

Sistemas Distribuídos

Tendências, Enfoque no
compartilhamento de recursos e
Desafios

Tendências em Sistemas Distribuídos

- ❖ Tecnologia de redes pervasivas
- ❖ Computação móvel e ubíqua
- ❖ Serviços multimídia
- ❖ SD como um serviço público



Rede pervasiva

❖ Internet moderna

❖ Recurso pervasivo

❖ Permite que usuários façam uso de serviços:

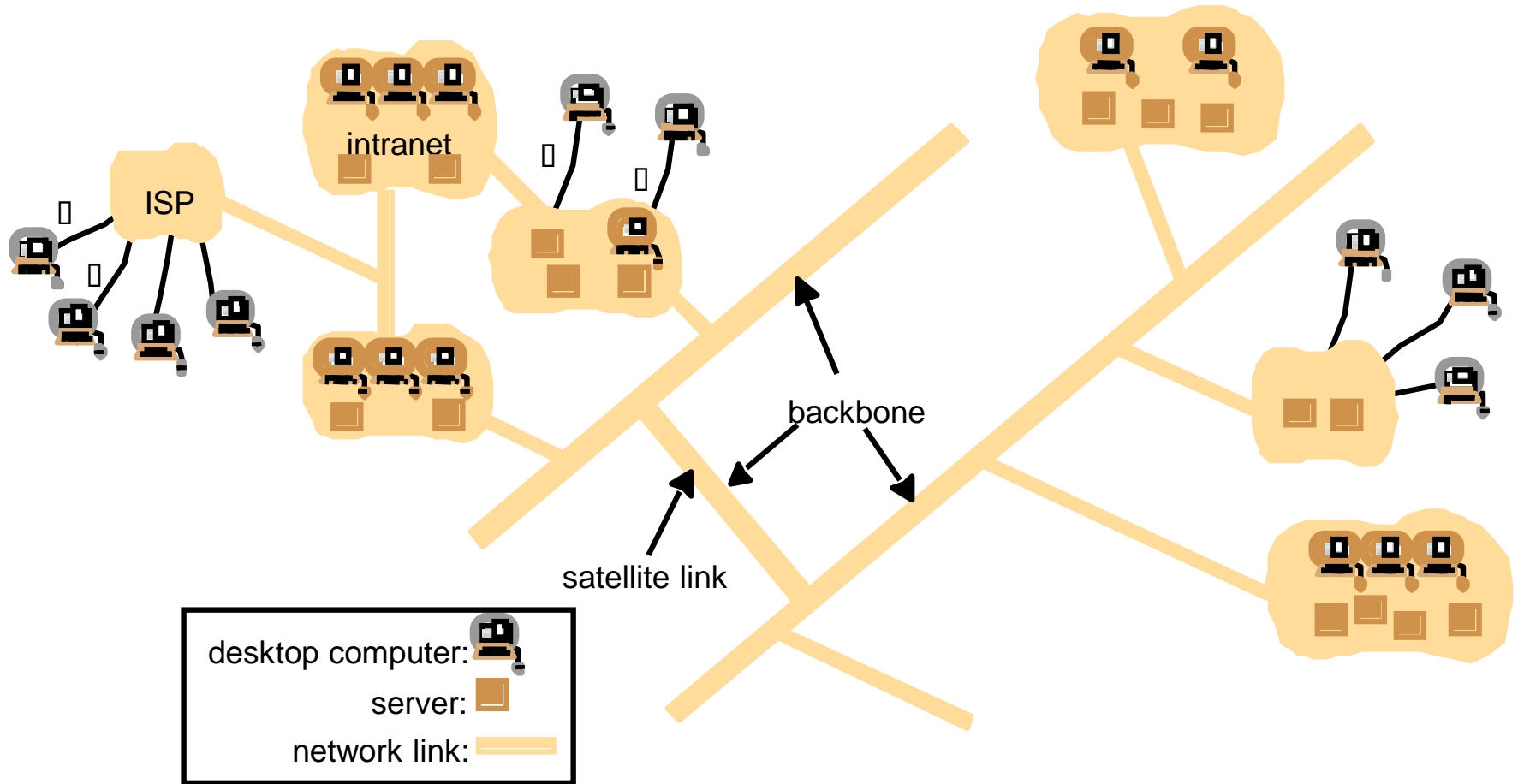
WWW, e-mail e transferência de arquivos.

❖ O conj. de serviços é aberto – adição de computadores servidores e novos tipos de serviço.

