Capítulo V

A Aplicação

Este capítulo descreve a aplicação que foi inicialmente idealizada para testar o Protocolo de Comunicações Multiponto-a-Multiponto desenvolvido neste trabalho. No entanto, apesar de relativamente simples, despertou grande interesse entre as pessoas que dela se utilizaram, principalmente entre os usuários acostumados a usar o *talk* do ambiente UNIX. Mostrou-se útil como ferramenta que permite rápidas reuniões via teclado, como se fosse um *talk* entre vários usuários. Na verdade, por ter sido desenvolvida inicialmente como uma aplicação de teste do protocolo, é permitido ao usuário a alteração de vários parâmetros importantes do protocolo, como valores de contadores de tempo, janela de emissão usada no controle de fluxo, probabilidade de perda de quadros (simulação de erros de transmissão), etc. Além disso, possui uma rotina que permite o envio de uma seqüência de N quadros de tamanho T, com finalidade de testes de vazão, onde tanto N como T podem ser escolhidos pelo usuário.

V.1. Para o usuário

Ao ser invocada na linha de comando DOS, a tela é divida em quatro janelas, cada uma destinada a uma estação. Uma quinta janela é aberta por cima destas, mostrando os valores atuais dos parâmetros do protocolo, com possibilidade de edição destes. As teclas chaves desta aplicação são algumas teclas de função. A única necessária de ser memorizada é a tecla F1, que abre uma janela com um *Help* sensitivo ao contexto: nesta tela inicial, a tecla F1 abre uma nova janela com explicações sobre cada um dos parâmetros, sobre os quais o usuário pode interferir. Esta tela de edição de parâmetros pode ser aberta sempre que não houver conferência em andamento (conexão fechada) e, enquanto esta janela estiver aberta a estação não responderá a pedidos de conexão, ou seja, não participará de conferências que estejam iniciando, podendo porém, entrar posteriormente numa conferência já em andamento.

Conforme dito no Capítulo II, o protocolo utiliza um único canal para transmissão dos dados do usuário, o qual é compartilhado no tempo entre todas as estações. Então, num

determinado instante, apenas uma estação está transmitindo e as demais estão atuando somente como receptoras de dados. A aplicação é avisada pelo serviço sobre quais estações estão na conferência e quando pode começar a enviar primitivas de dados ou parar. Uma aplicação pode estar em três estados básicos: conexão fechada, em transmissão ou, em recepção. Mas, mesmo durante a fase de recepção é permitido ao usuário continuar a digitar normalmente, sendo os dados guardados em um *buffer* para posterior transmissão.

Após a etapa inicial de ajustes de parâmetros, o usuário pode começar a digitar o seu texto. Como o protocolo é voltado para conexão, a aplicação se encarrega de pedir a conexão ao serviço da camada provedora através de uma primitiva, enquanto armazena em *buffer* próprio os caracteres que forem sendo digitados. Ao chegar a primitiva de confirmação de conexão que indique que a aplicação pode transmitir, ela envia uma primitiva de dados com os caracteres até então digitados. Nas demais estações, a chegada de uma primitiva de dados faz com que os caracteres sejam enviados para a janela correspondente à estação origem, cuja identidade é um parâmetro desta mesma primitiva.

As teclas de comando utilizadas e suas funções são:

F1 - *help*: se acionada na tela de edição de parâmetros, abre uma nova janela com explicação sobre cada parâmetro. Caso acionada na tela principal, abre uma janela com um resumo de cada tecla de função;

F2 - se a conexão estiver fechada, enviar um pedido de conexão;

F3 - enviar um pedido de desconexão, se houver conexão aberta;

F4 - enviar um pedido para transmitir. Este pedido também é enviado automaticamente quando a aplicação não está com permissão para transmitir e algo for digitado;

F5 - ceder o direito de transmissão antes que termine o tempo da estação. Isto causa o envio da ficha; Sua utilização é importante no caso do tempo máximo de transmissão ter valor elevado.

