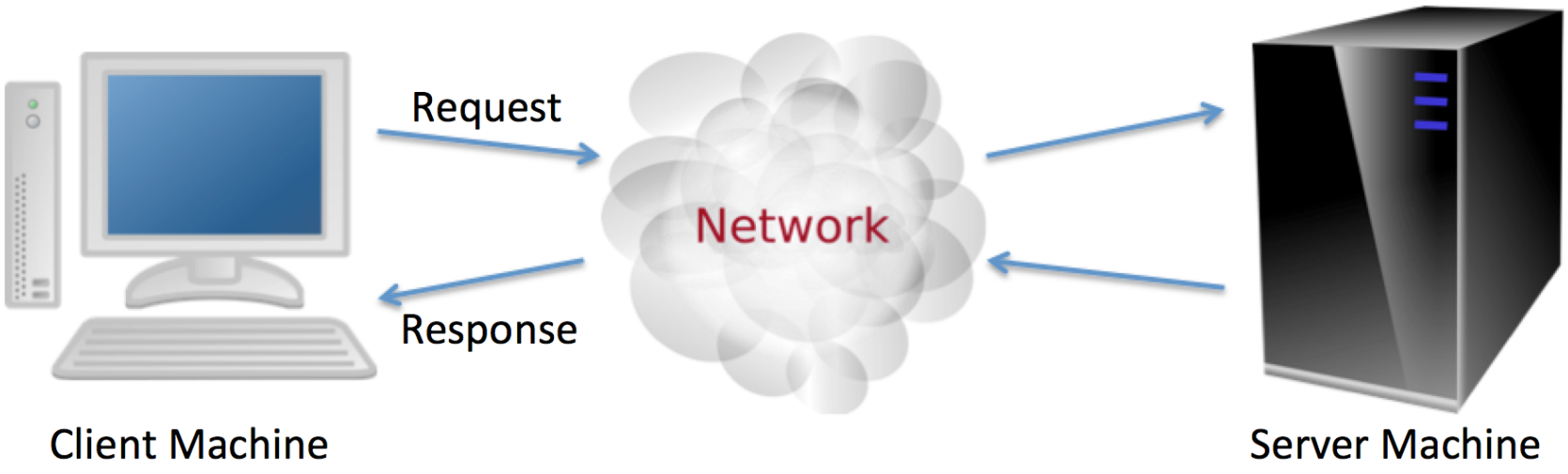
Comunicação Entre Processos: O Modelo Cliente-Servidor

Sistemas Distribuídos 2023/24

Modelo cliente-servidor típico



Camadas protocolares de rede

Aplicação

- HTTP, FTP, CORBA, e aplicações específicas

Transporte

- TCP e UDP
- Mensagens trocadas entre processos através de portos

Rede

- Transmissão de pacotes IP

Ligação

- Transmissão de pacotes entre nós ligados fisicamente

Endereçamento IP

- A maior parte do tráfego da Internet usa IP versão 4.
- Cada host é identificado por um endereço numérico de 4 bytes (32 bits).
  - Exemplo: 193.137.200.147
- Prevê-se que o IPv6 venha a substituir o IPv4 (endereços de 16 bytes permitem 10^{38} hosts, e o IPv6 já corresponde a mais de 45% do tráfego).

