

Uso de Comunicação Celular Digital GPRS para Sistemas de Automação da Distribuição de Energia Elétrica

V. C. Zambenedetti*, LACTEC; F. R. Coutinho, LACTEC; A. Barbiero, LACTEC;
R. P. Siqueira, COPEL DIS/CEFET-PR; J. G. Pereira, COPEL DIS/CEFET-PR; R. A. Hexsel, UFPR

Resumo- Este artigo trata do projeto de P&D desenvolvido pela COPEL DISTRIBUIÇÃO e o LACTEC de um sistema de comunicação utilizando telefonia celular com tecnologia GPRS. O desenvolvimento consta de duas partes: a primeira, um conversor serial-IP que faz a interface entre um dispositivo eletrônico inteligente (IED) e o telefone móvel, o lado cliente. A segunda, o programa para um Servidor junto ao Centro de Operação, o lado servidor. Ambos, o conversor serial-IP e o Servidor têm a finalidade de tornar o meio de comunicação transparente para o IED e o Centro de Operações.

Palavras-chave— Automação de redes; telefonia celular; TCP/IP.

I. INTRODUÇÃO

Com o início do serviço de telefonia celular da chamada geração 2.5 em 2002 esta tecnologia passou a disponibilizar ao usuário a comunicação por pacotes, utilizando protocolo da Internet, onde antes havia apenas o uso de modems ou serviços "short message" – SMS. Basicamente dois sistemas são oferecidos no Brasil, um o GPRS - General Packet Radio Service e outro o 1xRTT - Single Carrier Radio Transmission Technology. Nas tecnologias 2.5G o telefone móvel (celular) e a ERB – Estação Rádio Base têm como modo padrão a comutação por circuito (voz) sendo que apenas o cliente, ou seja, o telefone celular, pode solicitar à ERB a mudança para o modo comutação por pacote (IP). Desta forma o cliente, se estiver no modo padrão, não pode receber um quadro de dados no protocolo IP, ou seja, não está efetivamente conectado à Internet. No caso da automação de redes de distribuição, se o Cliente estiver conectado a um Dispositivo Eletrônico Inteligente - IED e o Servidor conectado a um Centro de Operação da Distribuição - COD, o COD não pode enviar um comando ao IED sem que o cliente (telefone móvel) tenha iniciado uma conexão.

O programa desenvolvido para o Cliente, identificado como Conversor Serial IP na Figura 1, deve ter uma tarefa que procure manter a conexão ativa a fim de poder receber comandos do servidor. Existem alternativas para esta situação onde, por exemplo, o Servidor conhece o número do telefone móvel do cliente e faz uma chamada telefônica com

o intuito de "acordá-lo". Além disto a forma de atuação das Operadoras de telefonia celular pode variar, com o acesso à rede da operadora, fazendo com que os clientes possam trabalhar com IP fixo; ou permitindo que o servidor forneça endereços IP válidos ou ainda utilizando NAT – Network Address Translation, onde a Operadora fornece ao cliente um endereço IP dinâmico conhecido apenas internamente à Operadora e para a Internet este cliente recebe IP dinâmico, válido no ambiente da Internet.

