são interfaces lógicas entre as entidades (N) e (N+1). Portanto, quando a entidade (N+1) precisa utilizar o serviço provido pela camada (N), ela busca este no SAP(N).

As informações entre entidades (N+1) são trocadas através de uma associação chamada conexão (N), estabelecida na camada (N) usando o protocolo (N). A figura 2.8 ilustra este conceito.

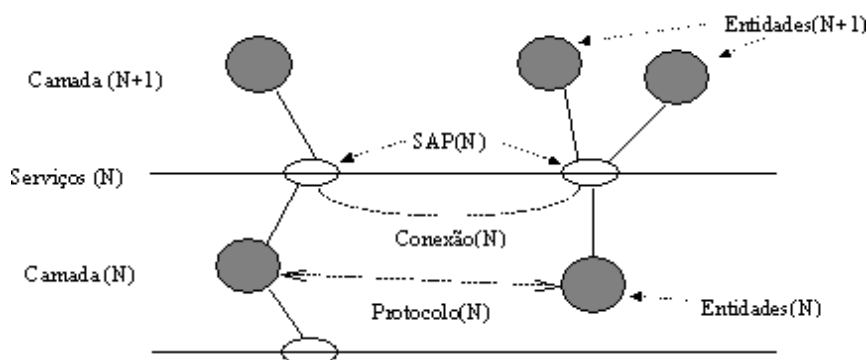


Figura 2.8: SAP's e conexões

1.4. Camada Física

A camada física é a única camada que possui acesso físico ao meio de transmissão da rede devendo, portanto, se preocupar com fatores como as especificações elétricas, mecânicas, funcionais e procedurais da interface física entre o equipamento e o meio de transmissão, ou seja, a camada física tem como função básica a adaptação do sinal ao meio de transmissão.

- mecânicas: propriedades físicas da interface com o meio físico de transmissão, incluindo, por exemplo, o tipo de conector utilizado;
- elétricas: relacionam-se com a representação de um bit em termos de, por exemplo, nível de tensão utilizado e taxa de transmissão de bits;
- funcionais: definem as funções a serem implementadas por esta interface;
- procedurais: especificam a seqüência de eventos trocados durante a transmissão de uma série de bits através do meio de transmissão.

A camada física possui as seguintes funções:

- estabelecimento/encerramento de conexões: ativa e desativa conexões físicas mediante a solicitação de entidades da camada de enlace;
- transferência de dados: a unidade de transmissão utilizada é o bit. O nível físico tem como função transmitir os bits na mesma ordem em que chegam da camada de enlace (no sistema de origem) e entregá-los à camada de enlace na mesma ordem que chegaram (no sistema de destino);
- gerenciamento das conexões: gerência da qualidade de serviço das conexões físicas estabelecidas. Deve monitorar taxa de erros, disponibilidade de serviço, taxa de transmissão, atraso de trânsito etc.

Os padrões de nível físico utilizados são, por exemplo, X.21, X.21 bis, V.24, V.28, RS-232 C, RS-430, RS-431 etc.

1.5. Camada de Enlace

A camada de enlace tem o objetivo de prover uma conexão confiável sobre um meio físico. Sua função básica é detectar e, opcionalmente, corrigir erros que por ventura ocorram no nível físico. As suas principais funções são:

- estabelecimento e liberação da conexão de enlace sobre conexões físicas ativas;
- splitting da conexão de enlace: desta forma pode haver uma conexão de enlace sobre várias conexões físicas;
- montagem e delimitação de quadros (framing): montagem de quadros a partir de unidades de quadros de serviços providas da camada de

rede e reconhecimento de quadros a partir da cadeia de bits vinda do nível físico;

- controle de fluxo: controla a taxa de transmissão dos quadros, evitando que o sistema transmissor envie dados a uma taxa maior do que o receptor consegue processar. Utiliza para isso mecanismos como stop-and-wait, positive acknowledgment e sliding window.
- controle de acesso: gerência do acesso ao meio de transmissão;
- controle de erro: a camada de enlace deve detectar erros de transmissão, de formato e de operação devidos a problemas de conexão física ou mau funcionamento da própria camada. Os erros mais comumente detectados são erros devido a perdas, duplicação, não-ordenação e danificação de quadros.
- controle de seqüência: as unidades de dados de serviço de enlace devem ser entregues à entidade de rede de destino na mesma ordem em que são recebidas da entidade de rede de origem;
- gerenciamento: a camada de enlace deve exercer algumas funções de gerenciamento relacionadas à qualidade de serviço prestado, caracterizada por: tempo médio entre erros irreversíveis, taxa de erro residual decorrente da alteração, perda, duplicação e não-ordenação dos quadros, disponibilidade do serviço, atraso de trânsito e throughput (vazão).

