

Protótipo de Rede Local baseado em Protocolo Poll-Select, utilizando Interface RS-232C como meio de acesso

Bruno Queiroz Jatene. É graduado em Tecnólogo em Processamento de Dados pela Universidade da Amazônia - UNAMA (1996), possuindo como área de interesse a Tecnologia da Construção de Redes de Computadores. Atua como analista de suporte e consultor em redes locais e de longa distância, atualmente (1997) cursa especialização em Redes de Computadores pela Universidade Federal do Pará.

RESUMO: Este trabalho apresenta um protótipo de rede local baseado em protocolo Poll-Select, utilizando Interface RS-232C como meio de acesso. O modelo Redes de Petri foi utilizado como descrição formal deste protocolo. Padronizado pela Burroughs (*hoje Unisys*), para controle de terminais em uma linha *multiponto*, a técnica de *polling* é utilizada como estratégia para o controle de acesso ao meio de transmissão. A linguagem *PASCAL* foi selecionada para codificação dos programas que integram a BJNET.

A rede proporciona a interligação das estações ao servidor através de uma topologia em *barra comum* cujo meio físico (media) de transmissão é o *par trançado*. Servidor e estações devem ser conectados à barra através de uma porta serial padrão *RS-232C*. Os serviços disponíveis são: compartilhamento de uma impressora instalada no nó primário, e a troca de mensagens, entre nós secundários, através de correio eletrônico.

1. INTRODUÇÃO

Não há dúvidas de que, com relação a sistemas de informação, *conectividade* e *interoperabilidade* têm se tornado palavras de ordem. Passou o tempo em que diversos usuários disputavam os recursos de um computador central (*mainframe*) de um famoso fabricante. A crescente disseminação da computação dentro das organizações alavancou o surgimento de um grande número de empresas no setor de informática. Acirrou-se a disputa de mercado e algumas dessas

novas empresas foram se especializando no desenvolvimento e/ou produção de produtos específicos, isso vem acontecendo tanto em relação ao hardware como ao software. Diante desse novo cenário, o que as organizações estão buscando hoje é aumentar a eficiência e reduzir custos, integrando produtos de diversos fabricantes em um mesmo sistema de informação. Em outras palavras, não basta ser famoso, vence quem tem o produto com a melhor relação custo/benefício.

Por representarem a própria materialização dos conceitos de conectividade e interoperabilidade as *redes de computadores* e *sistemas distribuídos* estão sem dúvida no cerne dessa questão.

Buscando atender essa nova demanda, as instituições de pesquisas, principalmente nos grandes centros, têm despendido enorme esforço no sentido de estudar temas relacionados a essas duas áreas. Dentro desse escopo, um tópico muito importante diz respeito aos *protocolos de comunicação*.

Entre os muitos desenvolvidos e que se prestam ao propósito especificação formal de protocolos, descreveremos o método conhecido como *Redes de Petri*. Um exemplo prático da aplicação do método será mostrado através da especificação e implementação do protocolo poll-select. Este protocolo foi padronizado pela Burroughs (hoje Unisys) para permitir a comunicação de terminais em uma rede com topologia em barra comum baseado em um método de acesso centralizado. Adaptamos o referido protocolo para ser utilizado em uma pequena rede local, a qual batizamos BJNET, objetivo maior de nosso trabalho.

2. INTERFACE RS-232C

No mundo da comunicação de dados, os equipamentos como computadores pessoais, terminais e portas de computador são chamados de Equipamentos Terminais de Dados (ETD). Modems e outros dispositivos de comunicação são conhecidos como Equipamentos de Comunicação de Dados (ECD).

A interface digital é um dispositivo de E/S que torna possível a compatibilidade entre um ETD e um ECD. A compatibilidade

é obtida pela padronização a nível internacional da interface.

O laboratório BELL e a EIA (Electronic Industries Association) especificaram RS-232C. Paralelamente, o CCITT (International Telegraph and Telephone Consultative Committee) padronizou a interface terminal-modem, através das Recomendações V.24/V.28, compatível com a RS-232 C.

2.1 Características de Sinal da RS-232

A Interface RS-232 especifica 25 circuitos de ligação, ou condutores, que controlam o fluxo de dados entre o ETD e ECD. O sinal de cada um desses condutores ocorre de acordo com uma transição de voltagem pré-definida. Portanto, os padrões estabelecem níveis de tensão para transmissão de dados, onde o estado lógico 1 (marca) é definido como sendo uma tensão negativa entre -3 e -15 volts, enquanto o estado lógico 0 (espaço) é definido como uma tensão positiva entre +3 volts e +15 volts, tudo referenciado ao terra de sinal com previsão de queda de tensão de 12V ao longo das linhas de transmissão.

