ACH 2147 — Desenvolvimento de Sistemas de Informação Distribuídos

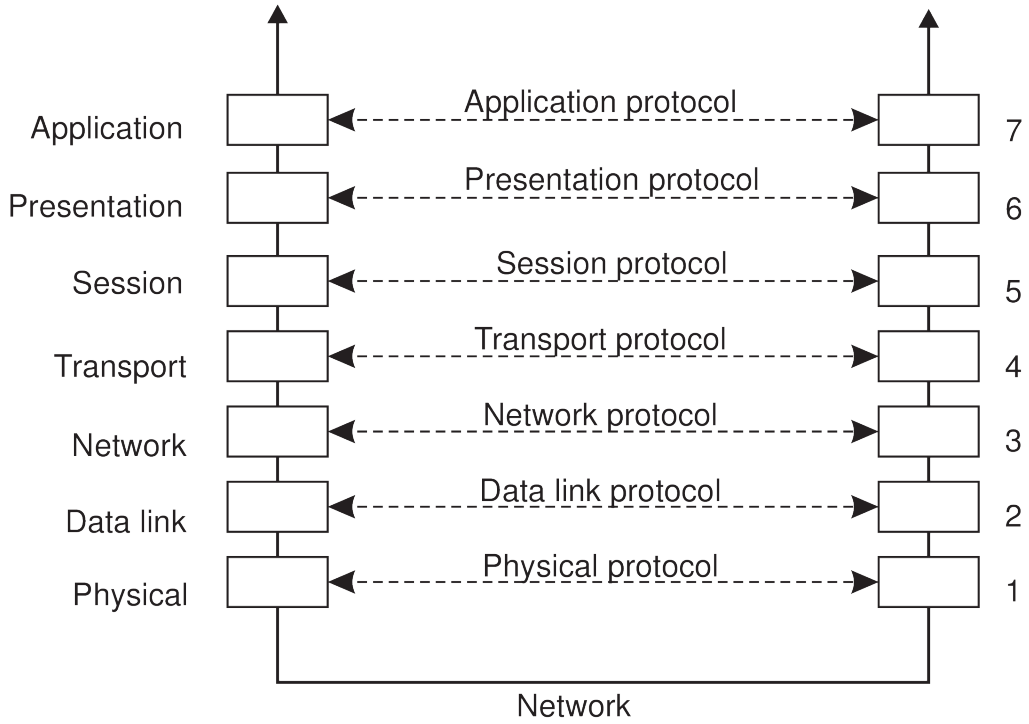
Aula 09: Comunicação (parte 1)