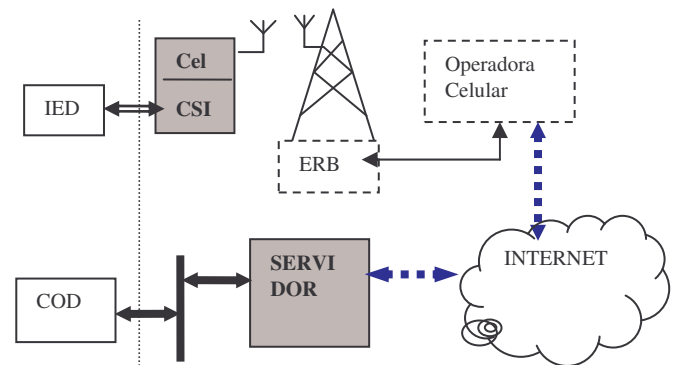


Figura 1 – Diagrama em blocos do sistema

A COPEL e o LACTEC, através de um projeto de Pesquisa e Desenvolvimento, estão desenvolvendo um sistema de comunicação utilizando telefonia celular 2.5G com tecnologia GPRS. O desenvolvimento consta de duas partes: inicialmente um Conversor Serial-IP (CSI) que faz a interface entre um dispositivo eletrônico inteligente (IED) e o telefone móvel, que será o cliente. A segunda parte é o programa para um Servidor junto ao Centro de Operação. Ambos, o Conversor Serial-IP e o Servidor, têm a finalidade de tornar o meio de comunicação transparente para o IED e o Centro de Operações. O CSI realiza as tarefas de encapsular o protocolo de automação no protocolo IP, manter a conexão ativa, solicitar autenticação ao servidor e criptografar mensagens.

A. Visão Geral do Sistema

A Figura 1 mostra uma visão geral de como o sistema está concebido. O IED – *Intelligent Electronic Device* - é um relógio ou chave automatizada da COPEL, utilizando protocolos aplicados no setor elétrico, como o DNP3,

*Vóldi C. Zambenedetti trabalha no Instituto de Tecnologia para o Desenvolvimento (voldi@lactec.org.br).

Roger P Siqueira trabalha na COPEL DIS (roger.siqueira@copel.com)

Roberto A Hexsel trabalha na Univ. Federal do Parana (roberto@inf.ufpr.br)

IEC60870-5-101, Harris, dentre outros. Estes equipamentos possuem como interface de comunicação uma porta RS232 ou alternativamente RS485. O celular é de tecnologia “geração dois e meio” ou 2.5G, GPRS - General Packet Radio Service, com as seguintes características:

- apenas inicia a conexão IP, nunca recebe pedido de conexão (por ser 2.5G)
- ao iniciar a conexão faz uma negociação com a operadora celular antes de poder enviar mensagens para a Internet.

O CSI - Conversor Serial IP - é um circuito eletrônico microprocessado, desenvolvido no primeiro ano do projeto, que utiliza o processador motorola 56F827 e memória externa de 64Kwords. Foi projetado e desenvolvido de forma a tornar transparente todo o meio de comunicação para o IED. Sua seqüência de atuação é inicializar o módulo celular, conectar-se com a operadora utilizando protocolo PPP, receber um endereço IP na rede da Operadora, estabelecer conexão com o Servidor [4,8]. A partir da conexão estabelecida deve encapsular e desencapsular o quadro de e para a porta RS232 para IP e deve também manter aberta a conexão com o Servidor. Este gerenciamento de conexão depende do modo de atuação da Operadora celular. No atual estágio está sendo utilizado apenas Operadoras GSM, embora a utilização de um modem CDMA não difira em essência de um modem GPRS. Como existem mais de uma operadora GSM e um mesmo módulo pode utilizar mais de uma operadora apenas com a troca de um cartão - SIM card - a versão final do CSI contará com dois SIM cards, a fim de possibilitar redundância de operadora, melhorando a qualidade de serviço.

O Servidor é um desenvolvimento do segundo ano do projeto, que está parcialmente concluído. Faz a ponte entre a o Firewall (este ligado diretamente à Internet) e o Centro de Operação da Distribuição da COPEL, através de rede intranet, gerenciando endereçamento IP dinâmico e conexão com o IED (via CSI), além de realizar todo processo de autenticação e segurança de comunicação através de criptografia das mensagens. Atualmente a criptografia está desativada a fim de agilizar os testes de comunicação. O Firewall deve ter um endereço IP e porta válidos na Internet, e deve ser fixo. É para este endereço que as mensagens do CSI devem convergir. Como alternativa, no CSI foi implementado um cliente DNS, testado com sucesso, que procura um endereço eletrônico através do nome (ex: www. automa.copel.com), embora este método cause um maior atraso no estabelecimento da conexão IP com o Servidor.

