

Nome : Riquelme Batista Gomes da Silva

Matrícula : 2021014317

Introdução:

Smart grid é um sistema inovador que utiliza tecnologia da informação para gerar, fornecer e consumir energia de maneira eficiente. Incorporando redes inteligentes às usinas, esse sistema coleta e gerencia dados para controlar as operações de forma eficaz, oferecendo benefícios como controle, eficiência operacional, resiliência da rede e integração de tecnologias renováveis. Ao integrar o Sistema de Supervisão e Controle (SCADA) à smart grid, a coleta, supervisão e administração de dados são realizadas. Unidades de controle centralizadas (MTUs) consultam o estado das redes smart grids, e Interfaces Humano-Máquina (HMIs) são essenciais para consultar dados de dispositivos em campo, como sensores, através da Internet. Nesse trabalho, foi desenvolvido um sistema de comunicação para os clientes HMIs, permitindo o monitoramento em tempo real das redes smart grids e otimizando a linha de produção. Para a realização desse trabalho, foram desenvolvidos dois programas que têm como função implementar as smart grids, operando num sistema cliente-servidor com um sistema peer-to-peer entre os servidores, os quais comunicam entre si numa rede própria.

Implementação:

O trabalho está dividido entre vários arquivos. Na pasta principal (main), encontram-se os programas principais que gerarão os arquivos executáveis. Na pasta src, está a implementação dos códigos-objetos que auxiliam na execução do servidor e do cliente. O arquivo "common" foi retirado do programa exemplo disponibilizado no Moodle.

O arquivo "Rede" foi criado para gerenciar toda a parte da logística de conexão dos clientes aos servidores e dos Peers, abrangendo desde a parte de conexão via sockets até o protocolo de participação na rede com o gerenciamento de IDs. Por sua vez, o arquivo "Dispositivo" é destinado ao gerenciamento dos sensores, obtendo seus dados, interpretando as mensagens provenientes da rede e executando seus comandos. Ele também automatiza a formação das mensagens do cliente e seu envio pela rede.

A maioria das funcionalidades está dividida em funções, visando uma programação mais modular, o que facilitou o desenvolvimento e a verificação do funcionamento do código.

Mensagens:

As mensagens foram concebidas em conformidade com a padronização do roteiro e através de discussões no fórum do Moodle. Todas as mensagens adotaram o formato padrão: COMANDO(INFO_ADICIONAIS), onde os comandos podem ser requisições

ou respostas de requisição, enquanto as informações adicionais fornecem dados complementares aos respectivos comandos.

Os comandos se categorizam em quatro tipos: controle, dados, erro e confirmação, desempenhando, respectivamente, funções de controlar a entrada e saída de clientes e peers, requisitar e responder dados, informar erros de controle e confirmar a saída do cliente ou peer de uma rede servidora. Os dados são, em sua maioria, inteiros, com exceção da potência individual de um sensor e da potência útil, que é um número com até duas casas decimais.

Arquitetura:

Foram desenvolvidos dois programas para um sistema simples de troca de mensagens. No programa destinado ao servidor, foram utilizados dois sockets: um para a conexão P2P com outro servidor e outro para a conexão passiva com os clientes, utilizando o **select()** para o gerenciamento de conexões simultâneas.

Servidor:

O servidor recebe um endereço IPv4 para tentar se conectar ativamente ao peer e dois números de porta na linha de comando, especificando em qual porta ele vai estabelecer a conexão peer-2-peer e em qual vai receber conexões dos clientes. A primeira porta é para a conexão peer-2-peer, e a segunda porta é para se conectar com clientes.

O servidor tenta primeiramente se conectar a outro que esteja escutando passivamente. Se não houver nenhum, ele se torna um servidor que escuta passivamente, aguardando outro servidor ou Peer. Em seguida, ele abre um socket passivo para conexão com os clientes e continua a se conectar com outros clientes pelo socket passivo, comunicando-se com os sockets ativos.

Na parte de comunicação, o servidor verifica inicialmente se é uma requisição de dados. Caso contrário, o servidor trata a mensagem como uma requisição de rede e chama uma função para interpretar se é uma requisição de entrada ou de saída.

Cliente:

O cliente deve receber o endereço IP e a porta do servidor ao qual deseja se conectar para estabelecer a comunicação. O cliente lê os comandos do teclado, gera suas respectivas mensagens de rede e as envia para o servidor. Ele também recebe as respostas do servidor e imprime na tela as informações correspondentes.

Escolhas singulares :

Para simplificação, os IDs dos clientes correspondem exatamente à posição no vetor de clientes. Quanto à escolha dos IDs dos peers, é selecionado um número aleatório entre 20 e 29 para evidenciar a diferença entre os IDs dos Peers e dos clientes.

O tratamento do protocolo para entrada de clientes e peers na rede foi realizado na parte inicial do programa, com exceção à entrada contínua de clientes e peers no servidor, que é tratada de forma contínua.

Conclusão:

O trabalho foi bem-sucedido no desenvolvimento de dois programas para um sistema de troca de mensagens em Smart Grids, utilizando funcionalidades da biblioteca de sockets POSIX. A arquitetura do servidor envolve o uso de dois sockets, um para conexão P2P e outro para conexão passiva com clientes, gerenciando múltiplas conexões através da função `select()`. O cliente se conecta ao servidor, lê comandos do teclado, gera mensagens de rede e processa as respostas do servidor.

A padronização das mensagens, discutida no fórum do Moodle, adotou o formato COMANDO(INFO_ADICIONAIS), categorizando os comandos em controle, dados, erro e confirmação. A comunicação eficaz entre entidades, como clientes e peers, foi garantida pela interpretação consistente dessas mensagens.

A implementação modular, dividida em arquivos principais, facilitou o desenvolvimento e a verificação do código. A arquitetura robusta do servidor, a lógica de conexão e o gerenciamento de mensagens proporcionam um sistema funcional e eficiente para a troca de informações em Smart Grids.