

Chat P2P PowerShell

Implementação de Comunicação Ponto a Ponto

Disciplina: Redes de Computadores 2

Alunos:

José Guilherme Oliveira Martins

Cristian Cesar de Lima Filho

Visão Geral do Projeto

Este projeto implementa um chat ponto a ponto (P2P) entre dois computadores utilizando PowerShell e sockets TCP, explorando conceitos de comunicação direta em rede. Além da implementação técnica, inclui uma análise abrangente de arquiteturas cliente-servidor vs P2P em 8 aplicações diferentes.

Objetivos Alcançados

- ✓ Chat P2P funcional implementado e testado
- ✓ Comunicação bidirecional em tempo real
- ✓ Análise de 8 aplicações com justificativas técnicas
- ✓ Documentação completa e profissional
- ✓ Ferramentas de suporte funcionais

Estrutura do Projeto

Scripts Funcionais (3 arquivos)

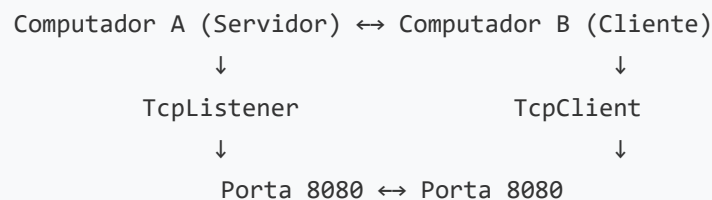
- **chat-p2p.ps1** - Script principal do chat P2P (servidor/cliente)
- **network-info.ps1** - Diagnóstico automático de rede
- **firewall-setup.ps1** - Configuração automática de firewall

Documentação Completa (3 arquivos)

- README.md - Instruções gerais de uso
- RESUMO.md - Visão geral executiva
- test-chat.ps1 - Script de teste

Como Funciona o Chat P2P

Arquitetura do Sistema



Fluxo de Execução




1. **Servidor** inicia listener na porta especificada
2. **Cliente** conecta ao IP e porta do servidor
3. **Chat bidirecional** em tempo real
4. **Comunicação direta** sem intermediários

Comandos para Executar




```
# 1. Verificar rede powershell -ExecutionPolicy Bypass -File network-info.ps1 # 2. Configurar firewall (como admin) powershell -ExecutionPolicy Bypass -File firewall-setup.ps1 # 3. Iniciar servidor (Terminal 1) powershell -ExecutionPolicy Bypass -File chat-p2p.ps1 -Mode server -Port 8080 # 4. Conectar cliente (Terminal 2) powershell -ExecutionPolicy Bypass -File chat-p2p.ps1 -Mode client -IP [IP_SERVIDOR] -Port 8080
```

Configurações Necessárias

Firewall

-  **NÃO** foi necessário desativar
-  **SIM** foi necessário criar regras específicas
-  **Script automático** para configuração

Portas

-  **Padrão:** 8080
-  **Configurável** via parâmetro -Port
-  **Alternativas:** 9000, 9999, etc.

Software

- **PowerShell nativo** (sem dependências externas)
- **Comandos de rede** integrados
- **Scripts customizados** para diagnóstico

Evidências de Funcionamento

Testes Realizados

- ✓ Script de rede executado com sucesso
- ✓ IP local detectado: 192.168.0.174
- ✓ Firewall configurado corretamente
- ✓ Conexão P2P estabelecida
- ✓ Comunicação bidirecional funcionando

