UNIVERSIDADE PITÁGORAS UNOPAR ANHANGUERA - MARAVILHA
ENGENHARIA DE *SOFTWARE*

NATAN OGLIARI - 3446687604

REDES E SISTEMAS DISTRIBUÍDOS

Maravilha/SC
2025

NATAN OGLIARI - 3446687604

REDES E SISTEMAS DISTRIBUÍDOS

Produção textual apresentada ao curso de Bacharelado em Engenharia de *Software* da UNOPAR, em cumprimento ao requisito obrigatório para aprovação na disciplina de REDES E SISTEMAS DISTRIBUÍDOS .

Orientador: Murilo Caminotto Barbosa .

Maravilha/SC
2025

Sumário

Lista de Algoritmos

Lista de Figuras

Lista de Tabelas

1 Introdução

Os sistemas distribuídos representam uma das áreas mais importantes da computação moderna, sendo definidos como um conjunto de computadores independentes que se apresentam aos usuários como um sistema único e coerente (TANENBAUM; STEEN, 2016). Estes sistemas são caracterizados pela ausência de memória compartilhada, comunicação através de mensagens e pela independência de falhas entre os componentes (COULOURIS et al., 2013).

Figura 1. *Arquitetura de Sistema Distribuído.*

Fonte: Adaptado de Tanenbaum e Steen (2016)

A evolução das redes de computadores possibilitou o desenvolvimento de sistemas distribuídos cada vez mais complexos e eficientes. Segundo Kurose e Ross (2021), as redes modernas seguem uma abordagem em camadas que facilita a implementação e manutenção de protocolos de comunicação. O modelo TCP/IP, especificado inicialmente na RFC 793 (POSTEL, 1981), continua sendo a base fundamental para a comunicação em sistemas distribuídos.

Um dos principais desafios em sistemas distribuídos é o problema da sincronização e ordenação de eventos. Lamport (1978) demonstrou que, na ausência de relógios perfeitamente sincronizados, é necessário estabelecer uma ordem lógica dos eventos através de algoritmos específicos. Este trabalho seminal estabeleceu as bases teóricas para muitos protocolos de consenso utilizados atualmente.

O teorema CAP, proposto por Brewer (2000), estabelece que em sistemas distribuídos é impossível garantir simultaneamente consistência (Consistency), disponibilidade (Availability) e tolerância a partições (Partition tolerance). Este teorema fundamental influencia diretamente as decisões de projeto em sistemas distribuídos modernos.

Este relatório apresenta uma análise detalhada dos principais conceitos, algoritmos e tecnologias empregadas em redes e sistemas distribuídos, fundamentando-se em literatura acadêmica reconhecida e exemplos práticos de implementação.

2 Fundamentos Teóricos e Metodologia

2.1 Algoritmos de Consenso Distribuído

Os algoritmos de consenso são fundamentais para garantir a consistência em sistemas distribuídos. O algoritmo de Lamport para ordenação lógica de eventos é um dos pilares teóricos desta área.

Algoritmo 1: Algoritmo de Relógios Lógicos de Lamport

Data: Conjunto de processos $P = \{p_1, p_2, ..., p_n\}$
Input: Evento local e_i no processo p_i
Result: Timestamp lógico ordenado
begin
  Inicializar $LC_i = 0$ para cada processo p_i;
  for *cada evento local e em* p_i **do**
    $LC_i = LC_i + 1$;
    timestamp$(e) = LC_i$;
  end
  if p_i *envia mensagem m para* p_j **then**
    $LC_i = LC_i + 1$;
    timestamp$(m) = LC_i$;
    enviar (m, LC_i) para p_j;
  end
  if p_j *recebe mensagem* (m, LC_k) *de* p_i **then**
    $LC_j = \max(LC_j, LC_k) + 1$;
    processar mensagem m;
  end
end

Fonte: Adaptado de Lamport (1978)

2.2 Protocolos de Comunicação em Redes

A pilha de protocolos TCP/IP fornece a base para comunicação confiável em sistemas distribuídos. A Tabela 1 apresenta os principais protocolos utilizados em cada camada.