Como os receptores são obrigados a reconhecer sinais de no máximo ± 3 volts, sobra uma margem de segurança (região de transição) de 6 volts entre os níveis 1 e 0, o que contribui para aumentar a imunidade a ruídos e a diferença de potencial de massa. O estado do sinal não necessariamente será identificado de forma única quando a tensão estiver na região de transição.

Verifica-se abaixo uma visão das faixas de voltagem do Circuito de Ligação definidas pelo padrão RS-232C:

+15V	Faixa Positiva - Função ON
+3V	Região de Transição
-3V	Faixa Negativa - Função OFF
-15V	

Tabela 2.01 - Voltagens do Circuito de Ligação do padrão RS-232C

A norma RS-232 recomenda o uso de cabos curtos, com comprimento de até 15 metros, embora ressalte que cabos mais longos são permitidos, desde que resultem numa capacitância de carga inferior a 2,5 nF.

2.2 Ligação Cross-Over

Quando se deseja ligar dois equipamentos de mesmo tipo (terminal-terminal, modem-modem) pelo seu lado digital, deve-se utilizar o cabo Cross-Over. Este cabo faz a inversão (cruzamento) entre os sinais.

A interligação dos pinos da interface pelo cabo Cross-Over é a seguinte:

O cruzamento dos pinos 2 e 3 é necessário, pois o que é transmissão num dispositivo é recepção no outro, e vice-versa;

O mesmo é aplicado para as bases de tempo de recepção e transmissão dos pinos 15 e 17. Quando a base de tempo para a transmissão é fornecida pelo terminal, o cruzamento é entre os pinos 17 e 24;

Os pinos 4 e 5 são jumpeados, pois não há necessidade do delay RTS/CTS (não se coloca portadora na linha), e ligados ao pino 8 da recepção do outro equipamento (simulação da detecção de portadora);

O pino 20 da transmissão e o pino 6 da recepção são conectados;

Os pinos 1 dos dispositivos são interligados;

Os pinos 7 dos dispositivos são interligados.

Caracter	Descrição	Significado
RTS	Request To Send	Transmissor solicita permissão para transmitir
CTS	Clear To Send	Receptor autoriza transmissão
DTR	Data To Receive	Existe dado no buffer de recepção
DTS	Data to Send	Existe dado no buffer de transmissão
ACK	Acknowledgment	Receptor recebeu sem erro o dado enviado pelo transmissor

Tabela 2.02 - Especificações de alguns pinos

3. PROTOCOLO POLL-SELECT

Através de um exemplo prático mostraremos a especificação e implementação do protocolo *poll-select*. Este protocolo, padronização da Burroughs (*hoje Unisys*) para controle de terminais em uma linha *multiponto*, utiliza a técnica de *polling* como estratégia para o controle de acesso ao meio de transmissão. A transmissão é *assíncrona* (*caracter a caracter*) em modo *half-duplex*.

Faremos a especificação do protocolo *poll-select*, inicialmente de maneira informal, posteriormente apresentaremos uma especificação formal utilizando o método de Redes de Petri. Finalmente, baseados nessa especificação implementaremos o mesmo e apresentaremos a aplicação prática, utilizando-o como protocolo de uma pequena rede local (Rede BJNET).

3.1 Especificação Informal

O protocolo *poll-select* incorpora procedimentos para controle do acesso

de nós secundários em uma rede multidrop (*em barra comum*).

A padronização deste protocolo apresenta os seguintes procedimentos:

Procedimento	Descrição
Burroughs Standard Poll-Select	Procedimentos de poll e de select padrões.
Fast Select	Este procedimento é utilizado sempre que se deseja transmitir uma mensagem a um nó secundário (<i>originalmente para um terminal de video</i>) sem verificar se o nó está pronto para recepção.
Modo Contenção	Este procedimento permite ao nó primário interrogar a rede depois de colocar todos os nós secundários em modo de contenção.
Broadcast Select	Este procedimento é utilizado quando se deseja transmitir uma mensagem a todos os nós secundários da rede.
Group Select	Semelhante ao Broadcast Select mais limitado a um grupo de nós secundários.
Group poll	Em vez de interrogar cada nó secundário individualmente, o nó primário pode dentro desta opção interrogar o grupo.

Tabela 3.01 - Procedimentos Padrão do Protocolo Poll-Select

Dos procedimentos apresentados acima, é interesse de nosso estudo: *Burroughs Standard Poll-Select*, *Fast Select* e *Broadcast Select*.

☞ Burroughs Standard Poll-Select

O procedimento *Burroughs Standard Poll-Select* incorpora os procedimentos de poll e de select padrões para controle do acesso de nós secundários em uma rede multidrop, em linhas privadas ou comutadas.