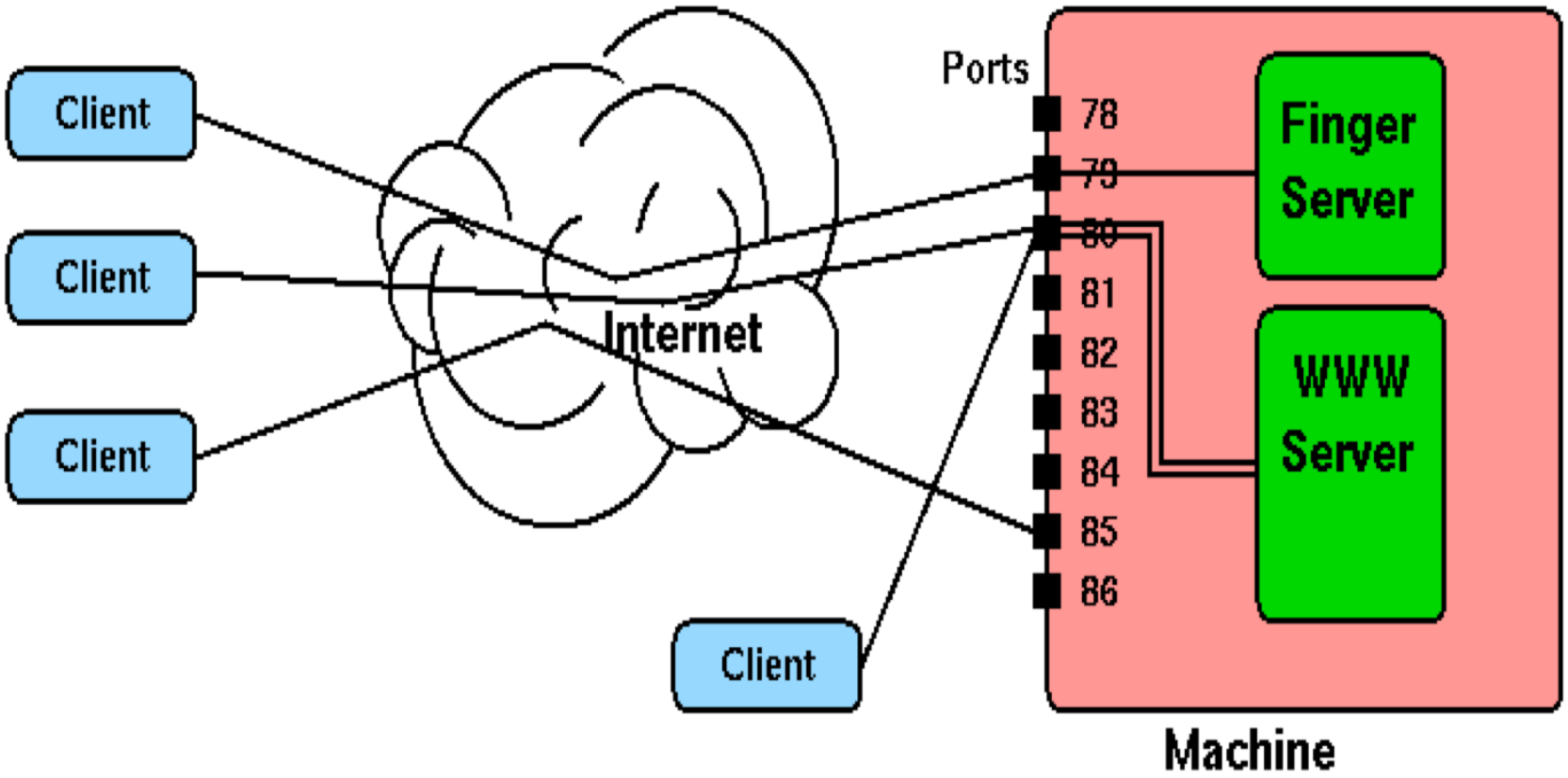
Domain Name System (DNS)

- A Internet suporta um esquema de nomes simbólicos para as máquinas, tais como www.uc.pt ou google.pt.
- Convenientes para nós, mas as máquinas têm de traduzir domínios para endereços IP, usando para isso o Domain Name System (DNS).
- Cada domínio corre um ou mais servidores de DNS.
- É um serviço distribuído (cada servidor de DNS conhece apenas uma pequena parte do total).

Endereçamento de sockets

- Cada máquina pode ter muitos processos a correr, portanto o endereço IP não é suficiente para identificar o destinatário.
- Um processo é identificado pelo endereço IP e pelo número do porto ao qual está associado.
- O identificador do porto tem 16 bits (até 65535).
- "Well-known ports" até 1023 (*e.g.*, HTTP no porto 80, SMTP no porto 25, FTP nos portos 20 e 21).

Exemplo


Client
Client
Client
Client
Internet
Ports
78
79
80
81
82
83
84
85
86
Finger Server
WWW Server
Machine

Protocolos de transporte

Protocolo UDP

- Não tem mecanismos de fiabilidade (para além do checksum).
- Não garante entrega, ordenamento ou filtragem de duplicados.
- Não há ligações, controlo de congestionamento, garantia de transmissão.
- Tem baixa latência.

Protocolo TCP

- Protocolo orientado por ligações.
- Garante a entrega dos dados, mantendo a ordem.
- Mecanismos de sequenciamento, controlo de fluxo, retransmissão e buffering.
- Não garante largura de banda nem tempo de transmissão.

