

Redes de computadores I

Aula 6 – Software de rede

Fontes: Andrew S. Tanenbaum; Kurose;

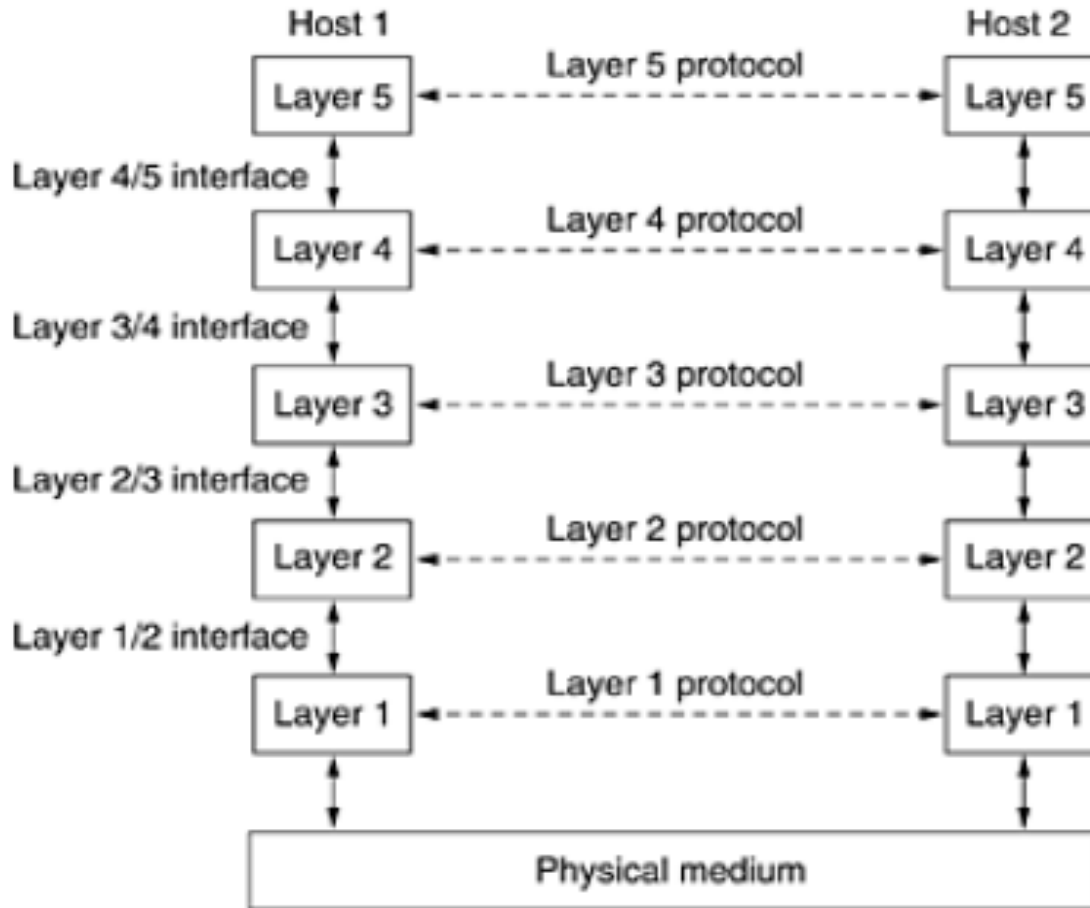
Software de Redes

- No projeto das primeiras redes de computadores, o hardware foi a principal preocupação e o software ficou em segundo plano.
- Atualmente, o software de rede é altamente estruturado.

Hierarquias 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. A quantidade de camadas, o nome, o conteúdo e a função difere de uma rede para outra.
- O **OBJETIVO** de cada camada é oferecer determinados serviços as camadas superiores, isolando essas camadas dos detalhes de implementação desses recursos.

Hierarquias de Protocolos



Hierarquias de Protocolos

- A camada **n** de uma máquina se comunica com a camada **n** de outra máquina.
- Coletivamente, as regras e convenções usadas nesse dialogo são conhecidas como o protocolo da camada **n**.
- Protocolo é um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação.