Da forma como está concebido, o sistema não utiliza serviços de rede privativa - VPN - com uma Operadora Celular específica, mas pode utilizar os serviços GPRS de qualquer Operadora, como foi demonstrado nos testes realizados - trocando o cartão SIM e o quadro de inicialização do protocolo PPP. Para o uso dos serviços de duas Operadoras não se notou nenhuma necessidade de se alterar configurações do Servidor nem do módulo CSI.

II. O CONVERSOR SERIAL-IP

O conversor serial-IP - CSI - foi concebido com duas finalidades: fazer a interface de comunicação entre uma porta serial e o protocolo TCP/IP ou UDP/IP e servir como base para futuros desenvolvimentos, principalmente a inclusão de processamentos de leitura de dados analógicos e digitais. Embora existam soluções prontas que realizam a tarefa de conversão serial-IP, estas não possuem a flexibilidade pensada no início do projeto.

A. Desenvolvimento do Circuito

O desenvolvimento do hardware do projeto Conversor Serial-IP foi dividido em duas fases, onde cada uma resulta em um protótipo funcional do hardware. Na fase inicial optou-se por desenvolver uma placa que utilizasse fonte e módulo de modem externo, com intuito de se agilizar o processo de desenvolvimento do software visto que a fonte e o módulo são as partes críticas do desenvolvimento.

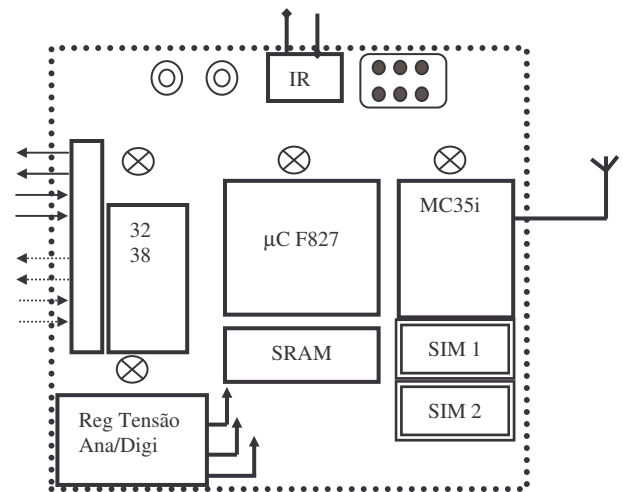


Figura 2 - Diagrama em blocos do CSI

Na segunda fase o módulo celular foi incorporado à placa do processador, ficando como mostrado na Figura 2. O Regulador de Tensão regula e abaixa a tensão da fonte de alimentação externa, fornecendo três saídas de tensão independentes, estabilizadas e contínuas. Uma delas é utilizada para uso específico de alimentação do núcleo do processador, uma para o módulo celular e a outra para os demais circuitos. A alimentação do módulo GPRS é feita em 4,2Vcc nominais. Este regulador mereceu um projeto especial pois embora seu consumo em regime seja relativamente baixo, menos de 100mA, quando em transmissão exige picos de até 2A, embora os picos tenham duração inferiores a 100μs. Além disso, possui sinal que reinicia a unidade de processamento no caso de ocorrer problemas com o nível da tensão fornecida pela fonte externa ao regulador.

A Memória Volátil (SRAM) fornece quantidade extra de memória para que se possa efetuar o processamento da pilha TCP/IP e PPP/IP. Além disso é utilizada também como memória de programa para agilizar o desenvolvimento do software visto que oferece velocidade de gravação superiores as da memória flash.