Prof. Renan Alves

Escola de Artes, Ciências e Humanidades — EACH — USP

01/04/2024

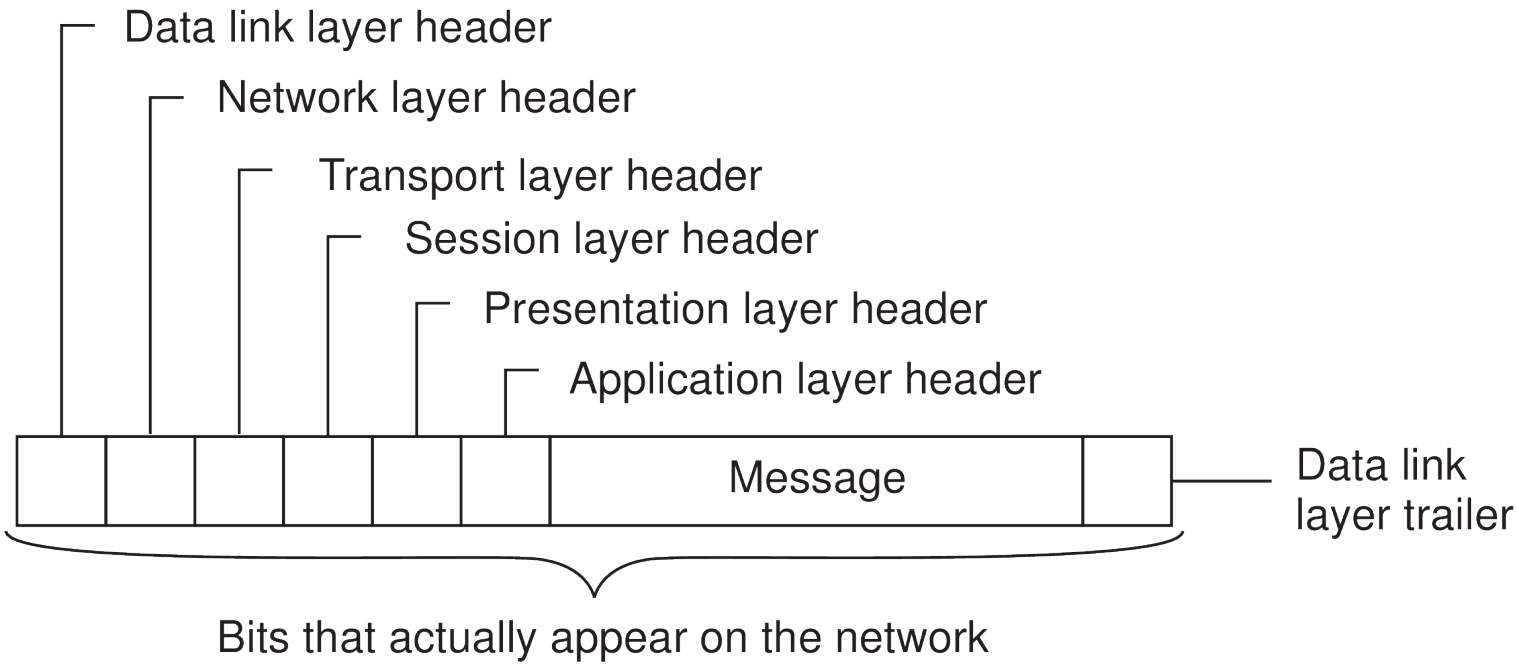
Modelo básico de rede



Desvantagens

- Foco apenas na troca de mensagens
- Funcionalidades frequentemente desnecessárias ou indesejadas
- Viola a transparência de acesso

Empilhamento de cabeçalhos



Camadas de baixo nível

Recapitulação

- Camada física: contém a especificação e implementação de bits, e sua transmissão entre remetente e receptor
- Camada de enlace: descreve a transmissão de uma sequência de bits em um quadro (frame), permite detecção de erros e controle de fluxo
- Camada de rede: descreve como os pacotes devem ser roteados.

Observação

Para alguns sistemas distribuídos, a interface de mais baixo nível é a da camada de rede.

Camada de Transporte

Importante

A camada de transporte fornece os recursos de comunicação para a maioria dos sistemas distribuídos.

Protocolos de Internet padrão

- TCP: comunicação orientada à conexão, confiável, orientada a fluxo
- UDP: comunicação de datagrama não confiável (melhor esforço)

Camada de middleware

Observação

O middleware foi inventado para fornecer serviços e protocolos comuns que podem ser usados por muitas aplicações diferentes

- (Des)empacotamento de dados, necessário para sistemas integrados
- Protocolos de nomeação, para permitir o compartilhamento fácil de recursos
- Protocolos de segurança para comunicação segura
- Mecanismos de escalabilidade, como replicação e armazenamento em cache