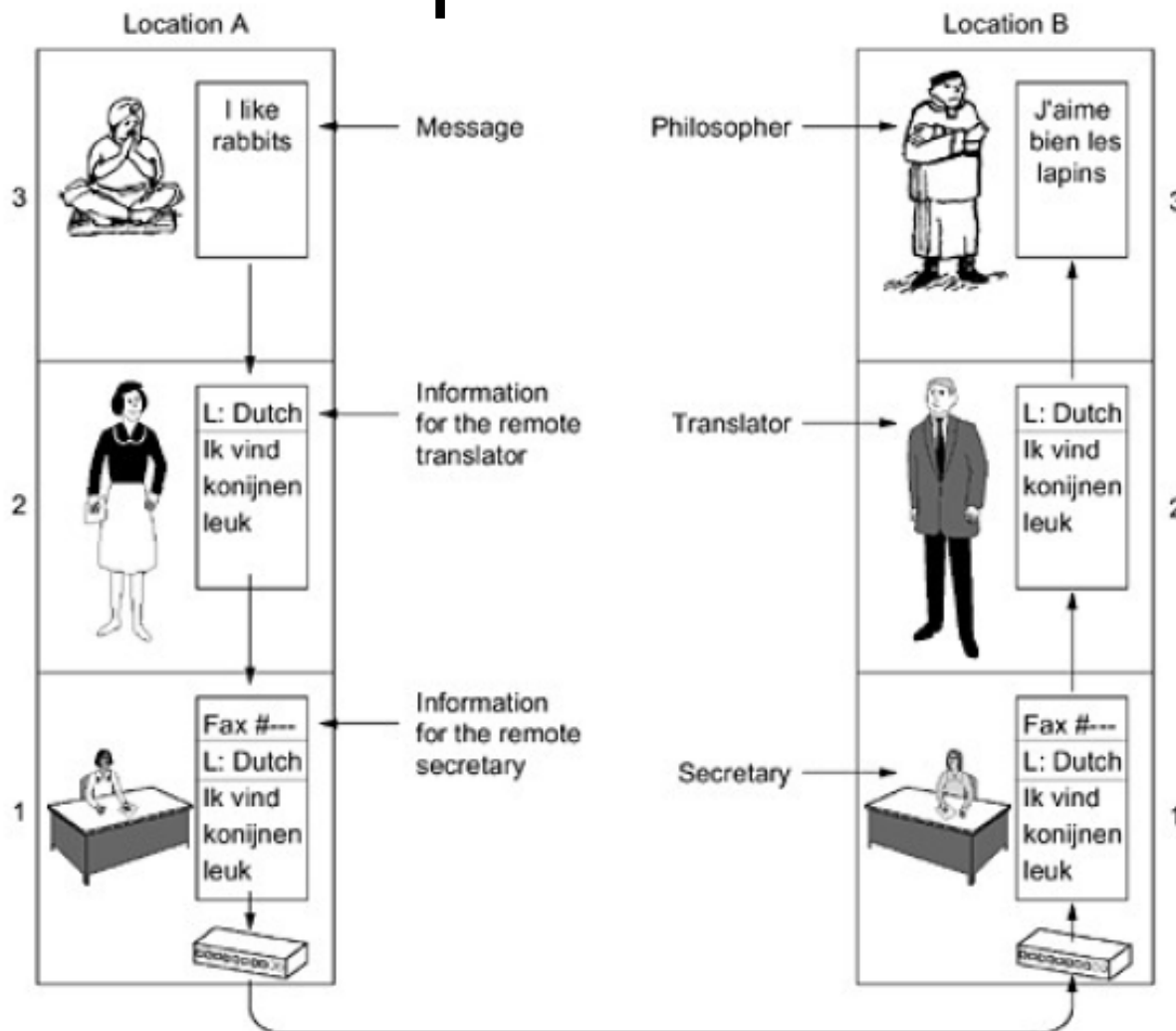
Hierarquias de Protocolos

- Na realidade, os dados não são transferidos diretamente da camada n de uma máquina para a camada n de outra máquina.
- Cada camada transfere os dados e as informações de controle para a camada imediatamente abaixo dela, até ser alcançada a camada mais baixa.
- Abaixo da camada I encontra-se o **meio físico** através do qual se dá a comunicação propriamente dita. Na Figura anterior, a comunicação virtual é mostrada por linhas pontilhadas e a comunicação física por linhas contínuas.