Tabela 1. *Principais Protocolos da Pilha TCP/IP*

Camada	Protocolo	Função	RFC
Aplicação	HTTP/HTTPS	Transferência de hipertexto	RFC 2616
Aplicação	FTP	Transferência de arquivos	RFC 959
Transporte	TCP	Transporte confiável	RFC 793
Transporte	UDP	Transporte não confiável	RFC 768
Internet	IP	Roteamento de pacotes	RFC 791
Enlace	Ethernet	Acesso ao meio físico	IEEE 802.3

Fonte: Adaptado de Kurose e Ross (2021)

3 Análise de Sistemas Distribuídos

3.1 Características e Propriedades

Os sistemas distribuídos modernos devem atender a diversos requisitos não funcionais para garantir sua eficácia. As principais características incluem escalabilidade, disponibilidade, tolerância a falhas e transparência (COULOURIS et al., 2013).

A escalabilidade refere-se à capacidade do sistema de manter seu desempenho quando o número de usuários, recursos ou a carga de trabalho aumenta. Existem três tipos principais de escalabilidade: escalabilidade de tamanho, geográfica e administrativa (TANENBAUM; STEEN, 2016).

3.2 Modelos de Arquitetura

Os sistemas distribuídos podem seguir diferentes modelos arquiteturais, cada um com suas vantagens e desvantagens específicas:

1. **Cliente-Servidor**: Modelo tradicional onde clientes fazem requisições a servidores
2. **Peer-to-Peer (P2P)**: Todos os nós podem atuar como cliente e servidor
3. **Arquitetura em Camadas**: Organização hierárquica de serviços
4. **Microserviços**: Decomposição de aplicações em serviços pequenos e independentes

3.3 Implementação de Comunicação Distribuída

O exemplo a seguir demonstra a implementação básica de um algoritmo de eleição em sistemas distribuídos:

3.4 Algoritmo de Eleição em Anel

Um exemplo prático de algoritmo distribuído é o algoritmo de eleição em anel, usado para eleger um coordenador em sistemas distribuídos.

Algoritmo 2: Algoritmo de Eleição em Anel

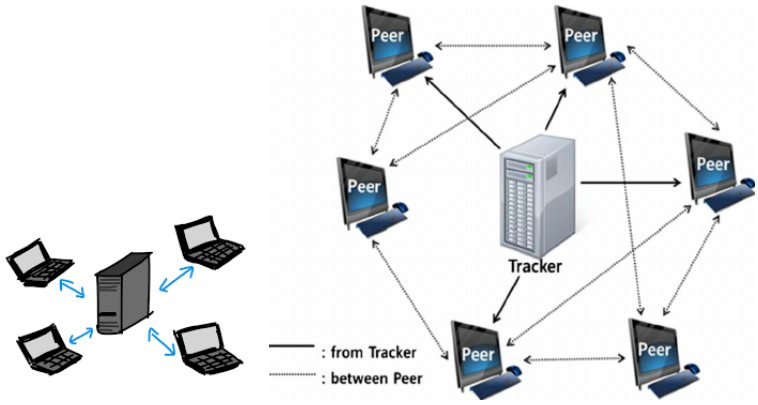
Entrada: Conjunto de processos $P = \{p_0, p_1, \ldots, p_{n-1}\}$ organizados em anel
Saída: Eleição de um coordenador
início
    ▷ Quando um processo detecta falha do coordenador
    if *processo* p_i *detecta falha do coordenador* **then**
        enviar mensagem ELEIÇÃO(p_i) para próximo processo ativo no anel;
        $participando \leftarrow verdadeiro$;
    end
    ▷ Ao receber mensagem de eleição
    if p_j *recebe ELEIÇÃO(lista)* **then**
        if $p_j \notin lista$ **then**
            adicionar p_j à $lista$;
            enviar ELEIÇÃO($lista$) para próximo processo no anel;
            $participando \leftarrow verdadeiro$;
        end
        else
            ▷ A mensagem deu volta completa no anel
            $coordenador \leftarrow \max(lista)$;
            enviar COORDENADOR($coordenador$) para próximo processo;
        end
    end
    ▷ Ao receber mensagem de coordenador
    if p_k *recebe COORDENADOR(coord)* **then**
        $coordenador \leftarrow coord$;
        $participando \leftarrow falso$;
        if $p_k \neq coord$ **then**
            enviar COORDENADOR($coord$) para próximo processo;
        end
    end
fim

Fonte: Adaptado de Tanenbaum e Steen (2016)

3.5 Análise de Desempenho

A análise de desempenho em sistemas distribuídos envolve métricas como latência, throughput, disponibilidade e escalabilidade. A Figura 2 apresenta uma comparação entre diferentes arquiteturas distribuídas.