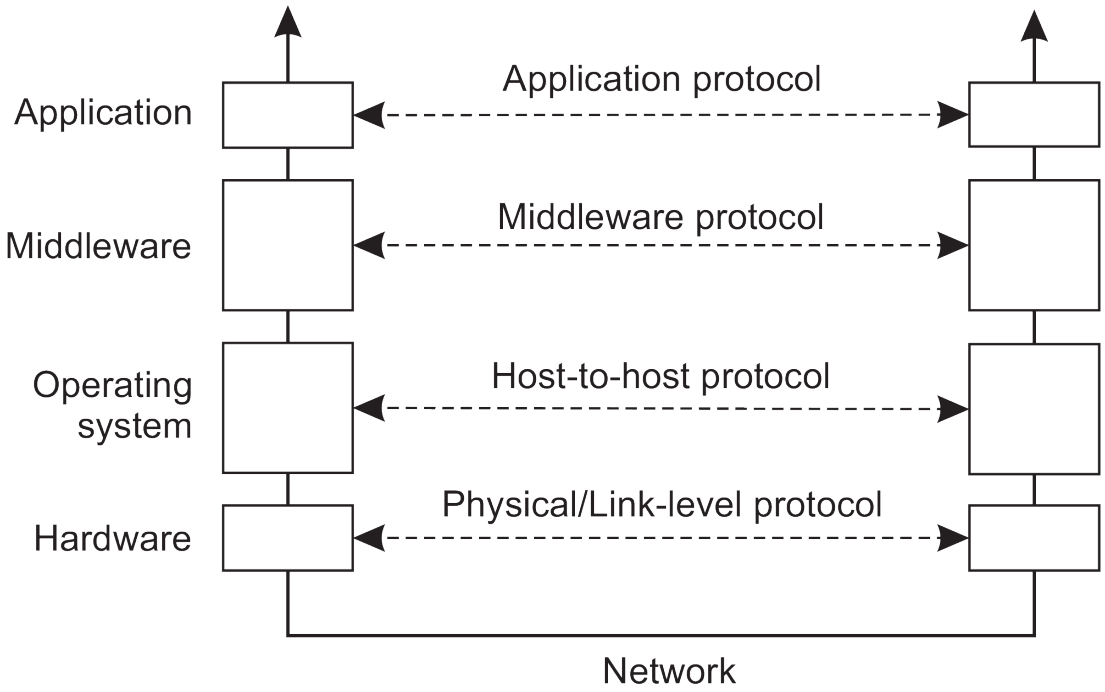
Exemplos

DNS, protocolos de autenticação, protocolos de lock distribuídos, RPC

Nota

O que resta são protocolos de aplicações específicas... tais como?

