

PROJETO DE UM PROTOCOLO LLC PARA REDE LOCAL

MINI-MAP USANDO A LINGUAGEM ESTELLE

Marcelo Moraes de Azevedo (1)

Maurício Assad Neder (2)

Aloysio de Castro Pinto Pedroza (2,3)

Hélio Schor (3,4)

Elton Lemos Santos (1,3)

RESUMO: Este trabalho apresenta o projeto de um protocolo LLC para rede local Mini-MAP, compatível com o padrão IEEE 802.2 Tipo 3. O protocolo foi especificado formalmente na linguagem Estelle e submetido a um processo de validação que incluiu a modelagem com redes de Petri e testes por simulação. O uso de uma especificação em Estelle permitiu uma abordagem formal para o desenvolvimento do protocolo IEEE 802.2 Tipo 3.

ABSTRACT: This paper presents the design of a LLC protocol for a Mini-MAP LAN which is compatible with the IEEE 802.2 Type 3 standard. The protocol has been formally specified in the Estelle language and has been submitted to a process of validation which included Petri nets modelling and simulation tests. The use of an Estelle specification provided a formal approach to the development of the IEEE 802.2 Type 3 protocol.

- (1) Amplus Informática S/A - Rua da Estrela, 79 - Rio de Janeiro - RJ
CEP 20251 - Tel: (021) 293-9737 - FAX: (021) 293-9698
- (2) Escola de Engenharia da UFRJ - Departamento de Eletrônica
- (3) COPPE/UFRJ - Programa de Engenharia Elétrica - Centro de Tecnologia
Bloco H - Sala H-321 - Cidade Universitária - Ilha do Fundão, S/N
Rio de Janeiro - RJ - CEP 21945 - Caixa Postal 68504
Tel: (021) 280-8832 R. 413 - FAX: (021) 290-6626
- (4) Elebra Computadores S/A - Praia de Botafogo, 228 - 3o. Andar
Rio de Janeiro - RJ - CEP 22250
Tel: (021) 551-0021 - FAX: (021) 552-1546

1 - INTRODUÇÃO

A arquitetura Mini-MAP é uma das implementações de redes locais previstas na especificação MAP 3.0 (Manufacturing Automation Protocol) [1] [2] [3] [4]. Trata-se de uma rede local para aplicações industriais, capaz de oferecer baixos tempos de resposta, acesso determinístico, alta imunidade a ruídos e baixo custo [5] [6].

O protocolo usado na subcamada controle de enlace lógico (LLC) de uma rede local Mini-MAP segue o padrão IEEE 802.2 Tipo 3 [7]. Este padrão define um protocolo orientado a não conexão, com reconhecimento de erros.

O projeto do protocolo IEEE 802.2 Tipo 3 faz parte de um conjunto de trabalhos que visa desenvolver uma rede local Mini-MAP com o uso de técnicas formais, o que permite a modelagem e verificação de propriedades do sistema em desenvolvimento. Entre os resultados já obtidos, destacam-se as implementações do hardware da rede local [8] [9] e do protocolo de aplicação MMS (Manufacturing Message Specification) [10], além da avaliação do uso de fibras ópticas em redes locais Mini-MAP [11].

O projeto do protocolo LLC foi iniciado com a construção de modelos em redes de Petri, extraídos do padrão IEEE 802.2 Tipo 3. A partir desses modelos, construiu-se uma especificação formal na linguagem Estelle [12] [13] para o protocolo. Tal especificação foi em seguida submetida a um processo de validação através de simulação. Após a conclusão dos procedimentos de modelagem e validação, utilizou-se a especificação em Estelle como base para implementação do protocolo IEEE 802.2 Tipo 3.

A seção 2, a seguir, descreve a especificação em Estelle construída para o protocolo IEEE 802.2 Tipo 3. A seção 3 descreve os procedimentos de modelagem e simulação usados para validação da especificação. A seção 4 apresenta a abordagem adotada para implementação do protocolo IEEE 802.2 Tipo 3. O Anexo apresenta um esboço da especificação em Estelle.

2 - ESPECIFICAÇÃO FORMAL DO PROTOCOLO LLC TIPO 3 EM ESTELLE

2.1 - ARQUITETURA DA ESPECIFICAÇÃO

A arquitetura adotada para a especificação é mostrada na Fig. 2.1.

