UNIVERSIDADE DO PORTO
FACULDADE DE ENGENHARIA

Dissertação de Mestrado

Modelação e Simulação de um Barramento de Campo utilizando uma Linguagem de Simulação Orientada ao Objecto.

Maria Margarida Carreira Pires Urbano

Setembro de 1999

ÍNDICE GERAL

Agradecimentos

Agradeço profundamente aos meus orientadores - Prof. Dr. José Marques dos Santos e ao Prof. Dr. João José Ferreira - pela preciosa orientação na execução desta dissertação.

Agradeço também ao Prof. Dr. José Grácio e ao Engº José Paulo Santos, da Secção Autónoma do Departamento de Mecânica da Universidade de Aveiro, pelas facilidades concedidas na utilização de recursos informáticos para a realização desta dissertação.

Ao Prof. Dr. José Alberto Fonseca do Departamento de Electrónica da Universidade de Aveiro, agradeço as suas preciosas ajudas técnicas.

Por fim, agradeço a todos os meus colegas da Escola Superior de Tecnologia e Gestão de Águeda, com os quais convivo diáriamente pelo apoio que me prestaram.

Introdução

Vivemos numa sociedade de informação que é fruto da utilização crescente dos computadores e que depende fortemente da sua evolução tecnológica. A vida quotodiana de uma empresa industrial não é excepção a esta realidade actual estando cada vez mais, dependente da utilização do computador e das redes informáticas.

No entanto, a evolução permanente e a complexidade crescente dos sistemas e redes informáticas tem exigido a progressiva utilização de ferramentas que facilitem o estudo do seu comportamento. Antes da sua concepção, é necessário ter a garantia, tanto quanto possível, que um sistema vai corresponder aos objectivos que lhe são propostos. Para isso, recorre-se a ferramentas (CASE("Computer Aided Software Engineering")) que permitem fazer a análise, a especificação e a concepção do sistema com o objectivo do seu desenvolvimento, validação e integração. A fase da análise, é realizada cada vez mais com a ajuda da modelação do sistema, por forma a simulá-lo antes de se passar à fase seguinte da validação. A modelação e a simulação têm cada vez mais um papel importante na concepção de e compreensão dos novos sistemas.

Por outro lado, a rede informática surgiu com a necessidade de partilhar os diferentes recursos informáticos, tanto físicos como de informação, permitindo a interligação de vários componentes por forma a que possam comunicar entre si. Num meio industrial existem diferentes tipos de rede, consoante o tipo de equipamento que interligam. Um dos tipos de rede é designado por barramento de campo que permite a interligação de equipamentos ao nível de sensores, actuadores e controladores.

O trabalho realizado tinha como objectivo fundamental demonstrar a possibilidade de utilizar uma linguagem de simulação geral, o Simple++, para modelizar e estudar o comportamento de barramentos de campo. O Simple++, sendo uma linguagem orientada ao objecto, permite um ambiente integrado de simulação, animação e interface gráfico. É muito aplicado na simulação de sistemas de fabrico, pelo que pareceu interessante estudar a possibilidade de o utilizar também na simulação de redes do tipo barramento de campo, evitando o recurso a outra ferramenta específica, ou seja, a simuladores de redes.

Preâmbulo

Esta dissertação é composta por seis capitulos cujos conteúdos apresentamos seguidamente.

O primeiro capitulo começa por fazer uma apresentação geral das redes locais, mais exactamente às redes locais industriais realçando a importância do modelo de referência OSI para a interligação de diferentes tipos de equipamento. O objectivo é o de apresentar as funções base de um barramento de campo e a sua importância para a implementação de sistemas distribuidos em tempo real.

O segundo capitulo faz uma apresentação do funcionamento do barramento de campo WorldFIP.

No terceiro capitulo, tem-se como objectivo fazer uma apresentação das linguagens de programação orientada ao objecto como ferramentas de ajuda à implementação e simulação de um sistema, para avaliar o seu desempenho. Assim, começa-se por fazer uma abordagem às técnicas de avaliação de desempenho de um sistema, expondo de uma forma mais detalhada a técnica da simulação para sistemas de eventos discretos. Por fim, apresentam-se as vantagens da programação orientada ao objecto relativamente às linguagens tradicionais, bem como, as suas caracteristicas fundamentais.

O quarto capitulo, faz uma apresentação do modelo criado no SIMPLE++ para o barramento de campo WorldFIP. Começa por fazer uma apresentação da filosofia de construção de um modelo no Simple++, utilizando os objectos da livraria de objectos do Simple++. Exemplifica a seguir quais os serviços do WorldFIP que vão ser modelizados e por fim, como se construiu o modelo.

No quinto capitulo são apresentados os resultados da simulação, de acordo com os critérios escolhidos.

O sexto e último será o capitulo onde se faz um resumo das conclusões principais do trabalho realizado, apresentando ainda, pistas para trabalho futuro.

Capítulo 1 - Redes locais - LAN

1.1 Introdução

Este capítulo tem como objectivo apresentar o conceito de rede local industrial ao nível da célula de fabrico, nomeadamente um barramento de campo, actualmente muito utilizado por garantir um conjunto de vantagens aos seus utilizadores relativamente às ligações ponto a ponto.

Em primeiro lugar, começa-se por fazer uma abordagem aos diferentes tipos de aplicação de redes, mais exactamente às redes locais industriais e à forma como os equipamentos comunicam entre si tendo como referência o modelo OSI. Em seguida, explica-se quais são as funções base de um barramento de campo e quais as suas vantagens em relação a um sistema clássico de entradas e saidas.

1.2 Redes Locais -LAN

Uma rede de comunicação é uma infraestrutura que permite que um conjunto de equipamentos interligados sejam capazes de comunicarem entre si.

Nas aplicações industriais os equipamentos integram interfaces de comunição que os tornam, muitas vezes, indissociáveis de uma rede específica. A vantagem principal das redes é a de permitir que os vários utilizadores partilhem os diversos recursos oferecidos pelos diferentes equipamentos.

As redes podem ser classificadas segundo diferentes critérios, sendo um deles a distância entre os equipamentos também conhecidos por sistemas, estações ou nós. De acordo com o critério anterior, a rede local ("Local Area Network" - LAN) é um sistema de comunicação que se restringe a uma área geográfica limitada (na ordem dos 10 quilómetros)[1.1].

A LAN é constituída pelos dispositivos próprios da rede, como os PC's, e as impressoras e por outro tipo de recursos como os repetidores, os "hubs" e as "bridge", que são necessários para que a comunicação se torne possível e fiável entre os diversos componentes.

Os diferentes equipamentos são interligados através da topologia mais adequada (barramento, estrela ou anel) e comunicam entre si através de mensagens ou pacotes

de informação utilizando como suporte físico o cabo coaxial, a fibra óptica ou cabo entrançado blindado[1.2]. Os pacotes de informação é um conjunto de bits que inclui, além da informação útil, informação de gestão de erros.

Um dos principais campos de aplicação da LAN é em sistemas de automação para escritórios, laboratórios e ambientes fabris. Inicialmente, este tipo de rede surgiu como um sistema fechado, onde a solução da rede era muito especifica à área de trabalho e dependente do fabricante do sistema automatizado, de tal modo que outro equipamento, que não fosse do mesmo fabricante, dificilmente se integraria na aplicação.

Actualmente, as LAN oferecem soluções de interligação "standard" para sistemas abertos e a escolha de equipamento para implementar uma rede deixa de estar dependente do fabricante. O conceito de interligação de sistemas heterogéneos tem como referência o modelo OSI ("Open Systems Interconnection")[1.3] que se descreve a seguir.

1.3 Principais Conceitos do Modelo OSI

Como se afirmou antes, o modelo OSI constitui uma referência para a interligação de sistemas abertos, permitindo que software e equipamentos de diferentes fabricantes possam comunicar entre si. Este modelo define uma arquitectura "standard" para qualquer sistema por forma a possibilitar a comunicação entre sistemas diferentes sem que haja prejuízo para o utilizador final. Um utilizador final pode ser uma pessoa responsável pela gestão da manutenção ou da qualidade, uma máquina CNC ou um PC.

A interacção entre sistemas constituintes da rede requer mecanismos de comunicação, de controlo e de sincronização. Por exemplo, uma aplicação industrial pode comportar a gestão da produção, a gestão da manutenção e do desenvolvimento de produto, sendo constituida por vários tipos de software que terão de comunicar e cooperar entre si. Os sistemas trocam entre si mensagens que contêm, para além da informação útil, também informação que permite o controlo de erros e de sincronização. Na comunicação são necessárias regras e procedimentos que definam o que fazer em determinadas situações e que determinem o caminho a dar à informação

O modelo OSI define uma arquitectura em sete camadas, em que cada camada tem uma função muito específica e oferece um serviço à camada imediatamente superior. A figura seguinte (figura 1.1) apresenta a estrutura do modelo OSI e descreve de uma forma sumária a função de cada uma das camadas [1.3],[1.4].

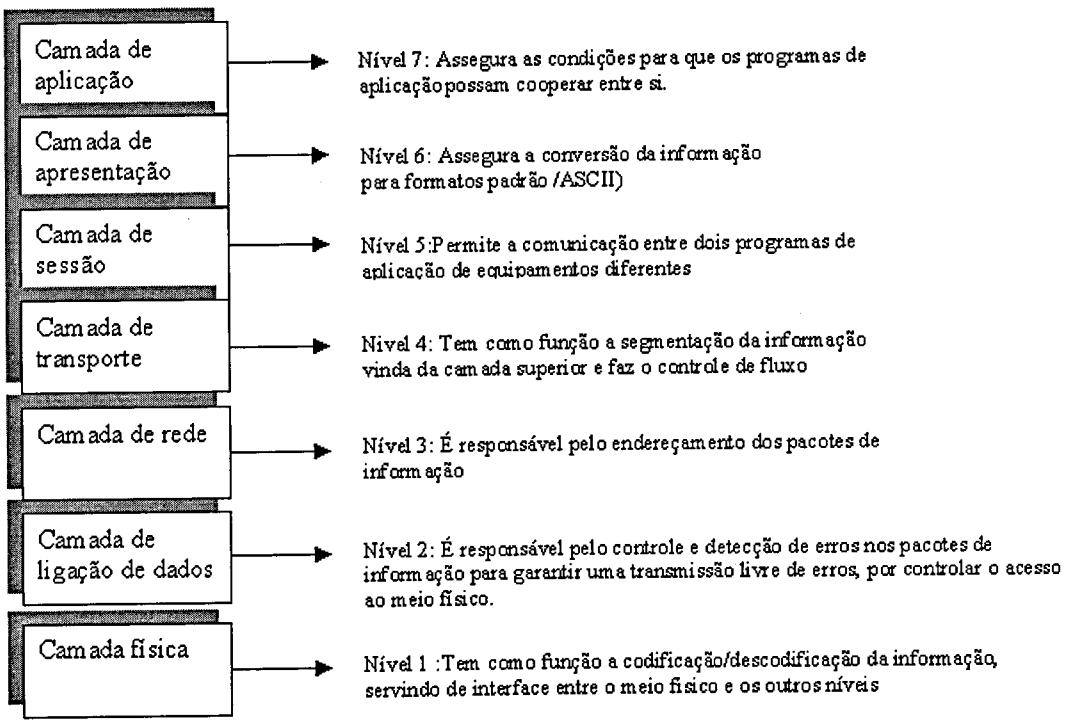


Figura 1.1 - As sete camadas do modelo OSI e as funções respectivas.

Analisando um sistema com uma estrutura em camadas, as entidades homólogas comunicam entre si, utilizando os serviços das camadas inferiores. Como se pode ver na figura 1.2, quando um programa utilizador A (nó emissor) necessita de uma informação do qual o outro sistema B (nó receptor) é responsável, envia uma mensagem via camada de aplicação (mensagem assinalada pelo percurso (1) na figura 1.2) que vai sendo sucessivamente transmitida às camadas imediatamente inferiores, até chegar à camada física. Por sua vez, a camada física é responsável por transmitir a mensagem ao meio físico e este por a transferir para o nó receptor. O percurso da mensagem, quando chega ao nó receptor, é idêntico ao anterior mas em sentido inverso (vai subindo pelas diferentes camadas). O programa utilizador B responde produzindo uma mensagem resposta (assinalada pelo percurso 2 na figura 1.2) que, é

encaminhada desde a camada de aplicação do nó receptor à camada de aplicação do programa utilizador do nó emissor, seguindo um processo análogo ao descrito antes.

A comunicação entre dois sistemas, considerando cada sistema como uma estrutura em camadas, é um conjunto de comunicações parcelares entre entidades homólogas. Cada camada fornece um serviço à sua camada imediatamente superior, comunicando com a camada semelhante do outro sistema com a qual troca informação (comunicação horizontal). A organização em camadas implica que cada uma seja independente das restantes, havendo assim, uma maior valorização de cada serviço oferecido, tornando o sistema mais flexível e mais fácil de implementar.

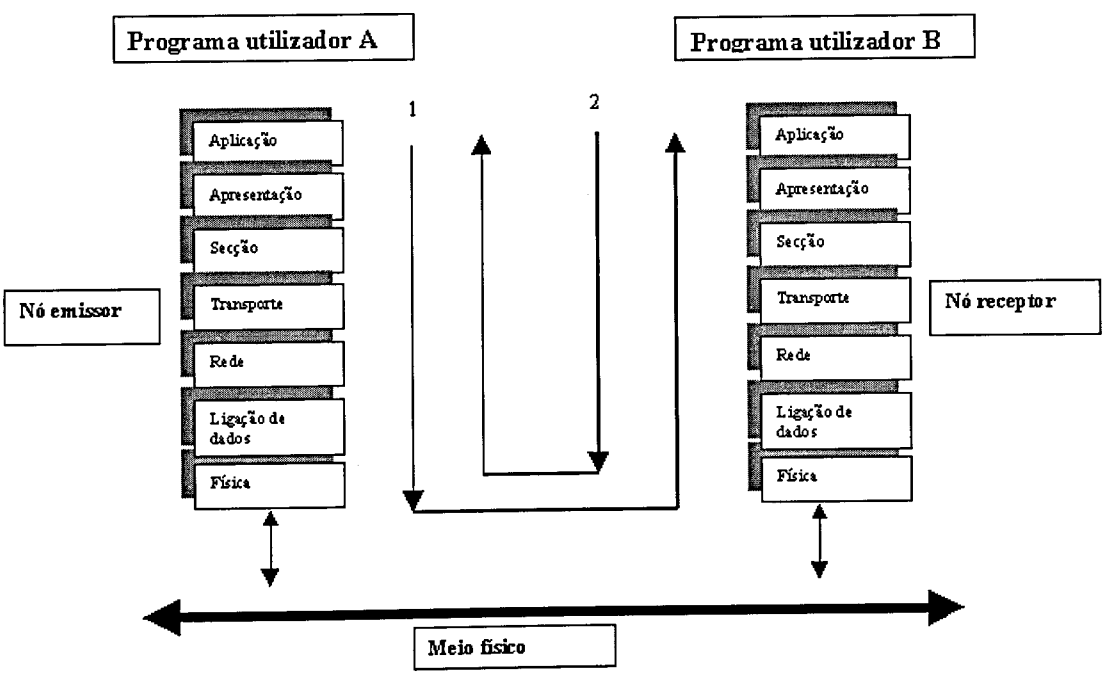


Figura 1.2 - A comunicação no modelo OSI

A seguir, apresentam-se alguns conceitos, nomeadamente o conceito de serviço e protocolo, que são utilizados nas várias camadas do modelo OSI, que permitem a comunicação entre os vários sistemas constituintes da rede [1.4],[1.5].

1.3.1 Serviço e Protocolo

O Serviço é uma função oferecida pela camada (N-1) à camada (N). Os serviços oferecidos pelas camadas OSI são acessíveis através de primitivas de comunicação que se agrupam em quatro tipos distintos, como se pode verificar na figura 1.3:

1. Pedido: pedido do serviço da entidade (N) a um fornecedor de serviços (N-1) para activar um serviço particular.
2. Indicação a indicação é enviada pelo fornecedor (N-1) a um utilizador (N) para lhe assinalar a activação do pedido.
3. Resposta: resposta ao serviço, sendo enviada por um utilizador a seguir a uma indicação.
4. Confirmação: a confirmação é enviada ao utilizador para lhe indicar o sucesso do seu pedido de serviço.

