

## ASPECTOS DE IMPLEMENTAÇÃO DE UM SERVIDOR MMS

Mônica V. C. Aguiar  
Elizabet Spohr  
Joberto, S. B. M.

Grupo de Redes de Computadores  
DSC/CCT/UFPB  
CP 10.117  
58.100 - Campina Grande - PB

### RESUMO

O objetivo deste artigo é apresentar um modelo de implementação de um servidor MMS que atue no ambiente de manufatura integrada. Esse modelo inclui um subconjunto de serviços que atendem, no mínimo, aos requisitos de comunicação de um dispositivo controlador lógico programável (CLP).

Será apresentada inicialmente uma visão geral dos serviços MMS, bem como do controlador lógico programável, seguida da definição dos serviços MMS a serem implementados com base nas funções de um CLP. Finalmente, será apresentado um modelo de implementação, incluindo a descrição da estrutura de processos e das estruturas de dados.

### ABSTRACT

This paper presents a MMS server implementation model that actuates in an integrated manufacturing environment. This model includes a subset of services that attends, at least, the communications requirements of a programmable logic controller device (PLC).

Initially, it presents a survey of MMS services and PLCs, followed by the definition of the MMS services that will be implemented based on PLC's functions. Finally, it presents the implementation model, including the process and data structures description.

### 1. INTRODUÇÃO

A partir da década de 70 houve um acelerado desenvolvimento das técnicas de comunicação digital, envolvendo dados, voz e imagem. Acompanhando essa evolução começaram a surgir as primeiras redes locais de computadores.

Particularmente na indústria, as redes locais têm tido penetração cada vez maior e isto se deve a vários fato-

res, entre eles a filosofia CIM ("Computer Integrated Manufacturing") [1]. Integrar uma manufatura por computador significa interligar todos os sistemas computadorizados existentes na empresa, incluindo: controladores lógicos programáveis (CLPs), dispositivos de comando numérico computadorizado (CNCs), dispositivos de comando de robôs (CNRs), etc., além da área administrativa e de projetos [2].

Os problemas associados com a integração operacional ocorrem principalmente em relação à interligação de equipamentos de diferentes fabricantes, uma constante na indústria. Isso tornou essencial o desenvolvimento de projetos que estabelecessem um padrão único de troca de mensagens num ambiente fabril.

Um desses projetos é o MAP ("Manufacturing Automation Protocol") [3], baseado no Modelo de Referência OSI ("Open Systems Interconnection") da ISO ("International Standards Organization") [4].

A camada de aplicação proposta para o MAP prevê o uso do protocolo MMS ("Manufacturing Message Specification") [5]. Este protocolo destina-se à transmissão de mensagens entre dispositivos programáveis da manufatura e controle de processos, entre eles o CLP.

Os aspectos específicos do MMS associados a cada tipo de dispositivo são descritos em documentos denominados "Companion Standards". Esses documentos descrevem, entre outras coisas, o mapeamento das funções do dispositivo em serviços MMS e a especificação das classes de serviços a serem suportados pelo dispositivo.

Neste artigo será apresentado um modelo de implementação de um subconjunto MMS que atenda aos requisitos funcionais de um CLP.

## 2. DESCRIÇÃO DOS SERVIÇOS E DO PROTOCOLO MMS

O protocolo MMS é baseado no conceito de **Dispositivo Virtual de Manufatura (VMD)**. O VMD modela o comportamento visível externamente da aplicação de um servidor MMS.

Na implementação de um servidor MMS deve-se efetuar o mapeamento do modelo VMD para a funcionalidade de um dispositivo real da manufatura. Como já foi dito, a orientação para seleção de um mapeamento particular pode ser encontrada nos "Companion Standards" [6].

No modelo adotado o VMD reside dentro de um processo de aplicação servidor MMS, tornando disponível (para controle e/ou monitoramento) um conjunto de recursos e funcionalidades associados com um dispositivo real. Um processo de aplicação pode conter zero ou mais VMDs, cada um representando determinado dispositivo virtual, logicamente independente dos demais. Se o processo de aplicação não define um VMD, ele não pode agir como um servidor MMS.