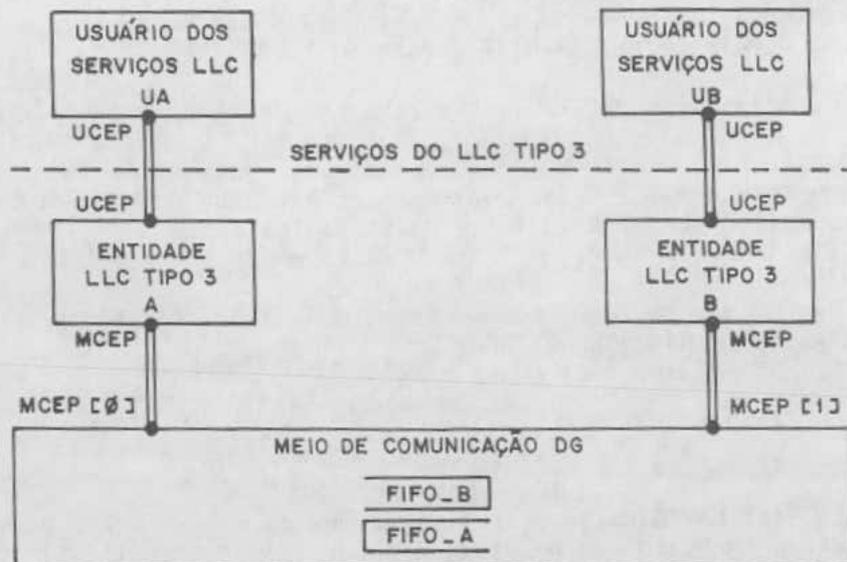


FIG. 2.1 - Arquitetura da especificação do protocolo LLC

A arquitetura da Fig. 2.1 modela a troca de dados entre dois usuários da camada de enlace (UA e UB). A troca de dados é controlada por duas entidades LLC Tipo 3 (A e B), conectadas através do meio de comunicação DG.

2.2 - DESCRIÇÃO DA ESPECIFICAÇÃO

A especificação em Estelle do LLC Tipo 3 inicia-se com um conjunto de declarações que definem constantes e tipos de estruturas de dados, funções e "procedures". Entre as constantes declaradas neste trecho da especificação, encontram-se o número máximo de retransmissões, período máximo de espera de reconhecimento e códigos para os diversos status de transmissão e recepção de PDU's. São também atribuídos tipos a funções e "procedures" externas usadas na especificação.

A especificação utiliza três tipos de canais para conexão de instâncias de módulos:

- Canal LLC_interface

Este canal é usado para comunicação entre instâncias de módulos do tipo USER_TYPE (UA e UB) e instâncias de módulos do tipo LLC_ENTITY_TYPE (A e B). O canal LLC_interface contém as primitivas de serviço do LLC Tipo 3.

- Canal MAC_interface

Este canal é usado para comunicação entre instâncias de módulos do tipo LLC_ENTITY_TYPE (A e B) e instâncias de módulos do tipo MEDIUM_TYPE (DG). O canal MAC_interface contém as primitivas de serviço MAC.

- Canal SYS_interface

Este canal é usado para entrada de interações com instâncias de módulos do tipo USER_TYPE (UA e UB). O canal SYS_interface não é especificado pelo padrão IEEE 802.2 Tipo 3 e foi utilizado para permitir a introdução de seqüências de testes de simulação pelo usuário.

A especificação contém três tipos de módulos: USER_TYPE, LLC_ENTITY_TYPE e MEDIUM_TYPE. A relação entre os diferentes tipos de módulos e canais com a arquitetura da especificação é mostrada na Fig. 2.2.

O módulo USER_TYPE representa o comportamento do usuário da camada de enlace. Na especificação, são criadas duas instâncias desse módulo (UA e UB), através da primitiva de Estelle init.

O corpo do módulo USER_TYPE é do tipo USER_BODY, possuindo dois pontos de interação: UCEP e USER. Através do ponto de interação UCEP, são passadas e recebidas primitivas de serviço para a subcamada LLC. Através do ponto de interação USER, são recebidos pedidos de transmissão (ENVIAR_MSG), requisição (POLLING), troca (TROCA) ou preparação (UPDATE) de dados, em tempo de simulação.

