

L.F.FRIEDRICH*, S.S. TOSCANI**, M.TORNQUIST***

SUMARIO

O artigo descreve o software de comunicação para uma rede local de micro-computadores que está sendo construída na UFRGS. A rede é do tipo barramento (ethernet) com controle de acesso distribuído. O software está operacional, em fase de testes.

ABSTRACT

The paper describes the communication software for a microcomputer local network that is being built at UFRGS. The network is ethernet like with distributed access control. The software is under testing.

*Engenheiro Mecânico (UPF,1978), aluno do Curso de Pós-Graduação em Ciência da Computação da UFRGS, concluindo sua dissertação de mestrado. Áreas de interesse: sistemas operacionais distribuídos e redes de computadores. CPGCC/UFRGS - Caixa Postal 1501 - Porto Alegre - RS - CEP 90.000.

**Engenheiro Eletricista (UFRGS,1967), Mestre em Informática (PUC/RJ,1969) Áreas de interesse: programação concorrente e sistemas operacionais. Professor Adjunto do DI/CPGCC da UFRGS. Caixa Postal 1501 - Porto Alegre-RS CEP 90.000.

***Tecnólogo em Processamento de Dados (UFRGS,1977), Mestre em Ciência da Computação (UFRGS,1982). Áreas de interesse: MUMPS, sistemas distribuídos e validação de sistemas. Professor assistente do DI/CPGCC da UFRGS. Caixa Postal 1501 - Porto Alegre - RS - CEP 90.000.

Este artigo apresenta o software de comunicação para uma rede local de microcomputadores que está sendo construída no Curso de Pós-Graduação em Ciência da Computação de UFRGS.

Os microcomputadores utilizados são os micros MAXXI da Polymax e a rede local é do tipo barramento (ethernet) com controle de acesso distribuído. O hardware de comunicação consiste basicamente de uma interface serial RS-232-C, com a inclusão de um pequeno circuito no conector DB-25, conforme descrito em [CLE 81]. O software foi escrito em linguagem de montagem Z-80, no sistema CP/M, ocupando cerca de 5 Kbytes de memória, e pode ser utilizado a partir de qualquer linguagem de alto nível que permita chamadas de subrotinas escritas em Assembler.

O software deverá ser embutido, mais tarde, no interpretador MUMPS/M, já implementado nos micros MAXXI [TOR 83], originando uma rede local MUMPS. Esta rede colocará à disposição dos usuários todas as facilidades para o desenvolvimento de aplicações normalmente oferecidas pelos sistemas MUMPS.

O software baseia-se nas recomendações da ISO para interconexão de sistemas abertos (OSI) [DAT 81] e foi influenciado pelo trabalho desenvolvido no projeto da rede local do IME [FER 84]. O sistema segue a filosofia de divisão em níveis, sendo composto por dois níveis apenas: Nível de Enlace (NE) e Nível de Transporte (NT).

O Nível de Enlace implementa as funções necessárias à transmissão/recepção das mensagens de enlace (quadros) e oferece estes serviços através de primitivas que são acionadas pelo Nível de Transporte.

O Nível de transporte tem como função oferecer ao Nível Superior (NS) os serviços necessários para o estabelecimento de circuitos virtuais entre os micros comunicantes. O Nível de Transporte assegura o transporte das mensagens do Nível Superior de forma ordenada e segura.

No restante deste trabalho o software de comunicação é apresentado de forma detalhada.

2. NÍVEL DE ENLACE

O nível de enlace (NE) tem como responsabilidade principal o controle da transferência das mensagens enviadas pelo nível imediatamente superior. Para que o NE possa enviar mensagens para a rede é necessário que exista uma política de acesso ao meio de comunicação. O método que foi usado é o CSMA/CD ("Carrier Sense Multiple Access/Colision Detect") [MET76].

Neste método a portadora inicialmente é sentida e, estando desocupada, o NE inicia a transmissão. A seguir é verificada a existência de colisão (há colisão se o que uma estação está enviando está sendo recebido por ela mesma de forma não consistente). No caso em que há colisão a transmissão é interrompida e uma nova tentativa de transmissão é realizada.

