1.Protocolos de rede

Os protocolos de rede definem um formato comum e um conjunto de regras para a troca de mensagens entre dispositivos. Eles são implementados por dispositivos finais e dispositivos intermediários em software, hardware, ou ambos. Cada protocolo de rede tem sua própria função, formato e regras para comunicações.

1.1 Protocolos de comunicação em rede

Estes protocolos permitem que dois ou mais dispositivos se comuniquem através de um ou mais redes.

1.2 Protocolos de segurança de rede

Estes protocolos protegem os dados para fornecer autenticação, integridade dos dados e criptografia de dados.

1.3 Protocolos de roteamento

Estes protocolos permitem que os roteadores troquem informações de rota, compare caminho, e em seguida, selecione o melhor caminho para o destino remoto.

1.4 Protocolos de descoberta de serviço

Estes protocolos são usados para a detecção automática de dispositivos ou serviços.

Os protocolos de comunicação de redes são responsáveis por uma variedade de funções necessárias para comunicações de rede entre dispositivos finais.

- **Endereçamento:** Identifica o remetente e o destinatário pretendido da mensagem usando um esquema de endereçamento definido.
- **Confiabilidade:** Fornece mecanismos de entrega garantidos em caso de mensagens serem perdidas ou corrompidas em trânsito.
- **Controle de fluxo:** Esta função garante que os fluxos de dados a uma taxa eficiente entre dois dispositivos de comunicação.
- **Sequenciamento:** Essa função realiza um rotulamento exclusivo para cada segmento de dados transmitido a partir das informações de sequenciamento para remontar as informações corretamente. Isso é útil se os segmentos de dados forem perdidos, atrasados ou recebidos fora de ordem
- **Detecção de erros:** Esta função é usada para determinar se os dados forem corrompidos durante a transmissão.
- **Interface de aplicação:** Esta função contém informações usadas de processo a processo de comunicação entre aplicações de rede.

Os protocolos devem ser capazes de trabalhar com outros protocolos para entregar uma boa experiência na comunicação de rede. **Os conjuntos de protocolos são projetados para trabalhar entre si**. Um conjunto de protocolos é um grupo inter-relacionados necessários para executar uma função de comunicação. Uma das melhores maneiras de visualizar como os protocolos interagem é ver a interação como uma pilha. Esta pilha mostra como os protocolos individuais são implementados.

Os protocolos são visualizados em termos de camadas, com cada serviço de nível superior, dependendo da funcionalidade definida pelos protocolos mostrados nos níveis inferiores. As camadas inferiores estão relacionadas com a movimentação de dados pela rede e o fornecimento de serviço às camadas superiores, que se concentram no conteúdo da mensagem que está sendo enviada.

2.Conjunto de protocolos padrão aberto

Isso significa que está disponível gratuitamente ao público e pode ser usado por qualquer fornecedor em seu hardware ou software. Os padrões abertos incentivam a interoperabilidade, a concorrência e a inovação, além de garantir que o produto de nenhuma empresa possa monopolizar o mercado ou ter uma vantagem sobre a sua concorrência.

3.Conjunto de protocolos com base em padrões

Isso significa que foi endossado pela indústria de rede e aprovado por uma organização de padrões. Isso garante que produtos de diferentes fabricantes possam interromper com êxito. As organizações padronizadoras geralmente são organizações sem fins lucrativos independentes de fornecedores estabelecidos para desenvolver e promover o conceito de padrões abertos. Elas são importantes na manutenção de uma Internet aberta, com especificações e protocolos acessíveis gratuitamente, que podem ser implementados por qualquer fornecedor.

4.Hierarquia de protocolos

Para reduzir a complexidade do projeto, a maioria das redes é organizada como uma pilha de camadas ou níveis, colocadas umas sobre as outras. O número de camadas, o nome, o conteúdo e a função de cada camada diferem de uma rede para outra. No entanto, em todas as redes o objetivo de cada camada é oferecer determinados serviços às camadas superiores, isolando essas camadas dos detalhes de implementação desses recursos. A ideia fundamental é que um determinado item de software (ou hardware) fornece um serviço a seus usuários, mas mantém ocultos os detalhes de seu estado interno e de seus algoritmos.