O protocolo de enlace mais conhecido é o HDLC, antecessor de outros como LAPB, LAPD e LLC.

1.6. Camada de Rede

A camada de rede deve tornar transparente para a camada de transporte a forma como os recursos dos níveis inferiores são utilizados para implementar conexões de rede. Deve também equalizar as

diferenças entre as diversas sub-redes utilizadas de forma a fornecer um serviço único a seus usuários (independente da rede utilizada).

Suas principais funções são:

- roteamento: determinação das rotas apropriadas para a transmissão dos dados entre dois endereços (origem e destino) através de algoritmos de roteamento;
- multiplexação da conexão de rede: várias conexões de rede podem ser multiplexadas sobre uma única conexão de enlace, a fim de otimizar a utilização desta última;
- segmentação e blocagem: caso as sub-redes envolvidas em uma comunicação fim-a-fim possuam diversos tipos e tamanhos de quadros, a camada de rede deve exercer funções de segmentação de quadros e remontagem destes no destino;
- controle de erro: detecta e, dependendo da qualidade do serviço exigida, até corrige erros de alteração, perda, duplicação e não-ordenação das unidades de dados;
- seqüenciação: a camada de rede é responsável por manter a ordem das unidades de dados de serviço de rede a serem transmitidas na rede e recebidas pela camada de transporte no destino;
- controle de fluxo: controle da taxa em que os dados são transmitidos, de forma que o transmissor não envie mais dados do que o receptor tenha capacidade de receber;
- transferência de dados expressos: a transmissão de dados expressos tem por finalidade estabelecer prioridade de transmissão para alguns dados (como sinalização e interrupção) sobre os dados normais;
- seleção de serviço: permite a escolha do serviço de rede, de modo a garantir que os serviços oferecidos pelas diversas sub-redes sejam equivalentes;
- gerenciamento: a camada de rede deve efetuar tarefas de gerenciamento relacionadas à qualidade de serviço oferecida, caracterizada pelos parâmetros citados acima.

A camada de rede pode prestar serviços orientados à conexão (CONS - Connection Oriented Network Service) ou serviços não-orientados à conexão (CLNS - ConnectionLess Oriented Network Service). Um exemplo de protocolo utilizado na camada de rede é o X.25.

1.7. Camada de Transporte

A camada de transporte provê mecanismos que possibilitam a troca de dados fim-a-fim, ou seja, a camada de transporte não se comunica com máquinas intermediárias na rede, como pode ocorrer com as camadas inferiores.

As principais funções da camada de transporte são:

- estabelecimento e liberação da conexão de transporte: para se estabelecer a conexão, devem ser negociadas a classe de protocolo a ser utilizada, o tamanho máximo das unidades de dados de protocolo, a utilização ou não do serviço de transferência de dados expressos, parâmetros de qualidade de serviço (throughput, atraso de trânsito, prioridade, taxa de erro residual, etc) etc;
- controle de seqüência e controle de erro: numeração e reconhecimento explícito dos dados a fim de evitar perdas, duplicação ou entrega fora de ordem;
- controle de fluxo: a técnica de controle de fluxo utilizada na camada de transporte é a técnica de alocação de crédito, parecida com a janela deslizando;
- segmentação, blocagem e concatenação fim-a-fim: adapta o tamanho da unidade de dados utilizadas para as camadas de transmissão;
- monitoração da qualidade do serviço: a monitoração da qualidade de serviço deve ser constante, caso contrário deve ser gerada uma notificação à camada de sessão;
- transferência de dados expressos: a camada de transporte deve possibilitar esta funcionalidade;
- gerenciamento: funções de gerenciamento relacionadas à qualidade de serviço prestado às entidades de sessão.

As funções implementadas pela camada de transporte dependem da qualidade de serviço desejada.

Foram especificadas, então, cinco classes de protocolos orientados à conexão:

- classe 0: simples, sem nenhum mecanismo de detecção e recuperação de erros;
- classe 1: recuperação de erros básicos sinalizados pela rede;
- classe 2: permite que várias conexões de transporte sejam multiplexadas sobre uma única conexão de rede e implementa mecanismos de controle de fluxo;
- classe 3: recuperação de erros sinalizados pela rede e multiplexação de várias conexões de transporte sobre uma conexão de rede;
- classe 4: detecção e recuperação de erros e multiplexação de conexões de transporte sobre uma única conexão de rede.

1.8. Camada de Sessão

A camada de sessão é a responsável pelo estabelecimento de sessões entre dois usuários permitindo o transporte ordinário de dados (assim como a camada de transporte), porém com alguns serviços mais refinados, que podem ser úteis em algumas aplicações.