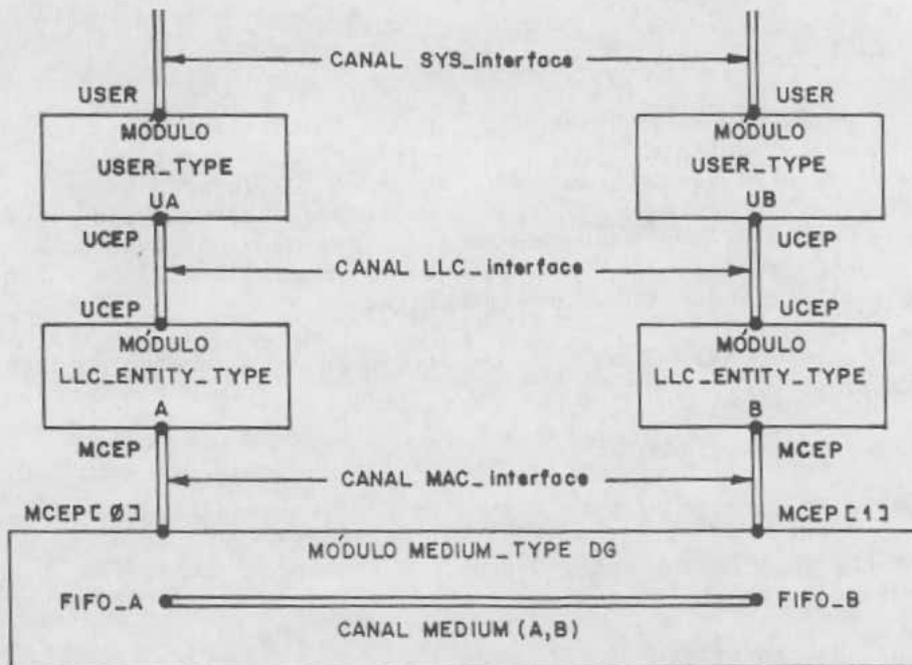


FIG. 2.2 - Relação entre a arquitetura da especificação e os tipos de módulos e canais utilizados

Foram declarados no corpo `USER_BODY` 5 estados distintos para representar o comportamento do usuário da camada de enlace. Estes estados são:

- . S0 - estado de repouso do usuário da camada de enlace;
- . S1 - estado atingido após uma transmissão de dados (`ENVIAR_MSG`), representando a espera da primitiva `DL_DATA_ACK_STATUS` indication;
- . S2 - estado atingido após uma troca de dados (`TROCA`), representando a espera da primitiva `DL_REPLY_STATUS` indication;
- . S3 - estado atingido após uma preparação de dados (`UPDATE`), representando a espera da primitiva `DL_REPLY_UPDATE_STATUS` indication;
- . S4 - estado atingido após uma requisição de dados (`POLLING`), representando a espera da primitiva `DL_REPLY_STATUS` indication.

O módulo `LLC_ENTITY_TYPE` representa a entidade LLC Tipo 3. Na especificação, são criadas duas instâncias desse módulo (A e B), através da primitiva de Estelle `init`.

O corpo do módulo `LLC_ENTITY_TYPE` é do tipo `LLC_ENTITY_BODY`, possuindo dois pontos de interação: `UCEP` e `MCEP`. Através do ponto de interação `UCEP`, são passadas e recebidas as primitivas de serviço da subcamada `LLC`. Através do ponto de interação `MCEP`, são passadas e recebidas primitivas de serviço para a subcamada `MAC`.

Embora o padrão IEEE 802.2 Tipo 3 utilize apenas 3 estados para descrever o comportamento do componente de transmissão, foram declarados no corpo `LLC_ENTITY_BODY` 5 estados distintos para este componente.

O estado `WAIT_A` do padrão IEEE 802.2 Tipo 3 foi desmembrado nos estados `W_ACK_DATA` e `W_ACK_SYNC`. O estado `WAIT_R` do padrão IEEE 802.2 Tipo 3 foi desmembrado nos estados `W_REPLY_POLL` e `W_REPLY_XCHG`.

Tal desmembramento não incorreu em nenhuma violação do padrão durante a construção da especificação em Estelle. O uso de uma maior quantidade de estados na descrição do componente de transmissão foi adotado apenas para maior inteligibilidade e facilidade de construção da especificação.