F6 - iniciar com zero o contador de "número de medidas". Este contador é utilizado somente nas medidas de vazão;

F7 - iniciar a transmissão de uma seqüência de N quadros de tamanho T para medida de vazão. N e T são definidos pelo usuário na tela de edição de parâmetros (tecla F9);

F9 - entrar no menu de modificações de parâmetros do protocolo. Esta tecla só funciona se a conexão multiponto a multiponto estiver fechada. É importante frisar que enquanto a janela de edição de parâmetros estiver aberta, a estação não responderá a

nenhum eventual pedido de conexão. Esta característica pode ser aproveitada quando o usuário estiver ocupado e não quiser participar de novas conferências: basta acionar a tela de edição;

TAB - apagar a janela de transmissão da estação e a correspondente em cada receptor.

As condições da conferência são sinalizadas pela cor da linha título de cada janela: quando não há conferência em andamento, todas as janelas tem título em fundo preto, exceto a janela pertencente à estação local, que possui fundo azul. Ao ser iniciada uma conferência as estações que responderam ao pedido de conexão aparecem com o título em fundo vermelho. O usuário também fica sabendo quando seus dados estão sendo imediatamente transmitidos através da cor do título da janela de sua estação: na fase de transmissão o fundo é verde.

V.2. As rotinas

A implementação da aplicação teve sua versão básica desenvolvida por membros do GTA. A versão final pode ser dividida em seis grupos: as rotinas que iniciam e finalizam; a rotina de alteração de parâmetros; as rotinas de montagem e envio de primitivas para o provedor do serviço; a rotina de tratamento de primitivas que chegam do provedor e executa a máquina de estados de recepção; a rotina que faz a interface com o usuário e executa a máquina de estados de transmissão da aplicação e; demais rotinas auxiliares (montagem de telas, manipulação das janelas, *help*). A seguir estas rotinas pertencentes a aplicação são descritas.

- **Iniciar_Aplicacao**(-) - esta rotina realiza as seguintes operações: cria um *buffer* para armazenar os caracteres digitados durante a fase em que a aplicação não pode transmitir (não está de posse do *token*), chama a rotina que monta a tela do usuário e coloca as cores de fundo dos títulos das janelas em cinza escuro, com exceção da janela pertencente a estação local, conforme visto anteriormente. Para terminar a etapa de iniciação da aplicação, é feita uma chamada à rotina que abre a janela de edição dos parâmetros do protocolo (Menu_Var);

- **Finalizar_Aplicacao**(-) - esta rotina apenas libera a memória ocupada pelo *buffer* alocado em Iniciar_Aplicacao;

- **Menu_Var(-)** - esta rotina é chamada durante a fase de iniciação da aplicação ou, se conexão estiver fechada, através do acionamento da tecla F9. Possibilita a edição dos parâmetros (variáveis) do protocolo e do sistema. Os parâmetros do protocolo alteráveis são: o valor inicial de cada contador de tempo, valor este que é usado a cada vez que um deles é criado ou reiniciado; o número máximo de tentativas (de envio de pedido de conexão, passagem da ficha, envio de pedidos de desconexão) e; o número

máximo de quadros transmitidos em avanço (janela de emissão). Os valores iniciais dos contadores de tempo devem ser expressos em quantidade de *timer ticks*, conforme visto na seção referente ao gerenciador de tempo. Para facilitar a interface com o usuário, é calculado o valor em segundos e mostrado ao lado do valor em *timer ticks*. Além dos parâmetros específicos do protocolo, algumas variáveis do sistema podem ser modificadas nesta tela de edição. São elas: o tamanho e a quantidade de quadros utilizados nas medições de vazão (utilizados na aplicação) e; a probabilidade de erros nos canais de transmissão (visto adiante). Vale mencionar o fato de que esta rotina toma como primeira providência a inibição da camada MAC, desabilitando o recebimento de quaisquer quadros. Através deste procedimento, garante-se que não haverá estabelecimento de conexão com esta estação enquanto as variáveis estiverem sendo alteradas. Ao final da edição (ou quando o usuário quiser habilitar novamente a estação a participar de conexões) o usuário pode fechar a janela pressionando a tecla Escape (Esc). Ao fechar a janela, a rotina habilita novamente a camada MAC, possibilitando o recebimento de quadros;

