

# **Computação Paralela e Distribuída**



**Cruzeiro do Sul Virtual**  
Educação a distância



# Material Teórico



Fundamentos Básicos

**Responsável pelo Conteúdo:**

Prof. Esp. Allan Piter Pressi

**Revisão Textual:**

Prof. Esp. Luciano Vieira Francisco





- [Introdução;](#)
- [Exemplos de Sistemas Distribuídos;](#)
- [Tendência em Sistemas Distribuídos;](#)
- [Resumo;](#)
- [Modelos de Sistemas;](#)
- [Resumo;](#)
- [Rede e \*Internetworking\*;](#)
- [Resumo.](#)



### OBJETIVO DE APRENDIZADO

- Conhecer os sistemas distribuídos, seus conceitos essenciais e a computação paralela;
- Compreender algumas definições e seus usos em aplicações distribuídas de sistemas, conceituando o contexto de computação paralela;
- Conhecer os conceitos de rede e mecanismos de proteção e integração de sistemas.





# Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:

Determine um horário fixo para estudar.

Mantenha o foco! Evite se distrair com as redes sociais.

Procure manter contato com seus colegas e tutores para trocar ideias! Isso amplia a aprendizagem.

Seja original! Nunca plágie trabalhos.

Aproveite as indicações de Material Complementar.

Conserve seu material e local de estudos sempre organizados.

Não se esqueça de se alimentar e de se manter hidratado.

## Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu “momento do estudo”;
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.

# Contextualização

Um sistema distribuído pode ser definido como aquele em que os componentes localizados em computadores em rede podem comunicar e coordenar as suas ações através da troca de mensagens. Esta definição leva às seguintes características significativas de sistemas distribuídos: concorrência de componentes e ausência de um mecanismo global de falhas independentes de componentes.

Ao analisarmos exemplos modernos de aplicativos distribuídos, incluindo jogos *on-line web multiplayer*, sistemas de negociação financeira e também analisar as tendências subjacentes que impulsionam os atuais sistemas distribuídos, constatamos as modernas redes de comunicação, o surgimento da computação móvel onipresente, a crescente importância dos sistemas multimídia distribuídos e a tendência para a visualização de sistemas como um produto utilitário.

Recursos podem ser gerenciados por servidores e acessados por clientes, ou podem ser encapsulados como objetos e acessados por outros objetos do cliente.

Os desafios decorrentes da construção de sistemas distribuídos são a heterogeneidade de seus componentes, abertura – permitindo que componentes sejam adicionados ou substituídos –, segurança, escalabilidade – a capacidade de funcionar bem quando a carga ou o número de usuários aumenta –, manipulação de falhas, concorrência de componentes, transparência e fornecimento de qualidade de serviço.

A *web* é discutida como um exemplo de uma grande escala de sistema distribuído e suas principais características são introduzidas.

# Introdução

Redes de computadores estão em todo lugar. A *internet* é uma rede assim como as muitas redes, constituídas por diversos componentes. Redes de telefonia celular, industriais, de fábricas, domésticas etc.; todas, separadamente e em conjunto, partilham das características essenciais que as tornam temas relevantes para estudar sob a ótica de sistemas distribuídos.

O objetivo é explicar as características das redes de computadores que impactam os projetos de sistemas e apresentar as principais técnicas e os conceitos que foram desenvolvidos para ajudar nas tarefas de projetar e implementar sistemas que são baseados nas quais.

Segundo Tanenbaum e Steen (2007), podemos definir sistema distribuído como aquele em que componentes de *hardware* ou *software* localizados em computadores em rede se comunicam e coordenam as suas ações apenas na troca de mensagens.

Essa simples definição abrange toda a gama de sistemas que podem ser implementados em computadores em rede.

Computadores conectados por uma rede podem ser espacialmente separados por distância, ou seja, estarem em continentes separados, no mesmo prédio ou na mesma sala.

A definição de sistemas distribuídos tem as seguintes características:

- **Concorrência:** é definido como a capacidade de execução simultânea de programas, capacidade do sistema para lidar com recursos compartilhados;
- **Cooperação:** os programas cooperam e coordenam as suas ações de troca de mensagens, compartilhamento do tempo e sincronização de seus objetos;
- **Falhas independentes:** computadores podem falhar e é de responsabilidade dos projetistas de sistemas planejar as consequências de possíveis falhas.

Sistemas distribuídos podem falhar na rede, gerando comportamentos diferentes aos sistemas interconectados, desde o isolamento à perda de performance ou erros, de modo que cada componente do sistema pode falhar independentemente, deixando os outros ainda em execução.

## Exemplos de Sistemas Distribuídos

Como mencionado, as redes estão em toda parte e sustentam muitos serviços cotidianos, tais como *internet*, pesquisas na *web*, jogos *on-line*, *e-mail*, redes sociais, *e-commerce* etc.

Os sistemas distribuídos abrangem muitos desenvolvimentos tecnológicos e, portanto, a compreensão da tecnologia é necessária ao conhecimento da computação moderna.

Nesse sentido, Google desenvolve um esforço significativo na infraestrutura de sistema distribuído para apoiar a pesquisa e representa uma das maiores e mais complexas instalações de sistemas distribuídos na história da computação. Os destaques dessa infraestrutura incluem:

- Uma infraestrutura física com um grande número de computadores em rede espalhados em data centers em todo o mundo;
- Um sistema projetado para suportar arquivos muito grandes e altamente otimizado ao estilo de uso exigido pela pesquisa e por outros aplicativos do Google.

Vejamos alguns exemplos de sistemas distribuídos para melhor ilustrar a atual diversidade e complexidade.

## Busca na Web

---

A busca na *web* emergiu como uma grande indústria de crescimento na última década, com recentes números que indicam que o número global de buscas subiu para mais de dez bilhões por ano.

## Finanças e E-commerce

---

O crescimento do *e-commerce* pode ser exemplificado pelo uso de sites de comércio eletrônico e com seu uso integrado com organizações que suportam pagamento via *internet*, integrando as empresas de cartões de crédito.

## Sociedade da Informação

---

O crescimento da *internet* como repositório de informações e conhecimento; o desenvolvimento de mecanismos de busca na *web* como o Google e Yahoo para pesquisar esse vasto repositório; o surgimento de bibliotecas digitais e a digitalização em larga escala de fontes de informação legadas tais como livros – por exemplo, o **Google Books** –; o aumento da importância do conteúdo gerado pelo usuário por meio de sites como **YouTube**, **Wikipedia** e **Flickr**; o surgimento de redes sociais por meio de serviços como **Facebook**, **Twitter** e **MySpace**.

## Educação

---

O emergente mercado de educação a distância através da *web* tem sido um grande ponto de compartilhamento e integração entre diferentes recursos tecnológicos.

## Ciência

---

O surgimento do *grid* como uma tecnologia fundamental para a e-ciência, incluindo o uso de redes complexas de computadores para apoiar o armazenamento, a análise e o processamento de – comumente, grandes quantidades de – dados científicos; o uso associado do *grid* a uma tecnologia para a colaboração mundial entre grupos de cientistas.

# Tendência em Sistemas Distribuídos

Sistemas distribuídos passam por mudanças e com isso podem surgir novas oportunidades de inéditas tendências e negócios, alguns que veremos a seguir.

## Uma Nova Internet

---

A nova *internet* permitirá uma vasta coleção de redes interconectadas, com ampla gama de tecnologias de comunicação e redes de terceira geração.

Programas em execução no sistema operacional dos computadores conectados ao qual interagem passando mensagens, empregando um meio comum de comunicação. O *design* e a construção da comunicação pela *internet* são grandes conquistas técnicas, permitindo programas em execução.