Cada VMD contém uma função executiva, um ou mais domínios, zero ou mais estações de operador e uma memória virtual de arquivos (opcional). Essa estrutura é mostrada na fig. 1 e descrita abaixo:

- Função Executiva:

Gerencia o acesso do MMS aos recursos do VMD (memória, processadores, portas de E/S, etc.), tornando esses recursos visíveis ao cliente. A função executiva exprime os recursos do VMD como "capacidades", as quais são usadas para descrever os elementos constituintes do domínio e para checar a validade de acesso a objetos MMS.

- Domínio:

O VMD tem no mínimo um domínio, chamado "M\_Executive", que contém todos os recursos do VMD.

Cada domínio representa um subconjunto dos recursos do VMD, o qual é usado com uma finalidade específica. Esse subconjunto refere-se a aspectos do VMD associados a um elemento específico (possivelmente todos) de uma estratégia coordenada de controle e/ou monitoramento.

A alocação de domínios pode ser estática ou dinâmica. Na alocação estática o domínio é pré-definido dentro do servidor MMS, não pode ser apagado e seu nome é conhecido. Na alocação dinâmica o domínio é criado e removido do VMD por serviços MMS. No VMD pode-se encontrar um dos dois ou ambos os tipos de domínio.

- Estação do Operador:

Responsável pela entrada e saída de informações de/para outra estação de operação, parte de outro VMD. A estação do operador é evocada como um objeto separado do VMD.

- Memória Virtual de Arquivos:

Consiste de uma coleção de arquivos com nome e age como um armazenador de dados e programas.

Como o nome indica, é um objeto virtual e, portanto, a implementação deve mapeá-lo para a memória real de arquivos.

Para que haja compatibilidade com o ISO 8571 FTAM, a memória virtual de arquivos do MMS é definida como um subconjunto do sistema de arquivo virtual do FTAM.

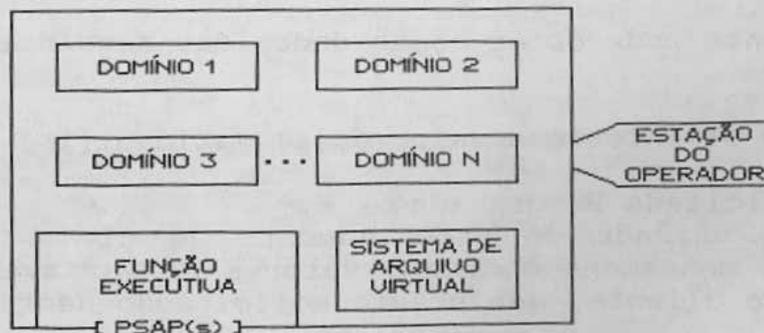


FIG. 1 - MODELO DO VMD MMS

Os serviços MMS são definidos com referência a um modelo geral cliente-servidor, sendo levados em consideração os mais diversos tipos e funções dos dispositivos programá-

veis da manufatura aos quais se destina o protocolo. Isso resulta numa grande quantidade de serviços oferecidos, sendo que as implementações deverão utilizar apenas os serviços necessários ao dispositivo em questão.

Os serviços suportados pelo MMS são agrupados em Unidades Funcionais [5].

### 3. O CONTROLADOR LÓGICO PROGRAMÁVEL

O controlador lógico programável é um dispositivo comumente utilizado no chão de fábrica e pode aparecer na situação de cliente ou de servidor. Como cliente, requisita funções dos servidores, que serão sempre CLPs; como servidor, executa funções requisitadas pelo cliente, o qual pode ser um CLP ou outro dispositivo.

Na implementação em questão, o subconjunto MMS será especificado de forma a prover, no mínimo, as funções de um equipamento CLP servidor. Essas funções são definidas pela IEC ("International Electrotechnical Commission") e serão descritas a seguir [6]:

#### - Verificação de Dispositivo:

Este serviço permite que o cliente verifique se o CLP (servidor) está apto a executar suas próprias funções. A verificação de dispositivo inclui três tipos de informação: informação de estado, detecção de falhas e isolamento de falhas. Essas informações referem-se à situação de cada subsistema do CLP (memória, sistema de E/S, unidade de processamento, etc.). O CLP pode enviar tais informações com ou sem prévia solicitação deste.

#### - Aquisição de Dados:

O CLP envia ao cliente dados provenientes de diversas fontes: processos, máquinas, resultados de cálculos de programas, etc.

O cliente pode obter esses dados das seguintes formas:

##### . "Polled"

O cliente lê o valor da(s) variável(is).

