

Redes de computadores II

Aula 08 – Camada de Aplicação

Camada de Aplicação

- As aplicações são o motivo para existirem as redes de computadores;
- Nos últimos 40 anos foram criadas inúmeras aplicações:
 - Correio eletrônico;
 - Transferência de arquivos;
 - *World Wide Web (WWW)*;

Camada de Aplicação

- Além dessas aplicações, surgiram outras de enorme sucesso no final do milênio:
 - Mensagens instantâneas;
 - Compartilhamento de arquivos P2P;
 - Aplicações de áudio e vídeo (TV e voz sobre IP);

Princípios de Aplicações em Rede

- O desenvolvimento de aplicações em rede pode ter vários objetivos;
 - Ficar rico;
 - Suprir uma necessidade particular;
 - Agradar o chefe;
- Requisitos:
 - Os programas devem rodar em sistemas e plataformas diferentes;
 - Não desenvolver programas para rodar no núcleo da rede;

Arquiteturas de Aplicações em Rede

- **Cuidado:** A arquitetura de aplicação é diferente da arquitetura de rede;
- As arquiteturas mais utilizadas são:
 - Cliente/Servidor;
 - Ponto – a- Ponto (P2P);

Arquiteturas de Aplicações em Rede

- Cliente/Servidor:
 - Hospedeiro (servidor) responsável por atender as requisições dos clientes;
 - Clientes não se comunicam diretamente;
 - Servidores possuem o endereçamento IP fixo;
 - Aplicações mais conhecidas: Web, FTP, Telnet e e-mail;

Arquiteturas de Aplicações em Rede

- Cliente/Servidor:
 - Nesta arquitetura, muitas vezes um único servidor pode ser incapaz de atender a todas as solicitações;
 - Assim um grande conjunto de hospedeiros (*datacenter*) são utilizados para criar um grande servidor virtual;

Arquiteturas de Aplicações em Rede

- Ponto-a-Ponto (P2P)
 - Comunicação direta entre os pares (dando origem ao nome);
 - A confiança nos servidores é mínima;
 - Muitas aplicações atuais de tráfego intenso são baseadas na arquitetura P2P;
 - Exemplos de Aplicações: eMule, BitTorrent, Skype, PPLive;

Arquiteturas de Aplicações em Rede

- Híbridas
 - Exemplos: Algumas aplicações de mensagens instantâneas, servidores rastreiam o IP dos usuários;
 - As mensagens são enviadas diretamente de usuário para usuário;

Comunicação entre processos

- Processo: programa rodando dentro de uma máquina;
- Na mesma máquina, dois processos se comunicam usando a comunicação entre processos (definida pelo SO).
- Processos em máquinas diferentes se comunicam trocando mensagens;

Comunicação entre processos

- Processo cliente: processo que inicia a comunicação;
- Processo servidor: processo que espera para ser conectado;
- Aplicações com arquiteturas P2P têm processos clientes e processos servidores;

Comunicação entre processos

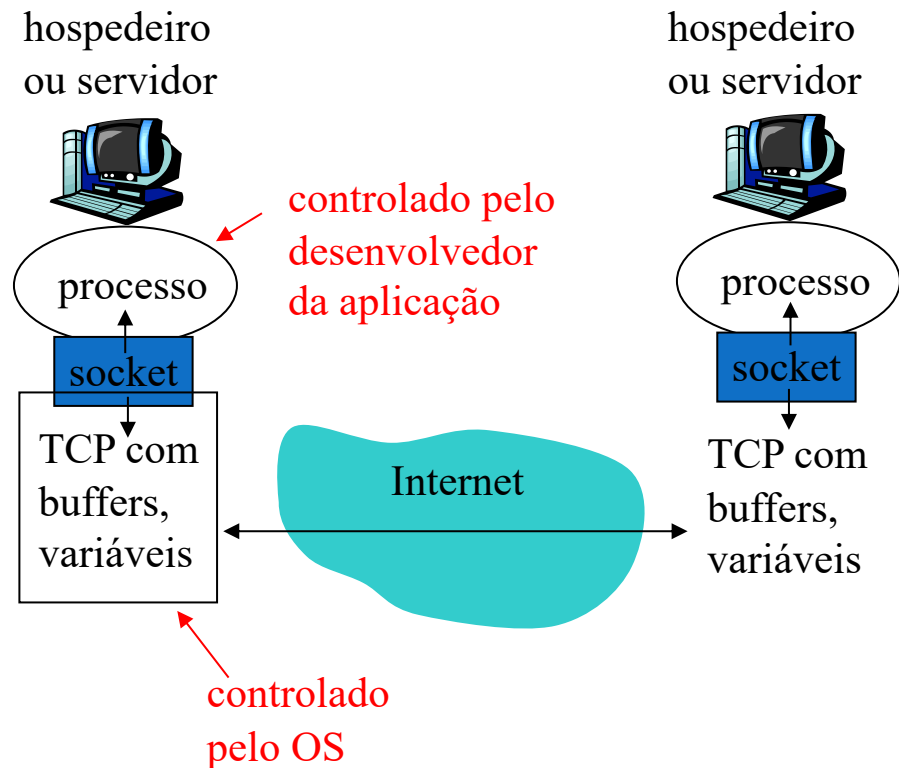
- Interface entre o processo e a rede de computadores:
 - Um processo envia e recebe mensagens pela rede através de uma interface de software denominada de *socket*;
 - Analogia: Processo = casa;
socket = porta;

Comunicação entre processos

- Quando o processo deseja enviar uma mensagem, manda através do *socket*;
- O processo emissor assume que há uma infraestrutura de transporte que leva a mensagem ao *socket* do receptor;
- *Socket* é a interface entre a camada de aplicação e a de transporte dentro de um *host*.

