

Computação Distribuída

Prof. Marco Aurélio S Birchal
PUC Minas

Plano de Ensino (2º semestre de 2020)

Curso: 38 - Ciência da Computação

Disciplina: 50633 - COMPUTAÇÃO DISTRIBUÍDA

Período: 8

Turno: MANHÃ

Carga Horária:

TEÓRICA 68 horas (GRADE 68) TOTAL 68 horas

Ementa:

Conceitos e classificações de arquiteturas distribuídas. Algoritmos para: coordenação e acordo, exclusão mútua distribuída, eleições, disseminação, controle de transações e concorrência distribuídas, replicação, escalonamento de tarefas, tolerância a falhas, comunicação síncrona e assíncrona. Objetos distribuídos e invocação remota. Arquivos e Bancos de Dados distribuídos. Análise de complexidade de algoritmos distribuídos.

Objetivo

Apresentar os problemas, desafios e algoritmos clássicos de computação distribuída. Capacitar o aluno a implementar soluções distribuídas para diversos tipos de problemas computacionais.

Processo de Avaliação

As avaliações da disciplina estão divididas da seguinte forma:

Prova I - 30 Pontos

Prova II - 35 Pontos

Trabalho Prático - 30 Pontos

Avaliação de Desempenho ADA - 5 Pontos

Reavaliação 100 pontos. O aluno que obtiver rendimento com valor maior ou igual a 30 e menor que 60 pontos poderá submeter-se à Prova de Reavaliação com o valor de 100 pontos. A Nota final do aluno será obtida pela média aritmética das Provas I, II, III, dos trabalhos e da Reavaliação. O aluno será considerado aprovado caso obtenha uma nota total superior ou igual à 60 pontos.