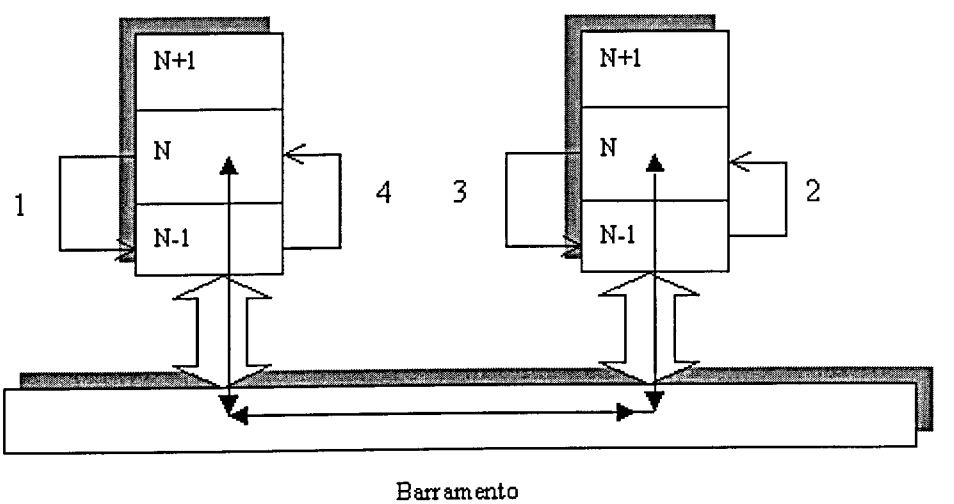


Figura 13 - Primitivas de comunicação no modelo OSI

O protocolo traduz-se num conjunto de regras, semânticas e sintáticas, que determinam as características de comunicação entre as camadas de um mesmo nível pertencentes a entidades OSI distintas.

Em seguida, são apresentados outros conceitos, tais como, o ponto de acesso ao serviço, o endereço, a comunicação com conexão, os diferentes tipos de transmissão, o controle de erros e, por fim, o controlo de fluxo.

1.3.2 Ponto de Acesso ao Serviço (SAP)

O SAP é o ponto onde os serviços (N) são fornecidos por uma entidade (N) a uma entidade (N+1).

1.3.3 Endereço (N)

O endereço (N) é o endereço do ponto de acesso dos serviços (N). Cada entidade deve ser identificada de maneira única para o conjunto do sistema distribuido. Um endereço (N) identifica univocamente um SAP (N) ao qual se liga uma entidade, na separação entre a camada (N) e a camada (N+1).

1.3.4 Comunicação com Conexão

A comunicação com conexão é um dos serviços que a camada (N) oferece à camada (N+1) que consiste na transferência de informação entre camadas (N+1) cooperantes. No caso de se pretender realizar uma transmissão de blocos de dados com conexão entre duas camadas (N+1), começa-se por estabelecer uma conexão de modo a verificar se a outra estação está presente. A comunicação com conexão executa-se em três fases: abertura de conexão, transferência de dados e fecho da conexão.

1.3.5 Transmissão Ponto a Ponto, Multiponto e por Difusão

A transmissão diz-se ponto a ponto quando há apenas duas entidades, uma a transmitir e outra a receber (emissor/receptor).

A transmissão diz-se multiponto quando há comunicação entre mais que duas unidades (1 emissor / vários receptores).

A transmissão por difusão corresponde ao caso em que todos os receptores são informados simultaneamente (o próprio emissor recebe a informação que enviou).

1.3.6 Controlo de Erros e Detecção de Colisão

Os serviços de controle de erros do modelo OSI têm como objectivo detectar se as mensagens emitidas não chegam alteradas. Quando uma entidade (N) envia um pacote de informação a várias entidades (1→N), não há a certeza se este chega correctamente ao seu destino. A mensagem pode chegar modificada ou destruida ao receptor.

Um protocolo de gestão do meio de transmissão tem como objectivo a definição de regras para gerir um conflito. Podem surgir conflitos de acesso ao meio e colisões, sempre que haja mais do que um emissor a enviar pacotes de informação para o meio de transmissão Por exemplo, o controlo centralizado por escrutínio é um dos métodos mais antigos de partilha do suporte de comunicação. Este método consiste em responsabilizar um utilizador de dar a "palavra" a cada um dos outros utilizadores para puderem emitir, consoante uma ordem pré-determinada. É um processo centralizado num dos utilizadores que é definido na inicialização do sistema.

1.3.7 Controlo de Fluxo

O controlo de fluxo é independente da camada e permite informar ao emissor qual a quantidade de informação que o receptor autoriza a emitir, evitando o congestionamento da rede.

1.4 Redes Industriais Locais : RLI

As redes locais foram inicialmente implementadas nos gabinetes, sendo progressivamente aplicadas aos diferentes níveis do sistema industrial.

A fábrica dos nossos dias é cada vez mais caracterizada pelo sucesso da integração do fluxo de informação com o fluxo de materiais [1.6]. Assim os métodos de ajuda à produção por computador (CAD, CAM, CAP. FMS[i]), são integrados num único processo: o CIM ("Computer Integrated Manufacturing"). O CIM é uma arquitectura aberta que tem a capacidade de integrar equipamento e software heterogéneo. Para que haja um fluxo eficiente e um acesso transparente à informação, a comunicação é um factor vital para o sucesso da implementação do CIM num sistema industrial. Dada a complexidade de aplicações que o CIM pode comportar, surgiram os modelos de arquitecturas funcionais hierarquizados que decompõem em diferentes níveis as diferentes aplicações [1.7]. O modelo NBS surgiu para uma aplicação industrial colocando em evidência cinco níveis que se apresentam na figura 1.4.

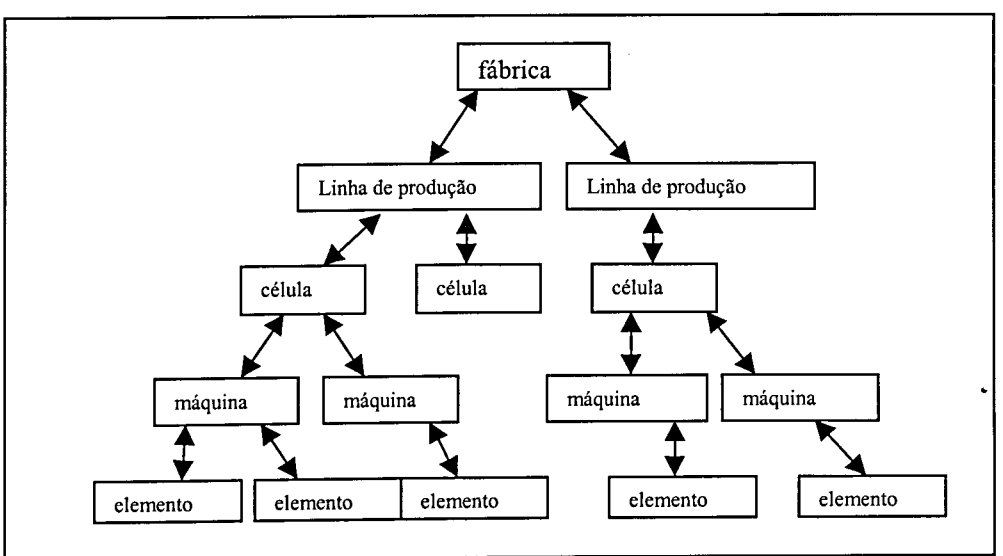


Figura 1.4 - A fábrica agrupa um conjunto de linhas de produção, de serviços de gestão, de serviços comerciais, de serviços técnicos e outros serviços comuns a um sistema de produção. A linha de produção agrupa um conjunto de células e responde a um pedido de mudança de gama de produção. As máquinas que permitem a execução parcial ou completa de uma etapa de um processo estão agrupadas em células.

[i] CAD: Computer Aided Design; CAP:Computer Aided Production; FMS: Flexible Manufacturing System.

Pode-se dizer que, em cada nível da arquitectura CIM [1.6], existe uma rede local industrial (RLI). Uma RLI suporta diferentes tipos de informação consoante o nível em que se encontra, desde a monitorização da produção até ao controlo de estados de variáveis, de programas, de parâmetros e de alarmes.

Actualmente, a questão que os utilizadores de uma rede colocam já não é "utiliza-se ou não um RLI" mas sim "como tirar o melhor proveito de uma RLI". O número de aplicações destes sistemas foram aumentando gradualmente, dada a possibilidade de alterar as suas configurações e de permitirem o diálogo entre equipamentos com a mesma origem ou de origem diferente, com um número variável de componentes e de funções.

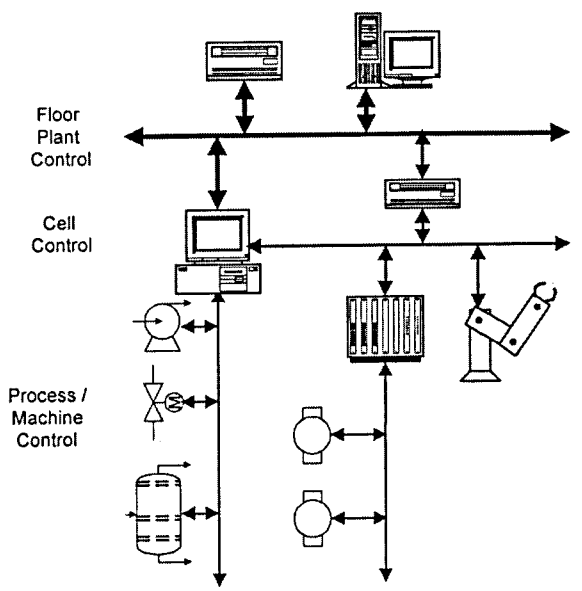


Figura 1.5 - A arquitectura CIM com três niveis.

Se considerarmos uma arquitectura CIM a três níveis, como se ilustra na figura 1.5, a frequência, o tipo e a estrutura de informação transmitida e as exigências temporais vão ser diferentes consoante o nível que pertence a RLI [1.8]. Pode dizer-se, de uma forma sucinta, que ao nível da linha de produção ("Shop Floor Control") teremos a monitorização e a gestão de produção, não sendo a muito relevantes a informação estruturada e as exigências temporais. A este nível a rede é constituída por PC's com diferentes tipos de software para a gestão da produção, da qualidade, e de controlo de stocks.

Ao nível de uma célula, surge o controlo das máquinas. Estas máquinas podem estar equipadas com autómatos, com Comando Numérico, robots, podendo o controlo ser executado por PC's. A este tipo de rede chama-se *barramento de campo.* Como exemplos destas redes enumera-se o WorldFIP, o Profibus/FMS e o Mini-MAP [1.8].

Ao nível da máquina, temos uma maior exigência de resposta temporal (crítica) e informação mais simples como, por exemplo, a posição de um eixo numa máquina de comando numérico. A este nível da máquina, a rede é composta por um autómato (controlador) a gerir um conjunto de sensores e de actuadores (ver figura 1.5). Como exemplos destas redes temos o CAN ("Controller Area Network") e o Device WorldFIP (WorldFIP reduzido) [1.8].

No que respeita a restrições temporais, as redes têm requisitos diferentes consoante o nível onde se inserem. Por exemplo, ao nível da linha de produção utilizam-se redes com taxas de transmissão elevadas por necessidade de transferência, de ficheiros, por exemplo. Não existem, no entanto, restrições que obriguem estas redes a possuir características de tempo real, o que ao nível da máquina tal é imperioso. Há portanto incapacidades que fazem com que as redes dos vários níveis não consigam cumprir restrições dos níveis que não são os seus.

1.5 As Funções Base de um Barramento de Campo.

Um barramento de campo permite a comunicação entre terminais inteligentes, que podem ser PC's, PLC's, robots, transdutores de pressão e sensores de velocidade. Os serviços oferecidos por um barramento de campo têm, no mínimo, de ser idênticos aos fornecidos por um sistema clássico de entradas/saídas, isto é, operações de comando de dispositivos, de pedido de um determinado estado, de entrada e saída de valores. Além disso, o barramento de campo tem que fazer a gestão das funções implementadas nos diferentes terminais, garantindo um bom desempenho da rede e uma comunicação isenta de erros [1.9],[1.10]. Esta troca de fluxo de informação deve ser controlada e gerida de modo a que não haja problemas de conflito no acesso ao meio e para que se garanta a chegada correcta da informação ao seu destino. O funcionamento do barramento deve ser transparente para o utilizador, isto é, o utilizador não tem que se preocupar com o funcionamento do barramento mas sim tem que ter a garantia de que tem respostas claras e precisas às suas solicitações. Esta gestão de informação, que define o funcionamento da rede, é realizada através de protocolos.

Um barramento de campo pode ser considerado um sistema distribuído em tempo real dado que suporta a comunicação entre terminais inteligentes e autónomos que processam e memorizam a informação, estando distribuidos e separados fisicamente [1.11]. Além disso, esta comunicação é efectuada em tempo real, isto é, opera com restrições temporais, uma vez que a informação tem um período de validade limitado que tem de ser respeitado. Todo o processamento e transferência de informação terá de estar limitado a uma janela temporal para que se possa fazer uma gestão coerente de toda a informação distribuída pelos diferentes terminais [1.8]. Por sua vez, o tempo de resposta dos diferentes terminais também tem que ser limitado, para que o sistema possa dar uma resposta num determinado tempo preciso. Num sistema deste tipo (sistema em tempo real), não só é importante o resultado final mas também saber em que instante no tempo é dado esse resultado [1.11].

1.6 As Vantagens do Barramento de Campo

As vantagens de um barramento de campo são as inerentes às de uma rede local industrial. Antes do aparecimento do produto rede local, o desenvolvimento dos autómatos programáveis e de comandos numéricos já permitia a elaboração de arquitecturas de comando distribuído. No entanto, os utilizadores aperceberam-se que as redes de ligação ponto a ponto, apresentavam diversos inconvenientes tanto a nível da cablagem como da manutenção. Além de ter custos elevados, o sistema torna-se pouco flexível para permitir qualquer alteração ou inserção de um novo elemento. Em termos de manutenção, por exemplo, a ligação ponto a ponto torna-se bastante morosa e consequentemente dispendiosa, enquanto que numa utilização com uma rede, o diagnóstico de uma avaria é mais fácil assim como a respectiva substituição ou reparação de um terminal[1.12].

Além disso, o barramento de campo permite um aumento da descentralização e distribuição das funções implementadas no sistema. Permite ainda a utilização de sensores inteligentes que, tendo incorporado um microprocessador com capacidade de memória e de cálculo, lhes permite executar funções de auto-diagnóstico, controlo e de manutenção [1.13].

É do senso comum que, se um sistema[ii] estiver dependente de um só recurso para dar resposta às diferentes solicitações (sistema centralizado), tem um tempo de resposta maior e, se o recurso avariar, o sistema deixa de funcionar. Num sistema descentralizado, as diferentes funções são implementadas pelos diferentes terminais, sendo cada um responsável por uma função. Neste caso, terá que se garantir que a informação fornecida a um dos terminais é também fornecida aos restantes e que o acesso ao meio é gerido por forma a evitar colisões.

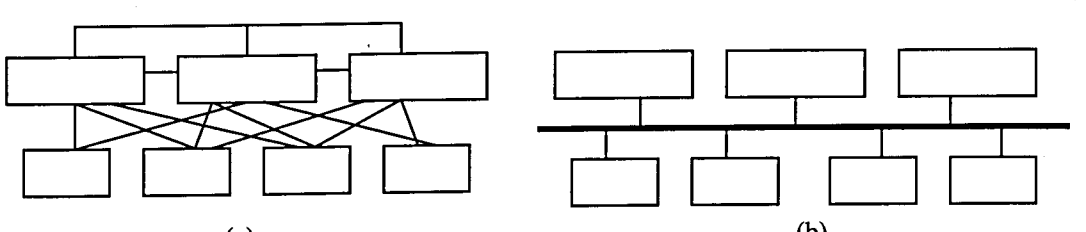


Figura 1.6 - Cablagem clássica ponto a ponto (a) *versus* cablagem RLI utilizando um barramento (b).

[ii] Seja um sistema de qualquer tipo :admnistrativo, informático.

A aplicação de um barramento de campo deve ser justificada em termos económicos e, em cada caso, deve fazer-se um estudo de implementação para verificar se há vantagens na aplicação de um barramento de campo [1.14].

Resumindo, as vantagens de utilização de um barramento de campo são as seguintes:

- a redução do custo de instalação inicial.
- a simplificação do sistema, implicando uma redução no tempo de manutenção necessário para detectar uma avaria e para a reparar e
- um aumento no desempenho do sistema dado que os diferentes terminais podem comunicar entre eles, sem que a informação tenha que passar forçosamente pelo controlador.