A função de um *firewall* é proteger, impedindo que conexões não autorizadas saiam ou entrem através da rede ou sistemas, sendo implementado filtrando as mensagens recebidas e enviadas; em uma nova *internet* esse recurso passa a ser onipresente nos diferentes dispositivos existentes.

Os *firewalls* também podem ser problemáticos em sistemas distribuídos, impedindo o acesso legítimo a serviços quando é necessário o compartilhamento de recursos entre usuários internos e externos.

A implementação da *internet* e dos serviços que essa suporta implicou no desenvolvimento de soluções práticas para muitos problemas do sistema distribuído.

## Computação Móvel

---

Os avanços tecnológicos na miniaturização de dispositivos e redes sem fio levaram cada vez mais à integração de dispositivos computacionais pequenos e portáteis em sistemas distribuídos. Esses dispositivos incluem:

- Computadores e aparelhos portáteis e inteligentes, tais como *smartphones*, câmeras digitais etc.;
- Dispositivos embutidos em eletrodomésticos, tais como máquinas de lavar roupa, carros, geladeiras, televisores e casas inteligentes.

A presença de computadores em todos os lugares se torna útil apenas quando podem se comunicar entre si. Por exemplo, pode ser conveniente para os utilizadores controlarem a sua máquina de lavar roupa ou o seu sistema de entretenimento a partir de seu telefone ou de um dispositivo de “controle remoto universal” da casa.

## As Intranets estão Conectadas ao Resto da Internet

---

O usuário tem acesso a três formas de conexão sem fio: os computadores pessoais podem se conectar à LAN sem fio, oferecendo cobertura de algumas centenas de metros – no andar de um prédio, digamos. Conecta-se ao restante da *intranet*

do *host* por meio de um *gateway* ou ponto de acesso. O usuário também possui um telefone celular – *smartphone* – conectado à *internet*, o qual dá acesso à *web* e a outros serviços da *internet*, restrito apenas ao que pode ser apresentado em sua pequena tela, podendo fornecer também informações de localização através da funcionalidade integrada e denominada *Global Positioning System (GPS)*.

Com uma infraestrutura adequada de sistema, o usuário pode realizar diferentes tarefas em distintos recursos usando os dispositivos que esses carregam. Esse tipo de cenário demonstra a necessidade de suportar as associações entre diferentes dispositivos.

O principal desafio que se aplica a essas situações é tornar a interoperação rápida e confiável, mesmo que o usuário esteja em um ambiente que talvez nunca tenha visitado. Significa permitir que o dispositivo do visitante se comunique na rede do *host* e associe tal dispositivo a serviços locais adequados.

## Sistemas Multimídia Distribuídos

---

Outra tendência importante é o requisito de suporte a serviços multimídia em sistemas distribuídos. O suporte multimídia pode ser definido como a capacidade de suportar uma variedade de tipos de mídia de maneira integrada.

Os benefícios da computação multimídia distribuída são consideráveis, pois uma ampla gama de novos serviços e aplicativos – multimídia – podem ser fornecidos na área de trabalho, incluindo o acesso a transmissões televisivas ao vivo ou pré-gravadas, o acesso a bibliotecas de filmes e oferta de serviços de vídeo sob demanda, acesso a bibliotecas de música, fornecimento de instalações de áudio e videoconferências e recursos de telefonia integrados, incluindo telefonia IP ou tecnologias relacionadas, tal como o **Skype**, uma alternativa ponto a ponto à telefonia IP.

## Computação Distribuída como um Utilitário

---

Com a crescente maturidade da infraestrutura de sistemas distribuídos, várias empresas têm promovido a visão de recursos distribuídos como mercadoria ou utilidade, desenhando a analogia entre recursos distribuídos e outras utilidades, tais como água ou eletricidade. Com esse modelo, os recursos são oferecidos por fornecedores de serviços apropriados e alugados, em vez de serem de propriedade do usuário final – tal modelo se aplica a recursos físicos e a serviços. Isso oferece maior flexibilidade ao fornecedor de serviços em termos de gerenciamento de recursos, tais como serviços de *software* e computação em nuvem.

## Compartilhamento de Recursos

---

Os usuários estão tão acostumados com os benefícios do compartilhamento de recursos que podem facilmente ignorar o seu significado. Compartilhamos rotineiramente recursos de *hardware* – tais como impressoras –, recursos de dados – tais como arquivos – e recursos com funcionalidades mais específicas – tais como mecanismos de pesquisa.

Enquanto desafio, o escopo dos sistemas distribuídos é resolver os problemas que surgem em computação distribuída.

## Heterogeneidade

A *internet* permite que os usuários acessem serviços e executem aplicativos em um ambiente heterogêneo, uma coleção de computadores e redes. Heterogeneidade – isto é, variedade e diferença – aplica-se a todos os itens, tais como redes, *hardwares* de computador, sistemas operacionais, linguagens de programação e implementações de diferentes desenvolvedores.

## Abertura

A abertura de um sistema de computador é a característica que determina se o sistema pode ser estendido e reimplementado de várias maneiras. A abertura de distribuição de sistemas é determinada principalmente pelo grau em que novos serviços de compartilhamento de recursos podem ser adicionados e disponibilizados para uso por uma variedade de programas clientes.

## Segurança

Muitos dos recursos de informação que são disponibilizados e mantidos em sistemas têm alto valor intrínseco para os seus usuários. Sua segurança é, portanto, de importância considerável. Segurança para recursos de informação possui três componentes: confidencialidade, integridade e disponibilidade.

Ademais, em segurança algumas questões devem ser resolvidas:

- **Ataques de negação de serviço:** outro problema de segurança é que um usuário pode desejar interromper um serviço por algum motivo. Isto pode ser alcançado bombardeando o serviço com um número tão grande de solicitações inúteis que os usuários sérios são também incapacitados de usarem o serviço – condição chamada de ataque de negação de serviço;
- **Segurança do código móvel:** o código para dispositivos móveis precisa ser manuseado com cuidado. Considere alguém que recebe um arquivo executável como um anexo de *e-mail*: os possíveis efeitos da execução do programa são imprevisíveis – por exemplo, pode parecer exibir uma imagem interessante, mas, na realidade, acessa recursos locais, ou talvez faça parte de um ataque de negação de serviço.

## Escalabilidade

Os sistemas distribuídos operam de forma eficaz e eficiente em diferentes escalas, variando de uma pequena *intranet* para a *internet*. Assim, um sistema é descrito como escalável se permanecer eficaz quando há aumento significativo no número de recursos e de usuários.

O *design* de sistemas distribuídos escalonáveis apresenta os seguintes desafios:

- Custos de recursos físicos;
- Perdas e gargalos de desempenho;
- Impedir que os recursos de *software* esgotem o dispositivo.

## Manipulação de Falhas

Sistemas de computadores, por vezes, falham; de modo que quando ocorrem falhas no *hardware* ou *software*, os programas podem produzir resultados incorretos ou pararem antes da conclusão da computação. Assim, algumas técnicas podem ser implementadas para tratar essas falhas, vejamos:

- Detecção de falhas;
- Tolerância a falhas;
- Recuperação de falhas;
- Redundância.

Ademais, sistemas distribuídos fornecem alto grau de disponibilidade em face as falhas de *hardware*.

A disponibilidade de um sistema é uma medida da proporção de tempo que está disponível para usar. Assim, quando um dos componentes de um sistema distribuído falhar, somente o trabalho que foi usado no componente com falha é afetado. Um usuário pode mudar para outro computador se aquele que estava usando falhar; de modo que um processo do servidor pode ser iniciado em outro computador.