O processador central ou nó primário interroga continuamente os nós secundários da rede. A frequência de interrogação (*polling*) é função do sistema de programação, velocidade de transmissão, carga de trabalho, etc.

Quando um nó secundário tem uma mensagem a transmitir este deverá aguardar uma interrogação do nó primário para iniciar a transmissão. Terminada a transmissão o nó

primário responde com um único caracter ACK se não foi encontrado erro, ou um único NAK se a recepção não foi adequada.

Ao receber um caracter ACK o nó secundário responde com EOT, o que faz com que o processador central repita a rotina de poll para o próximo nó secundário e assim sucessivamente.

Se o nó primário tiver recebido uma mensagem com erro de paridade em qualquer caracter ou um BCC incorreto, responderá com um caracter NAK isolado. A recepção de NAK no nó secundário faz com que o mesmo retransmita a mensagem. A mensagem deve ser retransmitida por um número máximo de vezes, estourado o número máximo de retransmissões, o nó secundário deve enviar um EOT e então o processador central irá interrogar o próximo nó secundário.

Se o processador central não recebe nenhuma resposta de um nó secundário

interrogado, ocorre um time-out e o computador envia um EOT, seja na forma de caracter isolado, seja como parte da sequência poll-select seguinte. Se o nó secundário estava pronto para transmitir, continuará mantendo a mensagem pronta para ser transmitida ao se interrogar o nó secundário na oportunidade seguinte.

Na implementação adotamos um método simples para melhorar a performance da rede. O método consiste em retirar da lista de poll, por determinado número de ciclos (6 em nossa implementação), uma estação (ou nó secundário) que não responda a um procedimento de poll pois se isso realmente aconteceu é bem possível que a estação esteja inativa.

Se o nó primário tem uma mensagem a enviar para um nó secundário este deve transmitir uma mensagem de seleção. A seleção de um nó secundário que não se encontra pronto para receber uma mensagem faz com que o mesmo responda com um NAK ao nó primário caso contrário responderá com um ACK, com o que o nó primário inicia o envio da mensagem. Se o nó secundário receber a mensagem sem erro, responderá com um novo ACK. Existindo algum erro, será enviado NAK. Finalmente se o nó primário por algum motivo não enviar a mensagem completa (*isto é, com ETX e BCC*), o nó secundário não responderá, causando um time-out. Em todos os casos o nó primário pode terminar a transação enviando um EOT (*isolado ou encabeçando o poll-select seguinte*).

3.2 Especificação em Redes de Petri

Abaixo temos a especificação dos nós primários e secundários da rede BJNET, utilizando o método de descrição formal de

protocolos Redes de Petri. A versão completa deste trabalho apresenta todos os procedimentos descritos formalmente com suas análises, incluindo os procedimentos de Fast Select, Broadcast Select e Poll.

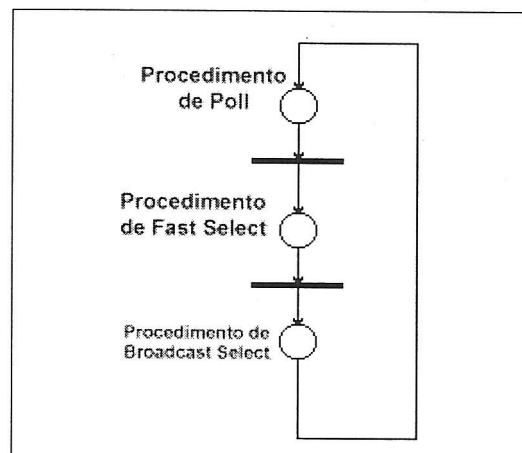


Fig 3.02 Especificação do Nó Primário

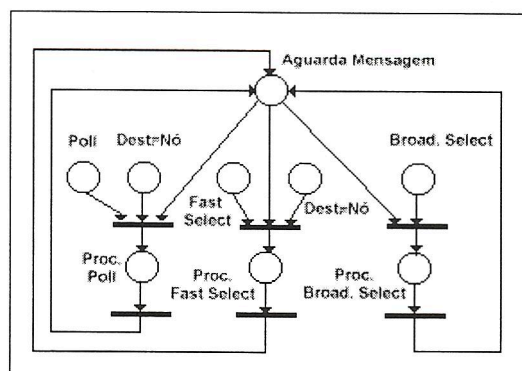


Fig 3.03 Especificação do Nó Secundário

4. A Rede BJNET

Trata-se de uma rede em barra comum que utiliza polling como estratégia para controle de acesso e cujos objetivos são possibilitar:

- O compartilhamento de uma impressora instalada no nó primário;
- A troca de mensagens, entre nós secundários, através de correio eletrônico.