Capítulo 2 : O Barramento de Campo WorldFIP

2.1 Introdução

Este capítulo tem como objectivo fazer a apresentação do funcionamento do protocolo WorldFIP ("Factory Implementation Protocol"). Como já foi referido no capítulo 1, o WorldFIP é um barramento de campo que assegura a comunicação entre o nível um de um sistema de automação (PLC's e controladores) e o nível zero (sensores e actuadores).

2.2 Conceitos Básicos Inerentes ao WorldFip

Num barramento WorldFIP os diferentes tipos de equipamento interligados têm funções diferentes tais como a supervisão, o controlo e o comando da célula (execução de programas) e a aquisição de dados (analógicos ou digitais). A figura seguinte (figura 2.1) ilustra as diferentes funções suportadas pelo WorldFIP e por outros tipos de barramentos de campo.

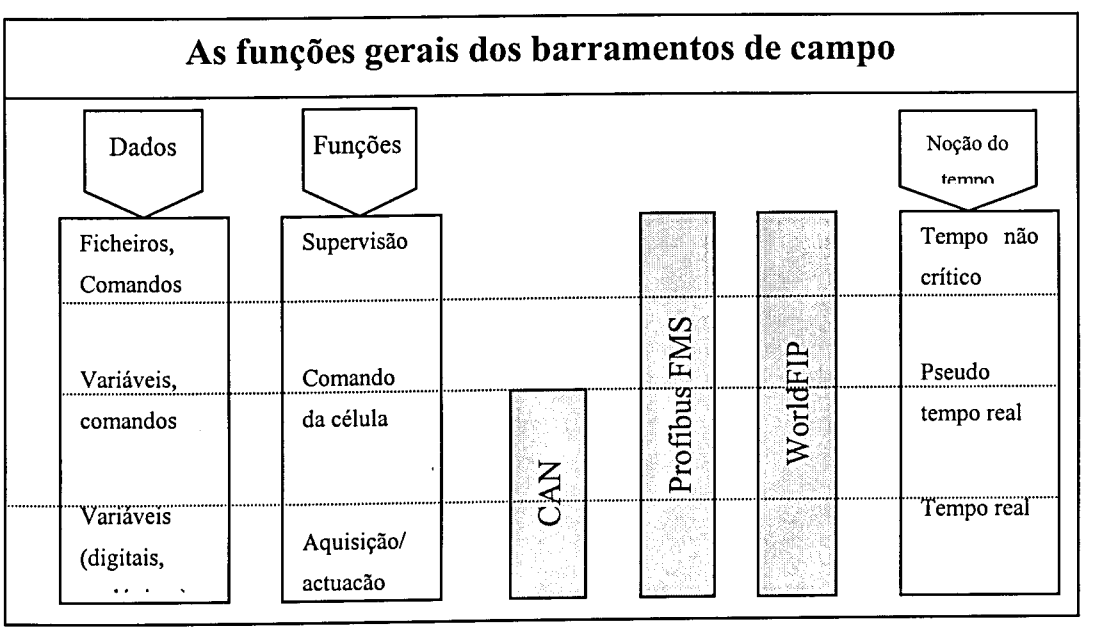


Figura 2.1 - As diferentes funções do WorldFIP[2.1].

O tamanho do pacote de informação necessário para a comunicação assim como o tempo de resposta por forma a garantir um sistema em funcionamento com respostas em tempo real é diferente conforme o tipo de função de cada equipamento.

2.2.1 Exigências temporais e tamanho do pacote de informação

O WorldFIP suporta o conceito de sistema aberto, tendo a capacidade de satisfazer as exigências de tempo real de um sistema de produção distribuído. A informação que circula num sistema destes tem um tempo de validade limitado, isto é, tem que ser transmitida ou processada durante um espaço ou janela temporal. Dependendo do tipo de aplicação, esta janela é maior ou menor.

A informação pode ser classificada como *crítica* ou *não crítica*, e a informação crítica pode ainda ser distinguida em *soft* ou *hard*. No primeiro caso toleram-se alguns atrasos temporais para além da janela temporal, sem que, no entanto, haja prejuízo para o processo produtivo. No segundo caso, não se tolera qualquer violação à janela temporal e um atraso na aquisição ou processamento de uma informação pode trazer consequências desastrosas para o processo produtivo. Como exemplo ilustrativo desta situação, o sensor que detecta a saturação numa linha de produção automatizada, tem de ter um tempo de resposta específico, habitualmente muito curto e, se o ultrapassar, pode danificar o braço de descarregamento da máquina anterior.

Outro aspecto temporal respeitante à informação de um sistema de produção é a periocidade ou frequência com que é produzida e processada. Assim, existe informação que é produzida periódicamente, em instantes de tempo bem conhecidos, como, por exemplo, a informação que é fornecida por um sensor ou transmitida a um actuador (associada a malhas de controlo, por exemplo). Por outro lado, existem sempre situações de emergência que têm de ser contempladas e que aparecem de uma forma aleatória (não periódica). Como, por exemplo, o sinal que é enviado ao controlador de uma máquina CNC quando um variador de frequência em defeito, é um sinal considerado crítico e não periódico. Num sistema de automação, a informação pode ser em tempo real ou não[2.2].

O tamanho do pacote de informação transmitida depende da função do equipamento. No barramento de campo WorldFIP, ao nível dos sensores/actuadores, as trocas de informação podem ser realizadas através de um número reduzido de bits ou, ao nível superior, por transferência de ficheiros que requerem um pacote de informação constituído por várias palavras.

É interessante salientar que, à medida que o tamanho da informação vai aumentando, a janela temporal associada vai sendo maior e menos exigente em relação às restrições temporais. É ao nível dos sensores/actuadores que o tempo de resposta é considerado mais crítico, variando entre 1 a 10ms. A este nível a informação é constituída por bits de E/S (entradas/saídas) ao contrário do que se passa ao nível superior, onde as funções requerem um grande número de pacotes de informação mas o tempo de resposta, embora importante [2.2], não constitui fonte de especial preocupação.

2.2.2 Controlo de acesso ao meio

O protocolo WorldFIP suporta todos estes tipos de informação e tem a capacidade de garantir/respeitar as diferentes restrições temporais, entendendo-se por restrições temporais a informação que deve ser enviada e recebida dentro de limites temporais. Para isso, este protocolo utiliza o controle de acesso ao meio (MAC), centralizado numa das estações, que é responsável por fazer um escalonamento do tráfego de informação e por estabelecer a ordem pela qual as restantes estações vão ter acesso ao meio físico para poderem comunicar umas com as outras. Deste modo, o protocolo WorldFip faz a interligação entre estações que podem ter dois tipos de funções: as que fazem de árbitro de barramento (AB), sendo a estação AB a que controla o acesso ao meio físico, e as que são produtoras ou consumidoras de informação. Qualquer estação pode fazer de AB mas, de cada vez, só pode haver uma a executar as funções de árbitro de barramento.

2.2.3 Modelo Produtor/Distribuidor/Consumidor

No barramento de campo WorldFIP, a troca de variáveis entre as diferentes estações, é baseada no modelo **Produtor/ Distribuidor/ Consumidor** que se apresenta na figura 2.2.

Neste modelo, cada variável só tem uma unidade produtora, podendo haver várias unidades consumidores dessa variável (necessitam do valor dessa variável para poderem executar uma acção). Como exemplo, um sensor produz uma variável e o seu valor será utilizado pelas estações interessadas. Cada variável tem um único identificador associado.

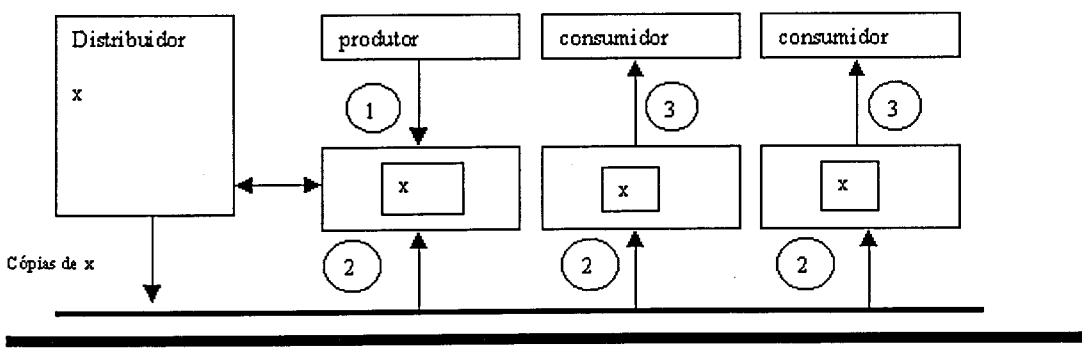


Figura 2.2 - Modelo Produtor/Distribuidor/Consumidor (PDC)

Tal como se esquematizou na figura 2.2, o produtor produz localmente o valor de uma variável(1), o distribuidor encarrega-se de distribuir esse valor a todos as estações(2) e, por sua vez, as que são consumidoras dessa variável, copiam esse valor (3) [2.3].

Este modelo, por um lado, dissocia a produção das variáveis da sua tranferência e da sua utilização por parte da estação consumidora e garante que todas as estações recebam ao mesmo tempo os valores actualizados das variáveis e, por outro lado, com o processo de escalonamento de transferência das variáveis, consegue cumprir algumas das restrições temporais [2.4].

No WorldFIP, o distribuidor não endereça as diferentes estações para transmitirem mas sim dá a palavra às diferentes entidades para comunicarem através dos identificadores das variáveis (no caso da figura 2.2 será a variável x). O WorldFIP

não identifica a estação que produz determinada variável, mas sabe quais são as variáveis que têm de ser emitidas e qual é o escalonamento entre elas.

2.3 Sistema de Comunicação WorldFIP

Num sistema de produção automatizado, cada equipamento assegura as funções do automatismo (processos de aplicação) definidas pelo utilizador e as funções de comunicação que permitem a comunicação com os outros equipamentos. Um processo de aplicação manipula os dados que podem ser produzidos localmente ou à distância pelos outros processos de aplicação [2.4]. O sistema de comunicação permite aos diferentes equipamentos trocarem informação através do barramento.

O sistema de comunicação WorldFIP está organizado em três camadas de acordo com o modelo OSI: a camada de aplicação, a camada de ligação de dados e a camada física ilustradas na figura 2.3.

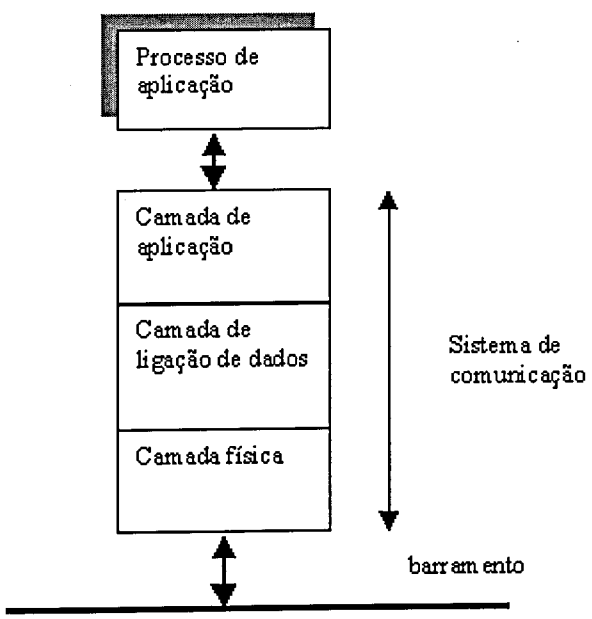


Figura 2.3 - Sistema de comunicação do WorldFIP[2.4]

A seguir apresenta-se, de uma forma resumida, os serviços fornecidos por cada uma das três camadas.

2.1.1 A camada de aplicação

A camada de aplicação oferece dois tipos de serviços a saber : os associados a variáveis simples e os associados a listas de variáveis. As variáveis produzidas ou consumidas pelo utilizador podem ser de dois tipos: simples ou estruturadas. As variáveis simples podem ser do tipo booleano e inteiro e as do tipo estruturada são estruturas simples, tais como listas ou tabelas.

Serviços associados a variáveis simples

A camada de aplicação oferece ao processo de aplicação os seguintes serviços :

- Leitura/escrita local
- Leitura/escrita à distância
- Indicações de transmissão/recepção de variáveis
- Informações de validade das variáveis
- Serviços de sincronização

Leitura/escrita local

Cada processo de aplicação possui uma imagem local das variáveis do sistema que permitem a leitura e escrita de valores locais. Neste processo de leitura e escrita local não há qualquer troca de informação entre as várias entidades ligadas ao barramento, não gerando tráfego no barramento. Utilizando o serviço **L_GET.req** lê localmente no registo da camada de ligação e com o serviço **L_PUT.req** escreve localmente do registo da camada de ligação. Este processo de leitura e escrita está descrito na figura 2.4.

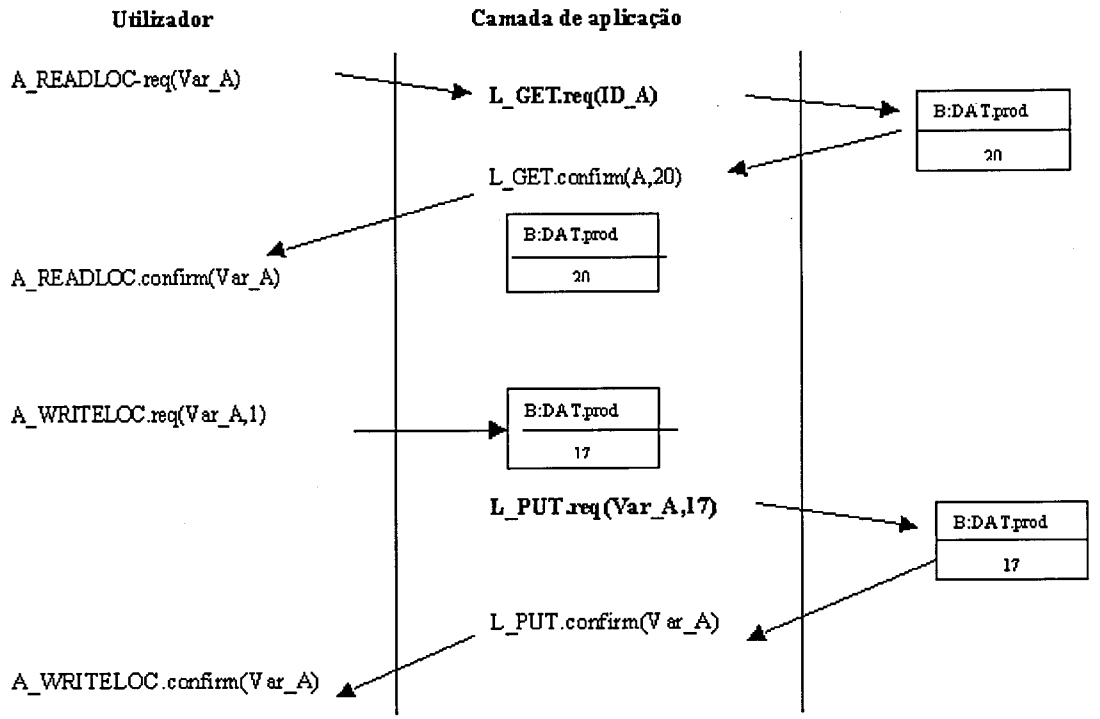


Figura 2.4 - Processo de leitura e escrita local

Leitura/escrita à distância

Uma leitura à distância consiste em pedir a tranferência do valor actual da variável em causa ao produtor seguido de uma leitura local. Se um utilizador necessitar de conhecer o valor de uma variável produzida por uma outra entidadade (produtor) então efectua um pedido de leitura à distância e, neste caso o distribuidor terá de fazer um pedido de actualização a todas os utilizadores do barramento. Quem for produtor dessa variável transfere o respectivo valor para o barramento, o qual será recebido por todos.

Indicações de transmissão/recepção de variáveis

As primitivas L_SENT.indication(Var_A) e L_RECEIVED.indication(Var_A) informam os utilizadores consumidores da recepção do valor de uma variável consumida ou os utilizadores produtores da emissão do valor de uma variável produzida como se ilustra na figura 2.5.