## Concorrência

Serviços e aplicativos fornecem recursos que podem ser compartilhados pelos clientes em um sistema distribuído. Existe, portanto, a possibilidade de que vários clientes tentem acessar um recurso compartilhado ao mesmo tempo – por exemplo, uma estrutura de dados que regista lances para um leilão pode ser acessada com muita frequência quando se chega perto do prazo final.

## Transparência

Transparência é definida como a ocultação do usuário e aplicativo programador da separação de componentes em um sistema distribuído, a fim de que o sistema seja percebido como um todo – e não como uma coleção de componentes independentes. As implicações da transparência possuem grande influência no *design* do *software* do sistema.

## Qualidade de Serviço

Uma vez que os usuários recebam a funcionalidade que necessitam de um serviço, tal como o serviço de arquivo em um sistema distribuído, podemos continuar a perguntar sobre a qualidade do serviço que o forneceu. As principais propriedades não funcionais dos sistemas que afetam a qualidade do serviço experimentado por clientes e usuários são: confiabilidade, segurança e desempenho.

Questões de confiabilidade e segurança são críticas no projeto da maioria dos sistemas de computador. Na verdade, a abreviação QoS – *Quality of Service* – foi definida para se referir à capacidade de os sistemas cumprirem tais prazos. Sua conquista depende da disponibilidade dos recursos de computação e de rede necessários nos momentos apropriados, o que implica um requisito para que o sistema forneça computação e comunicação garantidas enquanto recursos suficientes para permitir que os aplicativos concluam cada tarefa no prazo – por exemplo, exibir um quadro de vídeo.

## Resumo

Sistemas distribuídos estão em toda parte. A *internet* permite que usuários em todo o mundo acessem os seus serviços onde quer que estejam localizados. Cada organização gerencia uma *intranet*, esta que fornece serviços locais e de *internet* para usuários locais, comumente proporcionando serviços a outros usuários na *internet*. Pequenos sistemas distribuídos podem ser construídos a partir de computadores móveis e outros pequenos dispositivos computacionais conectados a uma rede sem fio.

O compartilhamento de recursos é o principal fator motivador para a construção de sistemas. Recursos como impressoras, arquivos, páginas da *web* ou registros de banco de dados são gerenciados por servidores do tipo apropriado. Por exemplo, os servidores da *web* gerenciam páginas da *web* e outros recursos também da *web*. Ademais, recursos são acessados por clientes – outro exemplo, os clientes da *web* servidores são geralmente chamados de navegadores.

A construção de sistemas distribuídos produz muitos desafios: heterogeneidade, abertura, segurança, escalabilidade, manipulação de falhas, concorrência, transparência e qualidade de serviço.

## Modelos de Sistemas

### Introdução

---

Os sistemas que são destinados ao uso em ambientes do mundo real devem ser projetados para funcionarem corretamente na maior variedade possível de circunstâncias e em face de muitas dificuldades e ameaças potenciais. A discussão e os exemplos já mencionados sugerem que sistemas distribuídos de diferentes tipos compartilham importantes propriedades subjacentes e dão origem a problemas comuns de projeto. Assim, veremos como as propriedades e os problemas de *design* de sistemas distribuídos podem ser capturados e discutidos pelo uso de modelos descritivos.

Cada tipo de modelo destina-se a fornecer uma descrição abstrata, simplificada, mas consistente de um aspecto relevante do *design* de sistema distribuído; por exemplo, os modelos:

- **Físicos** são a maneira mais explícita de descrever um sistema, pois capturam a composição de *hardware* de um sistema em termos de computadores – e outros dispositivos, tais como telefones celulares – e suas redes de interconexão;
- **Arquiteturais** descrevem um sistema em termos de suporte computacional e tarefas de comunicação realizadas pelos seus elementos computacionais – o computacional constitui elementos como computadores individuais ou agregados destes e suportados por interconexões de rede;
- **Fundamentais** adotam uma perspectiva abstrata para examinar aspectos de um sistema distribuído. Considere, por exemplo, os modelos fundamentais que contemplam três aspectos importantes de sistemas distribuídos: modelos de interação, abrangendo a estrutura e o sequenciamento da comunicação entre os elementos do sistema; modelos de falha, que consideram as maneiras pelas quais um sistema pode falhar ou operar corretamente; e modelos de segurança, que entendem como o sistema é protegido contra tentativas de interferir com o seu correto funcionamento ou de roubar os seus dados.

## Modelos Físicos

---

Modelo físico é uma representação dos elementos de *hardware* subjacentes de um sistema distribuído que abstrai significativamente os detalhes específicos do computador e tecnologias de rede empregadas.

Além desse modelo de referência, podemos identificar três gerações de sistemas, vejamos:

- Início dos sistemas distribuídos, os quais surgiram no final da década de 1970 e início da de 1980 em resposta ao surgimento de tecnologia de rede local;
- **Sistemas distribuídos em escala de internet:** com base nessa fundação, outros sistemas começaram a surgir na década de 1990 em resposta ao crescimento dramático da *internet* durante esse período;
- **Sistemas distribuídos na contemporaneidade:** os nós são tipicamente computadores em padrão *desktop* e, portanto, relativamente estáticos, discretos e autônomos.

## Arquitetura do Modelo

---

A arquitetura de um sistema é a sua estrutura em termos de componentes especificados separadamente e suas inter-relações. O objetivo geral é garantir que a estrutura atenda a demandas atuais e prováveis no futuro.

As principais preocupações são tornar o sistema confiável, gerenciável, adaptável e rentável. O projeto arquitetônico de um edifício tem aspectos semelhantes – determina não apenas a sua aparência, mas também a sua estrutura geral e o estilo arquitetônico – gótico, neoclássico, moderno – e fornece um quadro consistente de referência ao projeto.

## Componentes da Arquitetura

Para entender os blocos de construção fundamentais de um sistema distribuído, torna-se necessário considerar quatro questões-chave:

1. Quais são as entidades que estão se comunicando no sistema distribuído?
2. Como se comunicam ou, mais especificamente, qual paradigma de comunicação é usado?
3. Quais são os papéis e as responsabilidades que têm na arquitetura?
4. Como são mapeados para a infraestrutura distribuída física – qual é a sua colocação?

### Entidades Comunicantes

As duas primeiras questões são absolutamente centrais para a compreensão de sistemas distribuídos; o que se comunica e como essas entidades o fazem em conjunto a fim de definir um rico espaço de *design* ao desenvolvedor de sistemas distribuídos.

Objetos foram introduzidos para permitir e encorajar o uso de abordagens orientadas em sistemas distribuídos.

Desde a sua introdução, os componentes sofreram vários problemas significativos, os quais identificados com objetos distribuídos, de modo que o uso da tecnologia de componentes emergiu como uma resposta direta a essas fraquezas. Componentes se assemelham a objetos que oferecem abstrações orientadas a problemas para a construção de sistemas distribuídos e também acessados através de interfaces.

Os serviços da *web* representam o terceiro paradigma importante ao desenvolvimento de sistemas distribuídos, estando estreitamente relacionados a objetos e componentes, novamente adotando uma abordagem baseada no encapsulamento de comportamento e acesso através de interfaces.

Os protocolos de solicitação-resposta são um padrão imposto em um serviço subjacente de transmissão de mensagens para suportar a computação cliente-servidor.

O conceito de uma Chamada de Procedimento Remoto (RPC) representa um grande avanço intelectual na computação distribuída; afinal, no RPC procedimentos em processos em computadores remotos podem ser chamados como fossem procedimentos no espaço de endereço local.