- **Enviar_MPDL_XXX(-)** - Estas rotinas permitem a montagem e envio das seis possíveis primitivas para o provedor de serviço. As letras XXX podem ser substituídas por: ESTABLISH_Req, ABORT_Req, TOKEN_PLEASE_Req, TOKEN_GIVE_Req, DATA_Req e, STOP_TRANSMISSION_Resp. A única rotina que possui parâmetros é a Enviar_MPDL_DATA_Req que monta e envia a primitiva de dados. Seus parâmetros são: um ponteiro para a região de memória onde estão os dados e, um inteiro que indica o número de octetos (tamanho) dos dados;

- **Tratar_Pedido_Servico_MPDLN(-)** - esta rotina é o ponto de entrada do escalonador e é colocada na Fila do Escalonador de tarefas pelo provedor do serviço quando este envia uma primitiva. A primeira tarefa executada por esta rotina é verificar se a janela de *help* está ativa, caso em que ela é fechada. Em seguida a primitiva é recebida e uma declaração *switch ... case* identifica a primitiva pelo seu nome e executa operações que dependem da primitiva e do estado atual da máquina de estados da aplicação. Por sua relativa simplicidade, a máquina de estados é implementada por declarações *if ... then ... else*, colocadas dentro de cada *case*;

- **Executar_Aplicacao(-)** - esta rotina é chamada pelo *loop* do programa principal quando a Fila do Escalonador está vazia. Na verdade, esta rotina é uma função, e retorna *true* se o usuário quiser terminar a execução do programa e *false* caso contrário. A primeira tarefa desta rotina é verificar se existe alguma tecla que foi digitada e está pendente de tratamento. Em caso positivo, a tecla é lida. Se a janela de *help* estiver aberta, ela é fechada neste momento. Em seguida a tecla é tratada em uma

declaração *switch ... case* que, a exemplo da rotina anterior, implementa a máquina de estados referente à transmissão através de declarações *if ... then ... else.* Se não houver tecla a ser tratada, é então verificado se existe uma medida de vazão em andamento em que a estação local esteja ativa como transmissora. Neste caso, se o *token* estiver de posse da estação, um novo quadro será criado e transmitido. Nesta etapa também é verificado se o número de quadros estipulado já foi transmitido e/ou recebido, caracterizando o fim de uma etapa de transmissão. Se a variável de estado da medição indicar um fim de etapa de medição, então são calculados o tempo de duração (em segundos) e a vazão (em bits por segundo), ambos colocados na tela. Informações adicionais como o número de vezes em que não houve transmissão devido a um bloqueio por janela de emissão, o número de SREJ recebidos e, o número de erros de transmissão gerados (perda propositada do quadro para simular erros) são também colocados na tela.

V.3. A Máquina de estados da Aplicação

V.3.1. Estados

A aplicação inicialmente desenvolvida só permitia a execução do *talk* entre as várias estações conectadas. Esta aplicação foi dividida em três estados: conexão multiponto a multiponto fechada, aplicação em transmissão e, aplicação em recepção. Posteriormente foram desenvolvidas as rotinas que permitiram as medições de vazão. Os sub-estados que uma aplicação pode se encontrar com relação a uma medida de vazão são:

- nenhuma medida em andamento;
- aplicação participando de modo ativo em uma medida. Existe uma medida de vazão em andamento e a aplicação da estação local participa recebendo, contabilizando e também transmitindo mensagens;
- estação participando de modo passivo em uma medida. Existe uma medida de vazão em andamento mas, a aplicação da estação local participa apenas recebendo e contabilizando mensagens;
- estação aguardando para transmitir a primeira mensagem de uma medida. Ocorre se uma medição for iniciada pelo usuário (tecla F7) quando a aplicação estiver em estado de receptora. Um pedido de transmissão é enviado e a aplicação fica aguardando o recebimento do *token*, para poder transmitir a primeira mensagem da medida. Esta primeira mensagem é importante pois, seu conteúdo indica para as demais estações que

a aplicação da estação originária da mensagem está passando a ser uma estação ativa no processo de medição. Por isso, esta estação deverá ter suas mensagens contabilizadas pelas demais estações. Após a transmissão deste primeiro quadro, a aplicação está ativa na medição;

- medida em estado de finalização. Os valores de tempo de duração da medida e da vazão são calculados e mostrados na tela, assim como outras informações, conforme descrito anteriormente. A seguir assume-se que não há nenhuma medida em andamento. Uma medição só pode entrar em estado de finalização depois que receber, de todas as estações ativas, o número de mensagens especificadas para a medida. No caso de ser uma estação ativa, além de receber das demais, é necessário que também tenha terminado a sua transmissão.