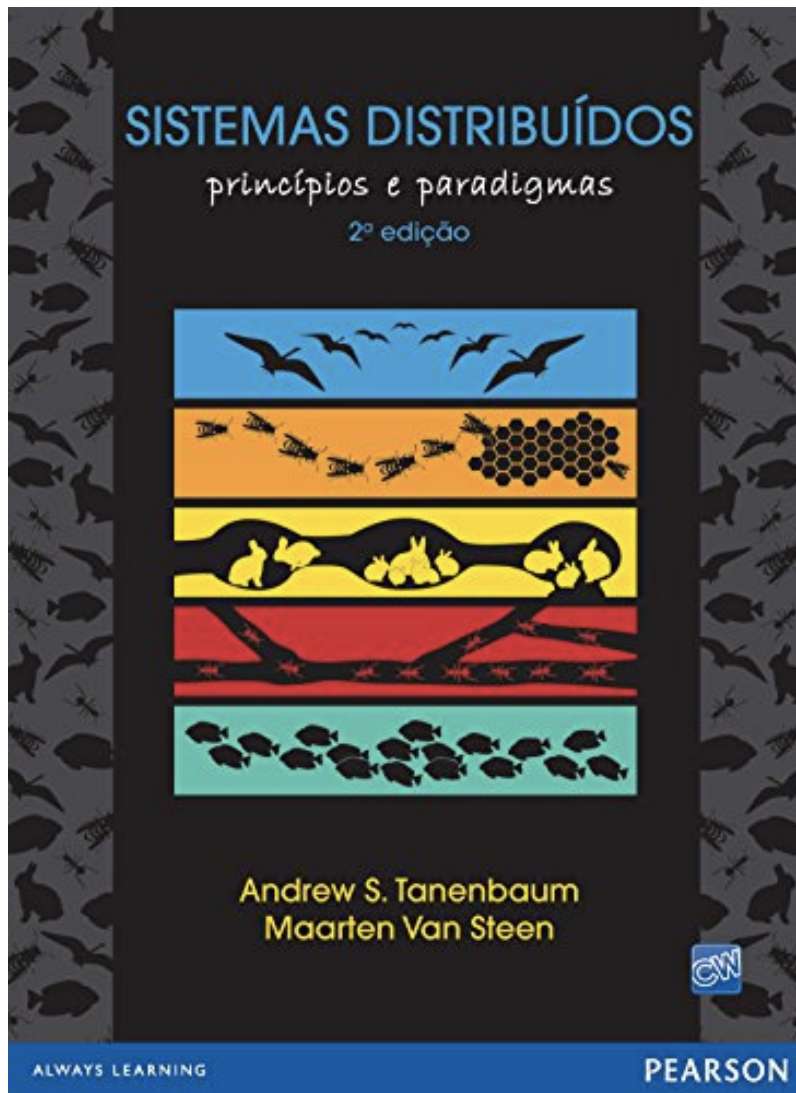
Bibliografia Básica

COULOURIS, G. et al. SISTEMAS distribuídos conceitos e projeto. 5. Porto Alegre Bookman 2013 ISBN 9788582600542.



Bibliografia Básica

TANENBAUM, Andrew S.; Steen, Maarten van. Sistemas Distribuídos: princípios e paradigmas - 2a edição. Pearson 416 ISBN 9788576051428.



Bibliografia Básica

KUROSE, James F.; Ross, Keith W. Redes de Computadores e a Internet: uma nova abordagem. Pearson 570 ISBN 9788588639102.



1 – Introdução aos Sistemas Distribuídos

Prof. Marco Aurélio S Birchal
PUC Minas

Sistemas Distribuídos (SD)

Sistemas distribuídos são sistemas, geralmente de grande porte e bastante complexos, que utilizam recursos de rede para se comunicar.

Tanenbaum (2008) os define como uma "coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente".

Sistemas Distribuídos (SD)

“Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens.” Coulouris

Características importantes:

concorrência de componentes, falta de um relógio global e falhas de componentes independentes.

Sistemas Distribuídos (SD)

Concorrência: em uma rede de computadores, a execução concorrente de programas é a norma. A capacidade do sistema de manipular recursos compartilhados pode ser ampliada pela adição de mais recursos na rede. Essa capacidade extra pode ser distribuída em muitos pontos de maneira útil. A coordenação de programas em execução concorrente e que compartilham recursos também é um assunto importante e recorrente.

Sistemas Distribuídos (SD)

Inexistência de relógio global: quando os programas precisam cooperar, eles coordenam suas ações trocando mensagens. A coordenação frequentemente depende de uma noção compartilhada do tempo em que as ações ocorrem. Existem limites para a precisão do relógios em uma rede – não existe uma noção global única do tempo correto. Essa é uma consequência direta do fato de que a única comunicação se dá por meio do envio de mensagens em uma rede.

1.1 Sistemas Distribuídos (SD)

Falhas independentes: Nos sistemas distribuídos, as falhas são diferentes. Os programas neles existentes talvez não consigam detectar se a rede falhou ou se ficou demasiadamente lenta. A falha de um computador ou o término inesperado de um programa em algum lugar no sistema (um colapso no sistema) não é imediatamente percebida pelos outros componentes. Cada componente do sistema pode falhar independentemente, deixando os outros ainda em funcionamento.