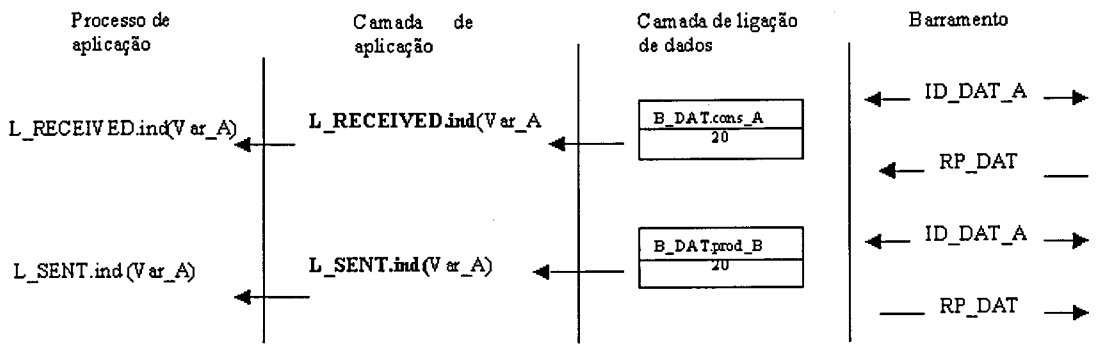


Figura 2.5 - Primitivas de indicação L_SENT.ind e L_RECEIVED.ind

Informações de validade das variáveis

No modelo Produtor/Distribuidor/Consumidor existem dois intervalos de tempo que convém realçar: o intervalo de tempo entre a produção de uma variável e a sua transferência e o intervalo entre a transferência e o seu consumo dessa variável. Pode acontecer, por diversas razões, que uma variável produzida não seja transferida ou uma entidade consumidora não tenha tempo de consumir todas as variáveis recebidas. As informações booleanas associadas às variáveis trocadas informam o utilizador da validade da informação produzida ou consumida e denominam-se por disponibilidade e refrescamento [2.4]. O estado de disponibilidade é elaborado pela camada de aplicação das entidades utilizadoras. O valor verdadeiro indica a um utilizador consumidor que a variável foi transferida pela rede num tempo inferior ao período de transmissão ou que o árbitro do barramento respeitou o periodo de "interrogação" da variável. Além disso, indica que o mecanismo de transferência de registo está a trabalhar correctamente. O estado de refrescamento é construído pelas entidades produtoras e indica às entidades consumidoras que a variável foi produzida num tempo inferior ao período de produção.

Serviços de sincronização

Os processos que compõem uma aplicação podem ser síncronos ou assíncronos. Um processo síncrono está directamente ligado à rede, isto é, a sua

execução depende de uma indicação vinda da rede. Um processo assíncrono surge independentemente dos processos em execução na rede.

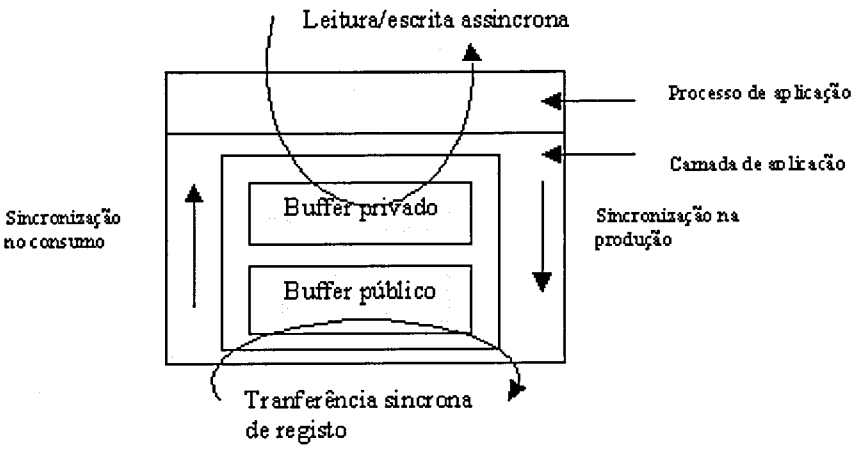


Figura 2.6 - Serviço de sincronização no WorldFIP[2.4]

Os serviços de sincronização permitem que os processos assíncronos possam participar numa aplicação distribuída sincronizada. Ao nível da camada de aplicação, o mecanismo de sincronização associa a cada variável um segundo registo. Assim, cada variável tem um registo privado que só é acessível pelo processo de aplicação e um registo público que só é utilizado pela rede. Como se mostra na figura 2.6, perante um processo de consumo, o conteúdo do registo público é copiado para o registo privado aquando da recepção de uma variável síncrona. Num processo de produção, o processo de sincronização consiste em copiar o conteúdo do registo privado para o registo público aquando da recepção de uma variável síncrona.

Serviços associados a listas de variáveis

A camada de aplicação oferece um serviço de leitura de listas de variáveis simples que, podem ser um valor de uma pressão, de um débito ou o estado de um sensor, disponíveis para um utilizador que, pode ser um autómato ou um posto de operador. Para os utilizadores da lista de variáveis também existem informações de validade bem como informações da coerência de produção e de coerência espacial [2.4] que se apresentam em seguida.

Coerência de produção (coerência temporal):

A coerência de produção é uma informação booleana elaborada pela camada de aplicação, informando o processo de aplicação utilizador que os produtores dos diferentes valores das variáveis da lista respeitaram o período de produção.

Coerência de transmissão (coerência temporal):

A coerência de transmissão é uma informação booleana elaborada pela camada de aplicação, informando o processo de aplicação utilizador que a rede respeitou os períodos de distribuição de cada variável que compõe a lista. É obtido através de um "e" lógico dos estados de disponibilidade de cada variável.

Coerência espacial:

A coerência espacial, tendo o valor "verdadeiro", indica que todas as instâncias de uma lista consumidora têm o mesmo valor. Indica a cada utilizador da lista se as diferentes cópias são iguais ou não. Cada entidade consumidora de uma variavel elabora uma variável de coerência. O estado de corência espacial é um "e" lógico de todos os valores das variáveis de coerência individuais.

2.3.3 A camada de ligação de dados

A camada de ligação de dados oferece dois tipos de serviços de transmissão à camada de aplicação: o serviço de troca de variáveis e o serviço de transferência de mensagens. Estes dois serviços são executados de uma forma cíclica, sendo os nomes das variáveis e os períodos fixados na altura da configuração da rede, ou de uma forma não cíclica (aperiódica) em que a troca de informação é efectuada a pedido do utilizador [2.6].

Endereçamento

O WorldFIP baseia-se num processo de difusão para a troca de variáveis. Todas as variáveis e os respectivos valores são colocados à disposição de todos os utilizadores da rede e cada entidade reconhece-se como sendo consumidora ou produtora dessa variável. Os identificadores são codificados usando 16 bits e, pelo menos teóricamente, podem ser identificadas 65536 variáveis. O mecanismo de endereçamento permite identificar as variáveis e as respectivas trocas, independentemente da localização física das entidades produtoras e consumidoras. As informações transmitidas não precisam de um endereço de destino.

Interface entre a Camada de Aplicação e a Camada física

A camada de ligação providencia serviços para servir a camada de aplicação, utilizando os serviços da camada física. É constituida por um conjunto de registos de produção e de consumo e, cada um destes registos contém o último valor actualizado pelo utilizador da rede. Um novo valor introduzido num registo de consumo ou de produção sobrepõe-se ao que lá foi colocado anteriormente.

O número de registos de cada estação é decidido na altura da configuração da rede, consoante o número de variáveis. A camada de aplicação escreve e lê os valores desses registos através das primitivas L_PUT.req(Var_A,20) e L_GET.req(VAR_A). Por outro lado, estes registos também podem ser acedidos pelo barramento, ao que se chama transferência de registos, como se mostra na figura seguinte.

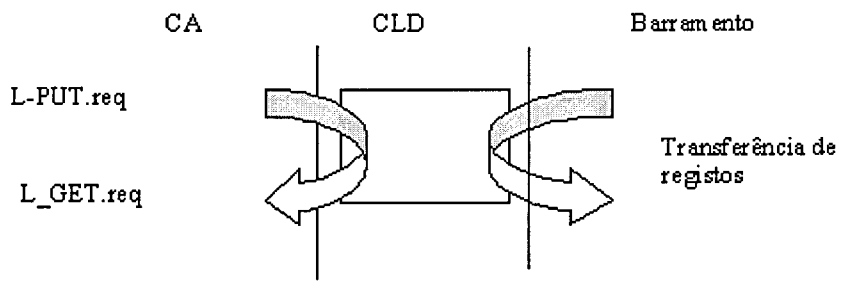


Figura 2.7 - A camada de ligação de dados (CLD) como interface entre a camada de aplicação (CA) e o barramento [2.6]

Assim, o árbitro do barramento para iniciar uma transferência de buffer começa por

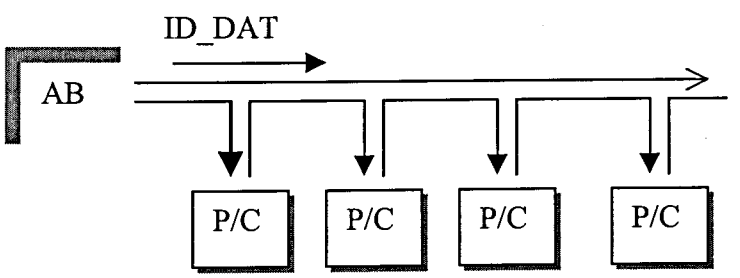
- transmitir um pacote de informação "ID_DAT" com o respectivo identificador. Este pacote é recebido por todos as estações da rede.

- a estação que se declara como produtora dessa variável, transmite um outro pacote de informação "RP_DAT" com o valor dessa variável. A camada de ligação de dados envia um "L_SENT.indicação" à camada de aplicação.

- as estações que se declaram como consumidoras dessa variável, actualizam os seus registos e enviam um "L_RECEIVED.indicação" à respectiva camada de aplicação (ver figura 2.5).

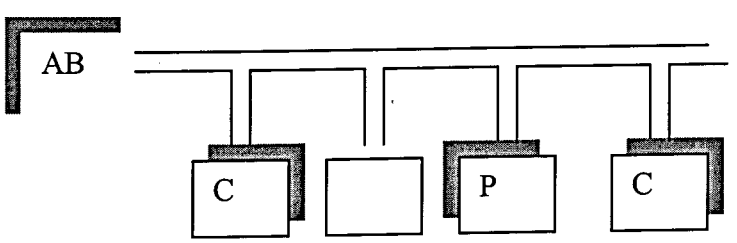
Técnica de acesso ao meio

No WorldFIP as entidades podem ter duas funções diferentes: a função de produtora/consumidora ou a de árbitro do barramento (AB). A entidadade AB tem como função gerir o acesso ao barramento, isto é, controlar o direito de acesso ao meio de cada produtor e consumidor de informação. A partir das variáveis que vão ser geradas na rede, o árbitro do barramento constrói uma tabela (tabela de escrutação) que define as necessidades de comunicação dos diferentes equipamentos em cada instante. Lendo esta tabela, o AB difunde os identificadores sequencialmente e, a unidade que se reconhece como produtora responde, difundindo o valor associado a essa variável e as unidades que se reconhecem como consumidores recebem-na. Ao receber a resposta, o AB passa ao identificador seguinte. Esta sequência é mostrada na figura 2.8.

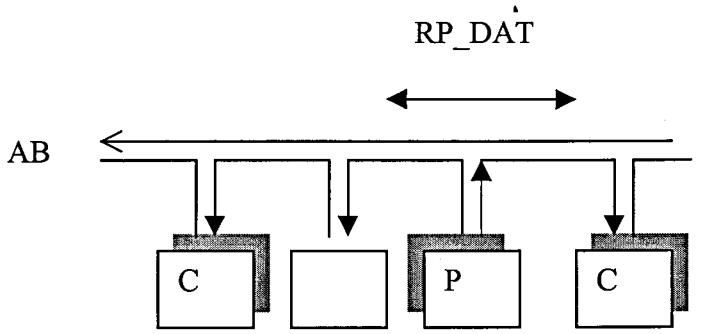


Etapa 1: AB difunde identificador

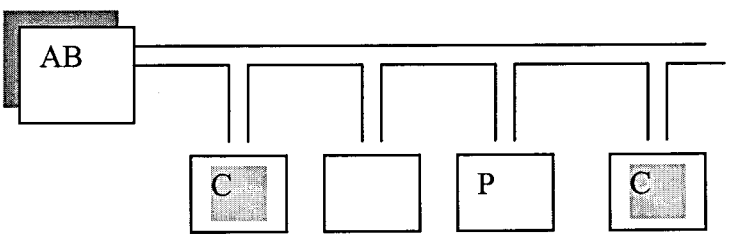
Em cada instante, só pode haver uma entidade a funcionar como árbitro do barramento activo.



Etapa 2: Reconhecimento pela entidade produtora e pelas entidades consumidoras.



Etapa 3: difusão do valor pela unidade produtora



Etapa 4: Recepção do valor pelos equipamentos interessados

Figura 2.8 - Técnica de acesso ao meio utilizada pelo AB

Após a etapa 4, o árbitro do barramento, passa ao identificador seguinte, consultando a tabela e o ciclo de pedido-resposta volta-se a repetir.

A tabela do árbitro do barramento

O árbitro do barramento é o gestor da rede e tem que ser informado quais são as variáveis que serão escalonadas e o respectivo tempo de escalonamento, assim como do tamanho do pacote de informação de cada variável. Esta informação está disponível numa tabela que é construída na altura da configuração da rede que, tem um aspecto idêntico à que se apresenta em seguida.

Variável	Periodo	Tamanho do pacote de informação(bits)	Pedido a que está associado
A	5	32	Especifico
B	10	64	Livre normal
C	15	32	
D	20	32	
E	20	48	Especifico aperiódico
F	30	32	

Tabela 1.1 - Exemplo de uma tabela de escrutação do AB

Por exemplo, utilizando a tabela 1.1, a variável B tem um tempo de escrutínio de 5ms e está associada ao pedido livre normal tendo um pacote de informação de 64 bits.

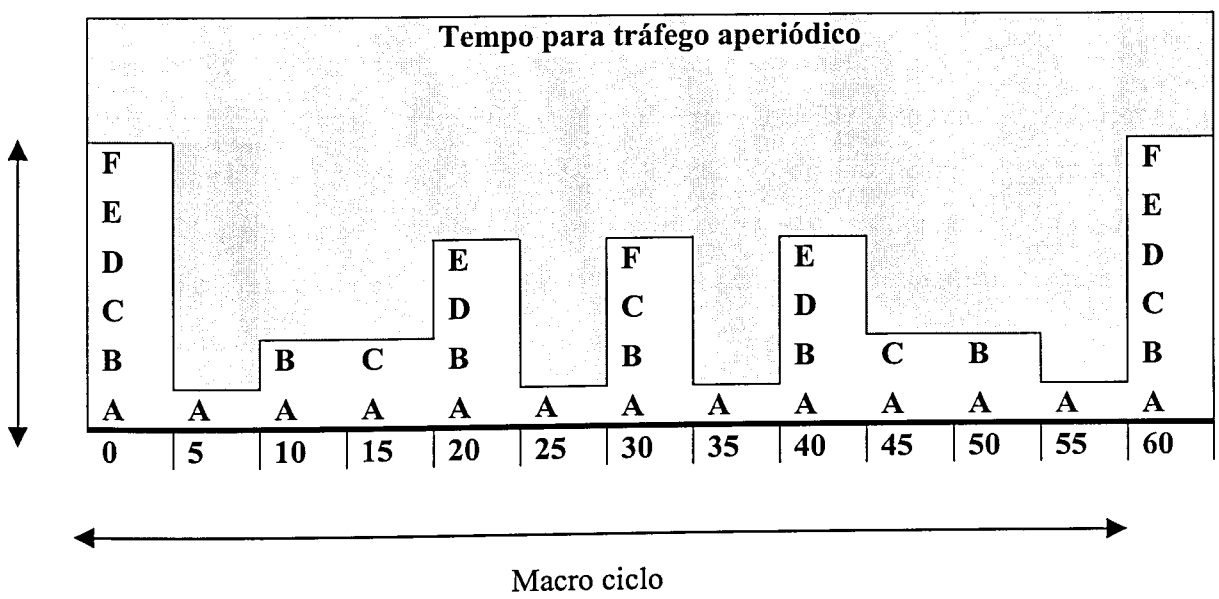


Figura 2.8 - tabela de escrutínio construida pelo AB[2.6]

O tempo de transmissão é partido em intervalos que se designam por ciclos elementares (EC). Os períodos das variáveis são sempre considerados múltiplos inteiros de duração do EC. Um macro-ciclo corresponde a um grupo de ciclos elementares, a partir do qual, os ciclos elementares começam a repetir-se. O número de ciclos elementares que corresponde a um macro ciclo é igual ao mínimo multiplo

comum (mmc) dos tempos de escalonamento a dividir pelo máximo divisor comum dos mesmos tempos (mdc). A duração de um ciclo elementar é dado pelo mdc dos tempos de escalonamento. Assim, neste exemplo, o "mmc" de (5,10,15,20,20,30) é igual a T_{mc}=60ms e o "mdc" de (5,10,15,20,20,30) é 5 sendo o nº de ciclos elementares igual a 12 e a duração máxima de um ciclo elementar de T_{ce}=5ms. Inicialmente, no primeiro ciclo elementar, o árbitro faz o escalonamento a todas as variáveis, no segundo ciclo, já faz o escalonamento só à variável A e assim sucessivamente, de acordo com os tempos de escalonamento. No entanto, se houver no mesmo ciclo elementar um número elevado de variáveis que ponham em risco a capacidade da linha (capacidade da rede é igual T_{ce} * débito da rede[1]), essas variáveis podem ser distribuidas de uma forma uniforme, pelos ciclos elementares vizinhos.