Hierarquias de Protocolos

- Entre cada par de camadas adjacentes existe uma interface. A interface define as operações e os serviços que a camada inferior tem a oferecer a camada que se encontra acima dela.
- Um conjunto de camadas e protocolos é chamado **arquitetura de rede**.
- Não é necessário que as interfaces de todas as máquinas de uma rede sejam iguais, desde que cada uma delas possa usar todos os protocolos de maneira correta. Uma lista de protocolos usados por um determinado sistema, um protocolo por camada, é chamada **pilha de protocolos**.

Hierarquias de Protocolos



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, alguns deles tem vários processos, é necessário um meio para que um processo de uma máquina especifique com quem ela deseja se comunicar.
- Através do endereçamento definimos um destino.

Questões de Projeto relacionadas às camadas

- Outra preocupação diz respeito a transferência de dados.
 - Em alguns sistemas, os dados são transferidos em apenas um sentido; em outros, os dados trafegam em ambos os sentidos.
- O protocolo também deve definir a quantos canais lógicos corresponde a conexão e quais são suas prioridades.
 - Muitas redes fornecem pelo menos dois canais lógicos por conexão, um para dados normais e um para dados urgentes.

Questões de Projeto relacionadas às camadas

- Outra questão importante é o controle de erros.
- 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.
- O receptor deve ter algum meio para informar ao transmissor quais mensagens foram recebidas corretamente e quais não foram.

Questões de Projeto relacionadas às camadas

- Nem todos os canais de comunicação preservam a ordem das mensagens enviadas a eles.
- O protocolo deve permitir explicitamente ao receptor remontar de forma adequada os fragmentos recebidos.
 - Uma solução óbvia é numerar os fragmentos, mas essa solução ainda deixa aberta a questão do que deve ser feito com os fragmentos que chegarem fora de ordem.

Questões de Projeto relacionadas às camadas

- Uma questão que afeta cada nível é como impedir que um transmissor rápido envie uma quantidade excessiva de dados a um receptor mais lento.
- Várias soluções foram propostas e serão discutidas a seguir. Esse questão é conhecida como **controle de fluxo**.

Serviços Orientados a Conexões e sem conexões

- As camadas podem oferecer dois tipos diferentes de serviços as camadas situadas acima delas: serviços orientados a conexões e serviços sem conexões.
- O serviço orientado a conexões se baseia no sistema telefônico. Para falar com alguém, você tira o fone do gancho, discar 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.

Serviços Orientados a Conexões

- O aspecto essencial de uma conexão é que ela funciona como um **tubo**.
- Em alguns casos, 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;
 - outras questões.

Serviços sem conexões

- Por outro lado, o serviço sem conexão se baseia no sistema postal.
- Cada mensagem (carta) carrega o endereço de destino completo e cada uma delas é roteada (encaminhada) através do sistema, independentemente de todas as outras.
- Em geral, quando duas mensagens são enviadas ao mesmo destino, a primeira a ser enviada é a primeira a chegar.

Serviços Orientados a Conexões e sem conexões

- Cada serviço pode ser caracterizado por uma qualidade de serviço. Alguns serviços são confiáveis, no sentido de nunca perderem dados.
- Em geral, um serviço confiável é implementado para que o receptor confirme o recebimento de cada mensagem, de modo que o transmissor se certifique de que ela chegou.
- Uma situação típica em que um serviço orientado a conexões confiável é apropriado é a transferência de arquivos.

Serviços Orientados a Conexões e sem conexões

- Um exemplo em que não se tem a necessidade de estabelecer uma conexão para enviar uma única mensagem curta com a necessidade de uma confiabilidade é o serviço de **datagramas com confirmação**;
- Funciona semelhante a enviar uma carta registrada e receber um aviso de recebimento. Quando recebe o aviso, o transmissor fica certo de que a mensagem foi entregue ao destinatário.