A Invocação Remota de Método (RMI) é muito semelhante às chamadas de procedimento remoto, mas em um mundo de objetos distribuídos. Com essa abordagem, um objeto de chamada pode invocar um método em um objeto remoto. Tal como acontece com o RPC, as implementações de RMI podem, no entanto, apoiar a identidade do objeto e a capacidade associada para passar objetos identificadores como parâmetros em chamadas remotas.

A comunicação diz respeito à entrega de mensagens para um conjunto de destinatários e, portanto, é um paradigma de comunicação multipartidária em suporte de um para muitos.

Muitos sistemas, como o exemplo de negociação financeira, podem ser classificados como de disseminação de informação, em que o número de produtores – ou editores – distribui itens de informação de interesse – eventos – para um número similarmente grande de consumidores – ou assinantes. Seria complicado e ineficiente para empregar qualquer um dos principais paradigmas de comunicação aqui discutidos para esse fim. Portanto, sistemas de publicação-assinatura surgiram para atender a essa importante necessidade.

Considerando que os sistemas oferecem um estilo de comunicação de um para muitos, as filas de mensagens fornecem um serviço ponto a ponto em que o produtor de processos pode enviar mensagens para uma fila específica e os processos do consumidor podem receber mensagens da fila ou serem notificados da chegada de novas mensagens na fila. As filas, portanto, possibilitam uma direção entre produtor e consumidor de processos.

Serviços de comunicação indireta podem apoiar um modelo em que os processos podem colocar itens arbitrários de dados estruturados, chamados de tuplas.

Sistemas de Memória Compartilhada Distribuída (DSM) fornecem uma abstração para compartilhar dados entre processos que não compartilham memória física.

Em um sistema distribuído que processa objetos, componentes ou serviços, incluindo os da web, figura a interação para executar uma atividade como, por exemplo, suportar uma sessão de *chat*. Ao fazê-lo, os processos assumem funções, de modo que esses papéis são fundamentais para estabelecer a arquitetura geral a ser adotada.

## Padrões de Arquitetura

Padrões arquiteturais construídos sobre os elementos aqui discutidos fornecem estruturas compostas para funcionarem em dadas circunstâncias. Não são necessariamente soluções completas, mas oferecem percepções parciais que, quando combinadas a outros padrões, levam o projetista à solução de um determinado domínio do problema.

O conceito de camadas é familiar e está intimamente relacionado à abstração. Assim, em uma abordagem em camadas, um sistema complexo é partitionado em várias dessas camadas, com determinada camada fazendo uso dos serviços oferecidos pela camada abaixo, a qual fornece, portanto, uma abstração de *software*, com camadas mais altas desconhecendo detalhes de implementação, ou mesmo de quaisquer outras camadas abaixo das quais. Em termos de sistemas distribuídos, isso equivale à organização vertical de serviço em camadas também de serviço.

Quadro 1 – Sistemas distribuídos (camadas de *software* e *hardware*)



Uma plataforma para sistemas e aplicativos distribuídos consiste no nível mais baixo de camadas de *hardware* e *software*. Assim, camadas de baixo nível fornecem serviços às camadas acima, sendo implementados de forma independente em cada computador.

O *middleware* é definido como uma camada de *software* cujo propósito é mascarar a heterogeneidade, além de fornecer um modelo conveniente para programadores de aplicativos.

Arquitetura em camadas é uma técnica para organizar a funcionalidade de uma determinada camada e colocar essa funcionalidade em servidores apropriados e, como uma consideração secundária, aos nós físicos.

Consideraremos a implementação de tal aplicativo usando a tecnologia cliente-servidor, de modo que as soluções de duas e três camadas associadas são apresentadas juntas e respectivamente. A vantagem desse esquema é a sua baixa latência em termos de interação, com apenas uma troca de mensagens para invocar uma operação. Já a desvantagem diz respeito à divisão da lógica de aplicação por meio de um limite de processo, com a consequente restrição de quais partes da lógica podem ser invocadas diretamente a partir de outra parte.

Na solução de três camadas há um mapeamento de um para um dos elementos lógicos para servidores físicos. Portanto e por exemplo, a lógica da aplicação é mantida em um lugar, por sua vez, podendo melhorar a manutenção do *software*. Cada camada também tem bem definida a função; outro exemplo: o terceiro nível é simplesmente um banco de dados que oferece uma – potencialmente padronizada – interface de serviço relacional. As desvantagens correspondem às complexidades de gerenciar três servidores e também o tráfego de rede adicionado e a latência associada a cada operação.

Note que essa abordagem generaliza para soluções de **n** camadas – ou multicamadas –, onde determinado domínio de aplicação é partitionado em **n** elementos lógicos, cada qual mapeado para dado elemento do servidor.

A tendência da computação distribuída é direcionar a complexidade para longe desde o dispositivo do usuário final até os serviços na *internet*. Isso é mais aparente no avançar para a computação em nuvem, mas também pode ser vista em camadas de arquitetura – como já discutido.

Tal conceito levou ao surgimento da Computação em Rede Virtual (VNC). O sentido inicial evoluiu para implementações como o **RealVNC**, que é uma solução de *software*, e o **Adventiq**, que é uma solução baseada em *hardware* que suporta a transmissão de eventos de teclado, vídeo e mouse sobre IP – KVM-over-IP. Outro VNC de implementação inclui Apple Remote Desktop, TightVNC e Aqua Connect.



Acesse o site do RealVNC e do Adventiq. Disponíveis em:

- RealVNC: <https://goo.gl/mmqUVk>;
- Adventiq: <https://goo.gl/rUAuwE>.

Embora seja direta, a implicação é que os usuários podem acessar as suas instalações de computação de qualquer lugar em uma ampla gama de dispositivos, representando um passo no sentido da computação móvel.

Ademais, a computação em rede virtual substituiu os computadores da rede, tentando realizar soluções *thinclient* por meio de dispositivos de *hardware* simples e baratos, os quais completamente dependentes de serviços em rede, baixando o seu sistema operacional e qualquer *software* de aplicativo necessário ao usuário de um servidor de arquivos remotos.

Desde os dados de aplicativos, o código é armazenado por um servidor de arquivos, de modo que os usuários podem migrar de um computador de rede para outro. Na prática, a computação em rede virtual provou ser uma solução mais flexível, razão pela qual atualmente domina o mercado.

Já *proxy* é um padrão comumente recorrente em sistemas distribuídos e projetados particularmente para apoiar a transparência da localização em chamadas de procedimento remoto ou invocação de método.

## Soluções de *Middleware*

A tarefa do *middleware* é fornecer um nível mais alto na abstração de programação ao desenvolvimento de sistemas distribuídos e, por meio de camadas, abstrair sobre a heterogeneidade na infraestrutura subjacente para promover a interoperabilidade e portabilidade.

Além de programar abstrações, o *middleware* também pode fornecer serviços de sistema distribuído de infraestrutura para uso em programas aplicativos ou outros serviços, os quais, se de infraestrutura, fortemente ligados à programação distribuída pelo modelo que o *middleware* fornece.

Inúmeros aplicativos distribuídos dependem inteiramente dos serviços fornecidos pelo *middleware* para suportar as suas necessidades de comunicação e compartilhamento de dados.

Muito foi conseguido na simplificação da programação de sistemas distribuídos através do desenvolvimento de suporte de *middleware*, mas alguns aspectos da confiabilidade de sistemas requerem suporte no nível do aplicativo.