##### . Não Solicitada Programada

A unidade de processamento do CLP é programada para enviar mensagens contendo valores de variáveis de processo para o cliente, sem prévia solicitação deste.

##### . Não Solicitada Configurada

A interface de comunicação do CLP é configurada pelo cliente para enviar mensagens, sem prévia solicitação deste.

#### - Controle:

O cliente controla a operação do CLP por dois métodos:

. Paramétrico  
O controle é feito escrevendo-se valores nas variáveis de controle residentes no CLP.

. Intertravamento  
O cliente solicita ao servidor que efetue determinada operação e o informe do resultado.

- Sincronização:

Este serviço é efetuado através do uso de semáforos negociados entre os usuários de aplicação.

- Alarme:

Alarmes são mensagens enviadas do CLP para o cliente quando ocorre determinada condição.

O serviço de alarme pode ser iniciado de duas formas:

. Programado

A unidade de processamento do CLP é programada para enviar ao cliente mensagens de alarme contendo os valores de interesse.

. Configurado

A interface de comunicação do CLP é configurada para enviar mensagens de alarme ao cliente.

- Interface com Operador:

Os dispositivos de interface com operador são usados pelo operador para monitorar e/ou modificar os processos controlados.

- Controle e Gerenciamento de Programa:

Inclui as seguintes funções: relatório do estado do programa de aplicação do CLP; controle de execução do programa de aplicação do CLP e transferência de conteúdo da memória programável do CLP para o cliente (acompanhada ou não da devolução deste conteúdo para o CLP).

#### 4. MAPEAMENTO DAS FUNÇÕES DO CLP PARA SERVIÇOS MMS

O IEC define 5 (cinco) classes de conformidade para os serviços de mensagens do CLP [6]. A tabela 1 mostra as classes de conformidade e as funções do CLP providas em cada classe.

CLASSE CLP	FUNÇÕES PROVIDAS POR CLASSE
1	VERIFICAÇÃO DE DISPOSITIVO SOLICITADA AQUISIÇÃO DE DADOS TIPO "POLLED" CONTROLE PARAMÉTRICO
2	GERENCIAMENTO DE PROGRAMA
3	VERIFICAÇÃO DE DISPOSITIVO NÃO SOLICITADA AQUISIÇÃO DE DADOS PROGRAMADA CONTROLE POR INTERTRAVAMENTO
4	AQUISIÇÃO DE DADOS CONFIGURADA ALARME PROGRAMADO SEMÁFOROS
5	ALARME CONFIGURADO

TABELA 1 - CLASSES DE CONFORMIDADE DO CLP (IEC)

As classes estão estruturadas hierarquicamente, de modo que cada classe engloba os serviços da classe anterior. Assim, a classe 2 inclui, além da função de Gerenciamento de Programa, as funções de Aquisição de Dados tipo "Polled", Verificação de Dispositivo Solicitada e Controle Paramétrico (definidas para a classe 1); a classe 3 inclui, além das funções de Aquisição de Dados Programada, Verificação de Dispositivo Não Solicitada e Controle por Intertravamento, as funções definidas para as classes 2 e 1; e assim por diante.

Os serviços MMS requeridos para atender a cada classe de conformidade são dados na tabela 2 abaixo:

CLASSES CLP	SERVIÇOS MMS
Classe 1	Initiate, Conclude, Abort, Reject, Identify, Status, Get Name List, Read, Write
Classe 2	Initiate Download Sequence, Initiate Upload Sequence, Download Segment, Upload Segment, Terminate Download Segment, Terminate Upload Segment, Delete Domain, Get Domain Attributes, Create Program Invocation, Delete Program Invocation, Start, Resume, Reset, Stop, Get Program Invocation Attributes
Classe 3	Unsolicited Status, Information Report
Classe 4	Cancel, Report Event Condition Status, Alter Event Condition Monitoring, Event Notification, Acknowledge Event Notification, Report Event Action Status, Report Event Enrollment Status, Get Alarm Summary, Take Control, Relinquish Control, Report Semaphore Status, Report Semaphore Entry Status
Classe 5	Define Event Action, Define Event Enrollment, Define Event Condition, Delete Event Action, Delete Event Enrollment, Delete Event Enrollment, Delete Event Condition, Get Event Action Attribute, Get Event Enrollment Attribute, Get Event Condition Attribute, Alter Event Enrollment, Define Semaphore, Delete Semaphore

TABELA 2 - MAPEAMENTO DAS FUNÇÕES CLP PARA SERVIÇOS MMS

A implementação MMS proposta prevê a utilização de serviços até a classe 3 do CLP; ou seja, serão especificados apenas os serviços MMS necessários ao atendimento das funções providas por um CLP classe 3. Entretanto, a estrutura de dados será definida de modo a suportar posteriores extensões para as classes 4 e 5.