Protocolo TCP

- As ligações TCP são fiáveis, full-duplex e ponto-a-ponto.
- É baseado em streams – uma ligação transporta um stream de bytes e não um stream de mensagens.
- Não existem fronteiras entre mensagens.
- O recetor não tem detalhes sobre a forma como os dados foram escritos.

Funcionamento do protocolo TCP

- Estabelecimento e término das ligações.
- Sequenciamento – o stream é repartido numa sequência numerada de pacotes IP.
- Fiabilidade ponto-a-ponto – acknowledgements, checksums, timeouts, retransmissões.
- À medida que os dados vão chegando, ficam num buffer até serem recebidos pelo processo recetor (overflow caso a receção seja mais lenta que o envio).

Protocolo UDP

- O protocolo UDP é interessante por reduzir o overhead associado às garantias de fiabilidade.
  - Manutenção do estado da ligação em ambos os lados.
  - Transmissão de mensagens adicionais.
- Aplicações que tenham um pedido e uma resposta únicos usam UDP.
  - O protocolo DNS beneficia da redução do overhead inicial; aplicações de Voice over IP toleram omissões ocasionais.
- Os pacotes podem perder-se por diversas razões, sendo que não é dada indicação ao emissor nem ao recetor (os pacotes desaparecem.)

Métricas relevantes

Perda de dados

- Algumas aplicações toleram perdas ocasionais (*e.g.*, audio).
- Outras aplicações (HTTP, FTP, SMTP) requerem transmissão fiável de dados.

Largura de banda

- Certas aplicações requerem uma largura de banda mínima.
- Outras aplicações são elásticas e usam a largura de banda que tiverem.

Timing

- Algumas aplicações são sensíveis aos atrasos (*e.g.*, Voice over IP, jogos interativos).
- Muitas aplicações são perfeitamente usáveis apesar do atraso.

Características das aplicações

Aplicação	Perda de dados	Largura de banda	Tempo
transferência de ficheiros	sem perdas	elástica	não
email	sem perdas	elástica	não
documentos web	sem perdas	elástica	não
audio em tempo real	tolerante	100kbps	100ms
vídeo em tempo real	tolerante	500kbps	100ms
vídeo/audio armazenado	tolerante	idem	1s
jogos interativos	tolerante	alguns kbps	50ms
mensagens instantâneas	sem perdas	elástica	5s ou mais

- Transferência de ficheiros, Web, e email usam TCP; multimedia, streaming, VoIP tipicamente usam UDP, ainda que haja exceções.

Comunicação entre processos

- Duas operações fundamentais: envio e receção.
- Um processo envia uma mensagem (uma sequência de bytes) e o destinatário recebe essa mensagem.
- Comunicação síncrona e assíncrona.

Desempenho da comunicação

Latência

- É o tempo que decorre entre a execução de uma operação de envio e o início da receção.
- Determinada pelos overheads (software, routing, acesso ao meio).

Taxa de transferência

- É a velocidade a que os dados são transmitidos entre duas máquinas (bits/s).
- Determinada pelas características físicas.

Erros na rede

- Perda de pacotes, maioritariamente devida a buffer overflows (atrasos no processamento nos switches e no destinatário).

Comunicação assíncrona

- No modelo assíncrono, há restrições mínimas em relação ao momento em que as mensagens são entregues e em relação aos passos computacionais.
- No modelo típico cliente-servidor, a interação pode seguir o modelo assíncrono.

Comunicação síncrona

- Num sistema síncrono, os processos executam as operações num determinado tempo, e as mensagens são entregues dentro de um limite de tempo.
- Se adotarmos este modelo, os protocolos são fáceis de desenhar, mas poderão surgir vulnerabilidades no mundo real (no qual poderá haver poucas garantias de tempo).

Comunicação entre processos

Modelo síncrono

- Os processos sincronizam após cada mensagem.
- Tanto o envio como a receção bloqueiam.

Modelo assíncrono

- Neste modelo, o envio não é bloqueante.
- A receção tanto pode ser bloqueante como não-bloqueante.

Num ambiente multi-threaded não há desvantagens em ter receção bloqueante (outra thread pode continuar ativa), sendo também mais simples, pelo que é o mais comum.