Considere, então, a transferência de grandes mensagens de correio eletrônico do *host* de correio do remetente ao destinatário. À primeira vista, trata-se de uma aplicação simples dos dados TCP do protocolo de transmissão. Mas pondere sobre o problema de um usuário que venha a tentar transferir um arquivo muito grande em uma rede potencialmente não confiável. Neste caso, TCP fornece alguma detecção e correção de erros, mas não pode recuperar interrupções da rede principal. Portanto, o serviço de transferência de *e-mail* adiciona outro nível de tolerância a falhas, mantendo um registro do progresso e retomando a transmissão usando um novo TCP conexão se o original “quebrar”.

## Modelos Fundamentais

---

Todos os modelos até então mencionados, significativamente diferentes entre si, compartilham, contudo, algumas propriedades fundamentais; em particular, todos são compostos de processos que se comunicam entre si no envio de mensagens por meio de uma rede de computadores. Ademais, todos os modelos compartilham o *design* e requisitos para alcançar as características de desempenho e confiabilidade dos processos e redes a fim de garantir a segurança dos recursos do sistema.

### Modelo de Falha

Em um sistema distribuído, ambos os processos e canais de comunicação podem falhar – isto é, podem se afastar do que é considerado como comportamento correto ou desejável, definindo as maneiras pelas quais a falha pode ocorrer, a fim de fornecer uma compreensão dos seus efeitos.

Canais de comunicação podem sofrer falhas como, por exemplo, mensagens cujos conteúdos possam estar corrompidos, mensagens inexistentes que possam ser entregues, ou mesmo mensagens reais que possam ser entregues mais de uma vez.

### Modelo de Segurança

O compartilhamento de recursos é um fator motivador para sistemas, de modo que o modelo arquitetural fornece a base ao nosso padrão de segurança, ou seja, a condição protegida de um sistema distribuído pode ser obtida resguardando os processos e canais utilizados para as suas interações, assim como os objetos que forem encapsulados contra acesso não autorizado.

As ameaças de um inimigo em potencial incluem agravos a processos e canais de comunicação.

Como um servidor pode receber invocações de muitos clientes diferentes, não pode necessariamente determinar a identidade do principal por trás de qualquer invocação particular.

Assim, um inimigo pode copiar, alterar ou injetar mensagens enquanto viajar através da rede e de seus *gateways* intervenientes. Tais ataques representam uma ameaça à privacidade e integridade das informações à medida que viajam pela rede e à integridade do sistema em si.

Todas essas ameaças podem ser derrotadas pelo uso de canais seguros porque baseados em criptografia e autenticação – descritos a seguir.

Suponha que um par de processos – por exemplo, um cliente e um servidor específicos – compartilhe um segredo; isto é, ambos conhecem o segredo, contudo, nenhum outro processo no sistema distribuído sabe disso. Então, se uma mensagem trocada por esse par de processos incluir informações que comprovem o conhecimento do remetente sobre o segredo compartilhado, o destinatário saberá com certeza que o remetente era o outro processo no par – claro, tomando-se o devido cuidado para garantir que o segredo compartilhado não seja revelado a um inimigo.

A criptografia é a Ciência de manter as mensagens seguras, embaralhando-as de modo a esconder o conteúdo. Assim, a criptografia moderna é baseada em algoritmos que usam chaves secretas – grandes números que são difíceis de adivinhar –, transformando dados de uma maneira que possam ser revertidos apenas com o conhecimento da chave de descriptografia correspondente.

O uso de segredos compartilhados e criptografia fornece a base para a autenticação de mensagens – provando as identidades fornecidas por seus remetentes. A técnica básica de autenticação é incluir em uma mensagem uma parte criptografada que contenha o suficiente do conteúdo da mensagem para garantir a sua autenticidade, condição esta que parte de uma solicitação para um servidor a fim de ler parte de um arquivo, por exemplo, incluindo a representação da identidade do principal solicitante, do arquivo e a data e hora da solicitação, todos esses criptografados com uma chave secreta compartilhada entre o servidor de arquivos e o processo de solicitação. O servidor, então, descriptografará e verificará se corresponde aos detalhes não criptografados especificados na solicitação.

Criptografia e autenticação são usadas para criar canais seguros como camadas sobre os serviços existentes de comunicação. Assim, um canal seguro é a comunicação conectando um par de processos, cada qual agindo em nome de um principal; de modo que um canal seguro tem as seguintes propriedades:

- Cada um dos processos possui confiabilidade quanto à identidade do principal cujo nome o outro processo é executado;
- Um canal seguro garante a privacidade e integridade – proteção contra adulteração – dos dados transmitidos através do qual;

- Cada mensagem inclui um registro físico ou lógico de data e hora para evitar repetição ou reordenação.

Canais seguros se tornaram uma importante ferramenta prática para garantir o comércio eletrônico e a proteção da comunicação – Redes Privadas Virtuais (VPN) e o protocolo *Secure Sockets Layer* (SSL) são instâncias desses requisitos de segurança.

Outras possíveis ameaças de um inimigo podem ser os ataques de negação de serviço e implantação de código móvel.

## Resumo

Os sistemas distribuídos são cada vez mais complexos em termos de suas características físicas subjacentes; por exemplo, em termos de escala de sistemas, o nível de heterogeneidade inerente a tais sistemas e as reais demandas para fornecer soluções de ponta a ponta em termos de propriedades como segurança. Isso aumenta a importância em ser capaz de entender e raciocinar sobre sistemas distribuídos em termos de modelos. Assim, consideramos os modelos físicos subjacentes com um exame aprofundado dos modelos arquitetônicos e fundamentais que sustentam sistemas distribuídos.

Ademais, vimos uma abordagem para descrever sistemas distribuídos em termos de um modelo arquitetônico abrangente que faz sentido desse espaço de *design*, examinando as questões centrais do que é informado e como essas entidades se comunicam, condição complementada pela consideração das funções que cada elemento pode desempenhar juntamente com estratégias apropriadas de colocação, dada a infraestrutura física distribuída.

Foi também introduzido o papel fundamental dos padrões arquiteturais ao possibilitar projetos construídos a partir dos elementos principais subjacentes, tal como o cliente-servidor modelo destacado nos principais estilos de *middleware* de suporte, incluindo soluções baseadas em objetos distribuídos, componentes, serviços *web* e na distribuição de eventos.

Em termos de modelos de arquitetura, a abordagem cliente-servidor é predominante – a *web* e outros serviços da *internet*, tais como *File Transfer Protocol* (FTP), notícias e correio, bem como serviços *web* e o *Domain Name System* (DNS) baseado nesse padrão, assim como o arquivamento e outros serviços locais. Voltando ao DNS, este possui um grande número de usuários e gerencia uma considerável quantidade de informações em vários servidores, usando a partição e replicação de dados para aumentar a disponibilidade e tolerância ao erro. Já o *cache* de clientes e servidores *proxy* é amplamente empregado para aprimorar o desempenho de um serviço.

No entanto, atualmente existe ampla variedade de abordagens para modelagem de sistemas distribuídos, incluindo filosofias alternativas, tais como *peer-to-peer*, computação e suporte para abstrações orientadas a problemas, considerando objetos, componentes ou serviços.

O modelo arquitetônico é complementado por padrões fundamentais que auxiliam no raciocínio sobre propriedades do sistema distribuído em termos de, por exemplo, desempenho, confiabilidade e segurança. Em particular, vimos modelos de interação, falha e segurança, os quais identificam as características comuns dos componentes básicos.