Pedidos de transferência de variáveis

Uma unidade pode necessitar de informação que não é contemplada na tabela de escrutação, existindo sempre a possibilidade de haver trocas de informação ocasionais. Utiliza então, os serviços de pedidos de tranferência, que lhe permite a tarnsferência de um conjunto de variáveis. O pedido é feito, de uma forma aleatória, pela camada de aplicação à camada de ligação de dados, ficando à espera de ser atendido na respectiva camada de ligação de dados. A estação quando solicitada pelo AB no ciclo periódico, verifica se houve ou não pedido e no caso afirmativo, responde ao AB, indicando que tem um pedido para ser atendido.

Os pedidos anteriores podem ser de dois tipos diferentes: pedidos explícitos e pedidos livres. Os pedidos explícitos permitem à camada de aplicação solicitar a difusão de uma ou mais variáveis, indicando a variável que permite a transmissão do pedido ao AB. Os pedidos livres são transmitidos ao AB através da primeira variável que chega e que tenha sido configurada para este tipo de pedido (o pedido não indica qual a variável, contrariamente ao que acontece com o pedido explicito). O WorldFIP permite ainda, pedidos explicitos periódicos ou aperiódicos e pedidos livres urgentes ou normais. Os pedidos explicitos periódicos são atendidos durante o escrutínio periódico (são os mais prioritários) enquanto os restantes pedidos são atendidos durante o escrutínio aperiódico.Tanto o árbitro do barramento como a camada de

[1] Débito da rede é igual ao nº máximo de bits transferidos num ciclo elementar

ligação de dados têm que ter os registos necessários para atender e executar estes pedidos, tanto periódicos ou aperiódicos, variando o processo de execução com o tipo de pedido.

2.3.4 A camada física

A camada física assegura a transferência de bits de informação entre os diferentes utilizadores da rede. O meio de transmissão pode ser fibra óptica ou cabo de cobre blindado [2.6].

Velocidade de transmissão

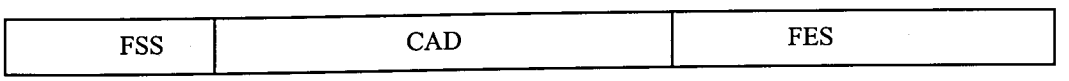
Para o suporte de transmissão do tipo cabo cobre blindado ("pair torsadée blindée"), a norma NF C46-604 especifica duas velocidades de transmissão :S1: para barramento "baixa velocidade": 50 Kb/s. e S2: barramento "alta velocidade": 1Mb/s. Um meio de transmissão de fibra óptica permite uma velocidade de transmissão da ordem dos 5Mb/s (norma C46-607) [2.6].

Codificação

A camada física codifica a informação transmitida à camada de ligação de dados utilizando o código de Manchester [2.7]. Este código permite transmitir simultaneamente sinais de sincronização temporal e informação.

Pacote de informação do WorldFIP

Qualquer pacote de informação é constituido por três campos: o de inicio de sequência (FSS), o campo de informação e de controlo (CAD) e, o campo de fim de sequência (FES)



O campo FSS é constituido pelo campo PRE("preamble") e pelo campo FSD("frame start delimiteur"). O campo PRE serve para a sincronização de relógios de recepção. O campo FSD indica à camada de ligação de dados o inicío da informação útil (CAD).O campo CAD("control and data") contém a informação lógica da camada de ligação de dados. O campo FES("frame end delimiteur") indica à camada de ligação o fim da informação útil.

2.4 A Gestão da Rede

No barramento de campo WorldFIP, a gestão da rede tem diferentes objectivos durante o ciclo de vida da rede. No arranque da implementação da rede, permite criar a configuração inicial que inclui a definição de parâmetros, identificadores, número de estações. Seguidamente, permite um conjunto de testes que verificam a funcionalidade da rede, bem como, testes às novas estações que vão sendo inseridas. Depois de implementada a rede, tem que se fazer a sua manutenção de modo a colocar fora de serviço estações que não estejam a funcionar convenientemente, repará-las e ligá-las novamente á rede sem, no entanto, provocar perturbações ao normal funcionamento da rede.

2.5 Componentes do FIP

Algumas empresas desenvolveram componentes específicos que suportam totalmente ou em parte, o protocolo do WorldFIP. Estes componentes incluem o FIPART, o FIPIU e , o FULLFIP (FIPLD1, FIPTR para o cabo blindado)

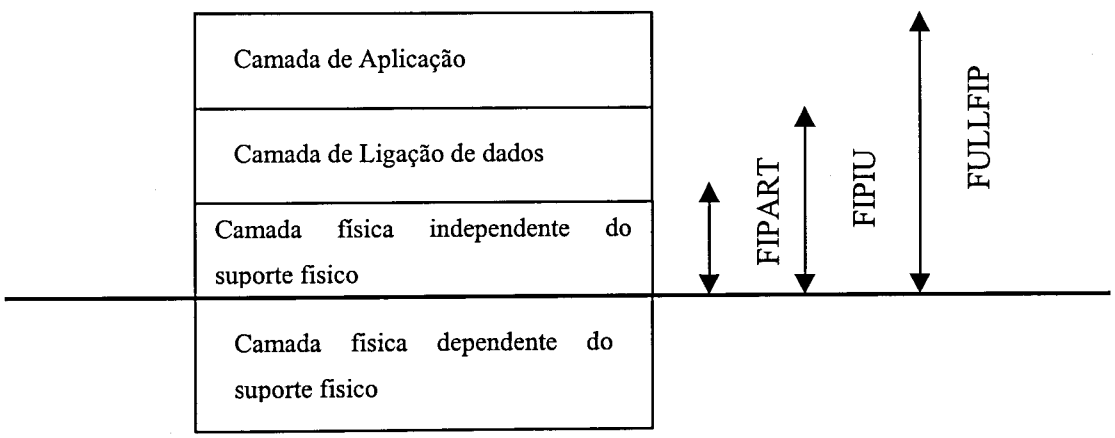


Figura 2.10 - Configuração dos componentes FIPART, FIPIU e FULLFIP

Os três componentes utilizam a mesma camada física, que é independente do suporte fisíco de comunicação. O **FIPART** inclui os serviços da camada de ligação de dados que são independentes do suporte utilizado (código Manchester, detecção de erros) e alguns serviços da camada de aplicação. O **FIPIU** inclui os serviços da camada física independentes do suporte de comunicação e todos os serviços da camada de ligação de dados e, consegue gerir :

- 2000 variáveis de 128-byte variáveis ou 4000 variáveis de 64 byte.
- até 1000 pedidos de transferência de mensagens
- até 128 pedidos urgentes e 128 pedidos normais.

O **FULLFIP** é o circuito do WorldFIP mais completo. Inclui os serviços da camada física independentes do suporte de comunicação e quase todos os serviços da camada de ligação de dados e camada de aplicação.

Nota: FIPART e FIPIU foram desenvolvidos pela TELEMECANIQUE. FULLFIP foi desenvolvido pela CEGELEC.

2.6 Normalização Francesa do FIP

O WorldFIP é uma rede local industrial normalizada, isto é, todo o seu funcionamento está descrito em normas. Qualquer utilizador da rede pode ter acesso à informação sobre o seu funcionamento consultando as respectivas normas.

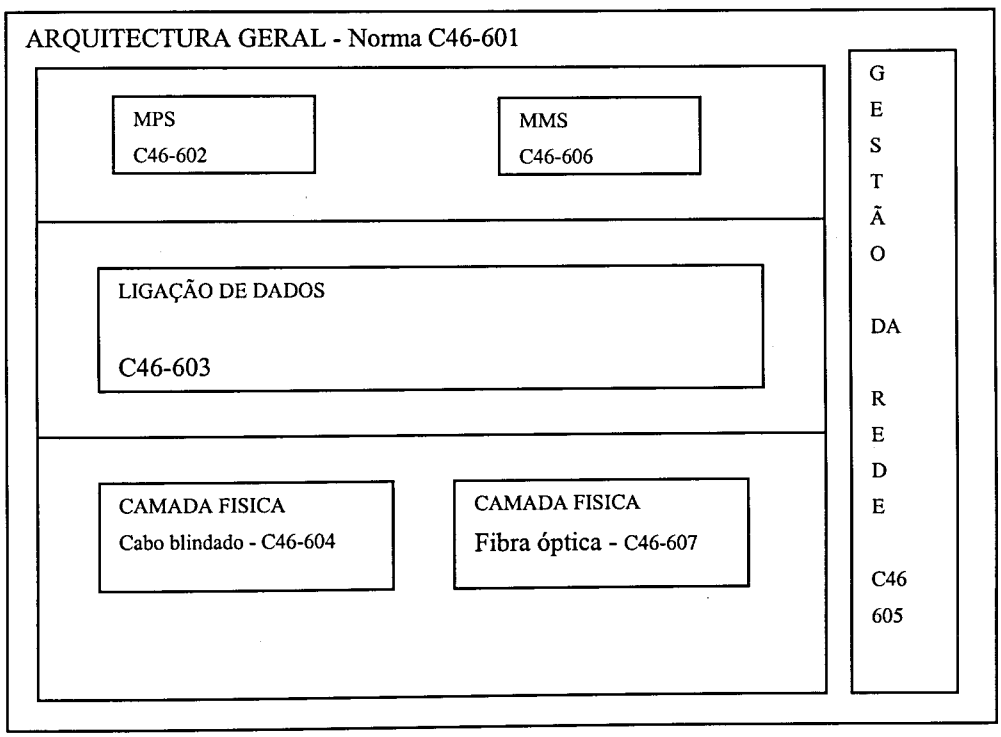


Figura 2.11 - Norma francesa para o WorldFIP

C46 601: barramento FIP para troca de informação entre accionadores, sensores e autómatos - Arquitectura geral do sistema

C46 602: barramento FIP para troca de informação entre accionadores, sensores e autómatos - Camada de aplicação MPS

C46-603: barramento FIP para troca de informação entre accionadores, sensores e autómatos - Camada de ligação de dados.

C46- 604: barramento FIP para troca de informação entre accionadores, sensores e autómatos - Camada física em cabo blindado

C46 605: Barramento FIP para troca de informação entre accionadores, sensores e autómatos - Gestão da rede.

C46-607: Barramento FIP para troca de informação entre accionadores, sensores autómatos- Camada física para suporte físico de fibra óptica.

Capítulo 3: Modelação e Simulação Orientada ao Objecto

3.1 Introdução

Este capítulo, tem como objectivo, fazer uma apresentação das linguagens de programação orientadas ao objecto como ferramentas de ajuda à modelação e simulação de um sistema para avaliar o seu desempenho.

O crescente aumento da complexidade das redes informáticas implica um crescente aumento da dificuldade de análise dos seus comportamentos e um aumento inerente dos seus custos. Assim, surgiu a necessidade de avaliar o desempenho de um sistema ainda na fase da sua concepção.

3.2 Introdução à Avaliação do Desempenho de um Sistema

A evolução permanente dos sistemas e redes informáticos obriga à necessidade crescente de ferramentas que facilitem o estudo do seu comportamento. Em particular, é necessário haver um apoio na fase de concepção, desenvolvimento e utilização dos sistemas e redes informáticos. Actualmente os custos de desenvolvimento e de implementação obrigam a que se faça uma avaliação do desempenho do sistema a estudar ainda durante a fase da concepção por forma a prever a sua resposta relativamente aos objectivos propostos inicialmente para o seu funcionamento. Esta avaliação permite comparar diferentes alternativas de concepção do sistema, tendo como suporte, os critérios de desempenho préfixados. A avaliação de desempenho de uma rede permite ainda fornecer aos seus utilizadores estimativas das operações de menor desempenho da rede, bem como, os meios de optimizar o seu funcionamento.

Para fazer o estudo de avaliação do desempenho de um sistema, é necessário que haja um conhecimento prévio e exacto do seu funcionamento, dos critérios de avaliação a estudar e qual técnica a utilizar [3.1].

Os critérios de desempenho são medições que permitem avaliar o desempenho de um sistema. Estes critérios dependem do tipo do sistema e dos objectivos a atingir sendo extremamente importante fixá-los à priori. Por exemplo, para um processador, o desempenho pode ser medido através da velocidade de processamento. Exitem três técnicas de avaliação do desempenho de um sistema e a sua escolha depende

essencialmente do tempo e requisitos necessários para resolver o problema. Estas técnicas de avaliação do sistema são apresentadas a seguir.

3.3 Técnicas de Avaliação do Desempenho de um Sistema

As três técnicas para avaliar o desempenho de um sistema são a medição, a analítica e a simulação [3.1].

A medição pressupõe a existência física do sistema e só depois de este estar implementado se podem fazer as medições para avaliar o desempenho do sistema. Esta técnica pressupõe a existência de um protótipo o que a torna dispendiosa. Outra desvantagem desta técnica é a pouca flexibilidade [3.2].

Os métodos *analiticos* baseiam-se em métodos matemáticos que caracterizam o sistema a ser estudado e envolvem a criação de um modelo matemático o que torna difícil avaliar o desempenho de um sistema complexo [3.3].

A *simulação* é identica ao método analítico mas obtém-se a solução a partir da utilização de um programa de computador que simula o comportamento do sistema a estudar. Relativamente ao método analitico, este método requer um grau de abstracção menor e é mais acertiva já que se baseia num menor número de suposições [3.3].

Actualmente, a simulação é a alternativa preferida para a análise do desempenho de um sistema, facto que se justifica pela grande evolução tecnológica computacional. A simulação permite tirar proveito da velocidade de processamento de um computador, da sua capacidade de memória e da disponibilidade de uma linguagem de programação.

3.4 Introdução à Simulação

O comportamento de um sistema real ao longo do tempo é estudado através de duas etapas básicas: a construção do modelo e a simulação.

O modelo é composto por um conjunto de hipóteses e de pressupostos que descrevem o funcionamento do sistema. Dependendo do tipo do sistema em causa, o modelo pode ser classificado em estático ou dinâmico, em determinístico ou estocástico e em contínuo ou discreto [3.6].

Uma vez desenvolvido e validado, o modelo pode ser utilizado para estudar a resposta do sistema a alterações das variáveis de entrada. Podem fazer-se ensaios ao sistema, por forma a prever o impacto das alterações no seu desempenho. Os resultados obtidos permitem não só compreender o sistema mas permitem também propôr alternativas ao modelo para que o seu comportamento reproduza o mais possível o real.

A simulação é um método que reproduz o funcionamento de um sistema real [3.4]. A simulação é uma ferramenta de análise, para prever os efeitos de alterações em sistemas já existentes, mas pode ser também uma ferramenta de projecto, para prever o desempenho dos sistemas antes de serem construídos. Esta técnica pode ser aplicada em diferentes tipos de aplicações. No entanto, para cada aplicação é necessário ter em consideração as vantagens e as desvantagens provenientes da utilização da simulação. Algumas das vantagens principais da simulação são [3.6]:

- ajudar a perceber como o sistema funciona de facto;
- permitir testar um sistema com entradas críticas, cuja probabilidade de ocorrência no sistema real é baixa mas que podem por em causa o bom funcionamento do sistema e a segurança de pessoas e bens e,
- uma vez construído, o modelo pode ser utilizado repetidas vezes para analisar o comportamento do sistema.

No entanto, na simulação podem ser identificadas algumas desvantagens, tais como, tempos de execução elevados, sempre que os modelos são mais complexos, exigir experiência de modelação e simulação por forma a encontrar o modelo mais adequado ao estudo em causa e obter resultados de difícil compreensão, podendo conduzir a uma falsa interpretação do funcionamento do sistema.

