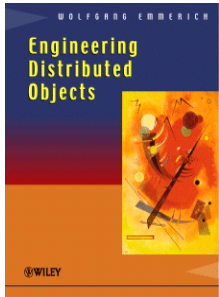


Capítulo I – Introdução aos Sistemas Distribuídos

From: Wolfgang Emmerich

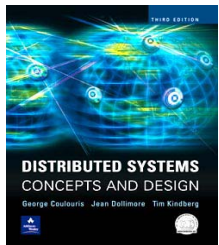
Engineering Distributed Objects

John Wiley & Sons, Ltd 2000



From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design

Edition 3, © Addison-Wesley 2001



Paula Prata,

Departamento de Informática da UBI

<http://www.di.ubi.pt/~pprata>

O que é um sistema distribuído?

Conjunto de computadores ligados em rede, com software que permita a partilha de recursos e a coordenação de actividades, oferecendo idealmente um sistema integrado.

Características:

- Comunicação através de mensagens
- Concorrência
- Partilha de recursos
- Sistema Assíncrono
- Falhas Independentes
- Heterogeneidade

O que é um sistema distribuído?

Características:

- Os componentes do sistema comunicam através de mensagens

(não existem variáveis globais partilhadas)

modelos de programação: cliente/servidor, modelo baseado em objectos, ...

- Concorrência: os vários utilizadores utilizam o sistema em simultâneo

(é necessário coordenar o acesso aos recursos partilhados: hw, sw, dados)

- Partilha de recursos

impressoras, discos, ferramentas para trabalho cooperativo, bases de dados

A partilha de recursos levanta questões de segurança

Gestores de recursos controlam o acesso a recursos partilhados

O que é um sistema distribuído?

Características:

- Sistema Assíncrono

não existe um relógio global

diferentes velocidades de processamento

não existe um limite para o tempo de comunicação

- Falhas Independentes

- falhas na rede (perda de mensagens, duplicação, reordenação)

- falhas em unidades de processamento

=> a falha de um componente não impede necessariamente os outros de funcionar

O que é um sistema distribuído?

Características:

- Heterogeneidade

Um sistema distribuído pode possuir:

- . diferentes tipos de rede

- . diferentes tipos de hardware

(diferentes representações de dados, diferente código máquina)

- . diferentes sistemas operativos

(diferentes interfaces para os protocolos de comunicação)

- . diferentes linguagens de programação

(diferentes representações de estruturas de dados como arrays ou registos,...)

Para tentar resolver o problema da heterogeneidade define-se uma camada de software intermédia: *middleware*

The Eight Fallacies of Distributed Computing

“Essentially everyone, when they first build a distributed application, makes the following eight assumptions.

All prove to be false in the long run and all cause big trouble and painful learning experiences.”