A camada N de uma máquina se comunica com a camada N de outra máquina. Coletivamente, as regras e convenções usadas neste diálogo são conhecidas como protocolo da camada N. Um protocolo é um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação. As entidades que ocupam as

camadas correspondentes em diferentes máquinas são chamadas de pares (***peers***). Os pares podem ser processos, dispositivos de hardware ou até mesmo humanos. Essa comunicação se dá quando cada camada transfere os dados e as informações de controle para a camada imediatamente abaixo dela, até ser alcançada a camada mais baixa.

Um conjunto de camadas e protocolos é chamado de arquitetura de rede. A especificação de uma arquitetura deve conter informações suficientes para permitir que um implementador desenvolva o programa ou construa o hardware de cada camada, de forma que ela obedeça corretamente ao protocolo adequado. Uma lista de protocolos usados por um determinado sistema, um protocolo por camada, é chamada de pilha de protocolos. A abstração de processos pares (***peers***) é fundamental para toda a estrutura da rede.

5.Questões de projeto relacionadas às camadas

Todas as camadas precisam de um mecanismo para identificar os transmissores e os receptores. Como em geral uma rede tem muitos computadores, e alguns deles tem vários processos, é necessário um meio para que um processo de uma máquina especifique com quem ela deseja comunicar. Outra preocupação que se deve ter em relação ao conjunto de decisões de projeto diz respeito à transferência de dados. Em alguns sistemas, os dados são transferidos em apenas um sentido; em outros, os dados trafegam em ambos os sentidos.

O controle de erros é uma questão importante, pois os circuitos de comunicação física não são perfeitos. Muitos códigos de detecção e correção de erros são conhecidos, mas as partes envolvidas na conexão devem chegar a um consenso quanto ao que está sendo usado. Nem todos os canais de comunicação preservam a ordem das mensagens enviadas a eles. Para lidar com uma possível perda de sequência, o protocolo deve permitir explicitamente ao receptor remontar de forma adequada os fragmentos recebidos. Esse assunto é chamado de controle de fluxo.

Quando for inconveniente ou dispendido o configurar de uma conexão isolada para cada par de processos de comunicação, a camada subjacente pode decidir usar a mesma conexão para diversas conversações não relacionadas entre si. Quando houver vários caminhos entre a origem e o destino, uma rota deverá ser escolhida. Algumas vezes, essa decisão deve ser compartilhada por duas ou mais camadas.

6.Serviços orientados a conexões

Este serviço se baseia no sistema telefônico. Para falar com alguém, você tira o fone do gancho, disca o número, fala e em seguida, desliga. Da mesma forma, para utilizar um serviço de rede orientado a conexões, primeiro o usuário do serviço estabelece uma conexão, utiliza a conexão e depois libera a conexão. Essencialmente, ela funciona como um tubo onde os bits são empurrados para outra extremidade.

Quando uma conexão é estabelecida, o transmissor, o receptor e a sub-rede conduzem uma negociação sobre os parâmetros a serem usados, como o tamanho máximo das mensagens, a qualidade do serviço exigida e outras questões. O serviço orientado à conexão confiável tem duas pequenas variações secundárias:

6.1 Sequência de mensagem

Os limites das mensagens são preservados. Quando duas mensagens de 1024 bytes são enviadas, elas chegam como duas mensagens distintas de 1024 bytes. Uma dessas aplicações é o tráfego de voz digital. Os usuários de telefone preferem ouvir um pouco de ruído da linha ou uma palavra truncada de vez em quando a experimentar um terado para aguardar confirmações.

6.2 Fluxo de bytes

Neste caso, os limites das mensagens não são preservados. Geralmente, acontece quando o usuário se conecta em um servidor remoto.

7.Serviços sem conexões

Este serviço se baseia no sistema postal. Cada mensagem carrega o endereço de destino completo e cada uma delas é roteada através do sistema, independente de todas as outras. Em geral, quando duas mensagens são enviadas ao mesmo destino, a primeira a ser enviada é a primeira a chegar.

É necessário apenas um modo de enviar uma única mensagem que tenha uma alta probabilidade de chegar, mas nenhuma garantia. O serviço sem conexão não confiável costuma ser chamado de serviço de datagramas, em uma analogia com o serviço de telegramas. Outro serviço é o serviço de solicitação/resposta. Nele, o transmissor envia um único datagrama contendo uma solicitação; a resposta contém a réplica.