A arquitetura da BJNET e a correspondência entre as camadas OSI são mostrados na figura abaixo.

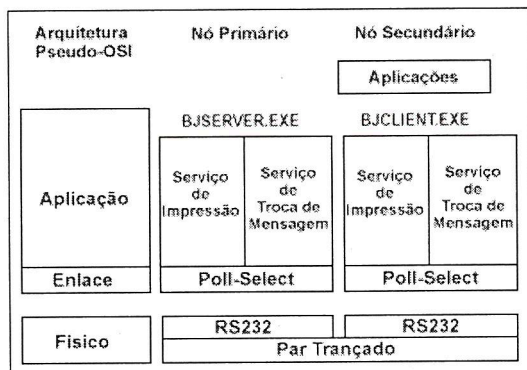


Fig.4.1 Arquitetura da Rede BJNET

Para um melhor entendimento, dividimos a figura em três blocos funcionais. O primeiro (*de baixo para cima*) é constituído pela media de transmissão (*par trançado*) e a interface RS232 que corresponde ao *nível físico*. O segundo bloco corresponde às camadas de *enlace de dados* (*protocolo Poll-Select*) e *aplicação* (*impressão e troca de mensagens*) do modelo de referência OSI, e é implementado pelos drivers da estação (*BJCLIENT.EXE*) e do servidor (*BJSERVER.EXE*).

Finalmente o terceiro bloco é constituído pelas aplicações carregadas pelos usuários nas estações (*ex: processadores de textos, planilhas eletrônicas, aplicações de base de dados, etc*).

Algumas camadas OSI não foram implementadas e por isso não aparecem. A razão disso é que nem todas são necessárias para a implementação em questão. Podemos citar como exemplo a camada de rede, totalmente desnecessária uma vez que não temos roteamento de mensagens, que é a principal função desta camada no modelo de referência OSI.

Para adequar nossa linguagem à das redes locais faremos uma referência ao nó primário como servidor e aos nós secundários como estações.

4.1 Características de hardware

Quanto ao hardware a rede BJNET proporciona a interligação das estações ao servidor através de uma topologia em *barra comum* cujo meio físico (media) de transmissão é o *par trançado*. Servidor e estações devem ser conectados à barra através de uma porta serial padrão *RS-232C* conforme a figura abaixo.

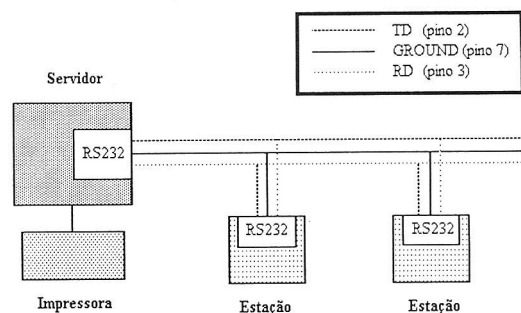


Fig. 4.2 Características Topológicas e de Hardware da Rede BJNET

Quanto à interface RS-232C, os pinos utilizados são o 2 (*SD-Send Data*), 3 (*RD-Receive Data*) e 7 (*Ground*). Os pinos 7 correspondem ao aterramento e devem estar todos interligados entre si. Os pinos 3 (*RD*) das estações devem estar conectados ao pino 2 (*TD*) do servidor. Os pinos 2 (*SD*) das estações devem estar conectados ao pino 3 (*RD*) do servidor.

Na implementação aqui apresentada utilizamos computadores do tipo AT 486 todos conectados à barra através da porta serial COM1.

Como a estratégia de controle do acesso à barra emprega a técnica de *polling*, um dos

microcomputadores atuará como nó primário (*servidor*). O nó primário então será responsável por prover serviços de impressão e troca de mensagens, logo, uma impressora deve estar conectada a porta paralela (LPT1) do mesmo.

4.2 Características de Software

As características de software dizem respeito aos aspectos envolvidos na implementação dos drivers *BJSERVER.EXE* e *BJCLIENT.EXE*. Como dissemos, estes drivers possibilitam a comunicação entre servidor e estações utilizando o protocolo *poll-select* a nível de enlace de dados. A transmissão é em modo *assíncrono* (*caracter a caracter*) à velocidade de 9600 bps, 1 start bit, 8 bits de dados e 1 stop bit sem paridade (*controle pelo BCC*).