❖ Soluções práticas para os problemas de SD



Intranet



Computação móvel e ubíqua

❖ Computação Móvel

- ❖ Execução de tarefas enquanto o usuário está se deslocando.
- ❖ Miniaturização de dispositivos
- ❖ Interligação de redes sem fio
- ❖ Podem acessar a Internet , os recursos da intranet



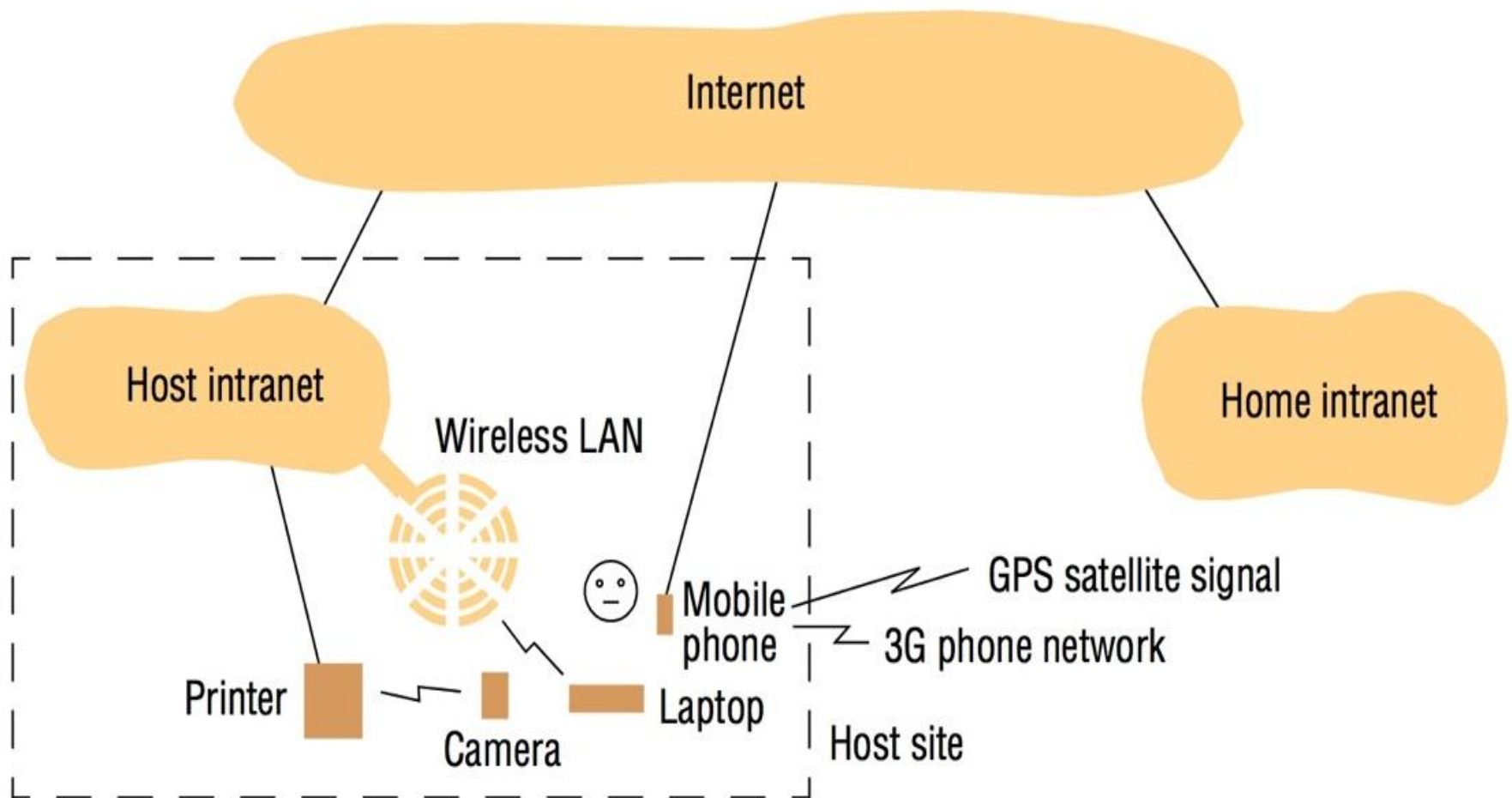
Computação móvel e ubíqua

❖ Computação Ubíqua

- ❖ Dispositivos computacionais pequenos e baratos presentes nos ambientes físicos dos usuários
- ❖ Acesso a serviços de computação em qualquer lugar.
- ❖ Ex. controlar a máquina de lavar, geladeira, som de um único dispositivo de controle remoto.