O estudo por simulação pode ainda ser dividido em doze etapas que, por sua vez, podem ser agrupadas em quatro fases: formulação do problema, construção do modelo, execução do programa e, por fim, a fase da implementação [3.6].

A **1ªfase - fase da formulação** - consiste na definição do problema (etapa 1) e dos objectivos a atingir (etapa 2) que indicam quais as respostas a que a simulação tem que responder. Nesta etapa, as variáveis de entrada do sistema e o seu modo de

funcionamento, as suas limitações e as medidas que melhor representam o seu desempenho têm que ficar bem definidas.

A **2ªfase - construção do modelo** - é constituída pela construção do modelo (etapa 3), pela selecção das variáveis de entrada (etapa 4), pela escrita do programa (etapa 5), pela verificação (etapa 6) e pela validação do modelo (etapa 7). Nesta fase,a escolha de um modelo para um determinado sistema depende essencialmente das suas características e do grau de conhecimento, por parte do projectista do modelo, do funcionamento do sistema. Existe uma relação constante entre a construção do modelo e as variáveis escolhidas para variáveis de entrada. À medida que o sistema se vai tornando mais complexo, também os dados de entrada se vão alterando. Os objectivos definidos na primeira fase, definem o tipo de dados de entrada necessários.

Depois de construído, o modelo, tem que ser convertido para um programa de computador para se obter um modelo computacional. A linguagem de modelação é seleccionada de acordo com suas as potencialidades, podendo pertencer a uma das quatro categorias seguintes, tendo cada uma vantagens e desvantagens: (1) linguagens de alto nível de uso geral (Fortran, C, Pascal); (2) extensão de linguagens de uso geral (3); linguagens de simulação (Simple++) e (4) pacotes de simulação que permitem ao utilizador construir um modelo através de menus de diálogo. O pacote de simulação é prático se a aplicação em estudo não se diferenciar muito das opções fornecidas, sendo, a falta de flexibilidade, a sua grande desvantagem. A linguagem de simulação é mais flexível e mais potente, podendo criar-se o que se pretende de raiz. Estas linguagens de simulação podem ser divididas em duas categorias: contínuas e discretas consoante o tipo de eventos que simularem.

Construído o modelo, é necessário verificar a coerência entre os dados considerados e os dados reais, verificando se o modelo representa o sistema em causa. A etapa da verificação assegura que o modelo desenvolvido tem em consideração toda a informação e modos de funcionamento previstos inicialmente.

A etapa de validação assegura que o modelo se comporta como o sistema real. Um modelo válido é um modelo em que o seu comportamento está de acordo com o que se pressupôs inicialmente. A obtenção de um sistema válido passa por um procedimento iterativo entre os resultados obtidos e os esperados, até obter o comportamento pretendido. A etapa da validação garante que o modelo seja, para o domínio em estudo, uma réplica do sistema real.

A **3ª fase - execução do programa** -é a execução do programa de simulação. Depois de construído, verificado e validado o modelo, determinam-se quais as alternativas possíveis para simular (etapa 8) que dependem essencialmente do tempo de inicialização e do tempo de execução. Na etapa seguinte (etapa 9), os resultados da simulação são analisados e interpretados para medir o desempenho do sistema em causa. Se os resultados forem satisfatórios, o estudo da simulação acaba, caso contrário, deverão ser identificadas as causas e executar novamente o programa (etapa 10). A simulação é um processo iterativo, podendo-se fácilmente "andar para trás" nas várias etapas da simulação, com o objectivo de se fazer um estudo mais exaustivo. As iterações proporcionam um maior conhecimento do sistema e do seu desempenho.

A **4ªfase - implementação** - consiste na elaboração da documentação (etapa 11) de todo o projecto e da implementação do sistema (etapa 12).

O sucesso na obtenção de resultados finais, depende da validade das suposições que se fizeram no inicio da construção do modelo e, um ponto crucial, é a fase de validação porque um modelo inválido conduz a resultados errados e, eventualmente, a conclusões erradas que depois de implementadas podem ser geradoras de prejuizo.

3.4.1 Simulação de sistemas de eventos discretos

Numa simulação por eventos discretos, o estado do modelo é alterado em determinados pontos discretos no tempo, onde ocorrem os eventos [3.6]. O objectivo da simulação por eventos discretos é o de estudar um sistema complexo, calculando os tempos associados a eventos reais de uma situação real. A simulação serve para calcular o mais rápidamente possível, os atrasos temporais que ocorrem em tempo real, sem necessidade imperiosa de esperar por esses atrasos entre eventos que ocorrem na situação real.

Em seguida apresentam-se alguns conceitos usados numa simulação de sistemas de eventos discretos [3.6].

Sistema: conjunto de entidades que interactuam entre si ao longo do tempo com a finalidade de alcançar determinados objectivos.

Modelo: representação lógica- matemática do sistema em função das entidadades e seus atributos, conjuntos, eventos, acções e tempos.

Estado do sistema: conjunto de valores das variáveis que definem o sistema num dado instante do tempo.

Entidade: objecto que corresponde a um componente do sistema e que requer uma representação no modelo.

Atributos: características de uma dada entidade.

Conjunto(Set): conjunto de entidades associadas.

Evento: ocorrência instantânea no tempo que altera o estado do sistema.

Acção: intervalo de tempo em que uma ou mais entidades interactuam responsável pelo cálculo de desempenho de uma função do sistema em estudo.

Atraso: intervalo de tempo cuja duração só é conhecida quando o tempo acaba.

Considerando o serviço de atendimento da camada de ligação aos pedidos da camada de aplicação do barramento de campo WorldFIP, como um exemplo, podemos dizer o seguinte:

- *estado do sistema pode ser definido pelo número de pedidos atendidos e pelo número de pedidos que são anulados.*
- *As entidades são os pacotes de informação que representam os pedidos. Os eventos são as chegadas aleatórias dos diferentes pedidos.*
- *Os atrasos são os intervalos de tempo que um pedido demora a ser atendido.*

3.4.2 Estrutura de uma linguagem de simulação para sistemas por eventos discretos

Como se disse no parágrafo anterior, a construção de um modelo para simulação implica sempre a escolha de uma linguagem de programação. Quando se diz "que um utilizador não tem de se preocupar com as especificações gerais de simulação" significa não tem de se preocupar com a programação do simulador. Uma linguagem de simulação (contrariamente ao que acontece com as linguagens de alto nível) já providencia o escalonamento automático dos eventos, a actualização automática dos conjuntos, a geração de variáveis aleatórias e a colecção automática de valores para estatistica. A estrutura de uma linguagem de simulação baseia-se em cinco conceitos: entidade, relações lógicas, executivo de simulação, relógio e o coleccionador de resultados [3.8].

As entidades são elementos reais que se encontram na vida real. Por exemplo, na construção de um modelo de uma célula de produção teremos como entidades as máquinas e os transportadores. Estas entidades podem ser temporárias ou permanentes (uma máquina é permanente e uma peça de fabrico é temporária). Um dos objectivos da simulação é o de analisar o comportamento das entidades temporárias ao serem processadas pelas permanentes.

As entidades estão relacionadas através de *relações lógicas* que definem o comportamento do modelo. Outros elementos, igualmente importantes, são o *executivo de simulação* e o *relógio*. O relógio mede o tempo e o executivo é o responsável pela coordenação entre a execução das relações lógicas ao longo do tempo, fazendo avançar o relógio para um tempo posterior.

Existem ainda o *gerador de valores aleatórios* e o *colecciònador de resultados*. O primeiro permite ter um comportamento estocástico e o segundo fazer a recolha de resultados para efectuar a respectiva análise.

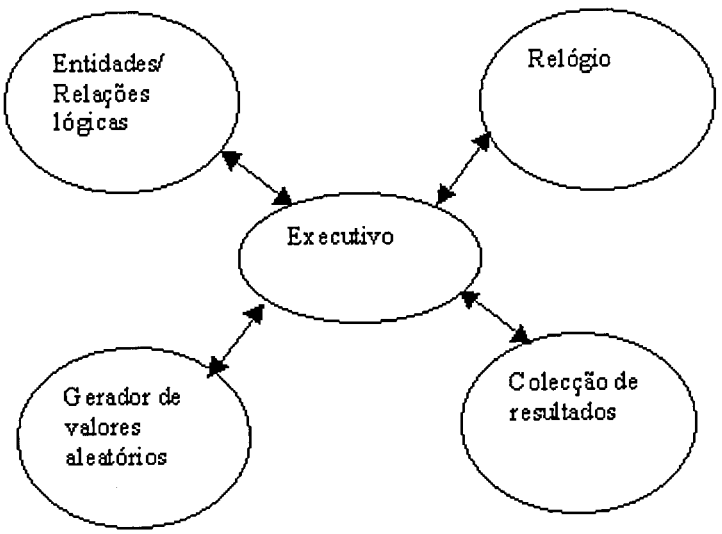


Figura 3.1 - Estrutura de uma linguagem de simulação.

A gestão do tempo do relógio pode ser feito ou por intervalos de tempo fixos ou por eventos. Por intervalos fixos, o tempo avança por exemplo de 5 em 5 segundos, se entretanto não estiver a decorrer nenhuma acção. Por eventos, o tempo avança para o tempo do evento seguinte. Esta gestão de tempo é da responsabilidade do executivo que transfere (avança) o modelo para o passo seguinte, executando as relações lógicas entre as entidades ao longo do tempo. Após o avanço do tempo, o estado do sistema deve ser actualizado de acordo com o respectivo evento.

As relações lógicas entre as várias entidades permitem construir o modelo de simulação de duas formas diferentes: por escalonamento de eventos ou por interacção de processos. O escalonamento por eventos requer uma identificação de todos os eventos que podem ocorrer, assim como, avaliar de que forma esses eventos alteram o estado do sistema. Neste caso, o escalonamento é feito de acordo com um calendário de eventos, que será uma lista que, para em cada instante "t", contém o estado do sistema, uma lista de actividades em execução e quando acaba cada uma delas, o estado das entidades. A lista de actividades contém todos os eventos que foram escalonados para ocorrerem em tempos posteriores. O mecanismo para avançar o tempo de simulação faz-se baseado na lista dos futuros eventos.

Um processo é uma colecção de eventos, actividades e atrasos temporais relativos a uma entidade. O modelo é construído com base nas entidades e na sequência de eventos e activades que lhe estão associados [3.6,3.8]. Quando se utiliza uma linguagem de alto nível de uso geral, como o PASCAL ou o FORTRAN, o simulador constrói o modelo baseado no escalonamento por eventos. Existem outras linguagens, tais como o SIMSCRIPT II.5, que utiliza a aproximação por processos para a construção do modelo[3.15]. Por exemplo, o ciclo de funcionamento de uma máquina. Uma abordagem possível é analisá-lo como um conjunto de eventos - inicio carregamento máquina, fim carregamento máquina, fecho portas, inicio ciclo máquina, avanço unidades, recuo unidades, rotação prato máquina, abertura porta, paragem maquina - e analisar como cada um deste eventos altera o estado do sistema.

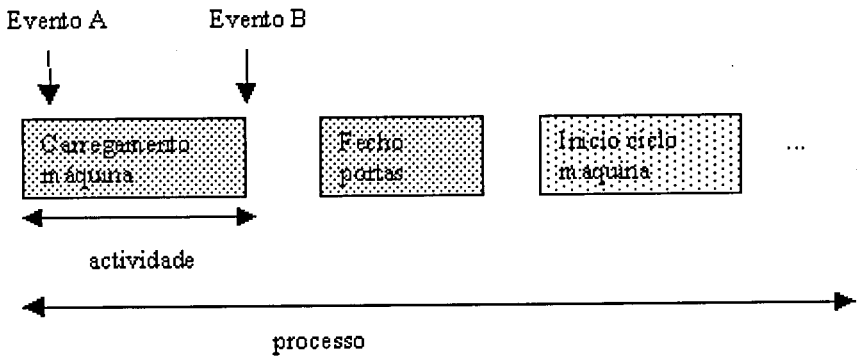


Figura 3.2 - Exemplo do ciclo de funcionamento máquina [3.8].

Outra abordagem é a de analisar o ciclo da máquina como um único processo, centralizando a atenção na entidade máquina. Neste caso, as actividades descrevem uma duração entre dois eventos, por exemplo, entre o início carregamento máquina e fim carregamento máquina(ver figura 3.4).

Seguidamente serão referidas algumas técnicas de construção de modelos para sistemas de eventos discretos.

3.5 Algumas Técnicas de Modelação de Sistemas de Eventos Discretos

Existem, actualmente, várias técnicas de resolução de modelos para sistemas de eventos discretos. Como já se disse anteriormente, o modelo a construir depende do tipo do sistema, conjugado com os resultados que se pretendem obter. Sendo o WorldFIP considerado um sistema de eventos discretos e como é objectivo deste trabalho avaliar o seu desempenho, apresentaremos em seguida, as Redes de Petri, a Teoria da Fila de Espera e da Modelação Orientada ao Objecto e a sua aplicabilidade à modelação do WorldFIP.

3.5.1 Redes de Petri

As Redes de Petri foram desenvolvidas nos anos sessenta. É uma linguagem gráfica que permite modelizar sistemas complexos, permitindo essencialmente, fazer uma análise qualitativa do sistema [3.9].

Sucintamente, a rede de Petri tem dois tipos de nós, nomeadamente a *etapa* e as *transições*. A *etapa* é representada por um círculo e a *transição* por uma barra (ou por um rectângulo). Os circulos e as transições são ligados por arcos (ver figura seguinte).

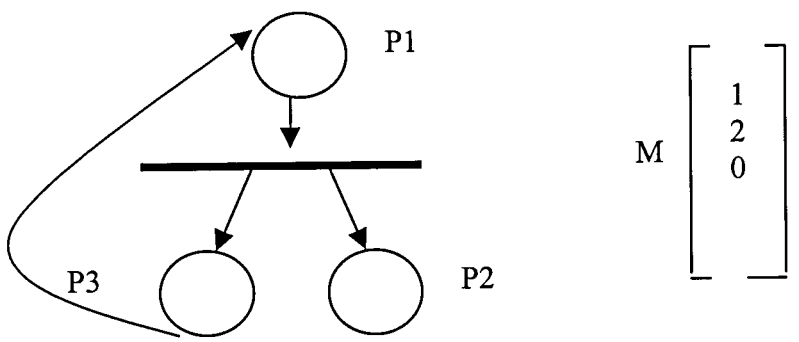


Figura 3.3 .- Representação de uma etapa e uma transição.

A marcação de cada *etapa* representa o estado do sistema em cada instante, sendo representada por um conjunto de valores do tipo inteiro que determina o número de marcas em cada *etapa.* As condições de evolução de uma marcação para outra são definidas pelas transições. Como exemplo, podemos considerar a paragem / arranque

de um motor. Estamos perante uma rede de Petri não autónoma, pois está dependente de eventos exteriores (arranque / paragem motor). A partir de uma dada marcação podem ser executadas várias transições. Normalmente, associa-se a cada transição uma temporização que especifica o tempo antes do disparo da transição. O tempo de espera pode ser nulo (transição imediata), uma constante (transição deterministica) ou pode seguir uma lei de distribuição exponencial e tem-se uma rede de Petri estocástica.

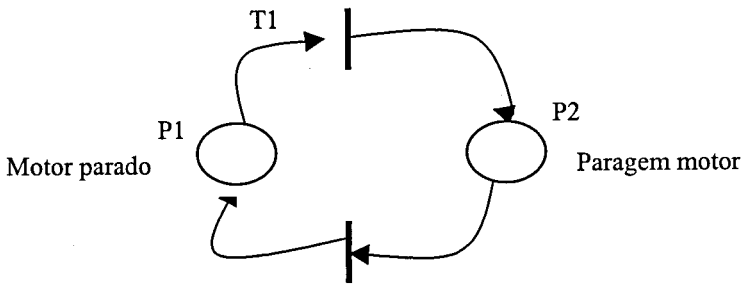


Figura 3.4 - Exemplo de um sistema arranque/paragem de um motor.

Aplicação das redes de Petri ao FIP

As redes de Petri já foram aplicadas ao barramento de campo WorldFIP. O modelo RdPTS- modelo de Réseaux de Petri Temporisés Stochastiques- possui características bem adptadas à modelação de sistemas distribuidos em tempo real. Foi aplicado à rede de terreno WorldFIP, para estudar e modelizar os seus mecanismos de refrescamento e disponibilidade da camada de aplicação [3.10].