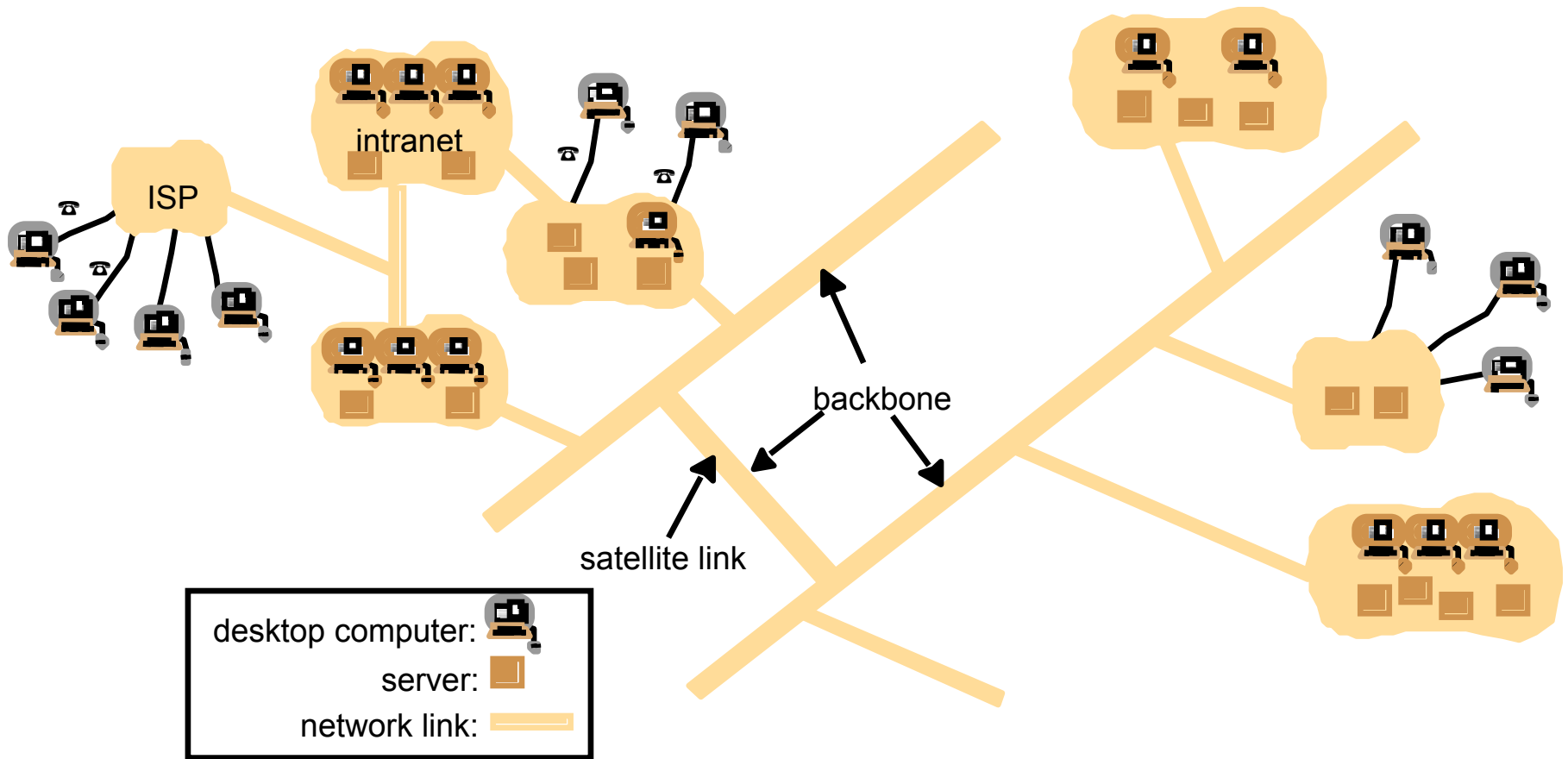
Peter Deutsch

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

first published the "8 Fallacies of Networking" internally while working at Sun Labs in 1991-92.

Exemplos de sistemas distribuídos:

A Internet



Exemplos de sistemas distribuídos:

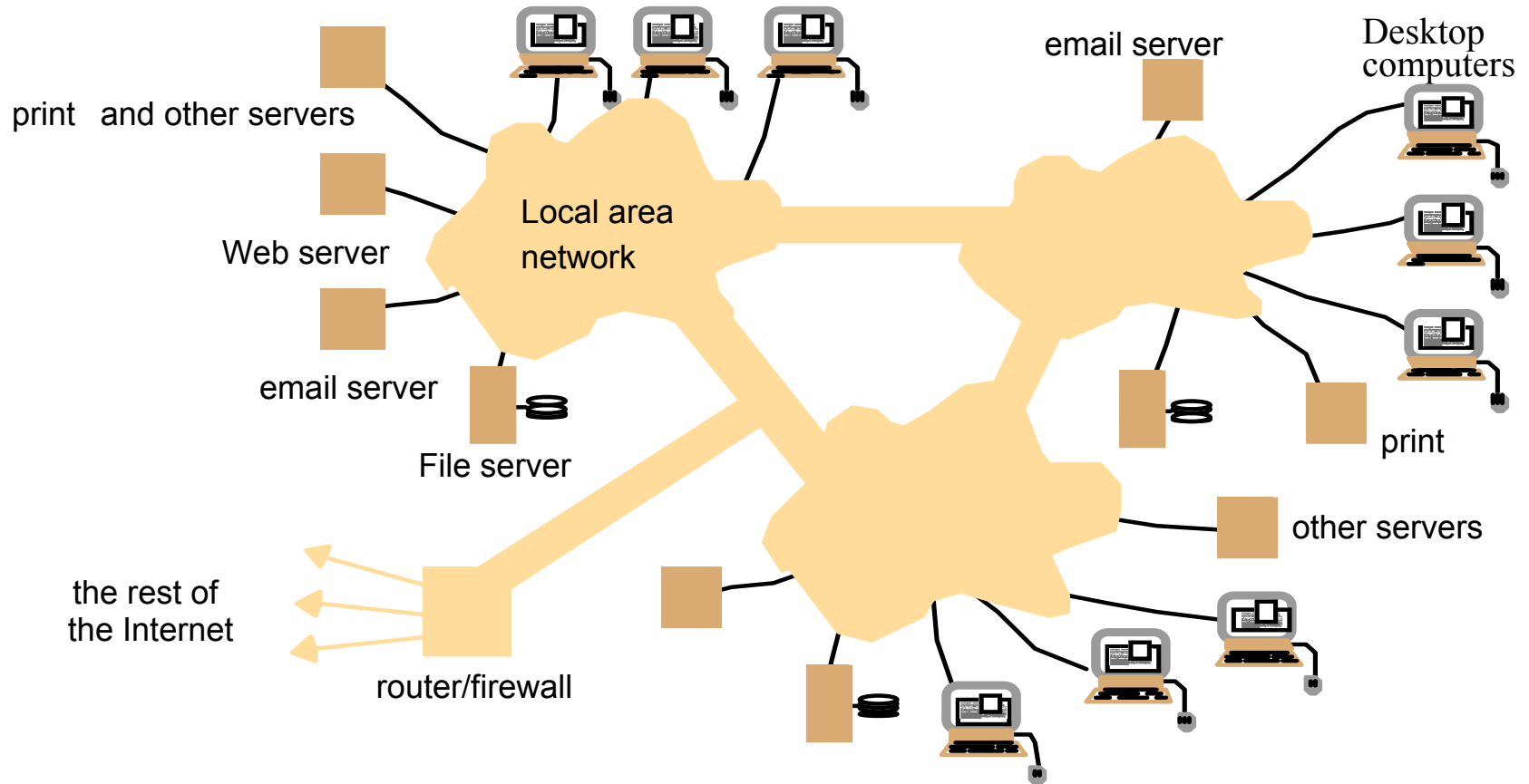
O desenho e construção dos mecanismos de comunicação da Internet (protocolos Internet) permitiu que um programa em execução num qualquer ponto da rede possa enviar mensagens a programas em qualquer outro lugar.