## 5. MODELO DE IMPLEMENTAÇÃO DO MMS

O modelo definido inclui a especificação de três tópicos principais: estrutura dos processos, interface com as camadas adjacentes e estruturas de dados.

A interação entre as estruturas especificadas é mostrada na fig. 2. Após a figura é apresentada a descrição de cada uma.

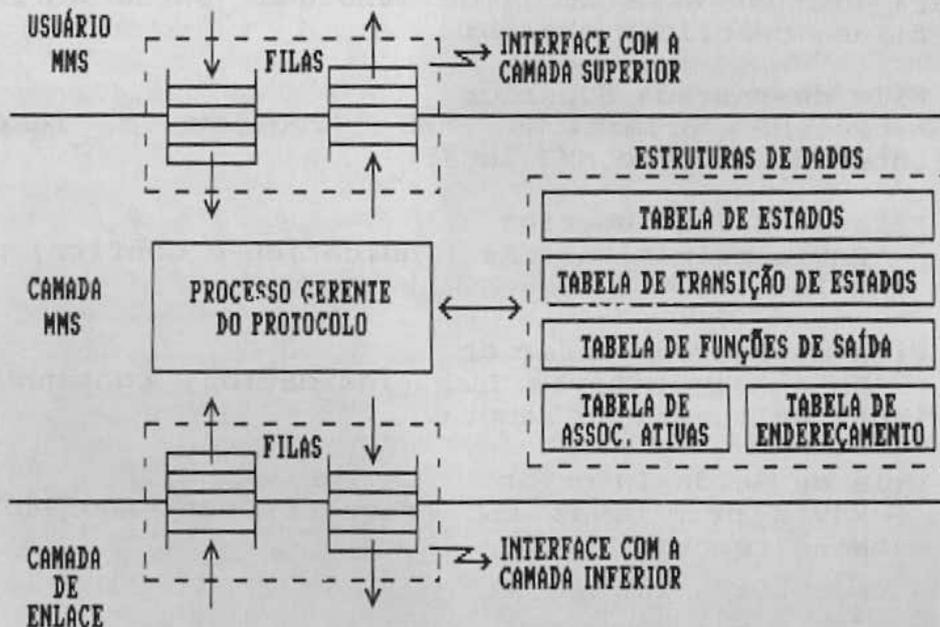


FIG. 2 - PROCESSO, ESTRUTURAS DE DADOS E INTERFACES PARA O MMS

### - Estrutura dos Processos

Definiu-se uma estrutura com apenas um processo. Este processo será encarregado de todas as funções de gerenciamento da camada:

- . Controle das interfaces com as camadas adjacentes;
- . Codificação/decodificação de PDUs;
- . Verificação de sintaxe e semântica das PDUs recebidas;
- . Manutenção e acesso às estruturas de dados;

- . Execução de serviços;
- . Invocação de primitivas.

Optou-se por esta estrutura tendo em vista que uma estrutura multi-processos poderia comprometer o desempenho do protocolo. Isto seria provocado pela utilização dos recursos de memória compartilhada e de filas, cujo acesso é lento, para prover a comunicação entre os processos (para troca de mensagens e parâmetros).

#### - Interface Entre as Camadas Adjacentes

Essa interface possibilita a troca de mensagens entre o processo e cada camada adjacente (superior e inferior). No caso deste projeto, a camada MMS é utilizada numa arquitetura mini-MAP [12], fazendo interface, portanto, com a camada de enlace lógico no nível mais inferior.

Para efetuar essa troca de mensagens serão utilizadas quatro filas, descritas abaixo:

- . Fila de Entrada Superior  
Recebe primitivas MMS (request e response) provenientes do usuário MMS local.

- . Fila de Saída Superior  
Envia primitivas MMS (indication e confirm) para o usuário MMS local.