Computação móvel e ubíqua



Sistemas multimídia distribuídos

- ❖ Capacidade suportar diversos tipos de mídia de maneira integrada.
- ❖ Diversas vantagens: TV ao vivo, vídeo sob demanda, bibliotecas de músicas, teleconferências, telefonia IP
- ❖ Webcasting
 - ❖ Suporte a uma variedade de formato de codificação e criptografia
 - ❖ Garantir a qualidade de serviço
 - ❖ Estratégia de gerenciamento de recursos
 - ❖ Estratégia de adaptação em que QoS não pode ser mantida

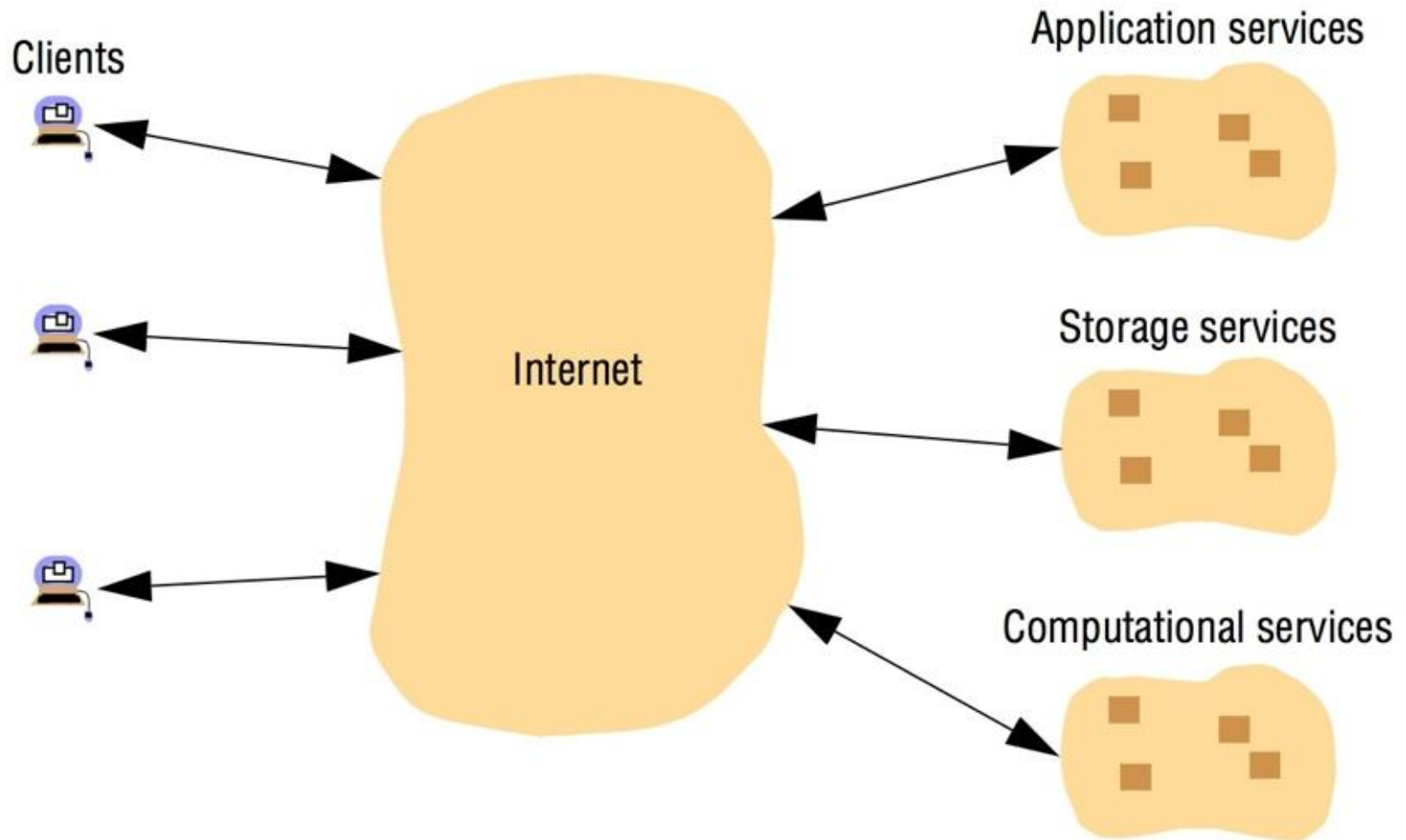


Computação distribuída com serviço público

- ❖ Analogia entre os recursos distribuídos e outros serviços públicos
- ❖ Recursos físicos disponíveis
- ❖ Serviços de software
- ❖ Computação em nuvem: clusters



Computação em nuvem



Enfoque no compartilhamento de recursos

- ❖ Compartilhamento de hardware
- ❖ Compartilhamento de informações
- ❖ Acesso através de serviços
 - ❖ Clientes
 - ❖ servidor



desafios

- ❖ Heterogenidade
- ❖ Sistemas abertos
- ❖ Segurança
- ❖ Escalabilidade
- ❖ Tratamento de falhas
- ❖ Concorrência
- ❖ Transparência