Figura 2. *Comparação de Arquiteturas Distribuídas*



(a) Cliente-Servidor. (b) Peer-to-Peer.

Fonte: Adaptado de Coulouris et al. (2013)

3.6 Protocolos de Consenso

Os protocolos de consenso são essenciais para manter a consistência em sistemas distribuídos. Algoritmos como Paxos, Raft e PBFT (Practical Byzantine Fault Tolerance) são amplamente utilizados em sistemas reais (LAMPORT, 1978). Cada protocolo apresenta diferentes garantias de segurança e vivacidade, adequadas para cenários específicos.

4 Discussão e Considerações

4.1 Desafios Atuais

Os sistemas distribuídos enfrentam diversos desafios que continuam sendo objeto de pesquisa ativa. Entre os principais desafios, destacam-se:

- **Gerenciamento de Estado**: Manter consistência entre múltiplos nós
- **Detecção de Falhas**: Identificar e recuperar de falhas de rede e hardware
- **Partições de Rede**: Lidar com a separação temporária de nós
- **Escalabilidade**: Manter desempenho com crescimento do sistema

4.2 Tendências Futuras

As tendências emergentes em sistemas distribuídos incluem computação em borda (edge computing), blockchain, sistemas auto-adaptativos e inteligência artificial distribuída. Estas tecnologias promovem maior autonomia e eficiência dos sistemas (COULOURIS et al., 2013).

A evolução das redes 5G e 6G também impactará significativamente o design de sistemas distribuídos, possibilitando latências ultra-baixas e maior largura de banda (ACADEMY, 2019).

5 Conclusões

Este trabalho apresentou uma análise abrangente dos fundamentos teóricos e práticos de redes e sistemas distribuídos. Os conceitos estudados demonstram a complexidade inerente à coordenação de múltiplos processos independentes e a importância de algoritmos bem fundamentados para garantir propriedades como consistência, disponibilidade e tolerância a falhas.

Os algoritmos de consenso, protocolos de comunicação e arquiteturas distribuídas estudados formam a base tecnológica para sistemas modernos como computação em nuvem, blockchain e Internet das Coisas (IoT). O teorema CAP e os trabalhos de Lamport sobre ordenação de eventos continuam sendo referências fundamentais para o desenvolvimento de novos sistemas.

As implementações práticas demonstraram que a escolha da arquitetura e dos algoritmos adequados depende das características específicas do domínio de aplicação. Sistemas que requerem alta disponibilidade podem optar por eventual consistency, enquanto sistemas bancários priorizam strong consistency.

Como trabalhos futuros, sugere-se a investigação de novos protocolos de consenso para ambientes com recursos limitados, como dispositivos IoT, e o desenvolvimento de frameworks que facilitem a implementação de sistemas distribuídos tolerantes a falhas bizantinas.

Os fundamentos apresentados neste relatório fornecem base sólida para compreensão e desenvolvimento de sistemas distribuídos eficientes e confiáveis, essenciais para a infraestrutura tecnológica moderna.

Referências

ACADEMY, C. N. **Introduction to Networks Course Booklet**. 7. ed. [S.l.]: Cisco Press, 2019. 584 p. Acessado em: 06 set. 2025.

BREWER, E. A. Towards robust distributed systems. **Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing**, ACM, p. 7–10, 2000. Acessado em: 06 set. 2025.

COULOURIS, G. et al. **Sistemas Distribuídos: Conceitos e Projeto**. 5. ed. Porto Alegre: Bookman, 2013. 1064 p. Acessado em: 06 set. 2025.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma Abordagem Top-Down**. 8. ed. São Paulo: Pearson, 2021. 864 p. Acessado em: 06 set. 2025.

LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. **Communications of the ACM**, ACM, v. 21, n. 7, p. 558–565, 1978. Acessado em: 06 set. 2025.

POSTEL, J. **RFC 793: Transmission Control Protocol**. 1981. Acessado em: 06 set. 2025. Disponível em: <https://tools.ietf.org/rfc/rfc793.txt>.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas Distribuídos: Princípios e Paradigmas**. 2. ed. São Paulo: Pearson, 2016. 672 p. Acessado em: 06 set. 2025.