Serviços Orientados a Conexões e sem conexões

Orientados a conexões	Serviço	Exemplo
	Fluxo de mensagens confiável	Sequência de páginas
	Fluxo de bytes confiável	Logon remoto
	Conexão não confiável	Voz digitalizada
Sem conexões	Datagrama não confiável	Lixo de correio eletrônico
	Datagrama confirmado	Correspondência registrada
	Solicitação/resposta	Consulta a banco de dados

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.
- Se a pilha de protocolos estiver localizada no sistema operacional, como ocorre com frequência, as primitivas serão normalmente chamadas do sistema.

Primitivas de Serviço

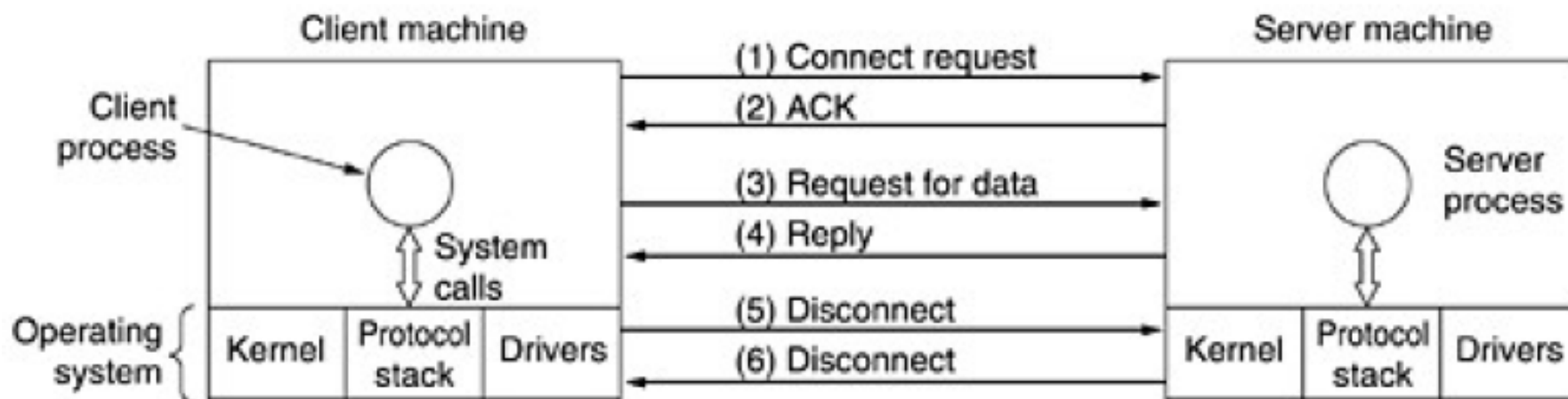
- As primitivas para um serviço orientado a conexões são diferentes das que são oferecidas em um serviço sem conexões. Como um exemplo mínimo das primitivas de serviço que poderiam ser fornecidas para implementar um fluxo de bytes confiável em um sistema cliente/servidor.

Primitiva	Significado
LISTEN	Bloco que espera por 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	Enviar uma mensagem ao par
DISCONNECT	Encerrar uma conexão

Primitivas de Serviço

- Primeiro, o servidor executa `LISTEN` para indicar que está preparado para aceitar conexões de entrada. Depois de executar a primitiva, o processo servidor fica bloqueado até surgir uma solicitação de conexão.
- 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.

Primitivas de Serviço



Primitivas de Serviço

- Em geral, o sistema operacional envia então um pacote ao par solicitando que ele se conecte. 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 (2). A chegada dessa confirmação libera o cliente. Nesse momento, o cliente e o servidor estão em execução e tem uma conexão estabelecida entre eles.

Primitivas de Serviço

- A próxima etapa é a execução de **RECEIVE** 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 **RECEIVE** bloqueia o servidor.
- Depois, o cliente executa **SEND** para transmitir sua solicitação (3), seguida pela execução de **RECEIVE** para receber a resposta.
- A chegada do pacote de solicitação a 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 (4). A chegada desse pacote desbloqueia o cliente, que pode agora examinar a resposta.