Em resumo, os estados adotados para representar o comportamento do componente de transmissão são:

- . `IDLE` - neste estado, o componente de transmissão aceita requisições do usuário da camada de enlace para envio de um novo PDU de comando;
- . `W_ACK_DATA` - estado de espera de um PDU de reconhecimento sem dados, atingido após a transmissão de um PDU de comando contendo um LSDU não nulo;
- . `W_ACK_SYNC` - estado de espera de um PDU de reconhecimento sem dados, atingido após a transmissão de um PDU de comando contendo um LSDU nulo;
- . `W_REPLY_POLL` - estado de espera de um PDU de reconhecimento com dados, atingido após a transmissão de um PDU de comando contendo um LSDU nulo;
- . `W_REPLY_XCHG` - estado de espera de um PDU de reconhecimento com dados, atingido após a transmissão de um PDU de comando contendo um LSDU não nulo.

No protocolo `LLC` Tipo 3, o comportamento dos componentes de recepção e transmissão são independentes. Para que a especificação do protocolo em Estelle refletisse essa característica, representou-se o comportamento do componente de recepção em todos os possíveis estados do componente de transmissão.

O corpo `LLC_ENTITY_BODY` possui ainda diversas funções e "procedures", usadas para a construção de PDU's, identificação do tipo de PDU, identificação do status de recepção de PDU's de resposta, verificação da existência de LSDUs nos PDU's, etc.

O módulo `MEDIUM_TYPE` representa o comportamento da subcamada `MAC` e do nível físico. Na especificação, foi criada uma única instância desse módulo (meio de comunicação `DG`).

O corpo do módulo `MEDIUM_TYPE` é do tipo `MEDIUM_BODY`. O módulo `MEDIUM_TYPE` possui dois pontos de interação: `MCEP [0]` e `MCEP [1]`. Através desses pontos de interação, são passadas e recebidas as primitivas de serviço da subcamada MAC. Os pontos de interação `MCEP [0]` e `MCEP [1]` conectam-se, respectivamente, aos pontos de interação `MCEP` das instâncias A e B do módulo `LLC_ENTITY_TYPE`.

O corpo `MEDIUM_BODY` possui um canal bidirecional (`MEDIUM`) que permite a troca de `PDU's` entre as instâncias A e B do módulo `LLC_ENTITY_TYPE`. Esse canal conecta dois pontos de interação do corpo `MEDIUM_BODY` (`FIFO_A` e `FIFO_B`).

3 - VALIDAÇÃO DA ESPECIFICAÇÃO EM ESTELLE

O projeto do protocolo LLC foi iniciado com a construção de modelos em redes de Petri capazes de evidenciar aspectos funcionais e de controle do protocolo. Tais modelos foram extraídos das tabelas formais de transição de estados existentes no padrão IEEE 802.2 Tipo 3.

Foram usadas redes de Petri do tipo predicado-ação [14] para a modelagem inicial do protocolo LLC Tipo 3. A partir deste modelo inicial foi obtido um modelo global do protocolo usando redes de Petri clássicas. Este modelo global foi analisado com o programa ARP [15], possuindo um total de 33 lugares e 65 transições. A análise realizada através do programa ARP indicou que a rede de Petri usada para modelar o protocolo LLC Tipo 3 é uma rede limitada, viva e reinicializável.

A especificação em Estelle foi construída a partir do modelo em redes de Petri do protocolo LLC Tipo 3. Dessa forma, obteve-se uma especificação previamente validada através de analogia com modelos em redes de Petri.

O método de validação através de redes de Petri possui a vantagem de testar exaustivamente a especificação, uma vez que todas as seqüências possíveis de disparo das transições do modelo são pesquisadas.

Este método, todavia, apresenta algumas desvantagens. Uma dessas desvantagens é a possibilidade de introdução de erros humanos durante a construção da especificação a partir de modelos em redes de Petri.

Uma outra desvantagem é a necessidade de simplificação dos modelos com redes de Petri, necessárias para permitir a análise com ferramentas computadorizadas. Frequentemente, esta dificuldade impede que a especificação seja validada por completo, exigindo a construção de modelos globais simplificados ou modelos separados para diferentes trechos da especificação.

No caso específico da modelagem do protocolo LLC Tipo 3 com redes de Petri, verificou-se a necessidade de introdução de simplificações para limitar a quantidade de marcações acessíveis pela rede. Sem tais simplificações, o processamento da análise através do programa ARP era abortado com erro.