8.Primitivas de serviço

Um serviço é especificado formalmente por um conjunto de primitivas (operações) disponíveis para que um processo do usuário acesse o serviço. Essas primitivas informam ao serviço que ele deve executar alguma ação ou relatar uma ação executada por uma entidade par. O conjunto de primitivas disponíveis depende da natureza do serviço que está sendo fornecido. As primitivas para um serviço orientado a conexões são diferentes das que são oferecidas em um serviço sem conexões.

Primitiva	Significado
LISTEN	Bloco que espera uma conexão de entrada
CONNECT	Estabelecer uma conexão com um par que está à espera
RECEIVE	Bloco que espera por uma mensagem de entrada

SEND	Envia uma mensagem ao par
DISCONNECT	Encerrar uma conexão

Essas primitivas poderiam ser usadas como a seguir. Primeiro, o servidor executa ***LISTEN*** para indicar que está preparado para aceitar conexões de entrada. Em seguida, o processo cliente executa ***CONNECT*** para estabelecer uma conexão com o servidor. A chamada de ***CONNECT*** precisa especificar a quem se conectar; assim, ela poderia ter um parâmetro fornecendo o endereço do servidor. Em geral, o sistema operacional envia então um pacote ao par solicitando que ele se conecte. O processo cliente é suspenso até haver uma resposta. Quando o pacote chega ao servidor, ele é processado pelo SO do servidor.

Quando o sistema observa que o pacote está solicitando uma conexão, ele verifica se existe um ouvinte. Nesse caso, ele realiza duas ações: desbloqueia o ouvinte e envia de volta uma confirmação. A chegada dessa confirmação libera o cliente. Nesse momento, o cliente e o servidor estão em execução e têm uma conexão estabelecida entre eles. É importante notar que a confirmação é gerada pelo próprio código do protocolo, e não em resposta a uma primitiva no nível do usuário. Se chegar uma solicitação de conexão e não houver nenhum ouvinte, o resultado será indefinido.

A próxima etapa é a execução de ***RECIEVE*** pelo servidor, a fim de se preparar para aceitar a primeira solicitação. Normalmente, o servidor faz isso imediatamente após ser liberado de ***LISTEN***, antes da confirmação poder retornar ao cliente. A chamada de ***RECIEVE*** bloqueia o servidor. Depois, o cliente executa ***SEND*** para transmitir sua solicitação, seguida pela execução de ***RECIEVE*** para receber a resposta. A chegada do pacote de solicitação à máquina servidora desbloqueia o processo servidor, para que ele possa processar a solicitação. Depois de terminar o trabalho, ele utiliza ***SEND*** para enviar a resposta ao cliente. A chegada desse pacote desbloqueia o cliente, que pode agora examinar a resposta. Se tiver solicitações adicionais, o cliente poderá fazê-las nesse momento.

Ao terminar, ele utilizará ***DISCONNECT*** para encerrar a conexão. Em geral, uma ***DISCONNECT*** inicial é uma chamada de bloqueio, suspendendo o cliente e enviando um pacote ao servidor para informar que a conexão não é mais necessária. Quando o servidor recebe o pacote, ele também emite uma ***DISCONNECT***, confirmando o pacote do cliente e liberando a conexão.

A resposta é que, em um mundo perfeito, esse tipo de protocolo poderia ser usado e, nesse caso, seriam necessários apenas dois pacotes. Entretanto, em face de mensagens externas, erros de transmissão e perda de pacotes, a situação se altera.

9.O relacionamento entre serviços e protocolos

O serviço define as operações em que a camada está preparada para executar em nome de seus usuários, mas não informa absolutamente nada sobre como essas operações são implementadas.

Já o protocolo é um conjunto de regras que controla o formato e o significado dos pacotes ou mensagens que são trocadas pelas entidades pares contidas em uma camada. As entidades utilizam protocolos com a finalidade de implementar suas definições de serviço. Em protocolos mais antigos, não havia muita distinção entre o serviço e o protocolo.

Na prática, uma camada normal poderia ter uma primitiva de serviço ***SENDPACKET***, com o usuário fornecendo um ponteiro para um pacote totalmente montado. Hoje, a maioria dos projetistas de redes considera tal projeto um sério equívoco.