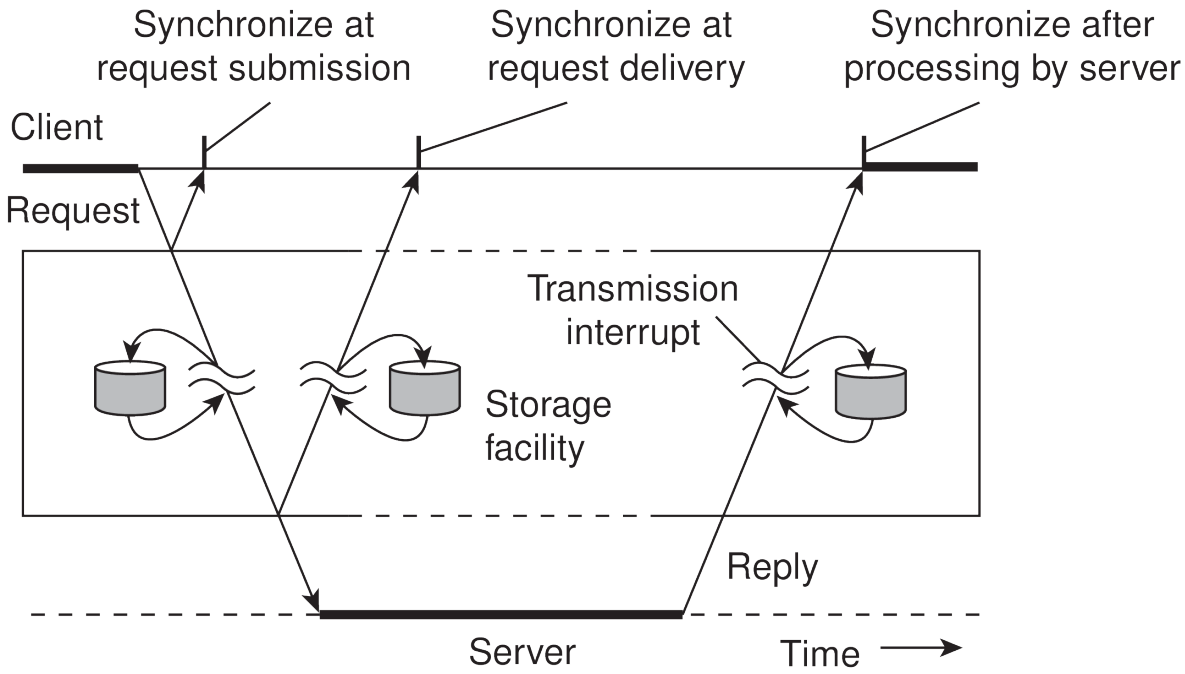
Um esquema de camadas adaptado



Tipos de comunicação

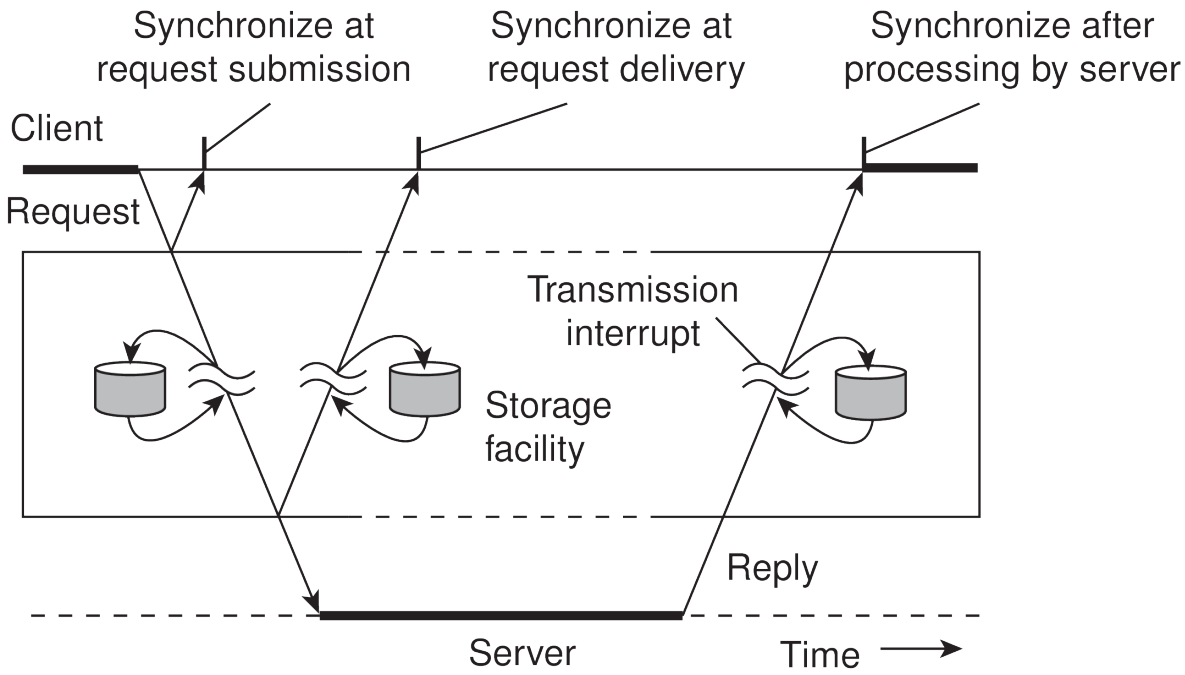
Distinguir...

- Comunicação transitória versus persistente
- Comunicação assíncrona versus síncrona



Tipos de comunicação

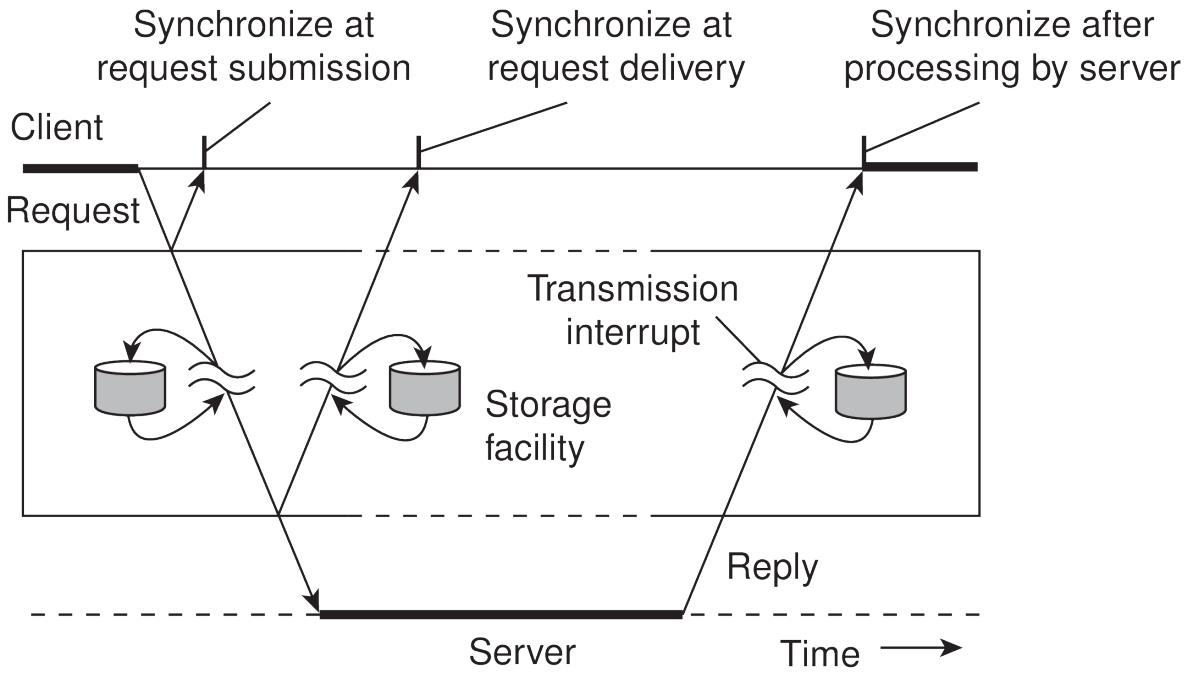
Transitória versus persistente



- Comunicação transitória: O servidor de comunicação descarta a mensagem quando não pode ser entregue no servidor ou no receptor.
- Comunicação persistente: Uma mensagem é armazenada em um servidor de comunicação até que seja entregue.