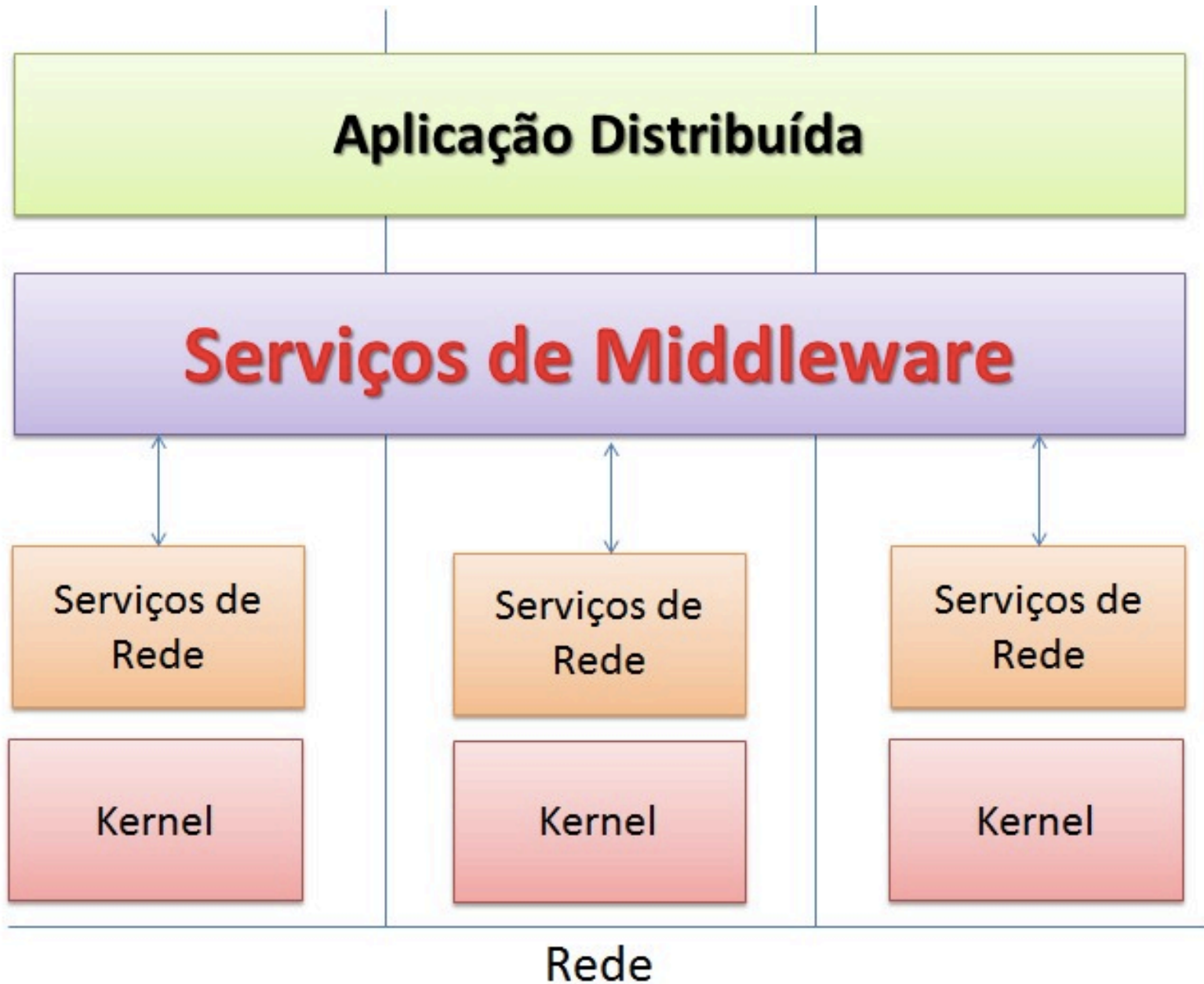
Componentes Típicos de um SD

- Rede
- Hardware
- Sistemas Operacionais
- Linguagens de Programação
- Múltiplas implementações

Middleware

Uma ou mais camadas de software que abstraem os desafios da computação distribuída, provendo um ambiente integrado e coeso.

Middleware



O Middleware CORBA

- *Common Object Request Broker Architecture*
- Desenvolvido pelo *Object Management Group* (OMG)
- Mantido por mais de 800 companhias
- Um arcabouço de *middleware* para lidar com múltiplos objetos
- Interage com um sistema distribuído
- Possui uma *Interface Definition Language* (IDL)

1.3 Tendências em Sistemas Distribuídos

- O surgimento da tecnologia de redes pervasivas (penetrantes, difusas).
- O surgimento da computação ubíqua (onipresente, que está em toda parte), combinado ao desejo de suportar mobilidade do usuário em sistemas distribuídos.
- A crescente demanda por serviços multimídia.
- A visão dos sistemas distribuídos como um serviço público.

1.3.1 Interligação em rede pervasiva e a Internet moderna

- A Internet moderna é um conjunto de redes de computadores interligadas, com uma variedade de tipos que aumenta cada vez mais e que agora inclui, por exemplo, uma grande diversidade de tecnologias de comunicação sem fio, como WiFi, WiMAX e redes de telefonia móvel.
- O resultado é que a interligação em rede se tornou um recurso pervasivo, e os dispositivos podem ser conectados (se assim for desejado) a qualquer momento e em qualquer lugar.
- um programa em execução em qualquer lugar pode enviar mensagens para programas em qualquer outro lugar através de uma grande variedade de tecnologias de comunicação.