O NE transmite os dados byte a byte para uma interface serial de comunicação com padrão RS-232C. A interação do NE com o hardware é feita através da monitoração dos sinais que são fornecidos por essa interface. Estes sinais são registrados na posição de memória correspondente ao byte de estado da interface de comunicação.

Uma outra posição de memória armazena os dados que são enviados para a interface, ou retirados da mesma.

No byte de estado são mantidas as seguintes informações necessárias ao NE:

RDRF - "Receive Data Register Full"

Indica o estado da recepção:

- 0 - dado recebido já foi lido, ou seja, não chegou dado novo;
- 1 - dado recebido ainda não foi lido.

TDRE - "Transmit Data Register Empty"

Indica o estado de transmissão:

- 0 - dado ainda não foi transmitido, nova transmissão não habilitada;
- 1 - dado já foi transmitido, habilitada nova transmissão.

DCD - "Data Carrier Detect"

Indica o estado da linha de transmissão:

- 0 - linha de transmissão ocupada;
- 1 - linha de transmissão livre.

Sendo este nível o responsável pela interação com o hardware, o seu desenvolvimento realizou-se de forma simples, procurando facilitar, assim, a execução de alterações quando da modificação do hardware.

O Nível de Enlace oferece seus serviços para o nível imediatamente superior através de duas primitivas (TRANSMISSÃO e RECEPÇÃO) que serão descritos adiante.

2.1. Estrutura de dados

Todas as mensagens que fluem no sistema implementado tem passagem pelo NE, o qual realiza a operação desejada (Transmissão ou Recepção). Pa-

ra isto é utilizada uma estrutura de dados bem definida onde as mensagens são armazenadas.

A área de dados existente no NE serve de buffer tanto na recepção quanto na transmissão dos dados. Esta área do NE é composta por duas partes, a saber: o cabeçalho da mensagem e o texto (dados) da mensagem.

O cabeçalho da mensagem é composto por no máximo seis campos, dos quais quatro apenas são vistos pelo NE; o restante do cabeçalho é de interesse do nível de transporte.

Os campos do cabeçalho de interesse do NE são:

<u>Campo</u>	<u>Tamanho</u>	<u>Conteúdo</u>
Destino	1 byte	endereço destino da mensagem (0 - 255)
Origem	1 byte	endereço origem da mensagem (0 - 254)
Bytestx	2 bytes	número de bytes que compõe a mensagem (cabeçalho do nível superior mais dados)

O endereço (nº de estação) 255 é reservado para indicar mensagem de difusão (todas as estações podem receber a mensagem).

O endereço de origem é obtido na inicialização do sistema, quando informa-se o nº da estação de trabalho.

O campo Bytestx, como indicado, controla a transmissão e/ou recepção do NE a partir do quinto campo do cabeçalho, onde estão informações que dizem respeito ao nível de transporte.

Todas as mensagens do sistema, independentemente do tipo (dados, controle), possuem um cabeçalho fixo que sempre é transmitido. Desta forma o NE transmite/recebe 4 bytes mais os bytes indicados no campo Bytestx.

Os Dados da Mensagem compõe uma parte variável da mensagem, podendo ter no mínimo 0 e no máximo 255 bytes.

Este tamanho foi escolhido em função das características da rede implementada, e pode ser mudado se necessário.

O NE tem ainda a função de anexar a cada mensagem a ele enviada o caracter de início de transmissão "SOH" (start of header).

2.2. Tratamento de Colisões

Na ocorrência de colisão quando da tentativa de transmissão de men

sagens, as estações envolvidas entram em um período de espera randômico, implementado através de um busy wait. Este tempo randômico é conseguido através do registrador R ("REFRESH") da máquina Z-80.

O NE decrementa o valor do registrador R até chegar a zero e, então, volta à tentativa de iniciar uma transmissão. À medida que as colisões vão ocorrendo o tempo de espera aumenta, porque passa a ser o número de colisões ocorridas multiplicado pelo valor do registrador R.