Tipos de comunicação

Pontos de sincronização



- Na submissão da requisição
- Na entrega da requisição
- Após o processamento da requisição

Cliente/Servidor

Algumas observações

A computação cliente/servidor é geralmente baseada em um modelo de comunicação síncrona transitória:

- Cliente e servidor precisam estar ativos no momento da comunicação
- O cliente emite a solicitação e bloqueia até receber a resposta
- O servidor essencialmente espera apenas por solicitações recebidas e, posteriormente, as processa

Cliente/Servidor

Algumas observações

A computação cliente/servidor é geralmente baseada em um modelo de comunicação síncrona transitória:

- Cliente e servidor precisam estar ativos no momento da comunicação
- O cliente emite a solicitação e bloqueia até receber a resposta
- O servidor essencialmente espera apenas por solicitações recebidas e, posteriormente, as processa

Desvantagens da comunicação síncrona

- O cliente não pode fazer nenhuma outra tarefa enquanto aguarda a resposta
- As falhas devem ser tratadas imediatamente: o cliente está esperando
- O modelo pode simplesmente não ser apropriado (e-mail, notícias)

Mensagens

Middleware orientado a mensagens

Visa uma comunicação assíncrona persistente em alto nível:

- Os processos enviam mensagens uns aos outros, que são enfileiradas
- O remetente não precisa aguardar uma resposta imediata, mas pode fazer outras coisas
- O middleware muitas vezes garante tolerância a falhas