São duas as interfaces seriais assíncronas. A Interface de Comunicação Infravermelha serve para permitir a reprogramação dos parâmetros de funcionamento da placa além também de permitir que se atualize todo o software embarcado por uma versão mais nova. É utilizada principalmente para troca de configuração ou programa em campo. A Interface Serial para comunicação com o IED é uma serial padrão RS-232. Uma terceira interface serial conecta o Processador diretamente ao módulo celular, não estando disponível externamente.

O Processador 56F827 possui arquitetura mista de processador de sinais com microcontrolador. Executa um Escalonador, descrito abaixo, fazendo o processamento do protocolo da remota para o protocolo TCP/IP sobre PPP, criptografia e gerenciamento da conexão. É dotado de memória flash interna para programa e memória volátil para dados.

III. NÚCLEO DE ESCALONAMENTO DE TEMPO REAL

Devido à necessidade de se realizar várias tarefas paralelamente (gerenciamento da conexão e transmissão de dados, por exemplo) e da dificuldade de se dar manutenção a uma máquina de estados, principalmente quando se insere ou se retira estados, decidiu-se construir um sistema de escalonamento de tarefas que facilitasse o desenvolvimento e a manutenção do sistema.

A. Tempos de Escalonamento

O escalonamento de tarefas é um método para se atingir um pseudoparalelismo na execução de tarefas, por permitir o compartilhamento do tempo de processamento entre as várias tarefas e sem atrapalhar a lógica escrita seqüencialmente pelo programador [3,7]. O núcleo desenvolvido é preemptível, de forma que nenhuma tarefa ocupa o processador além do tempo estipulado pelo escalonador para sua execução, aumentando a robustez do sistema no quesito de tolerância a falhas por permitir a recuperação do sistema mesmo quando as tarefas individuais sofrem alguns tipos de falhas críticas. Visando uma aplicação futura onde esse sistema poderia ser usado como uma unidade remota de supervisão e controle, decidiu-se trabalhar com duas faixas de tempo no núcleo de tempo real. Uma de 52 μ s para causar a preempção de cada tarefa, e outra de 520 μ s para marcar o reinício do ciclo de tempo real.

Isto quer dizer que cada tarefa só pode ser executada uma única vez a cada 520 μ s, e a cada vez que ela for executada terá no máximo 52 μ s de tempo de processamento, incluindo o tempo necessário para salvamento e restauração do contexto das tarefas (estado do processador durante a execução de uma tarefa). Este intervalo de tempo - 520 μ s - corresponde a 32 amostras de um ciclo de 60Hz. Depois de definido este tempo, considerou-se que 10 tarefas para este sistema seria um número suficiente, e dividindo-se o ciclo de tempo real por 10 tarefas, obtém-se o tempo de execução de 52 μ s. O ciclo de 520 μ s compreende também um *overhead* que é a

reinicialização das tarefas para execução mais algum tempo para preparar o conversor A/D. O tempo de execução de cada tarefa não é afetado por este *overhead*, pois o temporizador de preempção é reinicializado após a execução de todo o procedimento do temporizador de tempo real.

B. Estrutura dos Descritores

O escalonador se divide basicamente em três estruturas: o Descritores do Escalonador, o Descritores de Tarefas e o Contexto da Tarefa. Há apenas um Descritores de Escalonador que aponta para listas de Descritores de Tarefas e contém algumas informações como a Tarefa em execução. A FIGURA 01 mostra o relacionamento entre os Descritores, com uma lista encadeada de Tarefas prontas (*ready*). O Descritores de Tarefa possui como dados um ponteiro para o próximo Descritores (next), o ponto de entrada da tarefa (TaskX), o sinalizador de execução, a prioridade, o estado - pronta, executando, suspensa (*ready*, *running*, *suspended*) e o ponteiro para o Contexto da tarefa, além de outros dados. A Figura 3 mostra a concepção do uso dos Descritores.