1.3.1 Interligação em rede pervasiva e a Internet moderna

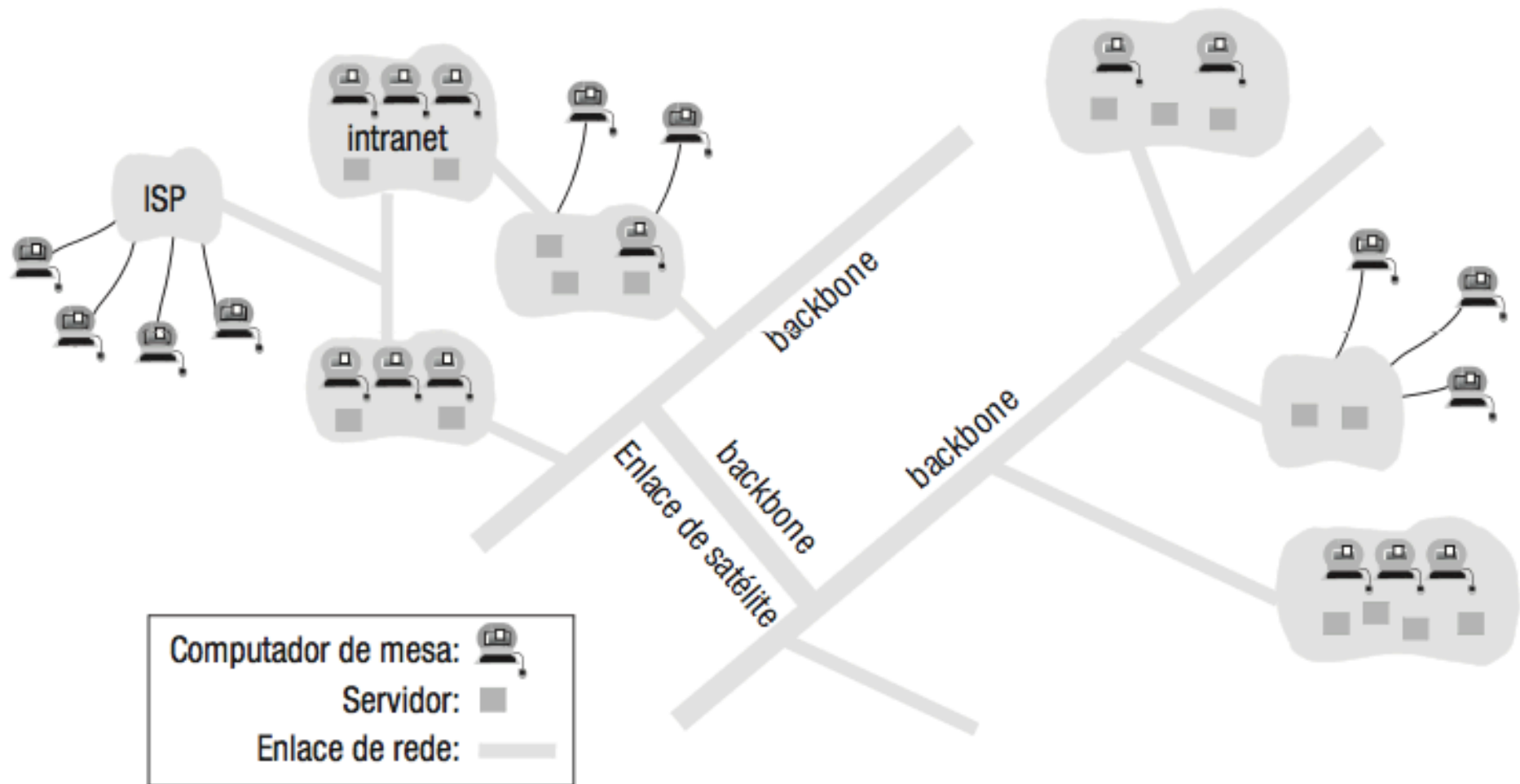


Figura 1.3 Uma parte típica da Internet.