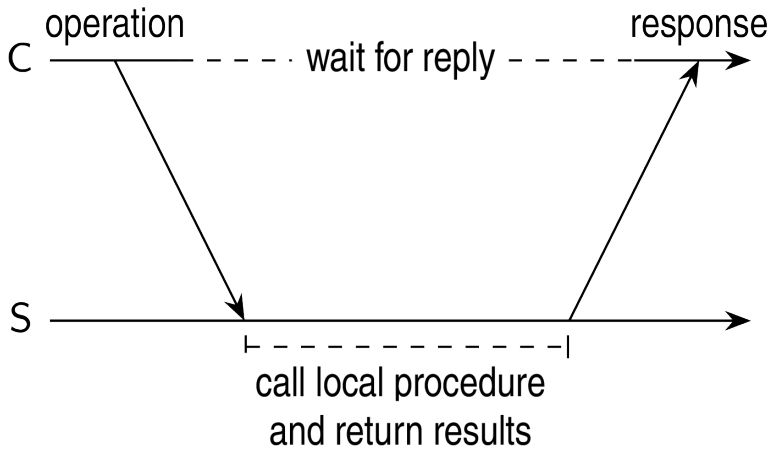
Funcionamento básico de RPC

Observações

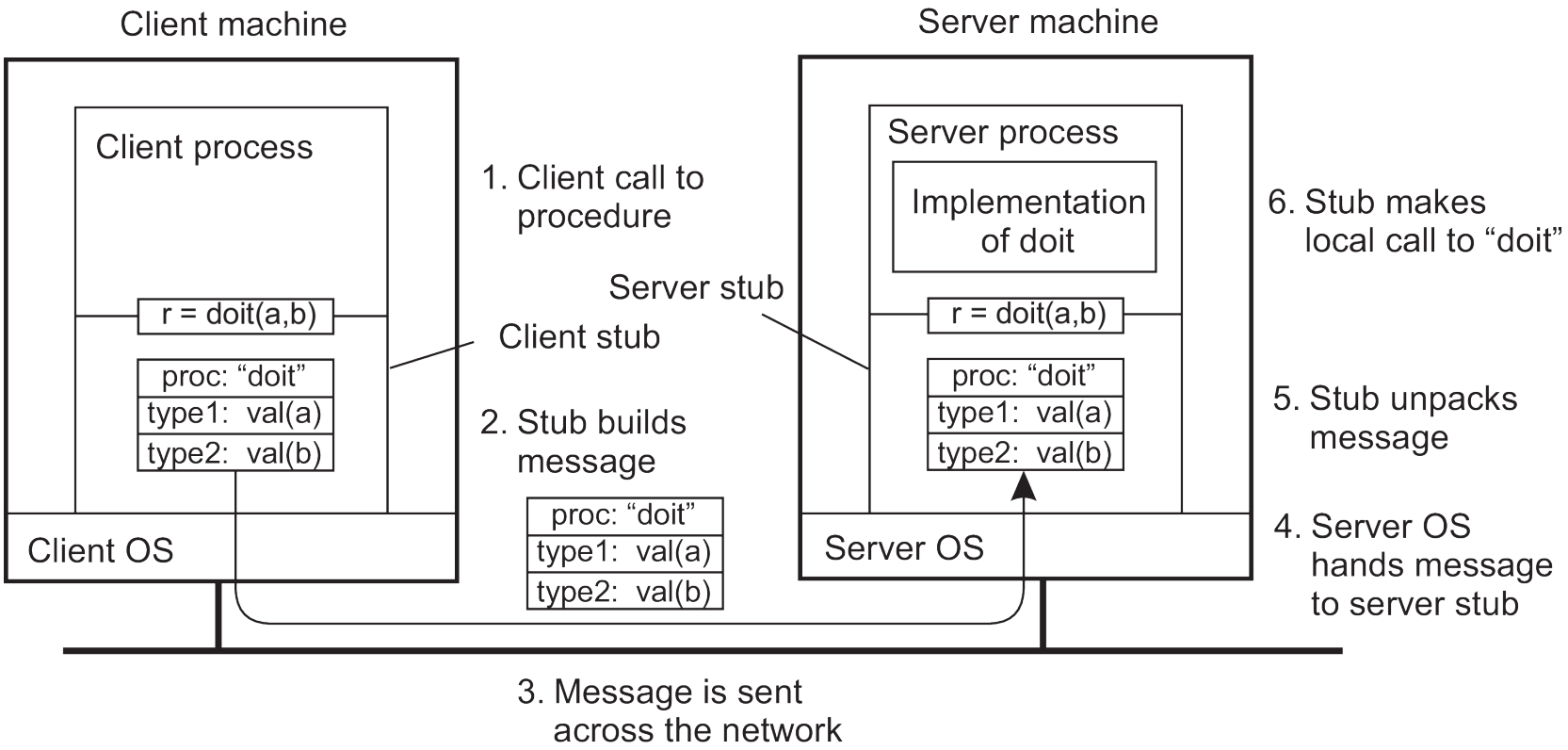
- Os desenvolvedores estão familiarizados com o modelo chamada de procedimentos.
- Procedimentos bem projetados operam isoladamente (caixa-preta).
- Não há nenhuma razão fundamental para não executar procedimentos em máquinas separadas.

Conclusão

A comunicação entre chamador e chamado pode ser ocultada usando o mecanismo de chamada de procedimento.