Heterogeneidade

- ❖ Variedade e diferença
- ❖ Internet: conj. heterogêneo de computadores em rede.
- ❖ A heterogeneidade se aplica aos seguintes aspectos:
 - ❖ Rede
 - ❖ Hardware de computadores
 - ❖ Sistemas operacionais
 - ❖ Linguagens de programação
 - ❖ Implementações de diferentes desenvolvedores



Heterogeneidade

❖ Rede

- ❖ Protocolos de internet – mascara as diferenças
- ❖ Cada tipo de rede tem sua implementação dos protocolos de internet

❖ Hardware: exemplo tipos de dados

- ❖ Representados de diversas maneiras em diferentes tipos de hardware.
- ❖ Deve ser levado em consideração se programas trocam mensagens desses tipo.



Heterogeneidade

❖ Sist. Operacionais

- ❖ Diferença nas interfaces de programação de aplicativos
- ❖ As chamadas de troca de mensagens do Unix e do Windows são diferentes.

❖ Linguagem de Programação

- ❖ Programas em diferentes linguagens podem se comunicar.
- ❖ Utilizar padrões comuns.



Heterogeneidade - middleware

- ❖ Camada de software que fornece uma abstração de programação.
- ❖ Solução do problema de heterogeneidade
- ❖ As diferenças são tratadas em nível de SO e HW
 - ❖ Corba, Java RMI e DCOM
- ❖ Sua implementação oculta o fato de que as mensagens passam por uma rede.



Sistemas Abertos

- ❖ Novos serviços podem ser adicionados e disponibilizados.
- ❖ As interfaces devem ser publicadas
 - ❖ Requests for Comments (RFCs)
 - ❖ Documentos técnicos
- ❖ Mecanismo de comunicação uniforme
- ❖ Independência de fornecedores



Segurança

- ❖ Valor da informação.
- ❖ Nos SDs os clientes enviam pedidos para acessar dados de um servidor
 - ❖ Médico – dados de pacientes
 - ❖ No comércio eletrônico e bancos – num. do cartão
- ❖ Ocultar o conteúdo das mensagens mas também a identidade do usuário.
 - ❖ Solução: criptografia



Segurança

- ❖ Dois problemas ainda não foram resolvidos totalmente
 - ❖ DoS
 - ❖ Segurança de código móvel. Exemplo recepção de um programa executável via e-mail.



Escalabilidade - desafios

❖ Controlar o custo dos recursos físicos

- ❖ Demanda por recurso aumenta, amplia-se o sistema
- ❖ Evitar gargalos. P. Ex. adicionar servidores de arquivos
- ❖ Parece ser óbvio, não é necessariamente fácil.

❖ Controlar a perda de desempenho

- ❖ Ex. DNS - algoritmo hierárquicos para busca



Escalabilidade - desafios

- ❖ Impedir que os recursos de SW se esgotem

- ❖ Ex. endereços IP (32 bits) – (128 bits) implica em modificações de componentes de SW

- ❖ Evitar gargalos de desempenho

- ❖ Em geral algoritmos descentralizados. Ex. predecessor do DNS era arquivo central feito por download



Tratamento de falhas

- ❖ Falha de HW ou SW podem causar resultados incorretos ou parar antes de ter concluído
- ❖ Em SDs as falhas são parciais
- ❖ O tratamento é particularmente difícil
 - ❖ Detecção de falhas – soma de verificação
 - ❖ Mascaramento de falhas – ocultação
 - ❖ Tolerância a falhas – usuário decide
 - ❖ Recuperação de falhas
 - ❖ Redundância



Concorrência

- ❖ Vários pedidos processados concorrentemente
 - ❖ Recurso encapsulado – threads
- ❖ Garantir que opere corretamente o recurso
 - ❖ Não assuma resultados inconsistentes
- ❖ Utilização de técnicas padrão
 - ❖ Semáforos



Transparência

- ❖ **Transparência:** ocultação do usuário final ou programador de aplicativos
 - ❖ **Acesso:** local ou remoto em operações idênticas
 - ❖ **Localização:** recurso acessados sem conhecimento de sua localização física
 - ❖ **Concorrência:** uso concorrente sem interferência do outro.
 - ❖ **Replicação:** aumentar a confiabilidade e o desempenho sem o conhecimento do usuário



Transparência

- ❖ Falhas: possibilidade de concluírem suas tarefas
- ❖ Mobilidade: movimentação de recursos sem afetar a operação
- ❖ Desempenho: reconfigurar o sistema a medida que as cargas variam.
- ❖ Escalabilidade: expansão sem alterar a estrutura ou algoritmos.