- . Fila de Entrada Inferior  
Recebe primitivas LLC (indication) contendo PDUs MMS enviadas pela estação remota.

- . Fila de Saída Inferior  
Envia primitivas LLC (request) contendo PDUs MMS para a estação remota.

#### - Estruturas de Dados

O modelo de implementação adotado baseou-se nas propostas encontradas em [7], [8] e [9]. Essas propostas sugerem estruturas de dados semelhantes, as quais são especificadas de forma disparar ações de saída como consequência da ocorrência de um evento (chegada de primitiva). Isso é feito através de pesquisas em tabelas, utilizando um mínimo de código de programação.

As estruturas de dados que implementam a máquina de protocolo do MMS compõem-se de três tabelas principais: Tabela de Estados, Tabela de Transição de Estados e Tabela de Funções de Saída. É dada, a seguir, uma descrição destas tabelas:

- . Tabela de Estados  
Esta tabela é um arranjo unidimensional indexado por conexão.

Cada requisição de serviço recebida pelo MMS gera uma máquina de estados, que corresponde a uma entrada na tabela. A máquina deixará de existir quando a instância de serviço pendente for encerrada.

São mantidos na tabela os seguintes dados:

- . identificação da instância de serviço pendente;
- . estado atual da conexão;
- . nome do serviço pendente;
- . tipo de PDU esperada;
- . endereços de pontos de acesso locais e remotos;
- . parâmetros MMS negociados durante o estabelecimento da associação;
- . etc.

Quando ocorre um evento, a tabela de estados é pesquisada para obtenção do estado atual da conexão. Com estes dados (evento e estado) efetua-se uma busca na tabela de transição de estados.

#### . Tabela de Transição de Estados

Esta tabela é uma lista encadeada, onde estão armazenadas apenas as intersecções evento/estado válidas.

Cada nó da lista armazena os seguintes dados: evento, estado, um apontador para função de teste de predicado e um número inteiro, usado como primeiro índice na tabela de funções de saída.

Os testes de predicado efetuam verificações semânticas dos eventos em relação aos estados das conexões, entre outras coisas. Cada função de teste de predicado retorna um número inteiro, que será usado como segundo índice na tabela de funções de saída.

#### . Tabela de Funções de Saída

Esta tabela é um arranjo bidimensional, indexado pelo inteiro encontrado na tabela de transição de estados e o inteiro retornado pela função de teste de predicado.

Nem todos os elementos do arranjo são entradas válidas na tabela, tendo portanto conteúdo nulo. As entradas válidas contêm um apontador para a função de saída correspondente ao evento/estado/teste de predicado.

Cada função executa uma série de ações de saída, tais como: montagem de PDUs, envio de primitivas, atualização do estado na tabela de estados, etc.

O exemplo a seguir ilustra melhor o mecanismo de pesquisa nas tabelas:

- O MMS recebe uma primitiva request proveniente do usuário (evento = X\_REQ);
- Após verificar, na tabela de estados, o estado atual da conexão, o MMS pesquisa a intersecção correspondente na tabela de transição de estados;
- Supondo que a conexão esteja inativa (estado atual = CLOSED), é efetuada uma pesquisa na tabela de transição de estados com a dupla X\_REQ/CLOSED. Como se pode observar pela fig. 3, a entrada da tabela cor-

respondente a essa intersecção contém os seguintes dados: função de teste de predicado = t\_pred\_2 e indice\_1 = 0;

EVENTO	ESTADO	CLOSED	WAIT_NULL_ACK	...
X_REQ		test_pred = t_pred_2 indice_1 = 0	NULL	
X_RESP		test_pred = t_pred_1 indice_1 = 1	NULL	
:				
:				

FIG. 3 - INICIALIZAÇÃO DA TABELA DE TRANSIÇÃO DE ESTADOS

- O teste de predicado t\_pred\_2 é executado e, com o valor retornado pela função e o indice\_1, será localizada na tabela de funções de saída a função de saída correspondente;
- Supondo que a primitiva recebida foi INITIATE\_REQ, a função t\_pred\_2 retornará o valor 1. Logo, a tabela de funções de saída será pesquisada com o índice (0,1), que corresponde à função de saída fsd\_1 (fig. 4);

```
int (* tfsaida [17][13])() =
{
  err_1, fsd_1 , fsd_2 , fsd_17, fsd_18, NULL , ....
  err_2, fsd_4 , fsd_5 , fsd_6 , fsd_7 , NULL , ....
  .
  .
  .
}
```

FIG. 4 - INICIALIZAÇÃO DA TABELA DE FUNÇÕES DE SAÍDA

- A função fsd\_1 executa as seguintes ações: cria uma máquina de estados para o serviço (na tabela de estados), envia uma primitiva LLC para a sincronização da rede e atualiza o estado da conexão na tabela de estados (novo estado = WAIT\_NULL\_ACK).