A partir do mercado os sistemas são construídos; de modo que o modelo de interação está preocupado com o desempenho de processos e canais de comunicação e com a ausência de um relógio global. Trata-se de identificar um sistema síncrono como aquele em que os limites podem ser o tempo livre de execução do processo, o tempo de entrega da mensagem e o desvio do relógio. Sinaliza ainda um sistema assíncrono como aquele que não tem nenhum limite, podendo ser alcançado na execução do processo no tempo de entrega da mensagem e no desvio do relógio – que é uma descrição do comportamento da *internet*.

Esse modelo classifica como falhas os processos e de comunicação de canais em um sistema distribuído. Já o mascaramento é uma técnica mais confiável, dado que o serviço é realizado a partir de um ano, mascarando algumas das falhas que exibe. Em particular, um serviço de informação confiável pode ser construído a partir de uma comunicação básica, canal, mascarando as suas falhas – por exemplo, suas falhas de omissão podem ser mascaradas pela retransmissão das mensagens perdidas.

Integridade é uma propriedade de comunicação confiável – exige que uma mensagem seja idêntica a outra enviada e que seja disparada duas vezes. A resistência é outra propriedade, dado que o buffer de saída pode ser solucionado pelo de mensagem recebida.

O modelo de segurança identifica as ameaças aos processos e comunicação de canais em um sistema abertamente distribuído. Outro problema de segurança é a autenticação principal – do usuário ou servidor – em uma mensagem que foi enviada, tal como existe uma criptografia para garantir privacidade de par a par de princípios de comunicação.

## Rede e *Internetworking*

Sistemas distribuídos usam redes LAN e WAN para comunicação. O desempenho, a confiabilidade, escalabilidade, mobilidade e qualidade de serviço são características das redes subjacentes que impactam o comportamento de sistemas distribuídos e, portanto, afetam o seu *design*. Assim, mudanças nos requisitos do usuário resultam no surgimento de redes sem fio e de alto desempenho com garantida qualidade de serviço.

Os princípios nos quais as redes de computadores são baseadas incluem camadas de protocolo, comutação de pacotes, roteamento e *streaming* de dados, de modo que técnicas de *internetworking* permitem redes heterogêneas a serem integradas. A *internet* é o maior exemplo; seus protocolos são quase universalmente utilizados em sistemas distribuídos.

Os esquemas de endereçamento e roteamento usados na *internet* resistiram ao impacto de seus enormes crescimentos, passando agora por revisão para acomodarem o crescimento futuro e para atenderem à nova aplicação de requisitos de mobilidade, segurança e qualidade de serviço.

O design de tecnologias específicas de rede é ilustrado em três estudos de caso: **ethernet**, **IEEE 802.11 – Wi-Fi** – e **rede sem fio bluetooth**.

## Introdução

---

As redes empregadas em sistemas distribuídos são construídas a partir de uma variedade de meios de transmissão, incluindo fio, cabo, fibra e canais sem fio; dispositivos de *hardware*, contemplando roteadores, *switches*, pontes, *hubs*, repetidores e interfaces de rede; e componentes de *software*, com pilhas de protocolo, manipuladores de comunicação e *drivers*. O resultado de funcionalidade e desempenho está acessível para sistemas distribuídos e programas de aplicativos afetados por todos esses. Assim, aplicaremos a nossa atenção à coleção de *hardwares* e *softwares* componentes que fornecem os recursos de comunicação a um sistema distribuído como subsistema de comunicação.

Os computadores e outros dispositivos que usam a rede para fins de comunicação são referidos como *hosts*. O termo **nó** é empregado para se referir a qualquer computador ou dispositivo de comutação conectado a uma rede.

*Internet* é um único subsistema que fornece comunicação entre todos os *hosts* conectados ao qual. É construída a partir de muitas sub-redes, estas que configuram uma unidade de roteamento – entrega de dados de uma parte da *internet* para outra –; é uma coleção de nós que podem ser alcançados na mesma rede física.

A infraestrutura da *internet* inclui uma arquitetura, *hardware* e *software* componentes que efetivamente integram diversas sub-redes em uma única comunicação de dados de serviço.

A concepção de um subsistema de comunicação é fortemente influenciada pelas características dos sistemas operacionais usados nos computadores dos quais a distribuição de sistema é composta, assim como as redes que os interconectam.

Veremos, então e de maneira geral, uma introdução sobre redes com referência aos requisitos de comunicação de sistemas distribuídos.

## Problemas de Rede para Sistemas Distribuídos

As primeiras redes de computadores foram projetadas para atender a uma aplicação relativamente simples de requisitos, de modo que aplicativos de rede, tais como transferências de arquivos, *login* remoto, correio eletrônico e grupos de notícias foram suportados.

O desenvolvimento subsequente de sistemas distribuídos deu suporte a programas aplicativos também distribuídos que acessam arquivos compartilhados e outros recursos, definindo um padrão mais alto de desempenho para atender às necessidades desses aplicativos interativos.

Mais recentemente, após o crescimento e a comercialização da *internet* e do surgimento de muitos novos modos de uso, apareceram requisitos mais rigorosos de confiabilidade, escalabilidade, mobilidade, segurança e qualidade de serviço.

As redes de computadores compõem uma parte indispensável da infraestrutura de sociedades modernas. O crescimento tem sido tão rápido e diversificado que é difícil encontrar estatísticas. O tamanho potencial futuro da *internet* é compatível com a população do Planeta, de modo que é realista esperar que inclua vários bilhões de nós e centenas de milhões de *hosts* ativos.

Nossa discussão dos modelos de falhas descreve o impacto dos erros de comunicação. Muitos aplicativos são capazes de recuperar a comunicação de falhas; portanto, não requerem comunicação garantida livre de erros. A confiabilidade da maioria dos meios físicos de transmissão é muito alta. Quando ocorrem erros, geralmente se devem a falhas no *software*, emissor ou receptor.

Os requisitos e as técnicas para obter segurança em sistemas distribuídos e o primeiro nível de defesa adotado pela maioria das organizações, protegendo as suas redes e os computadores conectados aos quais com um *firewall*.

Dispositivos móveis, tais como *laptops* e *smartphones*, com capacidade para *internet* são movidos com frequência entre locais e reconectados em rede conveniente de pontos de conexão ou até mesmo em movimento; de modo que redes sem fio fornecem conectividade a esses dispositivos. Contudo, os esquemas de endereçamento e roteamento da *internet* foram desenvolvidos antes do advento desses dispositivos móveis, de modo que não estão bem adaptados à necessidade de conexão intermitente para muitas sub-redes diferentes. À *internet*, tais mecanismos foram adaptados e ampliados para apoiar a mobilidade, mas o futuro crescimento esperado no uso de dispositivos móveis exigirá mais desenvolvimento.

Anteriormente, definimos qualidade de serviço como a capacidade para cumprir prazos durante a transmissão e o processamento de fluxos de multimídia em tempo real. Isso impõe novos requisitos importantes em redes de computadores. Assim, aplicações que transmitem dados multimídia exigem largura de banda garantida e latências limitadas aos canais de comunicação que usam. Algumas aplicações variam as suas demandas dinamicamente para especificar tanto uma qualidade de serviço mínima aceitável, quanto uma ótima.

A maior parte da comunicação em sistemas distribuídos é entre pares de processos, mas muitas vezes há também a necessidade de comunicação de um para muitos; enquanto isso pode ser simulado por envios para vários destinos, torna-se mais caro do que o necessário e pode não exibir as características de tolerância a falhas exigidas pelas aplicações. Por essas razões, muitas tecnologias de rede suportam a transmissão simultânea de mensagens para vários destinatários.