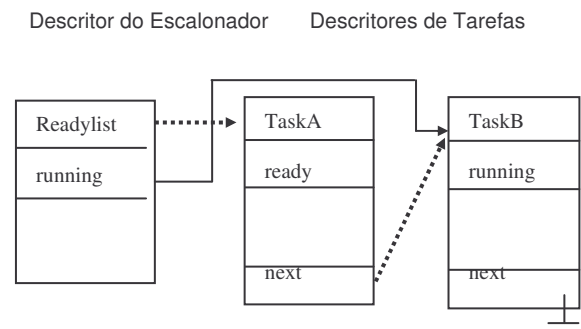


Figura 3 – Descritores do Escalonador e de Tarefas

O ponto de entrada da tarefa é a função a ser executada em paralelo e foi baseada no modelo das *p_threads* [6]. A prioridade da tarefa indica o quão antes a tarefa deve ser executada, ou seja, define sua posição na lista encadeada dos Descritores de Tarefas - quanto maior a sua prioridade, mais cedo ela será executada. Na Figura 3 pode-se ver que a Tarefa A (TaskA) tem maior prioridade que a Tarefa B (TaskB).

O estado da tarefa indica se a tarefa está pronta para executar, já está executando ou se ela está esperando algum recurso (um byte da porta serial, por exemplo). Somente uma tarefa pode estar no estado "executando" e que as outras tarefas só poderão entrar em execução se estiverem no estado "pronto" e ainda não tiverem sido executadas naquele ciclo. As Tarefas podem ser executadas apenas uma vez a cada ciclo, quando um sinalizador de execução marca como "não executada". Este Sinalizador é limpo em todos os Descritores ao final do ciclo. Algumas Tarefas podem, no entanto, serem executadas mais de uma vez em um ciclo, que é o caso da Tarefa Vazia (Idle), criada para testes e verificação do funcionamento do Escalonador.

Cada Descritores de Tarefa está necessariamente associado a uma área de memória chamada Contexto e um ponteiro indica uma área de memória previamente reservada pelo desenvolvedor e que é o repositório das variáveis locais da

tarefa e do contexto da tarefa ao se efetuar uma troca de tarefas. Quando ocorre uma interrupção no temporizador (*timer*) de preempção, o Contexto da tarefa é salvo na sua própria pilha e o controle é passado para o Escalonador, que seleciona a próxima tarefa a ser executada. Depois disso, é feita a restauração do contexto da tarefa selecionada, de forma a continuar a execução dela exatamente de onde havia sido parada. O contexto da tarefa é formado pelo valor da próxima instrução a ser executada, pelo registrador de status (estado) do processador e pelos valores dos demais registradores do sistema, empilhado nesta mesma ordem. A Figura 4 mostra o uso do Descritor de Tarefas, com ponteiros para o código (entry Task A) e o Contexto (Stack A). Como o processador tem arquitetura Harvard o código e a pilha estão em diferentes áreas de memória.

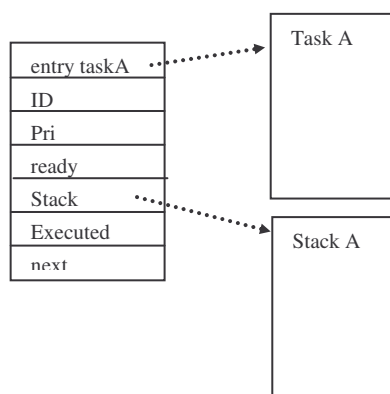


Figura 4 – Uso do descritor de tarefa

Somando-se a quantidade de instruções dos procedimentos de escalonamento, salvamento e restauração de contexto, tem-se o total de 187 instruções necessárias para efetuar a troca de uma tarefa, consumindo relativamente pouco tempo, cerca de 5 μ s, considerando-se que o processador esteja funcionando a 40MIPS (milhões de instruções por segundo).