1.3.2 Computação móvel e ubíqua

- A miniaturização de dispositivos e interligação em rede sem fio têm levado cada vez mais à integração de equipamentos de computação pequenos e portáteis com sistemas distribuídos.
- A portabilidade de muitos desses dispositivos, junto a sua capacidade de se conectar em redes em diferentes lugares, tornam a computação móvel possível.
- A computação ubíqua, também denominada computação pervasiva, é a utilização de vários dispositivos computacionais pequenos e baratos, que estão presentes nos ambientes físicos dos usuários, incluindo suas casas, escritórios e até na rua..

1.3.2 Computação móvel e ubíqua

O termo “pervasivo” se destina a sugerir que pequenos equipamentos de computação finalmente se tornarão tão entranhados nos objetos diários que mal serão notados. Isto é, seu comportamento computacional será transparente e intimamente vinculado à sua função física.

Por sua vez, o termo “ubíquo” dá a noção de que o acesso a serviços de computação está onipresente, isto é, disponível em qualquer lugar.

1.3.2 Computação móvel e ubíqua

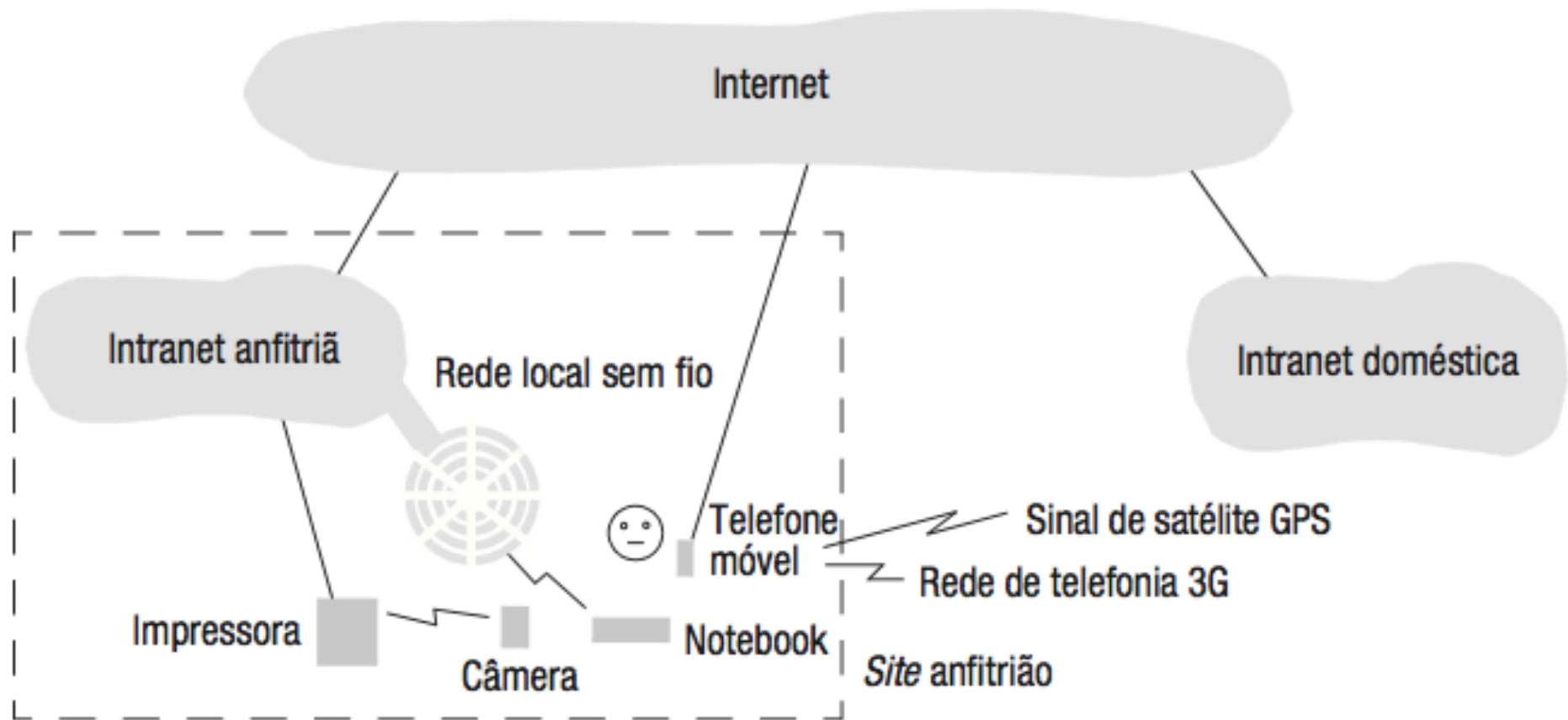


Figura 1.4 Equipamentos portáteis em um sistema distribuído.

1.3.3 Sistemas multimídia distribuídos

- Um sistema multimídia distribuído deve ser capaz de executar as mesmas funções para tipos de mídia contínuos
- A característica fundamental dos tipos de mídia contínuos é que eles incluem uma dimensão temporal e, de fato, a integridade do tipo de mídia depende fundamentalmente da preservação das relações em tempo real entre seus elementos.
- Webcasting é uma aplicação da tecnologia de multimídia distribuída. Webcasting é a capacidade de transmitir mídia contínua (normalmente áudio ou vídeo) pela Internet.

1.3.4 Computação distribuída como um serviço público

- o conceito dos recursos distribuídos como uma commodity ou um serviço público, fazendo a analogia entre recursos distribuídos e outros serviços públicos, como água ou eletricidade.
- Nesse modelo, os recursos são supridos por fornecedores de serviço apropriados e efetivamente alugados, em vez de pertencerem ao usuário final. Esse modelo se aplica tanto a recursos físicos como a serviços lógicos.
- O termo computação em nuvem é usado para capturar essa visão da computação como um serviço público.