O tratamento de colisão prevê um número máximo de colisões que podem ocorrer durante a transmissão de uma mensagem do NE.

O registrador R tem o seu valor modificado constantemente, sendo isto uma característica do microprocessador Z-80.

2.3. Primitivas

O NE fornece serviços ao seu nível imediatamente superior (NT) através de duas primitivas. Estas primitivas são controladas pelo próprio NE, o qual é chamado pelo NT, que fornece como parâmetro de entrada o código da primitiva que deve ser acionada. As primitivas do NE foram projetadas para prover os serviços básicos que são necessários para o nível de transporte poder executar a sua tarefa, sendo bastante simples.

2.3.1. Transmissão

A primitiva TRANSMISSÃO tem como função controlar o processo de transmissão de uma mensagem do nível de enlace e retornar o resultado de sua execução.

Inicialmente a primitiva faz a verificação do estado da linha de transmissão através da leitura do estado do interface serial de comunicação. Enquanto a linha estiver no estado ocupado e o tempo que a primitiva tiver para tentar a transmissão não se esgotar, irá permanecer verificando o estado da linha. Quando esgotar-se o tempo e a primitiva não conseguir a transmissão, voltará ao seu chamador indicando como código de retorno o valor 'FF'H (transmissão não realizada).

Quando a linha estiver no estado livre, a primitiva iniciará a transmissão do carácter de início de transmissão e dos três primeiros campos do cabeçalho da mensagem. Nesta etapa serão detectadas as colisões e tratadas da forma descrita anteriormente. Se a primitiva não conseguir executar de forma correta neste estágio, o código de retorno da primitiva será 'FF'H.

Após ter transmitido a parte referente ao cabeçalho da mensagem, a

primitiva TRANSMISSÃO entrará na sua última etapa que é a transferência do texto da mensagem. O número de bytes que devem ser transferidos é indicado no campo "Bytestx". Durante esta etapa qualquer impossibilidade de transmissão que ocorrer implicará no término da execução da primitiva e no retorno do código de erro 'FF'H.

A primitiva terá sucesso quando todos os bytes da mensagem forem transmitidos de forma correta. O código de retorno para a execução correta é o valor '06'H (transmissão realizada).

2.3.2. Recepção

O objetivo desta primitiva é o de controlar a recepção de uma mensagem. Ela recebe como parâmetro de entrada o endereço da estação origem da mensagem a ser recebida e rejeita mensagens originadas por outras estações. A chamada da primitiva com parâmetro de entrada igual a 'FF'H, indica que a mensagem a ser recebida pode ter origem em qualquer estação.

Em uma primeira etapa, a primitiva RECEPÇÃO verifica o estado da linha de transmissão. Se estiver livre, tenta a recepção do primeiro byte de uma mensagem. Esta tentativa se repetirá enquanto não receber nada e o tempo de recepção não se esgotar. O esgotamento do tempo sem que a primitiva tenha concretizado o recebimento ocasiona o término da execução e a devolução de um código de erro ('FF'H).

Na recepção do primeiro byte a primitiva verifica se o mesmo indica o início de uma mensagem ("SOH"). Não indicando, significa que o NE está interceptando uma mensagem pela metade. A primitiva então volta a tentar a recepção do indicador de início de uma mensagem.

Quando ocorre o recebimento de byte "SOH", a primitiva tenta a recepção do cabeçalho da mensagem, que é composto pelos próximos 4 bytes.

Do cabeçalho é extraída a informação referente ao tamanho da mensagem a receber (campo "Bytestx"), o qual controla, a partir de então, a execução da primitiva. Todos os bytes recebidos são colocados na área de dados do NE. No final da recepção é verificado se a mensagem recebida é endereçada à estação receptora, comparando o parâmetro da primitiva RECEPÇÃO e o campo "origem" da mensagem. A execução correta desta primitiva retorna ao chamador o código '06'H, significando sucesso na recepção.