4.3 Sistema Operacional

O driver das estações e do servidor foram codificados originalmente para serem executados sob o sistema operacional *MS-DOS 6.0* entretanto nada impede que drivers sejam implementados para outros sistemas operacionais, desde que obedeçam o protocolo *poll-select*. Já prevendo esta possibilidade desenvolvemos drivers de forma a tornar o código facilmente por-

Onde:

Campo	Tipo	Tam.	Descrição
SAP	Num	1	Este campo corresponde a um Service Access Point e no caso de impressão deve ter o valor 1
Mensagem	Alfa	Ilimitado	Mensagem a ser impressa

Tabela 4.02 Descrição dos Campos da Mensagem Para Impressão

O serviço de *troca de mensagens* possibilita ao usuário em qualquer estação enviar mensagens ou até mesmo arquivos tipo textos para outra estação. A mensagem é colocada em uma caixa postal que poderá ser

tável, bastando para isso modificar aquelas funções dependentes do ambiente operacional onde os mesmos deverão ser carregados.

4.4 Linguagem de Programação

A linguagem *PASCAL* foi a selecionada para codificação dos programas que integram a BJNET. Para tanto levou-se em consideração principalmente seu aspecto didático além da sua natural vocação para implementação de protocolos de comunicação.

4.5 Serviços Disponíveis

A rede BJNET provê os serviços de: Impressão remota e Troca de mensagens.

Impressão remota é o serviço através do qual as estações poderão compartilhar a impressora conectada ao servidor. Para isso toda impressão enviada à porta paralela, controlada pelo driver de rede da estação (LPT1, LPT2 ou LPT3), deverá ser capturada e enviada ao servidor. O servidor por sua vez deve receber a mensagem e imprimí-la.

O formato da mensagem de impressão é o seguinte:

SAP	Mensagem
-----	----------

Tabela 4.01 Formato da Mensagem de Impressão

consultada a qualquer momento. Todas as tarefas relativas a envio e recebimento de mensagens são implementadas através do programa *BJMAIL.EXE* que pode ser executado obedecendo-se a seguinte sintaxe:

BJMAIL [<End. Destino> <Arquivo>] [/?]

Onde:

Parâmetro	Descrição
End. Destino	Endereço da estação para a qual a mensagem deve ser enviada
Arquivo	Arquivo a ser enviado
?	Help

Tabela 4.03 Descrição dos Parâmetros para Execução do BJMAIL

Caso o usuário queira consultar a caixa postal ou mesmo editar uma pequena mensagem e enviá-la a uma estação, basta executar o programa BJMAIL.EXE sem parâmetros. O programa oferece uma interface simples, a caracter e orientada a menu.

```

*****
*                BJNET                *
*                Correio Eletrônico    *
*****
1. Enviar Mensagens
2. Ler Caixa Postal
3. Sair

Opção →

```

Fig. 4.3 Menu Principal do BJMAIL.EXE

O formato de mensagens para uma outra estação é o seguinte:

SAP	End. Destino	Mensagem
-----	--------------	----------

Tabela 4.04 Formato de Mensagem de uma estação a outra

Onde:

Campo	Tipo	Tam.	Descrição
SAP	Num	1	Este campo corresponde a um Service Access Point e no caso de mensagens deve ter o valor 2
End. Destino	Alfa	2	Endereço da estação para a qual a mensagem deve ser enviada. "00" se for uma mensagem tipo broadcast.
Mensagem	Alfa	Ilimitado	Mensagem a ser enviada

Tabela 4.05 Descrição dos Campos da Mensagem Para Correio Eletrônico

Para simplificar a implementação não existe um tamanho de mensagem máximo a ser enviado pela estação, permitindo que uma estação monopolize o barramento durante a transmissão de uma mensagem. Essa certamente é uma característica pouco interessante caso as mensagens a serem transmitidas através da rede sejam freqüentemente grandes. A solução desse problema pode ser um bom melhoramento a ser feito em trabalhos futuros.

4.6 Drivers de Rede.

Para possibilitar a comunicação entre o servidor e as estações da rede foram implementados dois drivers, um para ser carregado nas estações e outro no servidor. Os detalhes de implementação relacionados ao MS-DOS, como manipulação de interrupções, podem ser melhor esclarecidos pela bibliografia relacionada.

i) Driver do Servidor

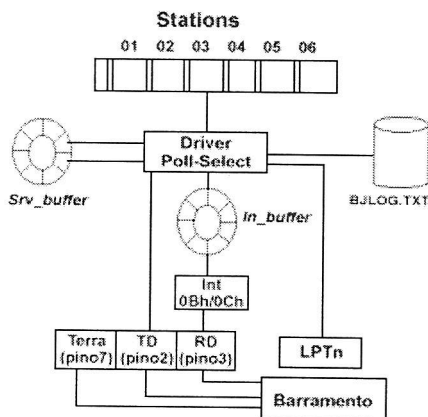


Fig. 4.4 Driver do Servidor (BJSERVER.EXE)