3.5.2 A teoria da fila de espera

A teoria de fila de espera lida com problemas que envolvam filas de espera, isto é, espera numa fila para ser atendido por um determinado serviço. É uma técnica bastante utilizada para resolver problemas da vida quotodiana (bancos, supermercados) e na informática (espera de uma resposta de uma impressora). A figura seguinte ilustra os elementos de uma fila de espera [3.5],[3.11].

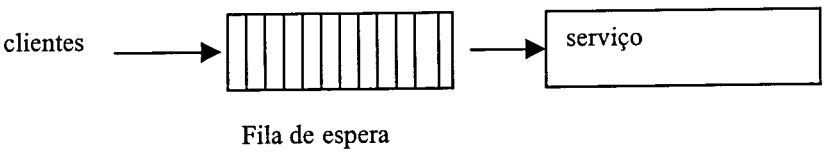


Figura 3.5 - Elementos de uma fila de espera.

De uma forma geral, para analisar o sistema é necessário conhecer:

- quantos clientes chegam e como chegam em função do tempo,
- como se processa o mecanismo do serviço: quais os seus recursos, qual a sua disponibilidade, quanto tempo demora a completar o serviço e,
- como é que o serviço recruta os seus clientes: por FIFO("first in first out"), LIFO ("last in first out") ou por selecção aleatória.

Aplicação das filas de espera no FIP

A teoria das filas de espera já foi aplicada ao estudo do barramento de campo WorldFIP. Foi construido o simulador *SimulFIP* que permite avaliar o desempenho da camada de ligação de dados. Este simulador foi desenvolvido utilizando a linguagem de simulação QNAP2, uma linguagem de descrição e análise de sistemas de filas de espera [3.12].

3.5.3 A Modelação orientada ao objecto

Os métodos de modelação orientados ao objecto parecem ter um futuro promissor, pois os sistemas tendem a ser concebidos como sistemas abertos e distribuidos. O conceito de objecto aparece como sendo o mais natural a adoptar, dada a sua capacidade de encapsulamento e de transferência de mensagens [3.13].

A modelação orientada ao objecto pressupõe a criação de um modelo com base nos conceitos utilizados nos métodos orientados ao objecto. Por métodos orientados ao objecto, entende-se programação orientada ao objecto, análise e concepção orientada ao objecto, ou de uma maneira geral, toda filosofia de base que suporta o desenvolvimento de sistemas orientados ao objecto [3.13].

Aplicação da modelação por objectos ao FIP

Com este trabalho pretende-se criar um modelo que permita avaliar o desempenho da camada de dados da rede WorldFIP utilizando uma linguagem de programação orientada ao objecto, o Simple++. Este modelo será apresentado no capítulo seguinte.

3.6 Linguagem de programação orientada ao objecto

3.6.1 Introdução

O conceito, programação orientada ao objecto, significa "*que o software é organizado como uma colecção de objectos discretos, incorporando tanto a estrutura de dados assim como o seu comportamento*"[3.17]. A programação orientada ao objecto tem como plataforma de construção de um programa o objecto, em que este integra não só um conjunto de operações que definem o seu comportamento como também um conjunto de informações que definem os seus atributos.

A primeira linguagem de programação orientada ao objecto (OOP) foi a SIMULA, projectadada e implementada entre 1962 e 1967. Nos anos de 70 foi introduzida a Smalltalk.[3.6]. Actualmente, existe uma grande variedade de OOP's para diferentes tipos de aplicações, nomeadamente para a simulação, como a Simple++.

3.6.2 Metologias OO versus metodologias funcionais

A grande diferença entre as linguagens tradicionais e a linguagem orientada ao objecto é a forma como a informação e as funções estão estruturadas. Nas linguagens, ditas tradicionais, como o C e o PASCAL, a declaração da informação e as funções do sistema estão separados em zonas distintas e a execução do programa começa na primeira linha do código e o resto é executado sequencialmente. Numa linguagem orientada por objectos, tudo que seja relativo a uma entidade (informação e função) é incorporado de maneira a construir uma classe. A execução do código não segue uma determinada orientação mas vai respondendo aos diferentes eventos, executando diferentes secções de código, de acordo com o tipo de evento.

De uma maneira geral, a utilização das linguagens tradicionais traz alguns problemas a nivel de coordenação, flexibidade e manutenção, quando aplicadas a projectos onde existe mais que um programador. É difícil, numa aplicação deste tipo, coordenar os diferentes programadores de forma a que cada um saiba onde começar e acabar a sua parte de programação e quais as variáveis que pode utilizar. Nestas linguagens, o código descreve as várias operações do sistema, especificando e decompondo as suas funcionalidades. Observou-se que estas linguagens não se

adaptam a uma variação das operações, isto é, se o sistema mantiver a mesma informação mas alterar o seu funcionamento, é necessário reescrever o código, tornando o sistema pouco flexível e aumentando a possibilidade de introdução de erros. Outro factor importante é o da manutenção: quando há necessidade de uma alteração por parte de um dos programadores, é dificil prever quais vão ser os efeitos, dessa alteração, nas outras partes do programa.

Numa linguagem orientada por objectos, cada programador tem de se preocupar com cada um dos seus objectos, dos quais é responsável, sendo cada objecto construído como sendo uma peça única, não interferindo na construção dos restantes. Em termos de coordenação de trabalho é mais fácil, dado que a comunicação entre objectos limita-se ao envio de mensagens entre eles. Se, por qualquer razão, for necessário alterar o funcionamento de um objecto ou inserir um novo, essas alterações não interferem no funcionamento dos restantes que se relacionam com o objecto modificado, dado que é mais fácil a reutilização do código se o programa estiver dividido em sub-módulos [3.15].

3.6.3 Os conceitos da linguagem de programação por objectos

O conceito fundamental é o de **objecto**. Um objecto é algo que existe e que executa um conjunto de operações que definem o seu comportamento[3.17]. Os objectos relacionam-se uns com os outros através de mensagens de tal modo que um objecto quando necessita algo de um outro objecto, envia-lhe uma mensagem. O receptor da mensagem, executa um conjunto de operações e depois de ter a resposta, envia-a para o objecto emissor. O controlo é dado ao objecto receptor até completar o pedido, retornando depois ao objecto emissor.

Cada mensagem tem um código associado que determina como o objecto se comporta quando recebe uma mensagem. Quando o objecto receptor recebe uma mensagem, o código é executado. Chama-se a este código, associado a cada mensagem, um *método.* Quando um objecto recebe uma mensagem, determina qual vai ser o método que terá de executar e passa o controle ao método. Os métodos são idênticos às funções ou procedimentos que se escrevem nas linguagens como o C ou o PASCAL. Enviar uma mensagem a um objecto é idêntico a chamar uma subrotina.

Capítulo 4: Aplicação da Linguagem de Modelação ao WorldFIP

4.1 Introdução

O objectivo deste capitulo é de fazer uma apresentação do modelo criado para o sistema WorldFIP utlizando a linguagem de programação Simple++. Para atingir o objectivo proposto, é ilustrada a filosofia base de construção de um modelo no Simple++, através de um exemplo genérico. Seguidamente, apresentam-se as primitivas de programação que permitem a construção de um modelo WorldFIP. Faz-se, então, uma descrição da configuração física de cada uma das primitivas assim como das respectivas relações lógicas. Por fim, descreve-se como se constrói um modelo para o WorldFIP.

4.1.1 Apresentação do Simple++

Como se referiu no capitulo anterior, o Simple++ é uma linguagem de simulação orientada ao objecto. É uma linguagem dedicada a projectar, simular e visualizar sistemas de produção, isto é, permite a criação de modelos e respectiva simulação para os diferentes níveis hierárquicos existentes num ambiente fabril. Assim, tanto podemos construir um modelo para uma simples linha de produção ou para um sistema de fabrico completo, proporcionando um conjunto de cenários que permitem a avaliação em termos de layout, desempenho e capacidade para cada um dos sistemas modelizados[4.1].

No Simple++, os blocos construtivos básicos (ver figura 4.1) são usados na construção de modelos particulares ou mesmo de novos modelos parciais (classes) que podem vir a ser instanciados na construção de futuros modelos particulares (o modelo do sistema que se deseja simular). Esta grande flexibilidade de modelação é finalmente enriquecida, através da associação de uma representação gráfica, que no momento de edição, pode ser personalizada pelo utilizador[4.2].

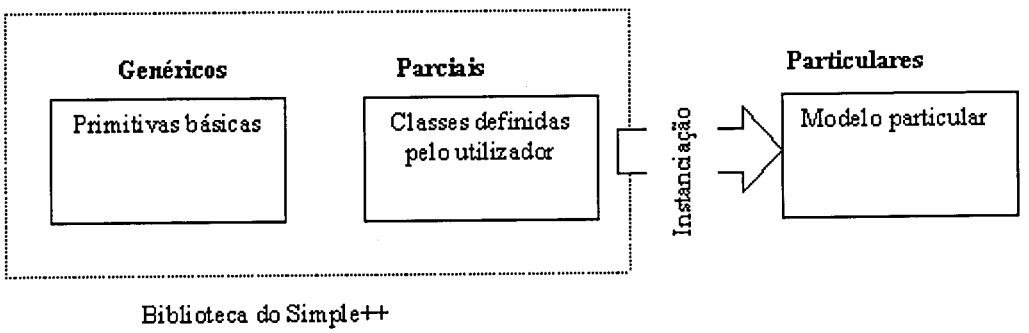


Figura 4.1 - Construção de um modelo no Simple++

Esta filosofia de modelação aplica-se, assim, aos blocos construtivos do Simple++, divididos em :

- Primitivas de modelação do sistema físico (usado na modelação de objectos físicos).
- Primitivas de modelação de configuração e controle (usados na modelação do

comportamento dos objectos físicos, bem como na gestão da informàção usada, produzida e visualizada quando decorre a simulação).

4.1.2 Apresentação do software Simple++

O Simple++ providencia um conjunto de ferramentas que criam um ambiente integrado entre um interface gráfico, a modelação orientada ao objecto, simulação e animação. A janela principal do Simple++ (ver figura 4.2) através dos diferentes menus (File, Environment, Debuger e Profiler) permite a criação e salvaguarda de modelos assim como alterar a configuração global do Simple++.

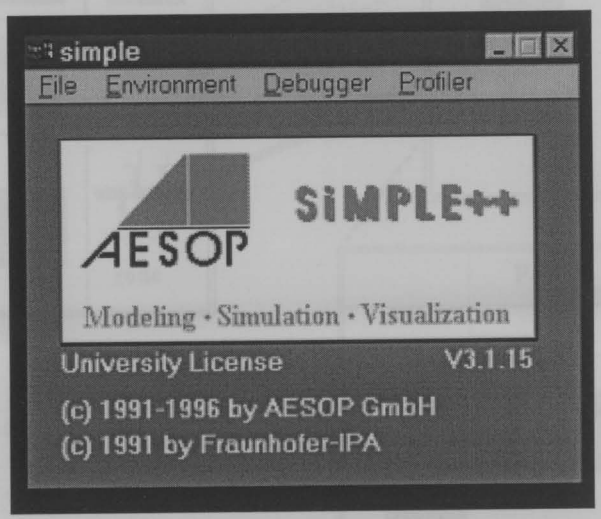


Figura 4.2 - Janela principal do Simple++

No Simple ++, a biblioteca de objectos standard, que é constituida pelos blocos construtivos básicos que permitem a construção de modelos parciais (classes) ou os modelos particulares, está representada na figura 4.3.

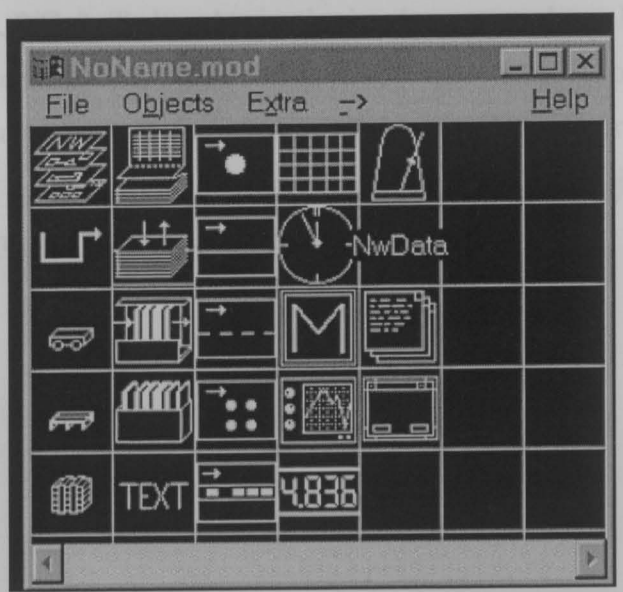


Figura 4.3 - A biblioteca de objectos do Simple++

Modelizar com o Simple ++ começa pela criação de novas classes de objectos, utilizando os objectos standard da biblioteca de objectos. Estes novos objectos construidos pelo utilizador podem, a seguir, integrar a biblioteca de objectos, sendo depois instanciados aquando da construção de um novo modelo. Com a figura seguinte pretende-se exemplificar a filosofia base do Simple++.

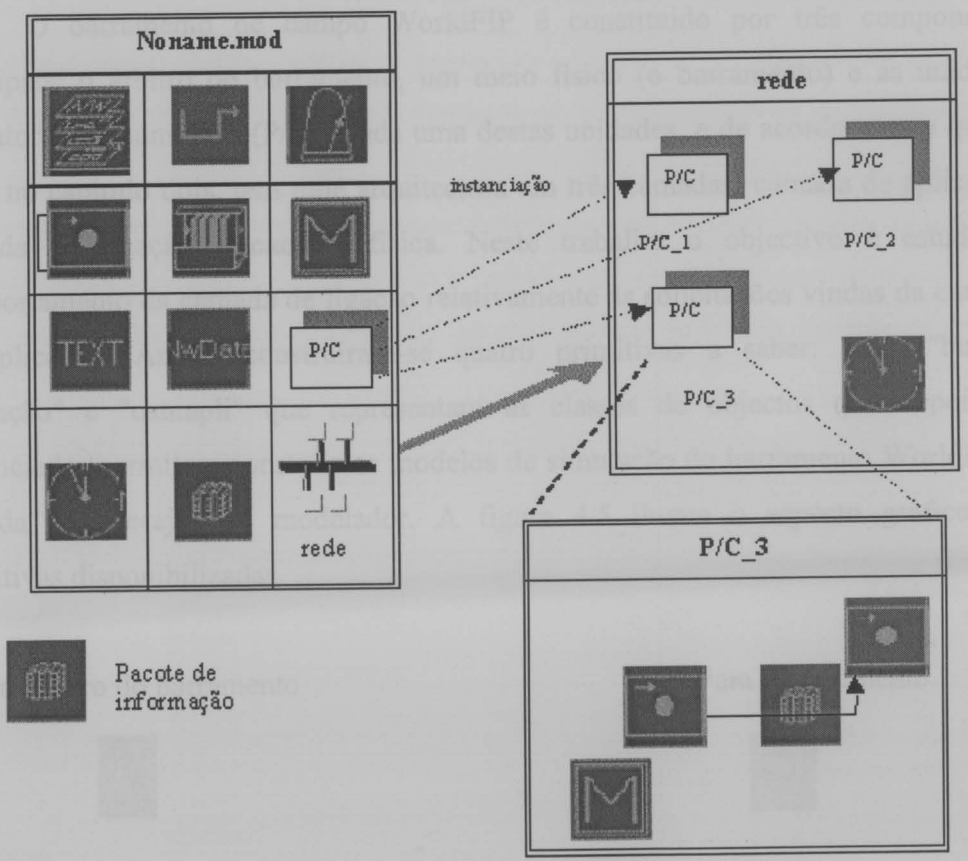


Figura 4.4 - Exemplo da construção de um modelo utilizando o Simple++

Neste exemplo as classes de objectos criadas pelo utilizador foram **P/C** e **rede**. A visão detalhada do P/C permite observar que é constituido por três objectos P/C_1, P/C_2, P/C_3. Todos eles instâncias da mesma classe P/C. A classe **P/C**, foi por seu lado, construida através da instanciação das classes de objectos básicas, pertencentes à biblioteca standard do Simple++. É interessante realçar que se houver necessidade de alterar o objecto P/C, bastará realizar a alteração desejada na classe, para que ela se verifique em todas as suas instâncias.