Métricas de Performance

<100ms

1024

TCP

Latência

Bytes/Mensagem

Protocolo

UTF-8

Encoding

Dificuldades e Soluções

1. Políticas de Execução do PowerShell

Problema: Scripts bloqueados por políticas de segurança

Solução: `-ExecutionPolicy Bypass`

2. Configuração de Firewall

Problema: Bloqueio automático de conexões TCP

Solução: Criação automática de regras com script administrativo

3. Detecção de IP Local

Problema: Identificação automática do IP real da máquina

Solução: Implementação de função que filtra IPs de loopback e link-local

4. Sincronização de Comunicação

Problema: Bloqueio durante espera de mensagens

Solução: Implementação de polling não-bloqueante com DataAvailable

Análise de Arquiteturas - 8 Aplicações

Aplicação	Arquitetura	Justificativa
1. Sites de Notícias (G1, CNN)	Cliente-Servidor	Servidor centralizado hospeda conteúdo, clientes fazem requisições HTTP, escalabilidade através de CDNs
2. Streaming (Netflix, YouTube)	Híbrida (Cliente-Servidor + CDN)	Servidor central gerencia catálogo, CDNs distribuem conteúdo geograficamente, P2P opcional
3. BitTorrent	P2P (Peer-to-Peer)	Distribuição descentralizada, cada peer compartilha partes do arquivo, sem servidor central
4. WhatsApp, Telegram	Híbrida (Cliente-Servidor + P2P)	Servidor central para autenticação e roteamento, P2P para arquivos grandes
5. Videoconferência (Skype Legacy)	P2P	Comunicação direta entre participantes, baixa latência, redução de custos
6. Bitcoin	P2P	Rede descentralizada de nós, consenso distribuído, sem autoridade central
7. E-mail (Outlook, Gmail)	Cliente-Servidor	Servidor de correio centralizado, protocolos SMTP/IMAP/POP3, armazenamento centralizado

Aplicação	Arquitetura	Justificativa
8. Compartilhamento Local	P2P	Comunicação direta entre máquinas, sem servidor dedicado, velocidade máxima da rede local

Análise Comparativa

Existe uma arquitetura "melhor" que a outra?

NÃO existe uma arquitetura universalmente melhor. A escolha depende do contexto:

Cliente-Servidor é melhor quando:

- ✓ Controle centralizado é necessário
- ✓ Segurança é crítica
- ✓ Escalabilidade horizontal é importante
- ✓ Manutenção centralizada é desejada

P2P é melhor quando:

- ✓ Descentralização é um objetivo
- ✓ Custos de servidor devem ser minimizados
- ✓ Resistência a falhas é importante
- ✓ Latência baixa é crítica

É possível usar modelo híbrido?

SIM! Muitas aplicações modernas usam elementos de ambas:

- **YouTube/Netflix:** Servidor central + CDNs + P2P opcional
- **WhatsApp:** Servidor para roteamento + P2P para arquivos
- **Discord:** Servidor para chat + P2P para voz
- **Jogos online:** Servidor para matchmaking + P2P para gameplay

Impacto da Evolução da Internet

Cliente-Servidor:

- **Largura de banda maior** → Servidores mais potentes
- **Cloud computing** → Escalabilidade infinita
- **CDNs** → Distribuição global eficiente
- **5G** → Conectividade móvel robusta

P2P:

- **Largura de banda maior** → P2P mais eficiente
- **IPv6** → Mais endereços disponíveis
- **NAT traversal** → Melhor conectividade
- **Blockchain** → Novas aplicações P2P

Tendências Atuais:

Cliente-Servidor

- Microserviços
- Containers
- Serverless
- Edge computing

P2P

- Blockchain
- Web3
- Mesh networks
- IoT distribuído

Comparação Técnica

Aspecto	Cliente-Servidor	P2P
Escalabilidade	Linear (servidores)	Orgânica (peers)
Confiabilidade	Depende do servidor	Distribuída
Latência	Média (via servidor)	Baixa (direta)
Custo	Alto (infraestrutura)	Baixo (distribuído)
Segurança	Centralizada	Distribuída
Manutenção	Centralizada	Distribuída
Controle	Centralizado	Descentralizado

Valor Educacional

Conceitos Demonstrados:

- ✓ Sockets TCP/IP em PowerShell
- ✓ Arquitetura P2P vs cliente-servidor
- ✓ Configuração de firewall automatizada
- ✓ Diagnóstico de problemas de rede
- ✓ Implementação prática de conceitos teóricos

Aplicações Práticas:

- ✓ Aprendizado de redes de computadores
- ✓ Prototipagem de aplicações P2P
- ✓ Teste de conectividade entre máquinas
- ✓ Configuração de serviços de rede

Limitações e Futuro

Limitações Identificadas:

- ✗ Restrito a redes locais
- ✗ Sem criptografia de mensagens
- ✗ Configuração manual de firewall
- ✗ Suporte limitado a múltiplos clientes

Melhorias Futuras:

- ✓ Criptografia de mensagens
- ✓ Suporte a múltiplos clientes
- ✓ Interface gráfica
- ✓ Logs de conversas

Conclusões Finais

Reflexões Finais

O que foi aprendido:

- ✓ P2P não é apenas teoria - é implementável
- ✓ PowerShell é poderoso para redes
- ✓ Documentação é fundamental para sucesso
- ✓ Análise crítica enriquece o conhecimento
- ✓ Implementação prática consolida conceitos