Para uma validação segura da especificação, torna-se necessário realizar um procedimento de simulação. O método de validação por simulação realiza

testes diretos com a especificação em Estelle, proporcionando portanto resultados bastante confiáveis.

O principal problema deste método reside no tipo de seqüência de testes usada para exercitar o simulador. Seqüências de testes geradas manualmente em geral não são capazes de realizar testes exaustivos da especificação. Para superar este problema, pode-se usar seqüências de testes geradas automaticamente com ferramentas computadorizadas.

O desenvolvimento de uma especificação em Estelle capaz de ser simulada exigiu a construção de "procedures", funções e corpos declarados como externos na versão preliminar da especificação. Em sua versão final, a especificação do protocolo LLC Tipo 3 atingiu cerca de 1500 linhas.

O procedimento de simulação da especificação utilizou os programas CE (Compilador Estelle) e SPDE (Simulador de Protocolos Descritos em Estelle) [16]. Estes programas fazem parte de um sistema de auxílio ao projeto de protocolos para a linguagem Estelle [17]. A Fig. 3.1 detalha a seqüência adotada para a simulação da especificação.

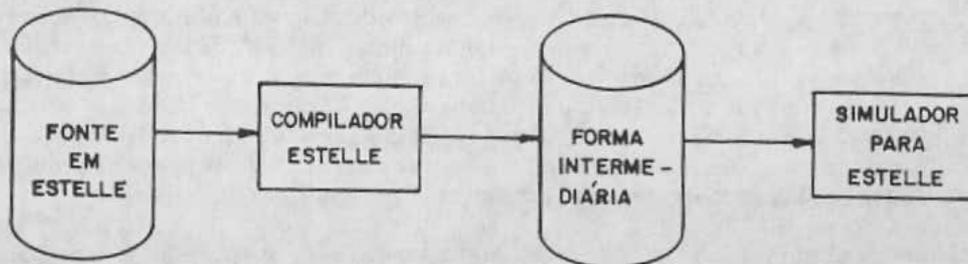


FIG. 3.1 - Simulação da especificação em Estelle

As seqüências de testes usadas no processo de simulação foram geradas manualmente, com o objetivo de validar o comportamento lógico da especificação.

Durante a primeira fase de testes, foram verificadas falhas no módulo LLC_ENTITY_TYPE, desenvolvido originalmente a partir de modelos com redes de Petri para o protocolo LLC Tipo 3. Tais erros foram originados pela falta de algumas transições, esquecidas durante a construção do módulo. Esse tipo de erro mostra de forma clara as limitações do método de validação indireta através de redes de Petri.

Após a correção dos erros existentes no módulo LLC_ENTITY_TYPE, procurou-se estabelecer a correção dos módulos, funções e rotinas desenvolvidos para a nova versão da especificação.

Uma vez verificada a correção do comportamento lógico da especificação, introduziu-se algumas alterações para tornar a entrada de interações e o procedimento de validação mais amigável.

O procedimento de simulação da especificação em Estelle foi realizado em um microcomputador do tipo IBM PC. A especificação foi submetida a diversas seqüências de teste, geradas manualmente pelo operador da simulação.

Verificando-se o comportamento da especificação através de interações com o operador e pela verificação dos traços de simulação, não se observou qualquer falha durante as diversas seqüências de teste realizadas.

Uma validação mais completa da especificação poderia ser obtida pelo uso de seqüências de teste geradas automaticamente. No momento, encontra-se em fase final de desenvolvimento um gerador de seqüências de testes [18] que se integrará ao sistema de auxílio ao projeto de protocolos para a linguagem Estelle [17].

4 - IMPLEMENTAÇÃO DO PROTOCOLO LLC TIPO 3

A especificação em Estelle do protocolo LLC Tipo 3 foi usada como referência para a implementação, aumentando os níveis de segurança do projeto e reduzindo a probabilidade de existência de erros. Para a obtenção de uma implementação eficiente, adotou-se o uso da linguagem de programação C.

Além de permitir uma implementação com reduzida incidência de erros e proporcionar uma base formal para construção de um código executável, o uso de uma especificação em Estelle apresenta como vantagem uma possível implementação semi-automática de protocolos. Embora o código em C do protocolo LLC Tipo 3 tenha sido escrito manualmente, pretende-se em breve realizar uma experiência utilizando uma ferramenta de implementação semi-automática atualmente em desenvolvimento [19].