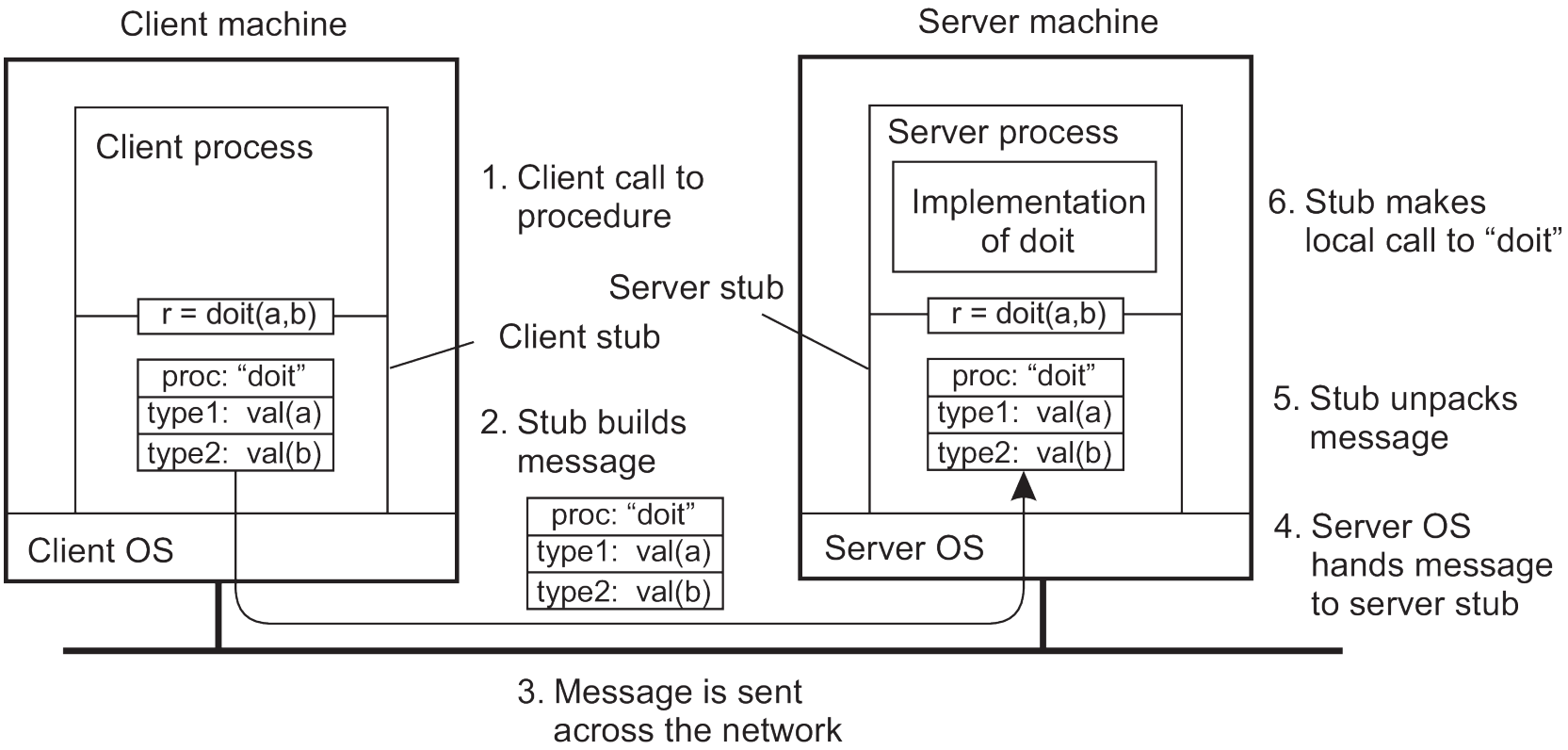


Funcionamento básico de RPC



1. O processi cliente chama o stub do cliente.
2. O stub constrói a mensagem; chama o SO local.
3. O SO envia a mensagem para o SO remoto.
4. O SO remoto manda a msg ao stub.
5. O stub desempacota parâmetros; chama o servidor.

Funcionamento básico de RPC



6. O servidor faz a chamada local; retorna o resultado ao stub.
7. O stub constrói a mensagem; chama o SO.
8. O SO envia a mensagem para o SO do cliente.
9. O SO do cliente manda a msg ao stub.
10. O stub do cliente desempacota o resultado; retorna ao cliente.

RPC: Passagem de parâmetros

Há mais do que apenas empacotar parâmetros em uma mensagem

- As máquinas cliente e servidor podem ter representações de dados diferentes (e.g. ordem de bytes).
- Empacotar um parâmetro significa transformar um valor em uma sequência de bytes.
- Cliente e servidor devem concordar com a mesma codificação:
- Como valores de dados básicos (inteiros, floats, caracteres) são representados
- Como valores de dados complexos (arrays, structs, unions) são representados

Conclusão

Cliente e servidor precisam interpretar as mensagens corretamente, transformando-as em representações dependentes de máquina.

RPC: Passagem de parâmetros

Algumas suposições

- Semântica de copy in/copy out: enquanto o procedimento é executado, nada pode ser assumido sobre os valores dos parâmetros.
- Todos as dados são passados por valor. Exclui passar referências a dados (globais).