Computadores na internet

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866

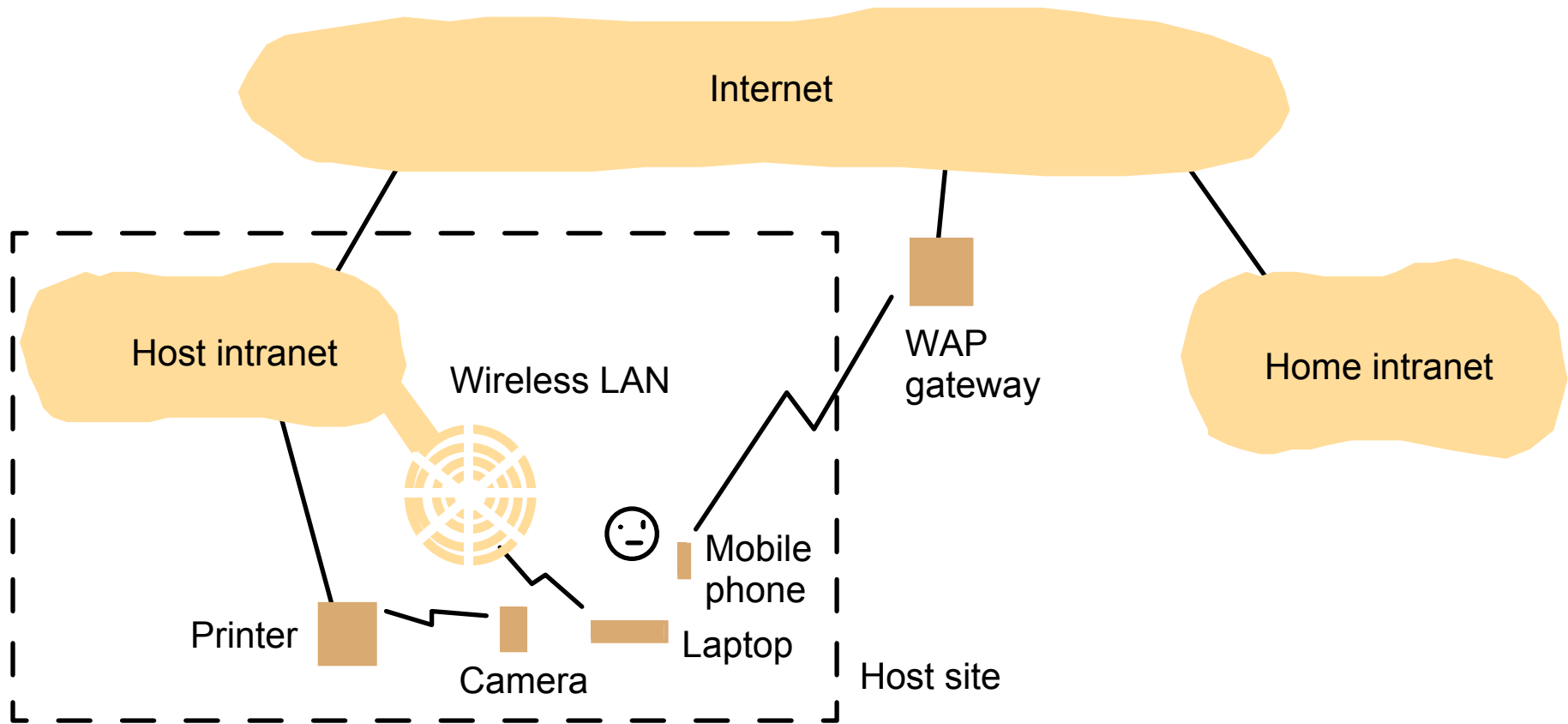
Exemplos de sistemas distribuídos:

Intranets



Computação Móvel e Ubíqua

Integração de equipamento portátil em sistemas distribuídos



Computação Móvel e Ubíqua

- . Computadores portáteis
- . Telefones móveis
- . PDA's - Personal Digital Assistants
- . Máquinas fotográficas digitais
- . Pagers
- ...
- . “wearable devices”, relógios com processador, ...
- . Sistemas embebidos (= computadores dentro de um produto)
electrodomésticos, carros, ...

=> O termo computação Ubíqua pretende designar sistemas cuja utilização está de tal forma integrada na funcionalidade do produto que é transparente para o utilizador.

=> Maiores restrições em termos de: custo, tamanho, potência/autonomia

Outros Sistemas Distribuídos:

- . Correio electrónico
- . World wide web
- . Sistemas de ficheiros distribuídos

Aplicações críticas:

- . Reserva de bilhetes em companhias de transportes
 - . Comércio electrónico
 - . Máquinas Multibanco
- => Exigem fiabilidade e segurança

Desafios na implementação de sistemas distribuídos:

- Escalabilidade
- Abertura
- Tolerância a Falhas
- Segurança
- Transparência

Escalabilidade

Capacidade de o sistema se manter a funcionar de forma correcta e à velocidade desejada independentemente do número de utilizadores.