A indisponibilidade inicial de uma quantidade suficiente de protótipos do hardware da rede local Mini-MAP fez com que o desenvolvimento fosse realizado sobre uma rede local Ethernet.

De forma a conferir um alto grau de portabilidade ao LLC Tipo 3 implementado, decidiu-se utilizar uma interface padronizada com a subcamada controle de acesso ao meio (MAC). A interface escolhida é a mesma definida pela Universidade de Clarkson para seus módulos de domínio público denominados "packet drivers" [20].

Um módulo "packet driver" proporciona uma interface de acesso padronizada para múltiplos usuários de uma mesma subcamada MAC. Algumas das primitivas disponíveis nessa interface são:

- . Send_pkt - primitiva para transmissão de pacotes;
- . Access_type - primitiva que permite ao usuário selecionar o modo de operação a ser utilizado para recepção de pacotes.

Até o momento, foram realizados testes com uma versão preliminar da implementação do protocolo LLC. Essa versão executa em um ambiente monousuário com sistema operacional DOS e requer uma interrupção de hardware para implementar o mecanismo de recepção de pacotes.

O interfaceamento do protocolo LLC com o "packet driver" é feito através de um módulo de emulação desenvolvido nas linguagens C e Assembler. A função desse módulo de emulação é converter a interface do "packet driver" para as primitivas de serviço MAC especificadas pelo padrão IEEE 802.2 Tipo 3. A Fig. 4.1 apresenta um esboço da arquitetura de implementação adotada.

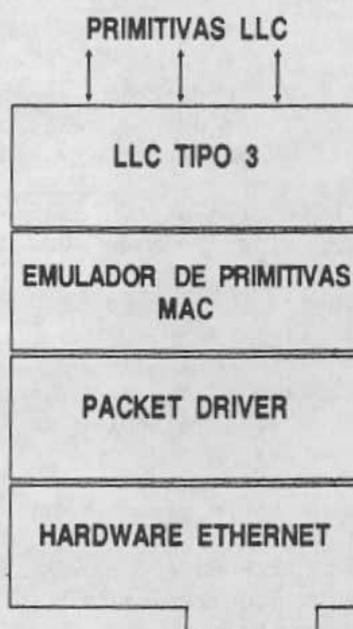


FIG. 4.1 - Arquitetura da implementação

Após ter sido inicialmente testada em um ambiente monousuário, a implementação do protocolo LLC Tipo 3 será transformada em um processo residente rodando sob a gerência de um núcleo multitarefas.

A adoção de uma interface padronizada para a subcamada MAC permitirá que a implementação do LLC Tipo 3 seja facilmente utilizada com outros tipos de hardware de rede local.

Para que o LLC possa ser utilizado com o hardware Mini-MAP, será necessário construir um módulo "packet driver" para o mesmo. Este módulo deverá ser desenvolvido em linguagem Assembler, que também foi adotada para a construção de programas de teste do hardware Mini-MAP.

O hardware Mini-MAP desenvolvido utiliza um controlador Motorola MC68824 para implementar o método de acesso ao meio da rede local. Esse controlador possui ainda a particularidade de proporcionar uma implementação em hardware da porção de recepção do protocolo IEEE 802.2 Tipo 3.

Tal característica apresenta como vantagens a obtenção de baixos tempos de resposta na rede local e a redução do código necessário para realização das funções da subcamada LLC.

A utilização do componente de recepção implementado pelo MC68824 pode ser feita através de rotinas desenvolvidas em linguagem ASSEMBLER. A

implementação em hardware do componente de recepção do protocolo IEEE 802.2 Tipo 3 poderá substituir parte do software já implementado em C, resultando em menores tempos de resposta e menor tamanho para o código.

5 - CONCLUSÃO

Este trabalho apresentou o projeto de um protocolo compatível com o padrão IEEE 802.2 Tipo 3. Tal protocolo será utilizado na subcamada controle de enlace lógico (LLC) de uma rede local Mini-MAP implementada através de técnicas formais de modelagem e validação.