Mesmo assim, os três estados relacionados apenas com o *talk* foram mantidos. O controle das medidas foi feito criando-se uma outra variável para, em paralelo, controlar o estado atual de uma medida de vazão. Desta maneira a máquina de estados da aplicação continuou simples e, o controle das medidas de vazão foi todo explicitado através de predicados e ações. A seguir são descritos as principais características de cada estado.

V.3.1.1. APL_CLOSED

A aplicação não se encontra conectada. Este é o único estado em que é permitido ao usuário a abertura da janela de edição de parâmetros do protocolo e do sistema. Durante o tempo em que esta janela estiver aberta todo o sistema estará desativado através da inibição da recepção de quadros (camada MAC). Neste estado da aplicação é permitido ao usuário iniciar um pedido de conexão acionando a tecla F2. Uma facilidade adicional é oferecida ao usuário: se este iniciar a digitação de uma mensagem qualquer, sem antes ter acionado a tecla F2 (pedido de conexão), a aplicação interpretará esta ação como um desejo de se conectar. Assim, a aplicação inicia automaticamente um estabelecimento de conexão e armazena os dados digitados em um buffer para posterior transmisssão.

V.3.1.2. APL_TX

A aplicação está conectada e possui a ficha, o que lhe permite realizar transmissões (a estação local é a Mestra da conferência).

V.3.1.3. APL_RX

A aplicação está conectada mas não possui a ficha. Neste estado a estação pode receber mensagens mas não pode transmistir. Assim, todos os caracteres digitados pelo usuário são guardados em um *buffer* próprio da aplicação para posterior transmissão.

V.3.2.Predicados

Os predicados da aplicação (PA), são relacionados a seguir:

PA1: existe medida de vazão em andamento;

PA2: todas as estações participantes da medida de vazão já enviaram o número de mensagens previsto. A aplicação local também já transmitiu o número de mensagens previsto.

PA3: a mensagem recebida indica o início de uma medição de vazão por parte da estação originária da mensagem.

PA4: não existe medida em andamento em estado de finalização;

PA5: o parâmetro da primitiva recebida indica que a conexão foi iniciada pela estação local;

PA6: a identificação da estação local é o parâmetro da primitiva recebida;

PA7: a janela de *help* está aberta;

PA8: a aplicação local está ativa em uma medida de vazão em andamento;

PA9: a aplicação local fez um pedido de transmissão (pediu o *token*);

PA10: a aplicação já transmitiu o número de quadros estipulado para a medida de vazão;

PA11: a medida de vazão está em fase de finalização;

PA12: a aplicação está pronta e aguarda para enviar a sua primeira mensagem para efetuar uma medida de vazão.

PA13: parâmetro da primitiva MPDL_STOP_TRANSMISSION.Indication é "TMT".

V.3.3. Ações

As ações da aplicação (AA), são relacionadas a seguir:

AA1: incrementar o contador de "mensagens recebidas" correspondente à estação originária;

AA2: guardar informação - uma medida está em fase de finalização;

AA3: guardar informação - a aplicação está participando de uma medida como elemento passivo (receptora apenas);

AA4: iniciar temporizador com a finalidade de cálculo de vazão;

AA5: tratar mensagem recebida como dado do usuário, isto é, enviar a mensagem para a janela de vídeo correspondente à estação origem;

AA6: liberar memória ocupada pelo parâmetro da primitiva;

AA7: liberar estrutura de parâmetros;

AA8: retirar primitiva da fila de primitivas;

AA9: indicar na tela de vídeo que a estação local é detentora do *token* (Mestra da conexão);

AA10: indicar na tela de vídeo que a estação local está conectada e está em estado de recepção;

AA11: enviar MPDL_STOP_TRANSMISSION.Response;

AA12: indicar na tela de vídeo as estações que estão presentes na lista recebida no parâmetro da primitiva;