1.3.4 Computação distribuída como um serviço público

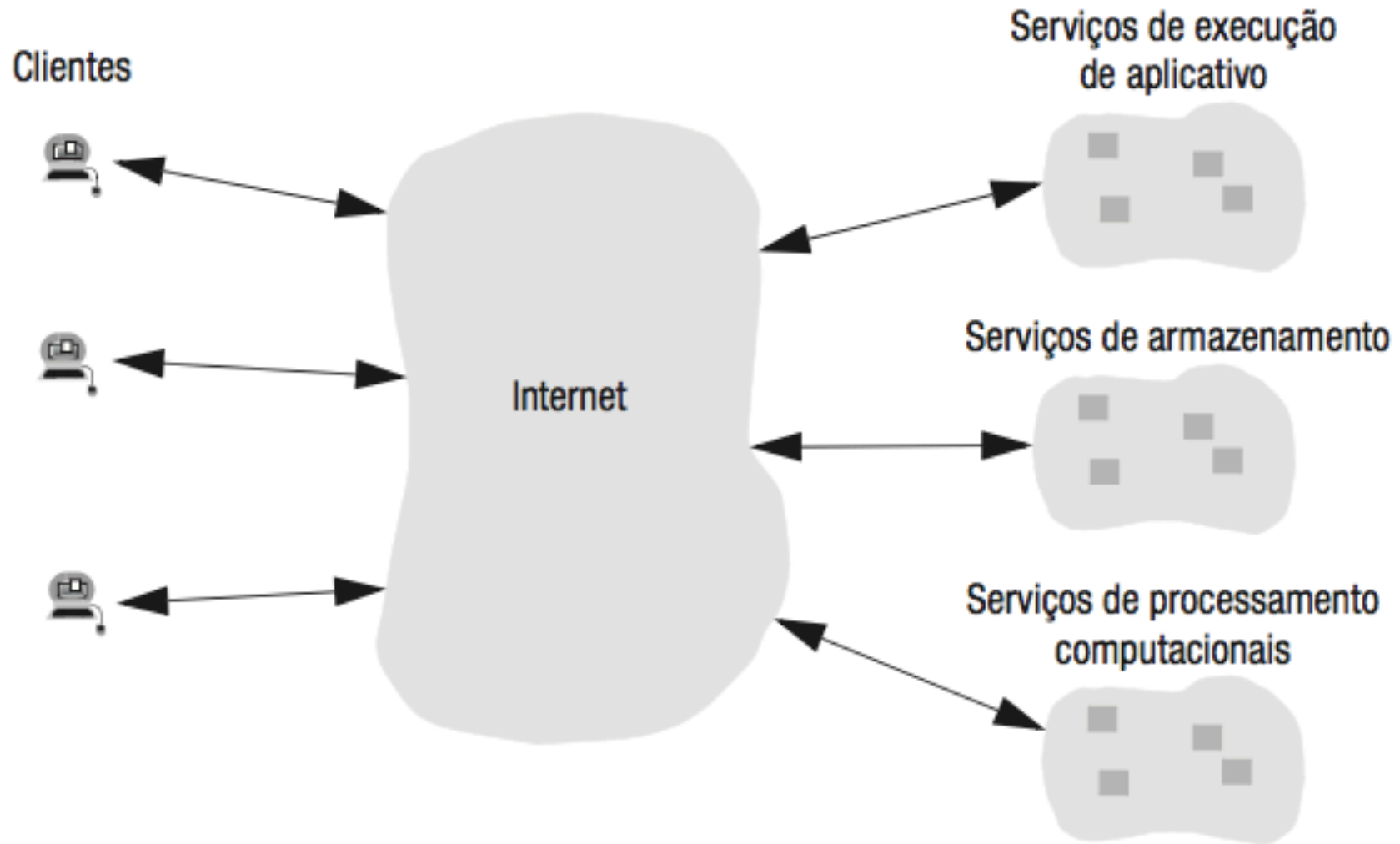


Figura 1.5 Computação em nuvem.