O projeto do protocolo LLC iniciou-se com uma modelagem em redes de Petri para o padrão IEEE 802.2 Tipo 3. Após uma análise desses modelos, construiu-se uma especificação formal na linguagem Estelle para o protocolo. Esta especificação foi em seguida submetida a um processo de validação através de simulação.

O procedimento aplicado à especificação em Estelle do protocolo LLC Tipo 3 não pode ser considerado uma validação definitiva da especificação. Para tanto, seriam necessários testes exaustivos com seqüências geradas por computador. De qualquer forma, procurou-se eliminar todos os erros detectados por seqüências de teste geradas manualmente.

A especificação do padrão IEEE 802.2 Tipo 3 na linguagem Estelle está sendo usada como base para uma implementação segura do protocolo. Tal abordagem apresenta como vantagem o uso de uma especificação formal, descrita em uma linguagem relativamente próxima de linguagens de programação. Uma implementação realizada a partir de uma especificação formal torna-se também menos sujeita a erros e inaderências ao padrão.

6 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MAP/TOP Users Group of SME, "Manufacturing Automation Protocol Specification Version 3.0", Julho de 1987
- [2] MENDES, M. J., "Redes Locais de Comunicação em Ambiente Industrial", Máquinas e Metais, Agosto de 1986
- [3] MENDES, M. J.; MAGALHÃES, M. F., "Redes Locais Industriais e o Projeto de Padronização MAP/TOP", SBA Controle e Automação, Março de 1988
- [4] MENDES, M. J., "Comunicação Fabril e o Projeto MAP/TOP", Escola Brasileiro Argentina de Informática, Santiago Del Estero, Argentina, Janeiro de 1989
- [5] GRAUBE, M., "The Carrier Band Network and Mini-MAP: Low-Cost Solutions", Control Engineering, Outubro de 1986, 2a. Edição
- [6] APPEZATO, L., "Projeto e Desenvolvimento de uma Interface de Rede Local para Aplicações em Automação Industrial", Tese de Mestrado, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica, Junho de 1990

- [7] IEEE 802.2, "Type 3 Operation, Acknowledged Connectionless Services", SC6N 4960, Agosto de 1988
- [8] AZEVEDO, M. M.; PEDROZA, A. C. P., "Projeto e Implementação de uma Rede Local Mini-MAP", Anais do VIII Congresso Brasileiro de Automática (VIII CBA), Setembro de 1990
- [9] AZEVEDO, M. M., "Rede Local Mini-MAP para Automação Industrial", Tese de Mestrado, COPPE/UFRJ, Programa de Engenharia Elétrica, Abril de 1991
- [10] ALVES, L. C. A., "Protocolo MMS para Comando de Robôs por Rede Local", Tese de Mestrado, COPPE/UFRJ, Programa de Engenharia Elétrica, Abril de 1991
- [11] AZEVEDO, M. M., PEDROZA, A. C. P., "Avaliação de Topologias com Fibras Ópticas para Redes Locais Mini-MAP e MAP/EPA", VII Simpósio Brasileiro de Redes de Computadores (VII SBRC), Março de 1989
- [12] ISO/TC97/SC21/WG16-1, "Estelle : a Formal Description Technique Based on an Extended State Transition Model", DP9074, 1987
- [13] COURTIAT, J. P.; DEMBINSKI, P.; GROZ, R.; JARD, C., "Estelle: un Langage ISO pour les Algorithmes Distribués et les Protocoles", Institute National de Recherche en Informatique et en Automatique (INRIA), Rapport de Recherche No. 595, Dezembro de 1986
- [14] AYACHE, J. M.; COURTIAT, J. P.; DIAZ, M.; JUANOLE, G., "Utilisation des Reseaux de Petri pour la Modélisation et la Validation des Protocoles", Technique et Science Informatiques, Vol.4, No.1, 1985
- [15] UFSC/DEE/LCMI, "Manual de Utilização - ARP2", 1989
- [16] VALIM, P. R. O., "Um Simulador de Protocolos de Comunicação para a Linguagem Estelle", Tese de Mestrado, COPPE/UFRJ, Programa de Engenharia Elétrica, Abril de 1990
- [17] PEDROZA, A. C. P.; VALIM, P. R. O.; GOULART, C. C.; OLIVEIRA, R. C., "Um Sistema de Auxílio ao Projeto de Protocolos de Comunicação para Redes de Computadores", Anais do Seminário Franco-Brasileiro em Sistemas Informáticos Distribuídos, Setembro de 1989
- [18] SILVA, G. S., PEDROZA, A. C. P., "Um Sistema para Geração Automática de Sequências de Teste para Protocolos de Comunicação", VIII Simpósio Brasileiro de Redes de Computadores (VIII SBRC), 1990
- [19] ALCANTARA, S. C.; PEDROZA, A. C. P., "Implementação Semi-Automática de Protocolos Descritos em Estelle", Trabalho submetido ao X Simpósio Brasileiro de Redes de Computadores (X SBRC)
- [20] NELSON, R., "User Documentation for the Packet Driver Collection, Version 8.x", Clarkson University, Janeiro de 1991