AA13: transmitir o *buffer* da aplicação, se houver mensagens nele;

AA14: guardar informação - não existe pedido de ficha da aplicação pendente;

AA15: guardar informação - existe pedido de ficha da aplicação pendente;

AA16: guardar informação - não há medida em andamento;

AA17: atualizar a tela de vídeo, indicando que não há conexão estabelecida e identificando a estação local;

AA18: atualizar a tela de vídeo, indicando que a estação cuja identificação está no parâmetro da primitiva recebida não está conectada;

AA19: atualizar a tela de vídeo, indicando que a estação cuja identificação está no parâmetro da primitiva recebida está conectada;

AA20: emitir sinal sonoro (*beep*), indicando ao usuário que a estação recebeu um pedido para transmitir;

AA21: fechar a janela de *help*;

AA22: abrir a janela de *help*;

AA23: enviar MPDL_ESTABLISH.Request;

AA24: enviar MPDL_ABORT.Request;

AA25: mostrar aviso na tela de vídeo - "Pedido de desconexão somente quando em transmissão";

AA26: enviar MPDL_TOKEN_PLEASE.Request;

AA27: enviar MPDL_TOKEN_GIVE.Request;

AA28: atribuir o valor zero ao contador do número de medidas realizadas;

AA29: atribuir o valor zero aos contadores de mensagens de medidas de vazão de todas as estações;

AA30: mostrar aviso na tela - "Abra a conexão antes de iniciar uma medida de vazão";

AA31: iniciar contador de tempo para posterior cálculo de vazão;

AA32: enviar mensagem com indicação de que esta estação está iniciando uma medida de vazão;

AA33: guardar informação de que a aplicação está aguardando para enviar o primeiro quadro de uma medida de vazão;

AA34: guardar informação de que a estação local é participante ativa em uma medida de vazão;

AA35: abrir janela e entrar em modo de edição dos parâmetros do protocolo;

AA36: mostrar aviso na tela - "Edição dos parâmetros do protocolo só é possível com a conexão fechada";

AA37: retornar *true* - fim do programa;

AA38: enviar o caractere correspondente à tecla pressionada para a janela da estação local;

AA39: guardar o caractere correspondente à tecla pressionada no *buffer* para posterior transmissão;

AA40: enviar MPDL_DATA.Request com o caractere correspondente à tecla pressionada como dado do usuário;

AA41: criar e enviar mensagem (MPDL_DATA.Request) com o tamanho estipulado para as mensagens de medida;

AA42: limpar janela (vídeo) da estação;

AA43: calcular tempo, vazão e mostrar na tela de vídeo os valores calculados, assim como as demais informações sobre as medidas realizadas;

V.3.4. Máquina de estados

A Tabela V.1 mostra a máquina de estados relativa à recepção da aplicação. Esta máquina de estados é implementada na rotina Tratar_Pedido_Servico_MPDLN. Seus eventos resumem-se a primitivas enviadas pelo provedor de serviço.

A Tabela V.2 mostra a máquina de estados referente à transmissão. Esta máquina de estados é implementada pela rotina Executar_Aplicacao, descrita anteriormente. Seus eventos são, em sua maioria, teclas pressionadas pelo usuário. Pelo fato desta rotina ser chamada em um *loop* do programa principal, um dos eventos possíveis é uma chamada sem nenhuma tecla pressionada a ser tratada. Este caso, que é muito freqüente, é usado para a implementação do controle e de transmissões com finalidade de medidas de vazão.

Tabela V.1 Máquina de estados de recepção da aplicação (recebimento de primitivas do provedor).