Alguns serviços que a camada de sessão deve prover são, por exemplo:

- gerência do controle de diálogo: a troca de informações entre entidades em um circuito half-duplex deve ser controlada através da utilização de tokens. A camada de sessão é responsável pela posse e entrega destes tokens, ajudando a controlar de quem é a vez de transmitir;
- sincronização: para se evitar, por exemplo, a perda de um volume de dados muito grandes que estão sendo transmitidos em uma rede não confiável, utiliza-se o conceito de ponto de sincronização. O ponto de sincronização corresponde a marcas lógicas posicionadas ao longo

do diálogo. Toda vez que um usuário recebe um ponto de sincronização, deve enviar uma resposta, confirmando que este foi recebido. Caso a transmissão, por algum motivo, seja interrompida, ela pode ser reiniciada a partir do último ponto de sincronização confirmado;

- gerenciamento da camada de sessão.

1.9. Camada de Apresentação

A camada de apresentação, ao contrário das camadas inferiores, já não se preocupa com os dados em nível de bits, mas sim com a sua sintaxe, ou seja, sua representação. Nela é definida a sintaxe abstrata, ou seja, a forma como os tipos e os valores dos dados são definidos, independentemente do sistema computacional utilizado e a sintaxe de transferência, ou seja, a maneira como é realizada esta codificação. Por exemplo, através da sintaxe abstrata define-se que um caractere A deve ser transmitido. A sintaxe de transferência especifica, então, como este dado será codificado em ASCII ou EBCDIC ao ser entregue à camada de sessão.

Outras funções que a camada de apresentação pode executar são a criptografia e compressão de dados.

1.10. Camada de Aplicação

Basicamente, as funções da camada de aplicação são aquelas necessárias à adaptação dos processos de aplicação ao ambiente de comunicação. A camada de aplicação é estruturada modularmente para permitir a flexibilidade das funções e de forma, para se determinar os requisitos de comunicação de cada aplicação distribuída. A camada de aplicação deve seguir o descrito na norma ISO 9545. Ela é formada por várias ASE's (Elemento de Serviço de Aplicação), que são os componentes básicos das AE's (Entidade de Aplicação). Uma AE é a função que um processo de aplicação utiliza para se comunicar com os seus pares. Um processo de aplicação pode utilizar diversas AE's, cada uma das quais provendo um conjunto de definições de cada uma das funções e das regras que governam o uso destas funções.

O componente básico de uma AE é um Elemento de Serviço de Aplicação (ASE). Um ASE é um elemento que define uma função ou um conjunto de funções que ajudam na realização da aplicação. Desta forma, pode-se imaginar que um AE é um grande programa feito de muitos sub-programas ou procedimentos, que são os ASE.

As principais funções da camada de aplicação são:

- Seleção do modo de diálogo (full duplex ou half duplex);
- Determinação da qualidade de serviço aceitável na conexão: retardo de transferência, taxa de erro tolerável, etc;
- Identificação dos futuros parceiros na comunicação: por nome ou endereço;
- Especificação de aspectos relativos à segurança: controle de acesso, integridade de dados, etc.

1.10.1. Elementos de Serviço de Aplicação de Propósito Geral

Aqui serão apresentados os elementos de serviço de aplicação que exercem funções básicas, podendo ser utilizados pela maioria das aplicações.

1.10.1.1. ACSE (Association Control Service Element)

O elemento de serviço de aplicação ACSE é o responsável pelo gerenciamento das associações de aplicação. Presta os serviços básicos de estabelecimento e liberação (ordenada e abrupta) de associações de aplicação. Desta forma, está presente em todas as entidades de aplicação.

1.10.1.2. ROSE (Remote Operations Service Element)

O ROSE é um elemento de serviço de aplicação que oferece suporte a aplicações interativas, sendo bastante útil em aplicações distribuídas. O ROSE não pode ser utilizado isoladamente. Em geral, ele é utilizado em aplicações envolvendo MHS (Message Handling System), DS (Directory Service) e gerenciamento OSI. No caso particular do gerenciamento, o ROSE é usado, por exemplo, para transportar operações de gerenciamento (como é o caso da leitura de atributos de um objeto gerenciado) do gerente para o agente e para transportar o resultado da operação realizada (valor dos atributos lidos) do agente para gerente.

1.10.1.3. RTSE (Reliable Transfer Service Element)

Como vimos anteriormente, uma das funções da camada de sessão é garantir que após uma interrupção em uma conexão estabelecida, a transferência dos dados possa se restabelecer a partir do ponto em que houve a interrupção, havendo uma perda mínima. O RTSE deve executar esta mesma função através de uma interface simples e ser capaz, então, de transferir as unidades de dados de uma forma confiável entre entidades de aplicação.