IV. TAREFAS DO CONVERSOR SERIAL-IP

O conversor serial-IP está trabalhando com 4 Tarefas: Conexão, Envio de Bytes, Recepção de Bytes, Gerência da Conexão IP. Uma quinta Tarefa, a de Inicialização não fica na Fila de Tarefas prontas pois é executada apenas quando da energização da placa CSI ou de uma possível falha física do módulo celular. O Escalonador só começa a funcionar efetivamente após o término com sucesso da Inicialização, que implica na verificação do funcionamento do regulador de tensão e do módulo celular. Ainda, uma sexta Tarefa, de Configuração, fica em suspenso até que um byte seja recebido pela porta de configuração – Infravermelha.

O funcionamento das Tarefas, como descrito abaixo, é feito com o uso do Escalonador. Para fim de testes das funcionalidades das rotinas dos protocolos PPP, IP, TCP, UDP e interface com as portas seriais foi produzida uma versão preliminar sem o Escalonador.

A. Tarefa de Conexão

Esta Tarefa inicializa o modem do módulo celular, envia mensagem em protocolo PPP para iniciar enlace com a Operadora e estabelece conexão em TCP com o Servidor. A Tarefa só avança nas etapas se a anterior estiver completada com sucesso.

O enlace PPP tem um quadro de inicialização diferente para cada Operadora e é praticamente só esta a diferença na Conexão. A troca de Operadora GSM se dá com a troca do “GSM card” e com a configuração da mensagem inicial do enlace PPP, o que torna muito flexível este sistema. Durante os testes pelo menos duas Operadoras GSM (GPRS) foram utilizadas com total transparência para o Servidor e para os equipamentos de automação (simulador de IED e simulador da Estação Central). Ao término do enlace PPP o CSI tem um endereço IP válido para a rede da Operadora. Este endereço usualmente é um 10.xx.xx.xx, não válido, portanto, para a Internet.

O Servidor, por sua vez, tem um endereço fixo, válido na Internet. A conexão TCP é estabelecida com este endereço. Para que isto ocorra uma mensagem de conexão para o endereço do Servidor é enviado para a Operadora. Esta faz com que o endereço IP intranet do CSI seja convertido em um endereço válido via um tradutor de endereços de rede (NAT – *Network Address Translator*). O Servidor recebe então uma mensagem de conexão deste endereço traduzido e deve reconhecer neste o CSI. A mensagem de autenticação é criptografada e uma vez reconhecida pelo Servidor o endereço recebido é atualizado em uma tabela interna. Este endereço será utilizado até que uma nova mensagem de autenticação seja recebida pelo mesmo CSI, geralmente ocasionado por uma queda de conexão.

O uso de rede privativa – VPN – pode simplificar este procedimento de conexão e mesmo permitir uma maior estabilidade na conexão. A implementação da alternativa VPN não implica em alterações no programa da CSI.

A Tarefa de Conexão permanece da fila de prontos e pode ser ativada por uma mensagem da Tarefa de Gerenciamento se esta perceber falha na conexão.

B. Tarefa Envio de Bytes

Esta Tarefa recebe bytes do IED através da porta serial, monta um quadro de dados, criptografa este quadro e agrupa em palavras de 16 bits. Esta etapa é necessário devido ao microcontrolador ser de 16 bits.

Após o agrupamento verifica se a conexão está aberta. Se não estiver envia uma mensagem de erro para o Gerenciador e aborta o envio dos bytes. Se houver conexão empacota os dados em TCP ou UDP, em IP e finalmente em PPP e envia ao módulo celular.

Se for utilizado TCP o Gerenciador se encarrega de garantir a entrega, com repetição de envio, se necessário. Se for utilizado o UDP, os dados são simplesmente enviados uma única vez. Todo o controle de fluxo fica a cargo do IED e do protocolo de automação. Embora aparentemente menos confiável o uso de UDP tende a ser muito mais econômico. Testes estão sendo realizados para determinar a diferença de desempenho e uso de banda.