É necessário:

- desenhar o software de forma a que o aumento de utilizadores não exija grandes alterações
- evitar algoritmos e estruturas de dados centralizadas (replicação de dados se necessário)

Desafios na implementação de sistemas distribuídos:

Escalabilidade (cont)

- controlar o aumento de custos devido à disponibilização de mais recursos
- controlar a perda de performance

(replicação de serviços)

- evitar o transbordo de certos limites de recursos (ex. Endereço IP com 32 bits, insuficiente)

Sistema Aberto (“Openness”)

Capacidade de o sistema ser extensível, quer em hardware quer em software

- novos componentes devem poder ser adicionados sem por em causa o funcionamento dos já existentes, e poder comunicar com eles

Desafios na implementação de sistemas distribuídos:

Sistema Aberto (cont)

Para isso é importante que:

- sejam conhecidas as interfaces dos novos componentes através da publicação da sua documentação
- utilizar protocolos e formatos standard

Exemplo de publicação de interfaces:

Request For Comment (RFCs) www.ietf.org

Contém as especificações dos protocolos internet desde o início dos anos 80

Tolerância a Falhas

Tolerar uma falha significa conter os seus efeitos de forma a que o sistema continue a funcionar.

Desafios na implementação de sistemas distribuídos:

Tolerância a Falhas (cont)

- Detecção da falha

Dados corrompidos (mensagens ou ficheiros) podem ser detectados através de somas de verificação.

- Localização da falha

Se não houve resposta a um pedido, o que significa?

- falha na rede
- falha no nó destino

como distinguir ?

Desafios na implementação de sistemas distribuídos:

Tolerância a Falhas (cont)

- Mascarar / Tolerar a falha

Algumas falhas podem ser ocultadas do utilizador se se utilizar redundância suficiente:

- quando uma mensagem não chega, pode ser retransmitida
- um ficheiro pode ser escrito em duplicado (um em cada disco)
- entre cada dois “routers” da internet, devem sempre existir dois percursos
- uma base de dados pode ser replicada em vários servidores

Desafios na implementação de sistemas distribuídos:

Segurança

Manter recursos computacionais seguros significa:

- Manter o nível de confidencialidade exigido pelos utilizadores
(protecção contra acessos não autorizados)
- Garantir a integridade dos dados
(protecção contra alteração ou corrupção de dados ou programas)
- Manter a disponibilidade do sistema
(protecção contra interferências com os meios de acesso aos recursos)

Alguns problemas por resolver:

- Ataques do tipo “negação de serviço” (denial of service)
- Segurança do código móvel

Desafios na implementação de sistemas distribuídos:

Transparência

O sistema deve ser visto como um todo e não como uma coleção de componentes distribuídos.

No standard de “Open Distributed Processing (ODP) foram definidos os seguintes tipos de transparência:

- Acesso
- Localização
- Concorrência
- Replicação
- Falhas
- Migração
- Desempenho
- Escalabilidade

Mais Importantes:

Acesso e Localização em conjunto são referidos como transparência de rede

Desafios na implementação de sistemas distribuídos:

Transparência de acesso:

permite que o acesso a recursos locais e a recursos remotos seja feito através das mesmas operações (i.é, usando a mesma interface).

(ex. Samba vs FTP)

Transparência de localização:

permite que os recursos possam ser acedidos sem o conhecimento da sua localização.

(ex. Programas de correio electrónico)

Transparência de concorrência:

permite que os vários clientes de um componente não necessitem de ter em conta o acesso concorrente ao componente.

Desafios na implementação de sistemas distribuídos:

Transparência de replicação:

permite que os clientes de um componente não se apercebam se existe replicação e estão a usar uma réplica e não o original; a utilização de várias instâncias de um componente pode ocorrer por razões de desempenho ou de fiabilidade; os utilizadores do componente não necessitem de saber que o componente possa ser replicado.

Transparência de Falhas:

permite que o sistema funcione na presença de falhas de hardware ou software sem que utilizadores e programadores saibam como as falhas foram ultrapassadas.

(por ex. um sistema de e-mail pode retransmitir uma mensagem até que a mesma seja entregue com sucesso)

Transparência de Migração:

permite que um recurso possa mudar de localização sem que isso afecte a sua utilização.

(ex. Telemóveis em movimento)

Desafios na implementação de sistemas distribuídos:

Transparência de Desempenho:

permite que o sistema seja reconfigurado para melhorar o seu desempenho sem que os utilizadores se apercebam.

Transparência de Escalabilidade:

permite que o sistema seja expandido sem que os utilizadores se apercebam de como isso foi conseguido.