1.10.1.4. CMISE (Common Management Information Service Element)

Cada recurso que está sujeito ao gerenciamento de sistemas é representado por um objeto gerenciado. Os objetos gerenciados podem ser criados, suprimidos e ter os valores dos seus atributos modificados. O gerenciamento de objetos descreve o serviço para relatórios de criação e supressão de objetos gerenciados, modificação de nomes de objetos gerenciados, modificação de atributos dos objetos gerenciados. Estas aplicações de gerência de rede são realizadas através do CMISE.

A descrição do serviço CMISE é detalhada na X.710 e na ISO 9595-1 versão 2. O Protocolo de Informação de Gerenciamento Comum (CMIP) é o protocolo utilizado para a troca de unidades de dados de protocolo de informação de gerenciamento comum entre o agente e o gerente. O CMIP especifica os elementos de protocolo que podem ser usados para prover as Operações e Notificações definidas na X.710, a qual define o Serviço de Gerenciamento Comum (CMIS). A X.711 define a sintaxe abstrata do CMIP.

1.10.1.5. TP (Transaction Processing)

O TP tem como objetivo prover a infra-estrutura necessária ao suporte do processamento de transações distribuídas envolvendo vários sistemas abertos. Uma transação distribuída deve ser caracterizada por:

- atomicidade: uma unidade de trabalho deve sempre ser executada de maneira completa, ou não deve ser executada;
- consistência: uma unidade de trabalho deve ser executada de uma maneira precisa, correta e válida em relação à semântica da aplicação;
- isolamento: os resultados intermediários de uma unidade de trabalho não devem tornar-se disponíveis a não ser para as operações que fazem parte de uma unidade de trabalho;
- durabilidade: os efeitos da realização de uma unidade de trabalho não podem ser alterados por nenhuma falha que venha a ocorrer posteriormente.

É responsabilidade do TP prover facilidades e mecanismos de forma a garantir que estes objetivos sejam alcançados.

1.10.1.6. CCR (Commitment, Concurrency and Recovery)

Para os casos em que o processamento se dá de forma distribuída, as facilidades são providas pelo TP. Já no caso do processamento ocorrer entre duas aplicações residentes em sistemas abertos distintos, ela deve ser coordenada de forma que haja a garantia de que a execução de unidades de trabalho ocorra somente quando tais aplicações estejam prontas e aptas a fazê-lo.

1.10.2. Elementos de Serviço de Aplicação de Propósito Específico

A função dos elementos de serviço de aplicação é prover suporte a aplicações de uma maneira genérica, porém algumas aplicações como a transferência de arquivos e o correio eletrônico se tornaram tão comuns que foram criados elementos de serviço específicos para estes casos.

1.10.2.1. MHS (Message Handling System)

O MHS é o serviço de correio eletrônico. Este tipo de serviço surgiu há vários anos, na década de setenta, porém funcionava de maneira proprietária, ou seja, sistemas de fabricantes diferentes não conseguiam trocar mensagens entre si. Na década de oitenta, o CCITT em conjunto com a ISO resolveram padronizar este tipo de serviço.

A operação do MHS se baseia no sistema store and forward, ou seja, não há nenhuma conexão entre o originador da mensagem e o seu destinatário durante o momento em que esta é enviada. As facilidades que o MHS oferece incluem:

- nomeação e endereçamento;
- entrega de mensagens a múltiplos destinatários;
- destinatário alternativo;
- tipo de serviço (prioridade);
- teste;
- notificação;
- conversão.

A versão de 88 do MHS contém, ainda, facilidades adicionais relativas a segurança, listas de distribuição e redirecionamento de mensagens.

1.10.2.2. FTAM (File Transfer Access and Management)

O FTAM tem por objetivo permitir a consulta, acesso, transferência e gerenciamento de arquivos em um ambiente aberto. É muito útil, por exemplo, no caso de transferência de arquivos de tarifação, que possuem um volume muito grande. As operações que podem ser executadas sobre arquivos podem ser classificadas em três tipos:

- transferência: operação de movimentação entre dois sistemas abertos, de parte ou de todos os dados contidos em um arquivo;

- acesso: operação em que o usuário pode ler, substituir ou remover os dados contidos em um arquivo;
- gerenciamento: operações de criação, remoção, inspeção e manipulação de atributos de um arquivo.

Os serviços prestados pelo protocolo FTAM deve servir de base para o desenvolvimento de sistemas que envolvam a manipulação de arquivos em vários níveis de complexidade. Por exemplo, a transferência parcial ou completa de arquivos entre sistemas interconectados através de uma rede local ou de longa distância; o acesso a um sistema de arquivos compartilhado entre múltiplas estações interconectadas através de uma rede; a transferência de arquivos para fins de impressão, etc.

1.10.2.3. DS (Directory Service)

O Serviço Diretório consiste em um repositório de informações mantido por um conjunto de sistemas abertos distribuídos. Estes sistemas possuem acesso a estas informações e podem, por exemplo, ler e modificar estas informações.