C. Tarefa Recepção de Bytes

Esta Tarefa recebe o quadro PPP/IP/TCP ou UDP da porta de comunicação com o módulo celular e verifica se trata de uma mensagem da rede ou de dados a serem enviados ao IED. No caso de serem dados, depois de desempacotar PPP/IP/TCP ou UDP, faz a descryptografia, separa de 16 para 8 bits e envia os dados, byte a byte, pela porta serial do IED.

D. Tarefa Gerenciadora de Conexão

Esta Tarefa verifica a existência ou não de conexão com o servidor além de processar informações recebidas das outras Tarefas. Executa funções do tipo solicitar reenvio no caso de falha de entrega de um quadro TCP e verificação periódica da conexão TCP.

Identifica também o funcionamento básico do módulo celular e em caso de falha deste pode solicitar seu desligamento (corte de alimentação) e religamento.

V. O SERVIDOR

A descrição do Servidor está detalhada a seguir. A fim de se ter um Servidor para testes do sistema foi desenvolvido um programa sob o sistema Windows Server 2003, construído em cima de duas funcionalidades principais: uma comunicação serial [5] e uma comunicação de sockets TCP/IP [2] através das quais blocos de dados são enviados e recebidos. Este software foi desenvolvido em C# [1,9] através do ambiente de desenvolvimento livre SharpDevelop.

A versão final está sendo desenvolvida em plataforma LINUX, Slackware 2.4.22, distribuição 9.1 ou superior e programada em C++.

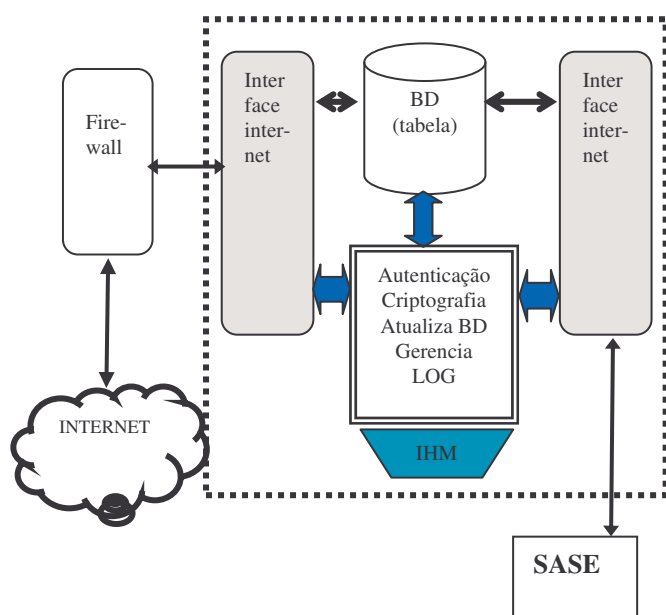


Figura 5 – Diagrama em blocos do Servidor

A Figura 5 mostra um diagrama em blocos do Servidor, com sua interface com o Firewall e o Centro de Operações, mostrando aqui o programa SCADA da COPEL, o SASE – Sistema de Automação de Ses. O Servidor possui interfaces ethernet com o Firewall e o SASE, uma tabela de configuração dos clientes, as rotinas de gerenciamento das conexões e uma interface homem máquina simplificada, utilizada apenas para inicialização do programa e definição dos arquivos de configuração.

A. Requisitos do Servidor

A seguir estão listados os Requisitos do Servidor.