O computador que atuar como servidor deve carregar o programa BJSERVER.EXE. Esse driver controla a porta serial COM1. O servidor controla a rede executando o

procedimento de poll. Caso uma estação interrogada deseje enviar mensagem esta deve ser capturada pelo driver e armazenada em uma lista circular (*in_buffer*) a mensagem é então analisada para saber se trata-se de uma mensagem de impressão ou uma mensagem a ser enviada para outra estação. No primeiro caso os caracteres são enviados para a porta paralela LPT1. No segundo caso se a mensagem for endereçada a uma estação específica os dados devem ser armazenados no buffer reservado à esta estação no array *station* e posteriormente transmitidos através da porta serial utilizando o procedimento de *fast select*. Caso seja uma mensagem tipo broadcast, esta deverá ser armazenada em um buffer apropriado (*srv_buffer*) e posteriormente transmitida através da porta serial utilizando o procedimento de *broadcast select*.

O programa BJSERVER.EXE pode ser executado em modo monitor, assim, todo protocolo pode ser monitorado, os caracteres que trafegam no barramento são apresentados no vídeo e também gravados no arquivo BJLOG.TXT.

A captura e armazenamento no buffer de entrada (*in_buffer*) dos caracteres recebidos através porta serial é feita pela substituição da rotina da interrupção original da BIOS (*0Bh para COM1 ou 0Ch para COM2*) pela rotina *Serial_to_buffer*.

A sintaxe para carga do driver é a seguinte:

BJSERVER [/M] [/?]

Parâmetro	Descrição
/M	Executa BJSERVER em modo monitor
?	Help

Tabela 4.05 Descrição dos Parâmetros para Execução do BJSERVER

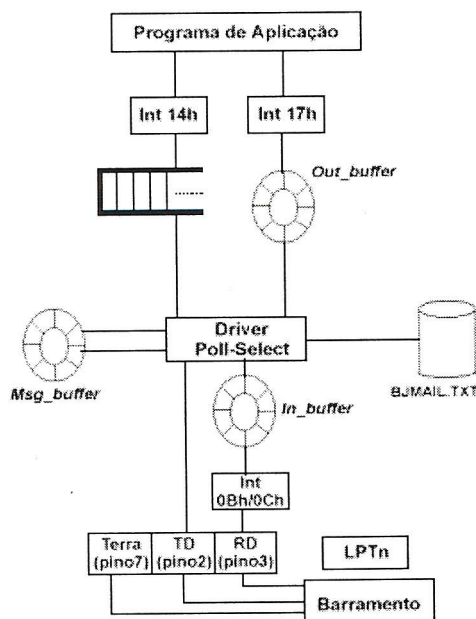
ii) *Driver da Estação.*

Fig. 4.5 - Driver da Estação (BJCLIENT.EXE)

Toda estação que deseje utilizar-se da BJNET deverá carregar o driver BJCLIENT.EXE. Este é um programa do tipo TSR (*Terminate But Stay Resident*) que implementa, além do protocolo poll-select, o conceito de multiprogramação. Na prática isto significa que o mesmo quando carregado na memória permite que o usuário permaneça no prompt do MS-DOS, desta feita compartilhando a CPU em 50% com o driver da estação. Assim sendo, o usuário tem condições de executar outros programas como editores de textos, planilhas eletrônicas e etc. Os programas deverão apresentar uma ligeira queda de performance, o que é perfeitamente normal, visto que não dispõem de 100% do tempo de CPU. Na carga do driver deve ser especificado o *endereço da estação* e a *porta paralela (LPT1, LPT2 ou LPT3)* a ser capturada pelo driver. Esse driver

controla a porta paralela especificada e a porta serial COM1. Toda impressão direcionada para a porta paralela controlada pelo driver é capturada pelo mesmo e armazenada em uma lista circular (*out_buffer*) os caracteres deste buffer devem ser enviados através da porta serial obedecendo o protocolo poll-select.

A captura e armazenamento no buffer de saída (*out_buffer*) dos caracteres enviados à porta paralela é feita pela substituição da rotina de interrupção 17h da BIOS pela rotina *Int_17h*.

A captura e armazenamento no buffer de entrada (*in_buffer*) dos caracteres recebidos através porta serial é feita pela substituição da rotina de interrupção (*0Bh para COM1 ou 0Ch para COM2*) da BIOS pela rotina *Serial_to_buffer*.