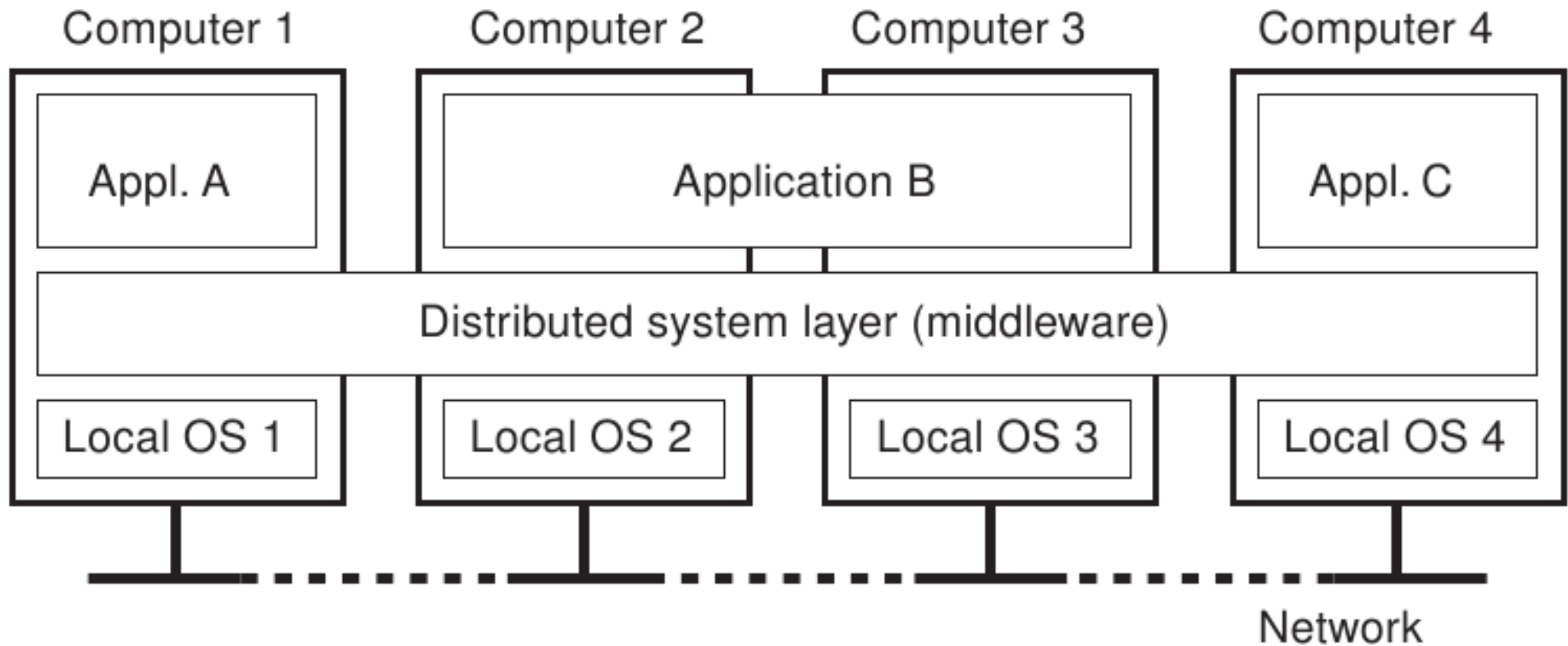
1.5 CARACTERÍSTICAS DE UM SISTEMA DISTRIBUÍDO

- Heterogeneidade: componentes heterogêneos precisam ser capazes de interoperar (instrumentos e dispositivos de diversas naturezas);
- Sistemas Abertos: As interfaces devem ser públicas e estarem disponíveis para facilitar a inserção de novos componentes; deve-se ter um ambiente “aberto”, com amplas possibilidades de integração.
- Segurança: O sistema somente deve ser utilizado da forma e para o fim que foi projetado; o ambiente deve oferecer segurança de acesso e proteção seletiva.

1.5 CARACTERÍSTICAS DE UM SISTEMA DISTRIBUÍDO

- Escalabilidade: O sistema deve operar satisfatoriamente com um número crescente de dispositivos; O desempenho do sistema deve aumentar à medida em que aumentem os seus recursos. o sistema deve ser escalável, na medida da necessidade.
- Tolerância à falha: A falha em parte do sistema não deve levar a falha do todo; erros devem ser tratados antes que provoquem maiores danos ao sistema.
- Concorrência: Deve ser possível acesso compartilhado aos recursos (políticas para permissão de acesso);
- Transparência: A distribuição deve ser o máximo possível transparente ao usuário; o sistema deve se comportar da mesma forma como se o usuário estivesse trabalhando localmente.

Heterogeneidade



Abertura

Significa o grau em que um sistema distribuído pode ser estendido e re-implementado.

Pressupõe a existência de:

- Interfaces públicas
- Padronização

Exemplo: CORBA

Segurança

- A segurança, em SD, apresenta três aspectos fundamentais:
 - Confidencialidade: somente o destinatário da mensagem deve vê-la.