3. NÍVEL DE TRANSPORTE

A interface entre o nível superior (aplicação) e o sistema de comunicação é responsabilidade das primitivas do NT. Estas primitivas imple-

mentam os serviços que são oferecidos para o desenvolvimento de aplicações na rede.

3.1. Estabelecimento da Conexão

Nesta fase da comunicação as entidades envolvidas criam um canal virtual entre si, e estabelecem parâmetros para as fases posteriores da comunicação. O estabelecimento de conexão se dá de duas maneiras: o nível superior requisita ao NT uma conexão, ou a estação escutando a linha recebe um pedido de conexão.

3.1.1. Primitiva CONECTA

Esta primitiva é requisitada pelo nível superior quando for necessário estabelecer um circuito virtual para troca de informações com outra estação remota. Para isto acontecer é necessário que a estação remota envolvida esteja em estado de ESCUTA na rede. Esta condição é conseguida quando o nível superior (na estação remota) ativa a primitiva PERGUNTA.

As conexões se realizam na relação uma para uma, ou seja, em um dado momento cada estação pode estar conectada com apenas uma outra estação remota. Desta forma para que o NT acione a primitiva CONECTA, é necessário que a estação envolvida esteja no estado LIVRE.

Como parâmetro de entrada a primitiva recebe o endereço da estação com a qual deve ser estabelecida a conexão. No retorno é desenvolvido ao nível superior o resultado da execução da primitiva CONECTA, o qual pode indicar:

- sucesso: conexão estabelecida;
- insucesso: conexão não estabelecida;
- estado inicial da estação não é LIVRE.

A ocorrência de um insucesso poderá se dar quando a estação remota não aceitar uma conexão, quando não estiver ativa ou quando a estação origem não conseguir enviar o pedido de conexão.

3.1.2. Primitiva PERGUNTA

Para que uma estação da rede local possa receber mensagens, é necessário (por falta de mecanismo de interrupção, no hardware) que a mesma esteja em estado de espera de mensagem. Isto corresponde a escutar a linha de comunicação a espera de pacotes. A primitiva PERGUNTA é ativada pelo nível superior para tentar a recepção de um pedido de conexão ou de uma mensagem de difusão (broadcast).

A execução da primitiva produz um resultado para o nível superior que pode ser:

- nenhuma mensagem foi recebida;
- mensagem de difusão recebida;
- mensagem pedido de conexão recebida e confirmada.

No segundo e no terceiro caso, a primitiva devolve como parâmetro de saída o endereço da estação que enviou a mensagem. No caso de mensagem de difusão são informados ao nível superior, a posição do buffer onde está a mensagem e o tamanho da mesma.

Na mensagem de pedido de conexão, a primitiva informa ao nível superior que o pedido foi recebido e confirmado, ou seja, a conexão foi estabelecida com a estação remota.

3.2. Transferência de Dados

Após o estabelecimento de conexão, as estações envolvidas estão aptas a trocar informações. Nesta fase de transferência de dados, as estações enviam e recebem mensagens.

3.2.1. Primitiva ENVIA

A primitiva ENVIA é ativada pelo NT quando o nível superior deseja a transmissão de informações para a estação remota com a qual se conectou. O endereço da estação que deve receber a mensagem já foi informado quando o nível superior realizou o pedido de conexão. A estação que requisitou a primitiva CONECTA, tem prioridade para enviar a mensagem inicial para a estação remota.

Toda mensagem que é enviada deve receber uma confirmação de recebimento (acknowledgment). Só após a confirmação do recebimento é que uma próxima mensagem pode ser enviada (ou seja, a janela de transmissão de mensagens é igual a 1).

Na chamada da primitiva o nível superior passa como parâmetros o endereço do buffer da mensagem e o tamanho da mensagem.

Recebe no retorno o resultado de execução da primitiva:

- Sucesso: mensagem transmitida e confirmada;
- insucesso: não recepção de confirmação ou problemas na transmissão;
- sem conexão: estação não está conectada.