## Tipos de Rede

---

Os principais tipos de rede que são usados para suportar sistemas são: PAN – redes de área pessoal –, LAN – redes de área local –, WAN – redes de área ampla –, MAN – redes de área metropolitana – e as variantes sem fio dessas opções – a *internet* é construída a partir de redes de todos esses tipos:

- **LAN:** transmitem mensagens a velocidades relativamente altas entre computadores conectados por um único meio de comunicação, tais como o fio de cobre trançado, cabo coaxial ou a fibra óptica;
- **WAN:** transportam mensagens a velocidades mais baixas entre nós que muitas vezes se encontram em organizações diferentes que podem estar separadas por grandes distâncias;
- **MAN:** Este tipo de rede se baseia no cabeamento de largura de banda de cobre e fibra óptica, sendo recentemente instalado em algumas cidades para a transmissão de vídeo, voz e outros dados em distâncias de até 50 km;
- **Redes locais sem fio (WLAN):** são projetadas para uso no lugar de cabos LAN a fim de fornecer conectividade para dispositivos móveis, ou simplesmente para remover a necessidade de infraestrutura com fio para conectar computadores dentro de casas e edifícios de escritórios para cada outro ponto e a *internet*;
- **Redes de longa distância sem fio (WWAN):** a maioria das redes de telefonia celular é baseada em tecnologias de rede sem fio digital, tal como o *Global System for Mobile* (GSM), empregado na maioria dos países.

Um conjunto de redes é um subsistema de comunicação no qual várias redes estão interligadas para fornecer recursos comuns de comunicação de dados que sobrepõem as tecnologias e os protocolos das redes de componentes individuais e métodos utilizados para a sua interconexão.

## Princípios da Rede

---

A base para todas as redes de computadores é a técnica de comutação de pacotes desenvolvida na década de 1960. Isso permite que pacotes de dados endereçados a diferentes destinos compartilhem uma única ligação de comunicação, ao contrário da tecnologia de comutação de circuitos subjacentes à telefonia.

## Transmissão de Pacotes

Na maioria das aplicações de redes de computadores o requisito é para a transmissão de unidades lógicas de informação ou mensagens – sequências de itens de dados de comprimento arbitrário. Mas antes que uma mensagem seja transmitida, é subdividida em pacotes.

## Transmissão de Dados

A transmissão e exibição de áudio e vídeo em tempo real é chamada de *streaming*. Requer muito mais larguras de banda que a maioria das outras formas de comunicação em sistemas distribuídos.

## Esquemas de Comutação

Uma rede consiste em um conjunto de nós conectados por circuitos. Assim, para transmitir informações entre dois nós arbitrários é necessário um sistema de comutação. Definamos, então, os tipos de comutação usados em redes de computadores:

- *Broadcasting* é uma técnica de transmissão que não envolve comutação;
- Comutação de circuitos;
- Comutação de pacotes.

Mas em uma rede de computadores, os pacotes podem ser armazenados e processados com rapidez suficiente para a ilusão de transmissão instantânea, mesmo que cada pacote precise ser encaminhado a muitos nós.

## Protocolos

O termo **protocolo** é empregado para se referir a um conjunto bem conhecido de regras e formatos a serem usados para a comunicação entre processos na execução de determinada tarefa. A definição de um protocolo tem duas partes importantes, pois pode servir como uma especificação:

- Da sequência de mensagens que devem ser trocadas;
- Do formato dos dados nas mensagens.

A existência de protocolos conhecidos permite que os componentes de *software* sejam separados em sistemas distribuídos a serem desenvolvidos de forma independente e implementados em linguagens de programação em computadores que podem ter códigos de pedido e dados com distintas representações. Os conjuntos de protocolos incluem uma camada de aplicação, outra de transporte e uma terceira camada de *internetwork*.

Quadro 2

Camada	Função
7 – Aplicação	Provê acesso aos serviços da rede para as aplicações. Por exemplo: quando um usuário for ler seus e-mails utilizando um programa de e-mail, na verdade está fazendo com que o seu programa de e-mail inicie uma transmissão de dados com a camada de aplicação do protocolo usado.
6 – Apresentação	Serve como tradutora dos dados na rede. Traduz os dados do formato enviado pela camada de aplicação para um formato comum a ser usado na transmissão desse dado, ou seja, um formato entendido pelo protocolo usado. Pode ter outros usos, como compressão e criptografia.
5 – Sessão	Proporciona a estrutura de controle para que as aplicações possam estabelecer, gerenciar e terminar sessões de comunicação. Sessão é uma comunicação que necessita armazenar estados. Estados são armazenados para permitir reestabelecimento da comunicação em caso de queda da comunicação, como por exemplo, na retomada da transferência de arquivos. Essa camada é relativamente pouco usada, pois muitos protocolos empacotam a funcionalidade dessa camada nas suas camadas de transporte.
4 – Transporte	Responsável pela transferência de pacotes de dados entre dois pontos de forma transparente e confiável com funções como controle de fluxo e correção de erro fim a fim.
3 – Rede	A camada de rede é responsável pelo endereçamento dos pacotes, convertendo endereços lógicos em endereços físicos, de forma que os pacotes consigam chegar corretamente ao destino. Essa camada também determina a rota que os pacotes seguirão para atingir o destino, baseada em fatores como condições de tráfego da rede e prioridades.
2 – Enlace	Responsável por empacotar os dados, fracionamento da mensagem em unidades de dados denominados quadros, que correspondem a algumas centenas de bytes. Um quadro é uma estrutura que contém informações suficientes para garantir que os dados sejam enviados com sucesso através de uma rede local até o seu destino. A entrega bem-sucedida significa que o quadro atingiu intacto o seu destino. Portanto, o quadro também deve ter um mecanismo para verificar a integridade do seu conteúdo na chegada.
1 – Física	Compreende as especificações do hardware que é utilizado na rede, ou seja, as especificações elétricas, mecânicas e físicas, que são documentados pelos padrões internacionais como por exemplo RS-232, Ethernet 802.3, V. 35.

## Roteamento

O roteamento é uma função necessária em todas as redes, exceto nas LAN, como *ethernets* – que fornecem conexões diretas entre todos os pares de hosts conectados. Em redes grandes, o roteamento adaptativo é empregado: o melhor caminho à comunicação entre pontos na rede é reavaliado periodicamente, levando em consideração o tráfego atual na rede e quaisquer falhas, tais como conexões ou roteadores quebrados.

## Controle de Congestão

A capacidade de uma rede é limitada pelo desempenho de seus *links* de comunicação e nós de comutação. Quando a carga em qualquer *link* ou nó específico se aproxima de sua capacidade, as filas se acumulam nos *hosts* tentando enviar pacotes e nos nós intermediários que segurarem pacotes cuja transmissão avançada é bloqueada por outro tráfego. Se a carga permanecer no mesmo nível alto, as filas continuarão a crescer até atingir o limite de espaço de *buffer* disponível. Quando tal estado é alcançado em um nó, este não tem outra opção a não ser abandonar pacotes de entrada. Assim, em vez de permitir que os pacotes percorram a rede até chegarem ao limite dos nós congestionados, onde terão de ser descartados, melhor seria mantê-los em nós anteriores até que o congestionamento fosse reduzido.

Em geral, o controle de congestionamento é obtido informando os nós ao longo de uma rota que esse ocorreu e que, portanto, apresenta taxa reduzida de transmissão de pacotes. Para nós intermediários, isso resultará no *buffer* de pacotes de entrada para um período mais longo. Já para *hosts* que são fontes dos pacotes, o resultado pode ser enfileirar tais pacotes antes da transmissão, ou bloquear o processo de aplicação que os está gerando até que a rede possa lidar com os quais.