A substituição da interrupção 14h da BIOS tem como objetivo principal, proporcionar a comunicação, das novas aplicações carregadas pelo usuário, com o driver da estação. As novas funções implementadas pela nova rotina são acessadas via registrador ah. Assim sendo as funções implementadas pela nova rotina a ser executada pela 14h são:

- Solicitação de retirada do driver BJCLIENT da memória (registrador ah = 90);
- Verificação da presença do programa BJCLIENT na memória (registrador ah = 91);
- Passagem de caracteres correspondentes a uma mensagem a ser enviada (registrador ah = 92);
- Autorização para transmissão da mensagem (registrador ah = 93).

A sintaxe para carga do driver é a seguinte:

BJCLIENT <End. Estação> [/1 /2 /3 /R /?]

Onde:

Parâmetro	Descrição
<End. Estação>	Endereço da estação
/1	Captura mensagens enviadas a LPT1
/2	Captura mensagens enviadas a LPT2
/3	Captura mensagens enviadas a LPT3
/R	Retira o driver da estação
/?	Help do comando

Tabela 4.06 - Descrição dos Parâmetros para Execução do BJCLIENT

CONCLUSÕES E EXTENSÕES

Conforme demonstramos neste artigo, a Rede BJNET foi desenvolvida e implementada utilizando o Protocolo Poll-Select e a interface RS-232C como meio de acesso.

A especificação de protocolos de comunicação requer uma modelagem em termos formais. O formalismo é a própria razão da existência de outras etapas no ciclo de desenvolvimento dos protocolos de comunicação (*verificação e implementação*). Para atender a essas exigências torna-se imperiosa a pesquisa de métodos que sejam aplicáveis a tarefa de modelagem. Diversas técnicas, como as citadas no capítulo V, foram desenvolvidas e ainda se encontram em desenvolvimento. A modelagem formal não invalida a utilização da especificação informal como forma de apoio, ao contrário, a mesclagem das duas tende a tornar mais consistente a especificação como um todo.

A escolha do melhor método para especificação de protocolos certamente deve

levar em consideração o seu potencial para modelagem, mas também deve considerar a existência de ferramentas que viabilizem a execução de todo o ciclo de desenvolvimento, ou mesmo parte dele, de forma automática ou semi-automática. Atualmente, Redes de Petri ainda é uma técnica utilizada, entretanto, há uma grande tendência para utilização, cada vez maior, das TDFs Estelle, LOTOS e SDL padronizadas pela ISO.

A Rede BJNET, mais propriamente o protocolo Poll Select, foi projetada para funcionar segundo uma Pseudo OSI, segundo o método formal de especificação de Redes de Petri ao qual foi baseada.

O Protocolo Poll Select, assim como a interface RS-232C, ainda possuem certas limitações. O Protocolo tem uma desvantagem importante de trabalhar com um modo de operação Half Duplex, enquanto que a interface RS-232C não é muito utilizada em projetos de Rede. Existem outras desvantagens comentadas durante o trabalho.

Esse trabalho levou em consideração a implementação de um Protótipo de uma Rede Local em uma Linguagem de Programação Orientada a Caractere, podendo-se notar que a interface gráfica com o usuário não é tomada como prioridade. Um outro detalhe importante na construção da BJNET é que a implementação não atentou para o fato de limitar o tamanho da mensagem a ser transmitida por uma estação, podendo ter como consequência a alocação de uma determinada estação durante um tempo muito elevado no meio de comunicação.

Sugerimos para aqueles dispostos a estudar e refinar esse trabalho, é importante considerar o fato de uma próxima versão ser implementada em uma Linguagem de Programação Orientada a Objeto,