Além destas tabelas principais, outras estruturas foram definidas:

- . Tabela de Associações Ativas  
Mantém informações sobre as associações ativas

(endereços de pontos de acesso locais, prioridade, etc.), necessárias mesmo após a execução dos serviços e enquanto a associação permanecer ativa.

O objetivo desta tabela é indicar as associações ainda ativas, uma vez que a tabela de estados mantém informações sobre as associações apenas enquanto o serviço estiver pendente. Quando o serviço é concluído, a máquina de estados é eliminada da tabela de estados, mas a associação permanece ativa até que seja encerrada, de forma normal (primitiva `conclude`) ou abrupta (primitiva `abort`).

#### . Tabela de Endereçamento

Contém os endereços dos pontos de acesso locais agrupados por tipo de comunicação: com associação, sem associação e resposta imediata. Assim, cada tipo de comunicação só tem acesso a determinado grupo de endereços.

As primitivas e PDUs MMS foram definidas através de um mapeamento manual da notação ASN.1 ("Abstract Syntax Notation One") [10], para uma codificação em linguagem C. A definição da estrutura das primitivas e PDUs em notação ASN.1 encontra-se no documento MMS.

As primitivas LLC foram codificadas de acordo com o documento LLC 3 [11].

## 6. CONCLUSÃO

Nesse artigo foram apresentados alguns aspectos essenciais da implementação de um servidor MMS para CLPs. O servidor é suportado por uma rede industrial segundo a arquitetura de protocolos mini-MAP e executa num supermicro com um núcleo orientado para implementação de protocolos de comunicação.

O modelo de implementação adotado utiliza uma estrutura simplificada em termos de processos e interfaces e, desta forma, procura estabelecer uma base de avaliação para a implementação de protocolos em ambientes com restrições de tempo de resposta (tempo-real).

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Mc GUFFIN, L.J. et al.; "MAP/TOP in CIM Distributed Computing". IEEE Network, v. 2, n. 3, maio, 1988.
- [2] MENDES, M.J. e MAGALHÃES, M. - "Redes Locais Industriais e o Projeto de padronização MAP/TOP". SBA: Controle e Automação, v. 2, n. 1, p. 56-70.
- [3] GENERAL MOTORS, "MAP Specification - Version 3.0", 1987.
- [4] ISO - "Open Systems Interconnection - Basic Reference Model", ISO-IS 7498, 1984.
- [5] EIA - "Manufacturing Message Specification - Part 1:

- Service Specification", Project 1393A, Draft 6, maio, 1987.
- [6] IEC - "Programmable Controller Message Specification", SC65A/WG6/TF7, fev., 1989.
  - [7] LIMA, C.P.A. - "Um Modelo para Implementação de um Serviço e Protocolo de Transferência de Arquivos em Rede". Tese de Mestrado em Informática, UFPB, Campina Grande - PB, maio, 1990.
  - [8] HALSALL, Fred - Data Communications, Computer Networks and OSI, Addison-Wesley, 1988.
  - [9] JOBERTO, S.B.M. and TURNELL, D.J.; "Implementing Multi-Layer Protocol Architectures on a Front-End Processor". In: IV Latin American Congress on Automatics. México, 1990.
  - [10] ISO - "OSI - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", ISO DIS 8825, ago., 1986.
  - [11] ISO/IEC - "Standards for Local Area Networks: Logical Link Control - Type 3 Operation, Acknowledged Connectionless Service", Project JTC 1/SC6, jun., 1988.
  - [12] TURNELL, D.J. e JOBERTO, S.B.M.; "Uma Implementação das Camadas MAC/LLC para a Interface Mini-MAP". Trabalho aceito para o 8o. SBRC, Campinas, 1990.