ANEXO - ESPECIFICAÇÃO DO PROTOCOLO LLC TIPO 3 EM ESTELLE

Este Anexo apresenta um esboço da especificação formal em Estelle construída para o protocolo IEEE 802.2 Tipo 3, destacando principalmente a arquitetura da especificação. A íntegra da especificação pode ser encontrada em [9].

```

specification LLC systemactivity;
default individual queue; timescale seconds;

(*$external NetUser*)

const . . . ;    type . . . ;

channel LLC_interface (user, provider);
  by user:
    Data_ack_request(SDU: data_type); Reply_request(SDU: data_type);
    Reply_update_request(SDU: data_type);
  by provider:
    Data_ack_status_indication(U_STS: status_type);
    Reply_status_indication(U_STS: status_type);
    Data_ack_indication(SDU: data_type);
    Reply_indication(SDU: data_type);
    Reply_update_status_indication(U_STS: status_type);

channel MAC_interface (user, provider);
  by user:
    Unitdata_request(PDU: pdu_type);
  by provider:
    Unitdata_indication(PDU: pdu_type);
    Unitdata_status_indication(P_STS: status2_type);

channel SYS_interface (user,provider);
  by user:
    ENVIAR_MSG;    POLLING;    TROCA;    UPDATE;

module LLC_ENTITY_TYPE activity;
  ip UCEP: LLC_interface(provider);    MCEP: MAC_interface(user);    end;

body LLC_ENTITY_BODY for LLC_ENTITY_TYPE;

var
  N_attempts: integer;    save_tx    : data_type;
  . . .

state
  IDLE, W_ACK_DATA, W_ACK_SYNC, W_REPLY_POLL, W_REPLY_XCHG;

```

```

(* procedures que montam os PDUS *)

procedure Build_DATA(U_DATA:data_type; var pdu_aux:pdu_type);

begin
    . . .
    pdu_aux.LSDU := U_DATA;
    pdu_aux.CODE := PDU_DATA;
end;
    . . .

initialize
to IDLE
begin
    save_rx := '';    N_attempts := 0;    Vs := 1;    Vr := 1;
end;

(*** TRANSMISSOR (IDLE): Atende primitivas de servico. *****)
trans
from IDLE
when UCEP.Data_ack_request
provided (Existe_LSDU(SDU))    (* Verifica existencia de LSDU *)
to W_ACK_DATA
name IdleData:
begin
    save_tx := SDU;    (* Salva SDU para eventual retransmissao *)
    Build_data(SDU,PDU);
    PDU.seq := Vs;
    output MCEP.Unitdata_request(PDU);
    N_attempts := 0;
end;
. . . end;
. . .

end;

module USER_TYPE activity;
ip    UCĒP: LLC_interface(user);    USER: SYS_interface(provider);    end;

body USER_BODY for USER_TYPE; . . . end;

module MEDIUM_TYPE activity;
ip    MCEP: array[0..1] of MAC_interface(provider);    end;

body MEDIUM_BODY for MEDIUM_TYPE; . . . end;

```

```
modvar
  A, B: LLC_ENTITY_TYPE;
  UA, UB: USER_TYPE;
  DG: MEDIUM_TYPE;

initialize
  begin
    init DG with MEDIUM_BODY;
    init UA with USER_BODY;
    init UB with USER_BODY;
    init A with LLC_ENTITY_BODY;
    init B with LLC_ENTITY_BODY;
    connect UA.UCEP TO A.UCEP;
    connect UB.UCEP TO B.UCEP;
    connect A.MCEP TO DG.MCEP[0];
    connect B.MCEP TO DG.MCEP[1];
  end;
end.
```