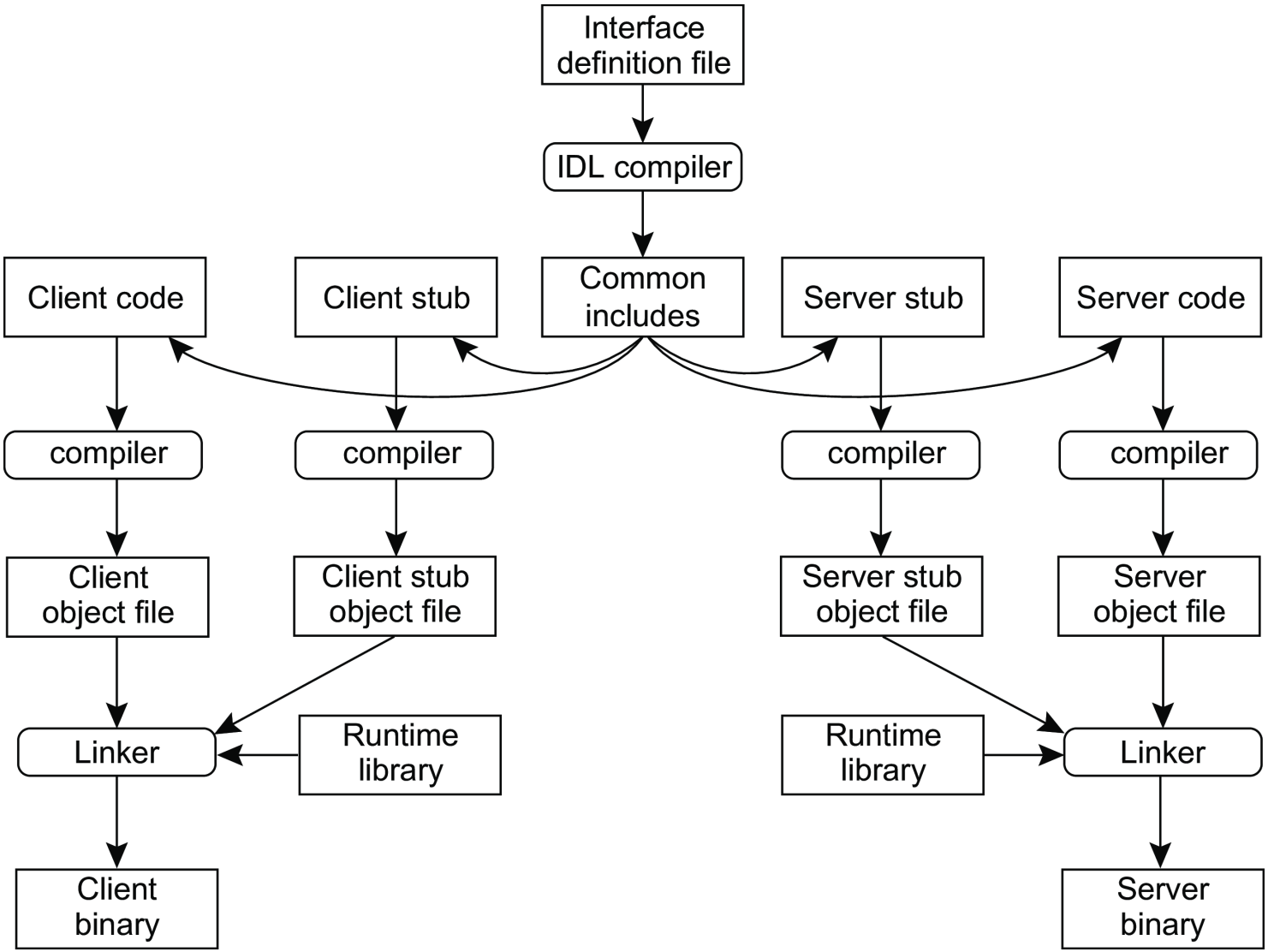
Conclusão

A transparência total de acesso não pode ser realizada.

Um mecanismo de referência remota aprimoraria a transparência de acesso

- A referência remota oferece acesso unificado aos dados remotos
- As referências remotas podem ser passadas como parâmetro em RPCs

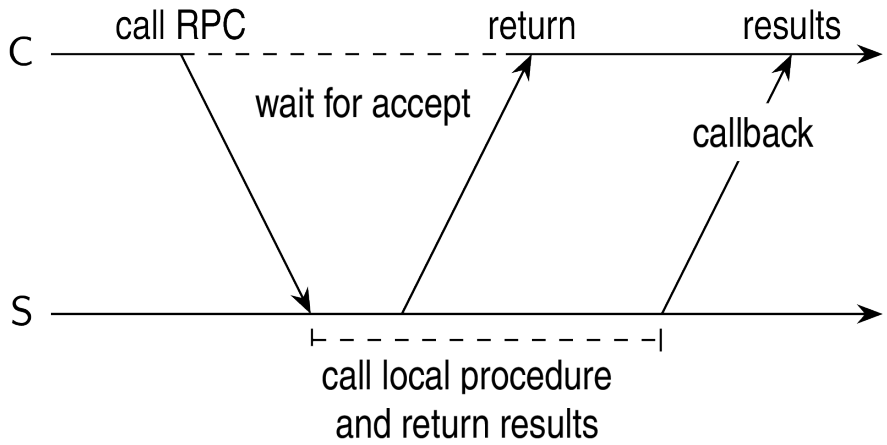
Interface Definition Language (IDL)



RPCs Assíncronos

Essência

Tentar se livrar do comportamento estrito de solicitação-resposta, permitindo que o cliente continue sem esperar por uma resposta do servidor.



Envio de múltiplos RPCs

Essência

Enviar uma solicitação de RPC para um grupo de servidores.