aconselhamos DELPHI (Produto da BORLAND), e também o tamanho da mensagem ser limitado de forma que uma estação não ocupe o meio de comunicação por um tempo muito elevado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ALMEIDA, Alvaro, Ferramentas Avançadas em Turbo Pascal, Ciência Moderna, 1992.
- [2] ALVES, Luiz. Comunicação de Dados. São Paulo, Makron Books, McGraw-Hill, 1992.
- [3] Artigos obtidos na INTERNET.
- [4] BOBOLA, Daniel T., Guia Incrível de Redes Locais. São Paulo, Ed. Makron Books, 1995.
- [5] BOCHMANN, Gregor V., A General Transition Model For Protocol and Communication Services, IEEE Transactions on Communications Vol 28 N. 4, April 1980.
- [6] CAMPBELL Joe, RS-232 Técnicas de Interface, EBRAS, 1986.
- [7] CARLOS, A. hauser, Modelagem Conceitual de Sistemas, EBAI, 1988
- [8] CCITT, Data Communication over the Telephone Network. Geneva, VII Plenary Assembly, Volume VIII, Fascicle VIII.1, november 1980.
- [9] DANTHINE, Andre A. S., Protocol Representation With Finite State Models, IEEE Transactions on Communications Vol 28 N. 4, April 1980.
- [10] DAVIS, William S., Sistemas Operacionais: Uma visão Sistemática. Rio de Janeiro, Ed. Campus, 1991.
- [11] DERFLER, Frank J., Guia de Conectividade; tradução [da 3. Ed. Americana] ARX Publicações. - Rio de Janeiro: Campus, 1995.
- [12] GIOZZA, William e Araújo José e Moura José e Sauv  Jacques, Redes Locais de Computadores Tecnologia e Aplicações, McGRAW-HILL, 1986.
- [13] HELD, Gilbert. Modem, o guia de referência completo. Rio de Janeiro, Editora Campus Ltda., 1992.
- [14] HERGERT, Douglas, Dominando o Turbo Pascal, Ciência Moderna, 1989.
- [15] HUGHES Larry, Low-Cost Networks and Gateways for Teaching Data Communications, ACM Press - SIGCSE - Bulletin V 21 n 1, Feb 1989, pp 6 a 10.
- [16] JACKMAN, J. e Medeiros D. J., A Graphical Methodology For Simulating Communication Networks, IEEE Transactions on Communications Vol 36 N. 4, April 1988.
- [17] LOWE, Doug, Redes Locais Para Leigos; Tradução Altair Dias Caldas de Moraes. São Paulo. Ed. Berkeley, 1994.
- [18] MATSUSHITA, Paulo Tetsuya, Instalação e Administração do Sistema Operacional Novell Netware 3.12. Ed. Érica
- [19] MONTORO, Fabio de Azevedo. Transmissão de dados e modem. São Paulo, Livros Técnicos Editora Ltda., 1990.
- [20] NOGUEIRA, José Marcos, Protocolos de Comunicação: Conceitos, Serviços, Especificação e Teste, Versão preparada para a V Escola Brasileiro-Argentina de Informática, 1991.
- [21] NORTON, Peter, Desvendando o PC e PS/2, Rio de Janeiro. Campus, 1991.
- [22] Novell Education, Course 200 - Network Technologies, Novell Press, 1995.
- [23] NUNES, José Renato Soares, Comunicação de Dados : conceitos básicos. Rio de Janeiro: LTC - Livros Técnicos e Científicos Ed. Sociedade Cultural e Beneficente Guilherme Guinle, 1989.

- [24] OLIVEIRA, Luís Antônio Alves de, Comunicação de Dados e Teleprocessamento: Uma abordagem básica. 3. Ed., São Paulo, 1989.
- [25] PETERSON, James, Petri Nets, Computing Surveys, Vol 9 n 3, Sep 1977.
- [26] REISIG, Wolfgang, Petri Nets, EACTS - Monographs on Theoretical Computer Science Vol 4, 1985.
- [27] RIBEIRO, Adagenor L., Uma Avaliação dos Métodos Para Definição de Requisitos em Engenharia de Software, Trabalho Individual CPGCC/UFRGS, Brasil 1990
- [28] SCOPUS Tecnologia, Manual do Terminal TVA 1800.
- [29] SILVA, Wladimir Lopes, Comunicação Entre Microcomputadores, Revista Microsistemas n 126, pp 46 a 55.
- [30] SILVA, Wladimir Lopes, Comunicação Entre Microcomputadores, Revista Microsistemas n 126, pp 46 a 55.
- [31] SILVEIRA, Jorge Luis da. Comunicação de Dado se Sistema de Teleprocessamento. São Paulo, Makron Books, McGraw-Hill, 1991.
- [32] SOARES, Luiz F. G. e LERUSALIMSKY R, Especificação de Protocolos Através de Redes de Petri com Temporização - O Protocolo de Acesso à Barra da Rede Local REDPUC, RBC V.3 n 3 1983;1984, pp 171 a 193.
- [33] SOARES, Luiz F. G., Redes Locais, Campus, 1986
- [34] SUNSHINE, C. e G. BOCHMANN, Formal Methods in Communication Protocol Design, IEEE Transactions on Communications Vol 28 N.4, April 1980.
- [35] TANENBAUM, Andrew S., Redes de Computadores (segunda edição), Campus, 1994
- [36] TAROUCO, Liane, Redes de Computadores Locais e de Longa Distância, McGRAW-HILL, 1986.
- [37] THIRY, C. C., M. e IBAGY, S. R.; SoftNET, Uma Rede Local de Baixo Custo; Trabalho de conclusão do Curso de Graduação, UFSC, 1990.
- [38] TISCHER, Michael, PC System Programing, Abacus, 1990.
- [39] YALLOUZ Carlos, Programas Residente no IBM PC, LTC, 1991.