- 1) O Servidor atuará de forma muito parecida com um NAT (*Network Address Translation*) servindo comunicações entre clientes (remotas) e o sistema SCADA da COPEL (SASE) através do protocolo IP
- 2) O Servidor irá operar na forma *client/server*, onde os clientes terão o papel de iniciar a comunicação com o gateway através de sockets TCP
- 3) As mensagens entre os clientes e o Servidor serão criptografadas
- 4) Haverá um contador de bytes transmitidos/recebidos (total e carga útil) na comunicação entre o servidor e os clientes. Este contador deverá ser salvo para posterior contabilização por um determinado período através de algum utilitário
- 5) O Servidor terá um arquivo de configuração de parâmetros de operação que será lido e criticado na inicialização do sistema, um arquivo de remotas válidas que será utilizado para autenticação e um arquivo de mensagens (LOG)
- 6) A gravação do arquivo de mensagens (LOG) deverá ocorrer em função de um nível de detalhamento e operar em modo circular. Os parâmetros que governam estas operações serão mantidos no arquivo de configuração
- 7) Ao final do processo de inicialização, o Servidor abrirá um socket TCP (em modo “*Listen*”) para atender a conexão com o SASE
- 8) Após o aceite da conexão com o SASE, o Servidor abrirá um novo socket TCP (Main Socket) no modo “*Listen*”, cuja função será atender novas conexões de clientes (CSI)
- 9) Após o aceite e autenticação de um novo cliente, o sistema fará a abertura de uma nova conexão TCP individual com este cliente, colocando o socket em modo “*Receive*”
- 10) Uma vez estabelecida esta conexão TCP entre o cliente e o Servidor, esta permanecerá ativa por tempo indeterminado ou até que haja algum evento de *reset* da comunicação desse socket
- 11) Também após este aceite, o Servidor atualizará o estado do socket cliente para conectado, contabilizando a quantidade atual de sockets conectados e gravando mensagem no arquivo de LOG conforme detalhamento solicitado pelo arquivo de configuração
- 12) Para cada socket criado para comunicação com o lado cliente, um socket TCP será criado também para atendi-

mento do lado do SASE, variando-se o valor da porta (uma porta por cliente)

- 13) Se o SASE tentar uma comunicação com um cliente, cujo socket correspondente do lado Internet não estiver ativo, esta comunicação será descartada.

VI. AGRADECIMENTOS

Este projeto está sendo desenvolvido dentro do âmbito de projetos de Pesquisa & Desenvolvimento ANEEL, Ciclo 2002/2003 e 2003/2004 da COPEL Distribuição, em conjunto com o Lactec e a UFPR. Participam deste projeto, além dos autores nomeados, o analista Reggie Vivekananda (Lactec – desenvolvimento do Escalonador); Eng. Álvaro H Costa (Lactec – contato com Operadoras Celular); Eng. Alu-ízio Grochoki; analistas Leslie H Watter e Ana Flavia B de Godoi (bolsistas Lactec – desenvolvimento do Servidor).

VII. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] M Barr, - "Programming Embedded System in C and C++" O'Reilly, 1999
- [2] D E Comer, D E., "Internetworking with TCP/IP, Vol. I – Principles, Protocols and Architectures", Prentice Hall, 2000
- [3] J-M Farines, J S Fraga, R S de Oliveira, "Sistemas de Tempo Real", Escola de Computação 2000 (USP), 2000 Disponível <http://www.lcmi.ufsc.br/gtr/livro/principal.htm>
- [4] E A Hall, "Internet Core Protocols – The Definitive Guide", O'Reilly, 2000.
- [5] J Hind, "Serial Comm - Use P/Invoke to Develop a .NET Base Class Library for Serial Device Communications", *MSDN Magazine*, Oct. 2002.
- [6] M Mitchell, J Oldham; A Samuel, "Advanced Linux Programming", New Riders Publishing, 2001
- [7] A C Shaw, "Sistemas e Software de Tempo Real", Editora Bookman, 2003
- [8] A S Tanenbaum, "Redes de Computadores", Editora Campus, 1994.
- [9] K Watson, "Beginning Visual C# - Programming", Maximum Press, 2002.
- [10] BluesPortTool, (15/set/2005) Disponível <http://www.bluebitter.de>
- [11] V C Zambenedetti, F R Coutinho, A Barbiero, R P Siqueira, J G Pereira, R A Hexsel, "Uso de comunicação celular digital utilizando 2.5G para sistemas de automação de energia elétrica", in Proc. IV SIMPASE, 2005