ESTADO ⇒ EVENTO ⇓	APL_CLOSED	APL_TX	APL_RX
MPDL_DATA.Indication			PA1: AA1 PA2: AA2 ÑPA1: PA3: AA3 AA4 ÑPA1: ÑPA3: PA4: AA5 AA6 AA7 AA8
MPDL_STOP_TRANSMISSION. Indication		AA10 PA13: AA11 ⇒APL_RX	
MPDL_ESTABLISH.Confirm	AA12 PA5: AA9 AA13 AA6 AA7 AA8 \⇒APL_TX ÑPA5: AA6 AA7 AA8 \⇒APL_•RX		
MPDL_TOKEN_GIVE.Indication			AA14 AA9 A13 ⇒APL_TX
MPDL_INDIVIDUAL_ABORT. Indication		PA6: AA6 AA7 AA8 AA17 \⇒APL_CLOSED ÑPA6: AA18 AA6 AA7 AA8	PA6: AA6 AA7 AA8 AA17 \⇒APL_CLOSED ÑPA6: AA18 AA6 AA7 AA8
MPDL_INDIVIDUAL_ABORT. Confirm		PA6: AA6 AA7 AA8 AA17 \⇒APL_CLOSED ÑPA6: AA18 AA6 AA7 AA8	PA6: AA6 AA7 AA8 AA17 \⇒APL_CLOSED ÑPA6: AA18 AA6 AA7 AA8
MPDL_ABORT.Indication		AA6 AA7 AA8 AA17 ⇒APL_CLOSED	AA6 AA7 AA8 AA17 ⇒APL_CLOSED
MPDL_ABORT.Confirm		AA6 AA7 AA8 AA17 ⇒APL_CLOSED	
MPDL_ESTABLISH.Indication	AA19 AA10 AA6 AA7 AA8 ⇒APL_RX	AA19 AA6 AA7 AA8	AA19 AA6 AA7 AA8
MPDL_TOKEN_PLEASE. Indication		P4: AA20 AA6 AA7 AA8	

V.4. Aplicação x parâmetros do protocolo

Esta aplicação, do ponto de vista do conforto do usuário, tem suas características bastante sensíveis em relação aos parâmetros do protocolo (editados através da janela ativada pela tecla F9), tais como: número de tentativas; tempo entre repetições (T_R); tempo entre envio de reconhecimentos positivos (T_{ACK}); tempo máximo de inatividade dos receptores (T_I); tempo de inatividade da Mestra (T_{IM}); tempo de inatividade no canal principal (T_{IP}). Algumas influências práticas da atribuição de valores a estes parâmetros serão brevemente discutidas nesta seção.

Durante a fase inicial de estabelecimento da conexão, dois parâmetros podem influir no tempo de espera (do usuário) para o estabelecimento da conexão. Se uma ou mais estações não estiverem habilitadas, por qualquer razão, a responder a um pedido de conexão, então o protocolo irá esperar o tempo determinado pela temporização T_R. Depois de findo este tempo,

irá tentar novamente o estabelecimento da conexão. O número de tentativas também pode ser escolhido pelo usuário. Somente após esgotadas todas as N tentativas previstas, a conexão se dará com as estações que responderam. Deste modo, valores elevados para T_R e N, traduzem-se em um longo tempo de espera para estabelecimento de conexão, no caso de uma estação (ou mais) não responder. No caso de todas as estações responderem, o tempo para conexão depende apenas do retardo de propagação do meio físico utilizado e do tempo de tratamento da conexão. Para o ambiente de redes locais, são preferidos valores pequenos, devido às baixas taxas de erros e ao baixo retardo de propagação do meio físico. Já no ambiente satélite, o retardo de propagação deve ser considerado, elevando-se o valor inicial de T_R para um (1) segundo, ou mais.

Tabela V.2 Máquina de estados de transmissão da aplicação (tratamento de teclas do usuário e tratamento de medidas de vazão).

ESTADO ⇒ EVENTO ⇓	APL_CLOSED	APL_TX	APL_RX
tecla F1	PA7: AA21 ÑPA7: AA22	PA7: AA21 ÑPA7: AA22	PA7: AA21 ÑPA7: AA22
tecla F2	AA23	AA23	AA23
tecla F3	AA25	AA24	AA25
tecla F4			AA26
tecla F5		AA27 ⇒APL_RX	
tecla F6	AA28	AA28	AA28
tecla F7	P4: AA29 AA30	ÑP8: AA32 AA34 P4: AA31	ÑP8: AA26 AA15 AA33 P4: AA31
tecla F9	AA35	AA36	AA36
tecla Esc	AA37	AA37	AA37
tecla Tab.	AA38 AA39 AA23 AA42	AA38 AA40 AA42	AA38 AA39 AA42 ÑPA9: AA26
outra tecla qualquer	AA38 AA39 AA23	AA38 AA40	AA38 AA39 ÑPA9: AA26
nenhuma tecla	PA11: AA43	PA11: AA43 PA8: AA41 PA10: PA2: AA2 ÑPA2: AA3 AA27 \\\⇒APL_RX PA12: AA32 AA34	PA11: AA43 PA8: ÑPA9: AA26