 - Integridade: A mensagem não deve ser corrompida.
 - Disponibilidade: o serviço deve ser provido mesmo em situações de comportamento anômalo.

Escalabilidade

- Um SD deve ter capacidade de crescer, com o aumento da demanda.
- Deve ser provido um controle entre os componentes do sistema, que pode custar caro.
- Devem ser evitados gargalos.
- É uma propriedade que muitas vezes é confundida.

Manipulação de Erros

- Um SD deve prover mecanismos de detecção de falhas e tratamento de erros.
- Algumas técnicas disponíveis são:
 - Failure detection (ex: message checksum)
 - Failure masking (ex: retransmissão de e_mail)
 - Failure tolerance (ex: servidores replicados)
 - Failure recovery (ex: arquivos de log)

Transparência

- A transparência inclui os aspectos:
 - Transparência de Acesso: esconde as diferenças entre representação de dados e mecanismos de invocação.
 - Transparência de Localização: permite que recursos sejam acessados sem que se conheça sua localização física.
 - Transparência de Concorrência: permite que diversos processos operem concorrentemente utilizando recursos compartilhados, sem interferência.

Transparência

- (...Cont):
 - Transparência de Replicação: permite que múltiplas instâncias de um recurso sejam utilizadas para aumentar a confiabilidade e o desempenho sem que os desenvolvedores e usuários tenham de conhecer essas réplicas.
 - Transparência de Falhas: permite que usuários e programas completem seus trabalhos, mesmo na ocorrência de uma falha.
 - Transparência de Migração: permite que recursos do sistema sejam movimentados, sem que sua operação seja afetada.

Estudo de Caso de SD: World Wide Web