3.2.2. Primitiva RECEBE

Para uma estação, que tenha estabelecido conexão com outra remota, poder receber dados enviados pela estação remota é necessário que o nível superior ative o NT, chamando a Primitiva RECEBE.

As mensagens de dados seguem uma sequência rígida, que é verificada na execução desta primitiva. Para cada mensagem recebida corretamente é enviada uma mensagem de confirmação de recebimento, liberando assim a estação remota da necessidade de repetir a mensagem.

Como parâmetro de entrada, o nível superior, na chamada da primitiva, informa o endereço do buffer onde a mensagem deve ser colocada. Como parâmetro de saída a primitiva devolve o código do resultado de execução, o qual pode ser:

- mensagem recebida;
- mensagem não recebida;
- desconexão pedida e realizada;
- estação não está conectada.

Quando a primitiva recebe alguma mensagem de dados, ela devolve para o nível superior o tamanho da mensagem que foi recebida.

3.3. Término da Conexão

Após a fase de transferência de dados entre as estações envolvidas em uma conexão, as mesmas devem se liberar para que novas conexões possam ser estabelecidas. Para isto o serviço de transporte oferece a primitiva DESCONNECTA.

3.3.1. Primitiva DESCONNECTA

O objetivo da primitiva DESCONNECTA é fazer com que as estações conectadas entre si sejam ordenadamente liberadas para novas conexões.

A estação que recebe o pedido de desconexão deve estar em um estado de recepção, para poder aceitar e confirmar a desconexão.

Como parâmetro de resposta a primitiva devolve para o nível superior o resultado da execução, o qual pode ser:

- sucesso: a conexão com a estação remota foi encerrada;
- insucesso: não foi possível encerrar a conexão (falta de recebimento de confirmação, ou impossibilidade de transmissão);
- a estação não está conectada.

3.4. Difusão de Mensagens

Para possibilitar o envio de mensagens para todas as estações que estejam ativas e em estado de escuta na rede, o NT oferece o serviço de difusão.

A primitiva DIFUNDE transmite uma mensagem para todos receptores da rede. Este tipo de mensagem não necessita da fase de estabelecimento de conexão e tampouco espera a recepção de mensagem de confirmação de recebimento.

Como parâmetros de entrada a primitiva recebe o endereço do buffer onde está depositada a mensagem a ser difundida e o tamanho da mesma. O parâmetro de saída informa ao nível superior o resultado da execução da primitiva DIFUNDE:

- sucesso: mensagem de difusão transmitida;
- insucesso: transmissão negativa.

4. CONCLUSÃO

O sistema aqui descrito está sendo testado e tem apresentado bons resultados. A inexistência de mecanismo de interrupção na interface de comunicação impede que se possa obter um melhor desempenho. Pretende-se, em uma próxima etapa, desenvolver um hardware mais adequado para a gerência da comunicação, no qual se tenha um microprocessador próprio na interface da rede, o que permitirá localizar o software aqui descrito nessa interface, liberando o micro principal de todo o gerenciamento do acesso à rede local.

BIBLIOGRAFIA

- /CLE 81/ CLEMENTS, Ken and DAUGHERTY, Dane. Ultra-low-cost network for personal computers. Byte, 6 (10), october 1981.
- /DAT 81/ DATA processing-open systems interconnection-basic reference model. Computer Networks, 5 (2): 81-118, April 1981.
- /FER 84/ FERNANDES, Lucio. Protocolo de transporte para redes locais. Rio de Janeiro, IME, Fevereiro 1984. (dissertação de mestrado).
- /MET 76/ METCALFE, R.M. and BOGGS, D.R. Ethernet: distributed packet switching for local computer networks. Communications of the ACM, 19 (7): 395-404, July 1976.
- /TOR 83/ TORNQUIST, M., TOSCANI, S.S. e FRIEDRICH, L.F. MUMPS/NET: uma rede local MUMPS para automação de escritórios. IN: CONGRES

SO NACIONAL DE INFORMATICA, 17., Rio de Janeiro, Outubro
1984. Anais. Rio de Janeiro, SUCESU, 1984.