Após o estabelecimento da conexão, esta pode ser instável, inclusive com aborto pelo serviço, devido a valores inadequados a quatro parâmetros: T_{ACK}; T_I; T_{IM} e T_{IP}. Estes contadores de tempo estão relacionados dois a dois: T_I com T_{ACK} e T_{IP} com T_{IM}. Se o valor inicial de T_I for menor ou igual ao de T_{ACK}, ou o de T_{IP} for menor ou igual ao de T_{IM}, certamente haverá problemas pois, T_I e T_{IP} "monitoram" o tempo máximo de silêncio permitido e, T_{ACK} e T_{IM} evitam a ocorrência de silêncio com tempo maior do que os seus valores iniciais. Na implementação realizada no ambiente de rede local, utilizou-se um valor

inicial para T_{ACK} de 2 *timer ticks* e, para T_{IP} um valor de 100 *timer ticks*. A discrepância entre os dois valores utilizados (2 e 100) é devida ao fato de que T_{ACK} está relacionado com envio da PDU RR, que faz parte do mecanismo de controle de erros, enquanto T_{IP} não.

Agora, com a conexão estabelecida e estável (do ponto de vista dos parâmetros), resta cuidar para que a conferência seja agradável. O tempo de resposta a cada frase não deve ser muito longo, principalmente se os usuários ainda não estiverem acostumados ao uso da tecla F5 (envio do *token*). Então, inicialmente pode-se usar valores pequenos para o tempo máximo de transmissão (T_{MT}), entre 5 e 10 *timer ticks* (0,3 a 0,6 segundos). Estes tempos darão a impressão ao usuário de que todos podem digitar "ao mesmo tempo". Entretanto, tempos maiores dão uma melhor organização à conferência (do ponto de vista do usuário), pois cada um será o Mestre da conferência durante um intervalo mensurável de tempo (do ponto de vista prático, 0,6 segundos não é nada). Então, valores maiores do que 100 *timer ticks* (5,6 segundos) são utilizáveis, se os usuários se acostumarem a utilizar o serviço prestado pela tecla F5.

Os demais contadores de tempo devem ser iniciados com valores baseados naqueles escolhidos para os contadores de tempo discutidos até aqui.

O contador de tempo T_{RQ} deve ter um valor inicial um pouco menor do que T_{IM}.

O contador de tempo T_{KPR} deve ter seu valor inicial um pouco maior do que o tempo máximo de transmissão já que, a cada passagem de ficha, uma estação que a pediu (e não recebeu) verifica se sua identificação está na Fila de Espera. Isto é possível pois a ficha é um quadro que é difundido entre todas as estações participantes.

O Apêndice D apresenta um manual para a utilização do sistema de texto-conferência.

V.5. Comentários

A aplicação descrita neste capítulo foi desenvolvida com a intenção inicial de somente servir de teste para o protocolo apresentado neste trabalho. No entanto, mostrou-se útil para realizar pequenas reuniões entre usuários da rede, através de sessões de texto-conferência.

Através de textos de ajuda, é possível a um usuário utilizá-la, sem necessidade de recorrer a manuais.

Devido a sua característica de ser uma ferramenta de teste, esta aplicação permite ao usuário alterar parâmetros importantes do protocolo, como tamanho da janela de emissão (utilizada no controle de fluxo) e valores iniciais das diversas temporizações utilizadas. Assim

sendo, é uma aplicação que, se mal usada, irá trazer instabilidade ao sistema, inclusive impossibilitando a conexão. Além disto, a aplicação permite gerar uma seqüência de quadros e transmití-los, com a finalidade de permitir a realização de medidas de vazão.

As conseqüências de atribuições e as relações entre valores iniciais das temporizações são discutidas no final do capítulo, possibilitando um maior controle do sistema.