## Internetworking

Existem muitas tecnologias com diferentes camadas de rede, *links* e físicas de protocolo, de modo que as redes locais são construídas com tecnologias *ethernet*, enquanto as áreas amplas de redes são elaboradas sobre opções telefônicas analógicas e digitais de vários tipos, *links* de satélite e redes ATM de área ampla.

Ademais, computadores individuais e redes locais estão ligados à *internet* ou a *intranets* por *modems* e conexões sem fio e DSL.

## Protocolos da Internet

Descreveremos aqui os principais recursos do conjunto de protocolos TCP/IP e discutiremos as suas vantagens e limitações quando utilizados em sistemas distribuídos.

Os protocolos da *internet* foram originalmente desenvolvidos principalmente para suportar aplicações de área, tais como transferência de arquivos e correio eletrônico, envolvendo latências relativamente altas entre computadores geograficamente dispersos; mas acabaram por se tornar eficientes o suficiente para suportar os requisitos de muitas aplicações distribuídas em redes locais e de longa distância, estando quase que universalmente empregados em sistemas distribuídos. Assim, a padronização resultante de protocolos de comunicação trouxe imensos benefícios.

Tais protocolos geralmente são colocados sobre outra tecnologia de rede, tal como a *ethernet*, que já fornece uma camada de rede que permite a troca de datagramas pelos computadores conectados à mesma rede.

O sucesso do TCP/IP se baseia na independência dos protocolos em relação à tecnologia de transmissão subjacente, permitindo que *internetworks* sejam construídos a partir de redes heterogêneas e *links* de dados. Usuários e programas aplicativos percebem uma rede virtual única com suporte a TCP, UDP e implementadores de TCP e UDP em uma rede IP virtual específica, ocultando a diversidade da mídia de transmissão subjacente.

## Endereçamento

Talvez o IP seja o aspecto mais desafiador do *design* dos protocolos da *internet*, pois está relacionado à construção de esquemas para nomear e endereçar *hosts* e rotear pacotes IP para os seus destinos.

## Protocolo IP

O protocolo IP transmite datagramas de um *host* para outro, se necessário via roteadores intermediários. O formato completo do pacote IP é bastante complexo, dado que existem vários campos que são usados pelos algoritmos de transmissão e roteamento.

A camada IP coloca os datagramas IP em pacotes de rede adequados para a transmissão em rede. Ademais, deve inserir também um endereço “físico” de rede da mensagem-destino para a rede subjacente.

## Roteamento IP

A camada IP roteia pacotes de sua origem para o seu destino. Cada roteador na *internet* implementa o *software* de camada IP para fornecer um algoritmo de roteamento.

## IPv6

Uma solução mais permanente para as limitações de endereçamento do IPv4 também foi buscada, e isso levou ao desenvolvimento e à adoção de uma nova versão do protocolo IP com endereços substancialmente maiores. A *Internet Engineering Task Force* (IETF) notou os problemas potenciais decorrentes de endereços de 32 bits do IPv4 já em 1990, de modo que iniciou um projeto para desenvolver uma nova versão do protocolo IP. Assim, o IPv6 foi adotado pelo IETF em 1994 e uma estratégia para migração foi recomendada.

## TCP e UDP

TCP e UDP fornecem as capacidades de comunicação da *internet* de uma forma que é útil para programas aplicativos. Os desenvolvedores de aplicativos podem desejar outros tipos de transporte, por exemplo, para fornecer garantias ou segurança em tempo real, mas os serviços geralmente exigiriam mais suporte na camada de rede do que o IPv4 fornece. Assim, TCP e UDP podem ser vistos como reflexões fiéis na programação da aplicação em nível das facilidades de comunicação que o IPv4 tem para oferecer.

IPv6 é outra história: certamente continuará a suportar TCP e UDP, mas inclui recursos que não podem ser convenientemente acessados através de TCP e UDP; de modo que pode ser útil introduzir tipos de serviços de transporte para explorá-los, uma vez que a implantação do IPv6 é para justificar o seu desenvolvimento.

A primeira característica a se notar é que, enquanto o IP suporta comunicação entre pares de computadores – identificados por seus endereços IP –, TCP e UDP, como protocolos de transporte, devem fornecer comunicação de processo a processo – isto é realizado pelo uso de portas. Por sua vez, números de porta são empregados para endereçar mensagens a processos dentro de um determinado computador, sendo válidos apenas em tal máquina.

UDP não oferece garantia de entrega, enquanto TCP fornece um serviço de transporte muito mais sofisticado, garantindo uma entrega confiável, isto porque a camada TCP inclui mecanismos adicionais – implementados sobre IP – a fim de atender às garantias de confiabilidade, a saber:

- **Nomes de domínio:** o DNS é uma forma na qual os computadores podem atender às solicitações na conversão aos nomes de domínio que os usuários especificam em endereços da *internet*;
- **Firewalls:** quase todas as organizações precisam de conectividade com a *internet* para o fornecimento de seus serviços aos seus clientes e outros usuários externos, permitindo que os seus usuários internos acessem as informações e tais serviços.

## Resumo

Concentramo-nos aqui nas técnicas e nos conceitos de rede que são necessários como bases para sistemas distribuídos, aproximando-os do ponto de vista de um *designer* de sistema.

Redes de pacotes e protocolos em camadas fornecem as bases para a comunicação em sistemas distribuídos. Redes de área local são baseadas em pacotes para transmitir em um meio compartilhado; *ethernet* é a tecnologia dominante. Amplas áreas de redes se baseiam na comutação de pacotes para encaminhá-los aos seus destinos através de rede conectada. Roteamento é um mecanismo-chave e uma variedade de algoritmos de roteamento é usada, em que o método do vetor de distância é o mais básico, mas eficaz. Congestionamento controle é necessário para evitar o “estouro” de buffers no receptor e intermediário de nós.

Ademais, as redes interconectadas são construídas colocando em camadas um protocolo de interconexão com coleções de redes interligadas por roteadores. Os protocolos TCP/IP permitem computadores na *internet* para se comunicarem uns com os outros de maneira uniforme, independentemente de estarem na mesma rede de área local ou em países diferentes. Os padrões da *internet* incluem muitos protocolos em nível de aplicativo que são adequados para uso em aplicações distribuídas de área ampla. IPv6 tem espaço de endereçamento muito maior ao necessário à evolução futura da *internet*, além do fornecimento de novos requisitos, tais como qualidade de serviço e segurança.

Os usuários móveis são suportados pelo *MobileIP* para *roaming* de área ampla e por *wireless LAN* baseados nos padrões IEEE 802 para conectividade local.

# Material Complementar

## Indicações para saber mais sobre os assuntos abordados nesta Unidade:

### Sites

#### Cultura Mix

O que é um sistema distribuído?

<https://goo.gl/gUpUCx>

#### Blog do Professor Manoel Veras

Modelos de sistemas distribuídos.

<https://goo.gl/o9fRGD>

#### Quora

Redes e inter-redes.

<https://goo.gl/cBxYkS>

#### Wikipedia

Computação paralela.

<https://goo.gl/U1r8rj>

## Referências

- GAGLIARDI, G. **Cliente/servidor**. São Paulo: Makron Books do Brasil, 1996.
- STALLINGS, W. **Arquitetura e organização de computadores:** projeto para o desempenho. 8. ed. São Paulo: Pearson Prentice Hall, 2009.
- TANENBAUM, A. S.; STEEN, M. V. **Sistemas distribuídos:** princípios e paradigmas. 2. ed. [S.l.]: Pearson, 2007.





**Cruzeiro do Sul**  
Educacional