Comunicação entre processos

- *Socket* é a interface de programação (API) pela qual as aplicações de rede são inseridas na internet.



Comunicação entre processos

- O desenvolvedor controla tudo na camada de aplicação através do *socket*;
- Na camada de transporte só tem acesso a:
 - Escolha do protocolo de transporte;
 - Capacidade de consertar alguns parâmetros;
- Dessa forma o desenvolvedor constrói a aplicação usando serviços oferecidos pela camada de transporte;

Endereçando processos

- Para receber mensagens o processo deve ter um identificador;
- Cada dispositivo possui um endereço IP exclusivo de 32 bits;
- Basta o endereço IP da máquina em que o processo é executado para identificar o processo?

Endereçando processos

- Muitos processos podem estar rodando no mesmo hospedeiro;
- *Identificador* inclui endereço IP e números de porta associados ao processo no hospedeiro.
- Exemplos de número de porta:
 - servidor HTTP: 80
 - servidor de correio: 25

Serviços de transporte

- Transferência de dados confiável:
 - Pacotes podem se perder dentro de uma rede;
 - Alguns aplicativos podem tolerar alguma perda;
 - Ex: Áudio e vídeo;
 - Outros, exigem transferência de dados 100% confiável;
 - Ex: Correio Eletrônico, transferência de arquivos, financeiras , etc.

Serviços de transporte

- Vazão:
 - É a taxa à qual o processo pode enviar bits ao processo destino;
 - Como outras sessões compartilharão a largura de banda, a vazão pode variar com o tempo;
 - Algumas aplicações exigem um mínimo de vazão para serem eficazes:
 - Ex: Telefonia por internet, que codifica a voz a uma taxa de 32 bits/s;

Serviços de transporte

- Vazão:
 - Se o protocolo não puder oferecer a vazão, a aplicação precisará codificar os dados a uma taxa menor ou desistir.
 - Essas aplicações são conhecidas como sensíveis à largura de banda;
 - Aplicações elásticas utilizam qualquer vazão que esteja disponível;
 - Ex: Correio eletrônico e transferência de arquivos;

Serviços de transporte

- Temporização:
 - Garantias de temporização;
 - Ex: Cada bit emitido pelo *socket* chega ao destino em menos de 100 milissegundos depois;
 - Utilizados em aplicações interativas em tempo real;
 - Longos atrasos tendem a prejudicar a aplicação;
 - Ex: Jogos e multimídia;

Requisitos de serviços

Tabela de requisitos:

Aplicação	Perda de dados	Vazão	Sensível ao tempo
transf. arquivos	sem perda	elástica	não
e-mail	sem perda	elástica	não
documentos Web	sem perda	elástica	não
áudio/vídeo tempo real	tolerante a perda	áudio: 5 kbps-1 Mbps vídeo: 10 kbps-5 Mbps	sim, centenas de ms
áudio/vídeo armazenado	tolerante a perda	o mesmo que antes	sim, alguns seg
jogos interativos	tolerante a perda	poucos kbps ou mais	sim, centenas de ms
Mensagem instantânea	sem perda	elástica	sim e não

Serviços de transporte

serviço TCP:

- *orientado a conexão*: preparação exigida entre processos cliente e servidor;
- *transporte confiável* :entre processo emissor e receptor;
- *controle de fluxo*: emissor não sobrecarrega receptor ;
- *controle de congestionamento*: regula emissor quando a rede está sobrecarregada;
- *não oferece*: temporização, garantias mínimas de vazão, segurança;

Serviços de transporte

Serviço UDP:

- transferência de dados não confiável entre processo emissor e receptor;
- Não oferece: preparação da conexão, confiabilidade, controle de fluxo, controle de congestionamento, temporização, garantia de vazão ou segurança ;
- Por que se incomodar? Por que existe um UDP?

Definições do protocolo

- Sabemos como os processos se comunicam, mas alguns questionamentos ainda são pertinentes:
 - Como essas mensagens são estruturadas?
 - O que significam os campos nas mensagens?
 - Quando os processos enviam as mensagens?

Definições do protocolo

- Conceito:
- Um protocolo de camada de aplicação define como processos que funcionam em sistemas finais diferentes, se comunicam entre si.

Definições do protocolo

- Tipos de mensagens trocadas,
 - Ex: requisição, resposta
- Sintaxe da mensagem:
 - que campos nas mensagens e como os campos são delineados;
- Semântica da mensagem:
 - significado da informação nos campos;
- Regras de quando e como processos enviam e respondem a mensagens

Aplicações

- HTTP:
 - Página Web consiste em objetos;
 - Objeto pode ser arquivo HTML, imagem JPEG, applet Java, arquivo de áudio, etc;
 - Página Web consiste em arquivo HTML básico que inclui vários objetos referenciados;
 - Cada objeto é endereçável por um URL;

Aplicações

- **FTP:**
 - Protocolo de transferência de arquivos;
 - Transfere arquivos de ou para computador remoto;
 - Modelo cliente/servidor;
 - Cliente: lado que inicia a transferência;
 - Servidor: computador remoto;

Aplicações

- Correio Eletrônico: Formado por três componentes principais:
 - Agentes do usuário ;
 - Servidores de correio;
 - Simple Mail Transfer Protocol: SMTP

Aplicações

- DNS: *Domain Name System*
 - Sistema de nomes de domínios;
 - Traduz o nome para endereço IP;

Aplicações

- Aplicações P2P
 - Distribuição de arquivos;
 - Banco de dados distribuídos;