4.2 A linguagem de modelação da rede WorldFIP

4.2.1 Apresentação das primitivas da linguagem de modelação WorldFIP

O barramento de campo WorldFIP é constituido por três componentes principais: o árbitro do barramento, um meio físico (o barramento) e as unidades produtoras/consumidoras(P/C). Cada uma destas unidades, e de acordo com o que se disse no capítulo dois, tem uma arquitectura em três camadas: camada de aplicação, camada de ligação e camada física. Neste trabalho o objectivo é estudar o comportamento da camada de ligação relativamente às solicitações vindas da camada de aplicação. Assim, construiram-se quatro primitivas a saber: "ab", "buses", "cligação" e "camapli" que representam as classes de objectos que depois de instanciadas permitem construir os modelos de simulação do barramento WorldFIP à medida dos desejos do modelador. A figura 4.5 ilustra o aspecto gráfico das primitivas disponibilizadas.

Para o árbitro do barramento

Para o barramento

Para a camada de ligação de dados

Para a camada de aplicação

Figura 4.5 - Representação gráfica das quatro primitivas de programação

4.2.2 Descrição da primitiva"ab":

A classe "ab" modeliza o árbitro do barramento do WorldFIP (ver capitulo 2).O ab tem como função fundamental a gestão do barramento, traduzindo-se essa função na realização cíclica de um conjunto de operações. Assim, o AB começa por ler a tabela de escrutação e enviar os respectivos pacotes de informação às diferentes unidades através do objecto "buses", ficando, em seguida, à espera de uma resposta. Na construção da classe "ab" foram identificadas componentes físicas, umas relacionadas com a inerente necessidade de interface ao meio (entidades ou objectos físicos) que recebem e enviam pacotes de informação, e outras, com a necessidade de existência de recursos físicos (filas de espera, tabelas) para satisfazer os diferentes pedidos. O conjunto de métodos criados (escritos na linguagem SIMTALK) definem o comportamento da classe "ab". A figura 4.6 faz uma ilustração detalhada da sua constituição

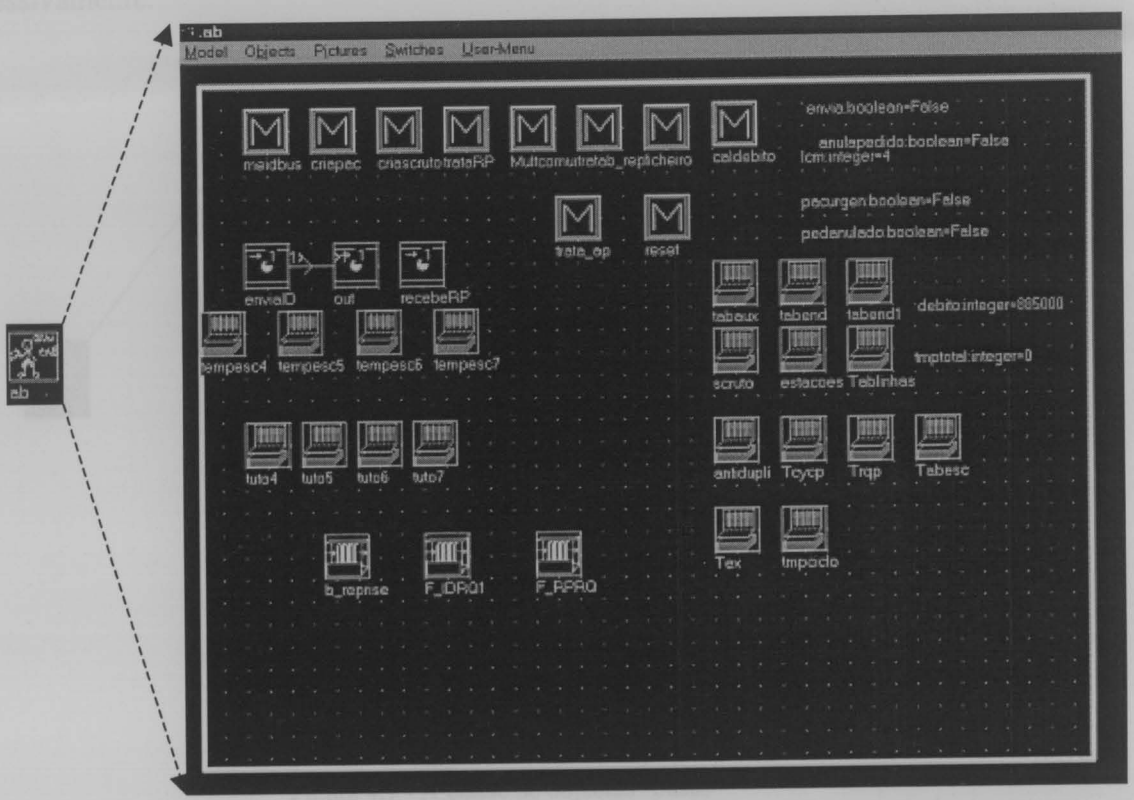


Figura 4.6 - A classe de objectos"ab"

4.2.3 Descrição da primitiva "buses"

A classe "buses" modeliza o meio físico do barramento de campo WorldFIP. Uma vez instanciada, num modelo particular, o objecto "buses" faz a interligação entre as instâncias da classe "ab" e as da classe "cligação", isto é, os pacotes de informação passam pelo "buses" quando circulam entre estes objectos. Na construção desta classe foram construidos tanto componentes físicas necessárias a realização da tranferência dos pacotes de informação, bem como métodos que definem o comportamento desta classe.

Como exemplo, o método "init" tem como função criar o número de entidades físicas que se chamam "bus", de acordo com o número de utilizadores (nutil) da rede. Assim, cria-se uma matriz de [nutil+1 , nutil+1] de forma a que a primeira linha da matriz pertence ao *ab* para enviar mensagens aos diferentes utilizadores, a segunda linha pertence ao primeiro utilizador para enviar as suas mensagens e assim sucessivamente.

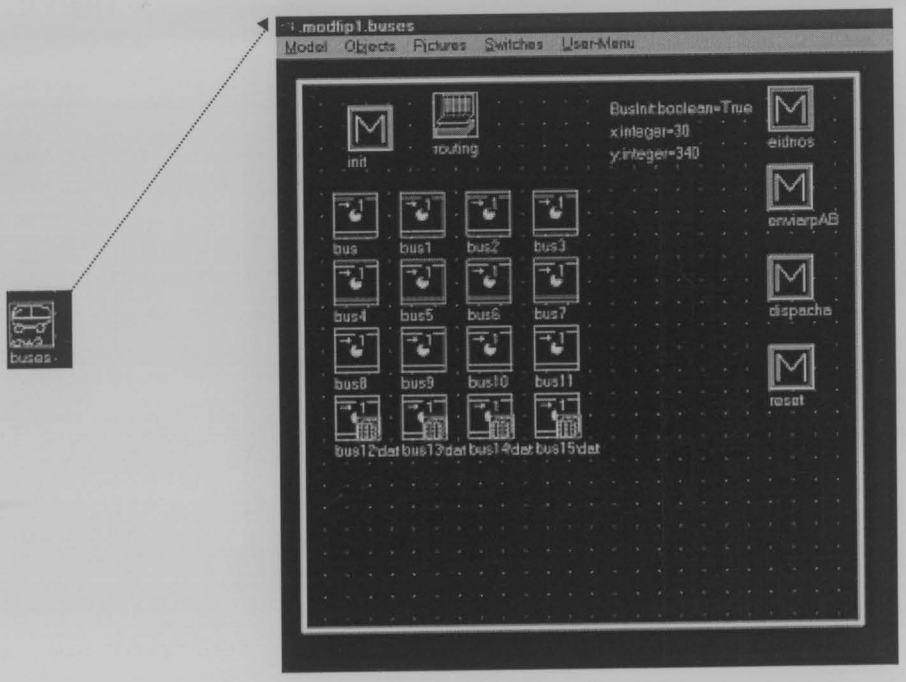


Figura 4.7 - A classe de objectos "buses"

4.2.4 Descrição da primitiva" cligação" e ""camapli"

As unidades produtoras/consumidoras P/C, são constituidas por duas camadas: a camada de ligação de dados e a camada de aplicação. Assim, cada unidade P/C é construida à custa da instanciação de duas classes, a "cligacao" e a "camapli", podendo ser representada pela figura 4.8.

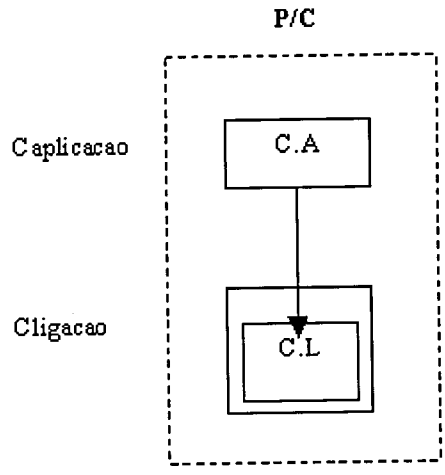


Figura 4.8 - A unidade P/C

Descrição da primitiva "cligacao"

A camada de ligação do barramento WorldFIP é modelada na classe "cligacao". Assim, esta classe uma vez instanciada, recebe os pedidos vindos da objecto "campli" e os pacotes de informação vindos do "ab" e, de acordo com as regras de funcionamento cria os diferentes tipos de resposta através dos métodos. Em cada camada de ligação é necessário criar um conjunto de componentes físicas (objectos fisicos) por onde passam os pacotes de informação (entidades), bem como, tabelas e filas de espera que lhe permita executar as suas funções. Em cada objecto "cligacao" existe uma tabela "prodcons" que contêm a informação respeitante às variáveis locais à unidade.

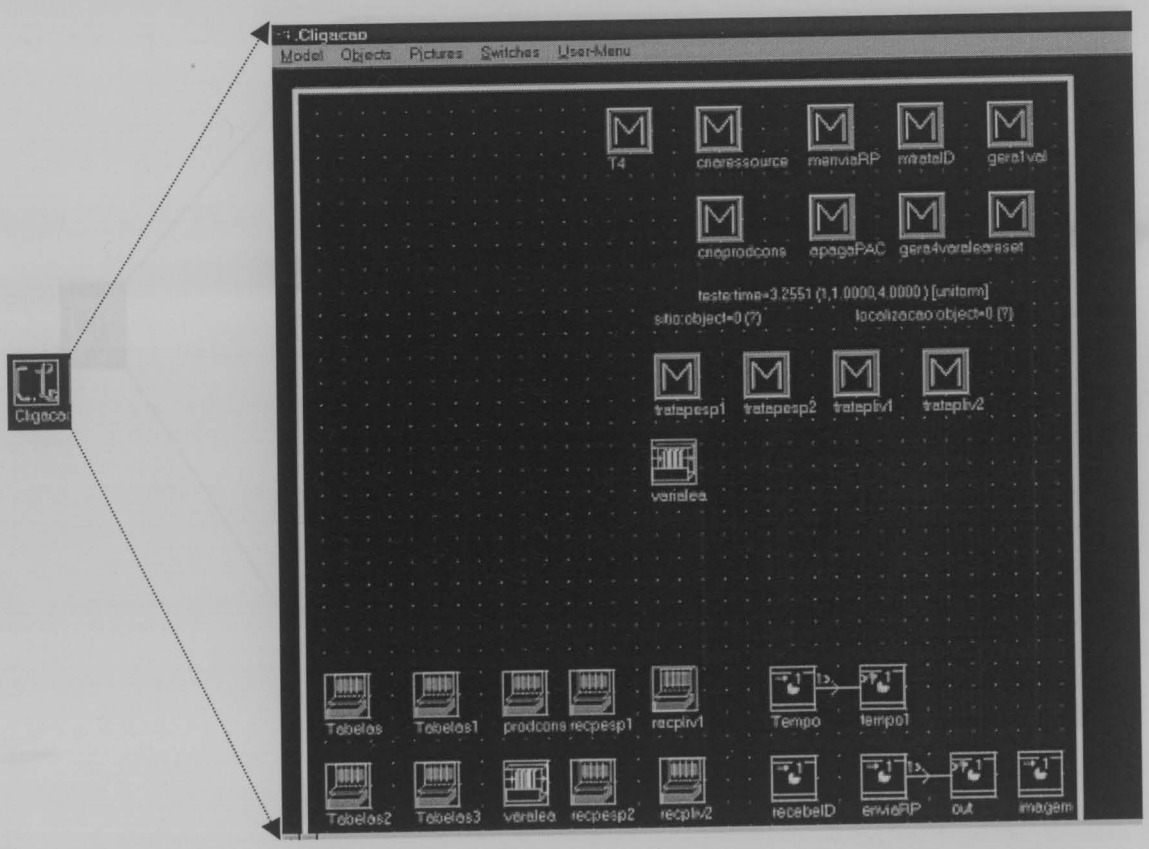


Figura 4.9 - A classe de objectos "cligacao"

Descrição da primitiva "camapli"

A classe "camapli" modeliza a camada de aplicação do barramento do WorldFIP. Tem como função, criar os diferentes pedidos e enviá-los à camada de ligação, ficando à espera de uma resposta. Tal como as outras classes, necessita de um conjunto de componentes físicos que permitam a recepção e envio de pacotes de informação, de filas de espera e tabelas que representam os recursos físicos necessários para cumprir a sua função e de um conjunto de métodos que definem o seu comportamento. Além disso, necessita de geradores de valores aleatórios, que podem estar associados a um evento, que neste caso, será a geração de um pedido.

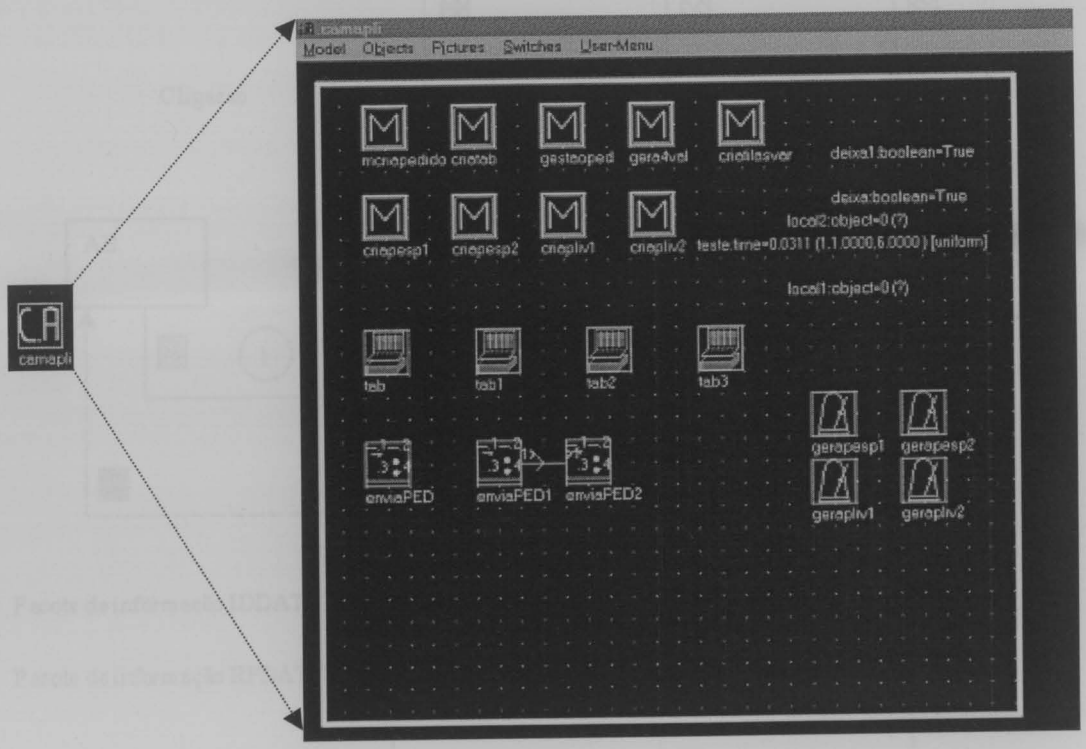


Figura 4.10 - A classe de objectos "camapli"

4.2.5 As relações lógicas do serviço de transferência de variáveis

A partir das classes de objectos "ab", "buses", cligacao" e "caplicacao" pode então ser construído o modelo. Vamos considerar um exemplo de uma rede constituida por três unidades P/C e o serviço de transferências de variáveis (ver capitulo 2).

As relações lógicas representam o comportamento de cada um dos objectos quando há a ocorrência de um evento, isto é, quando um objecto envia ou recebe um pacote de informação. A figura seguinte ilustra como se modelizou o serviço de transferência de variáveis:

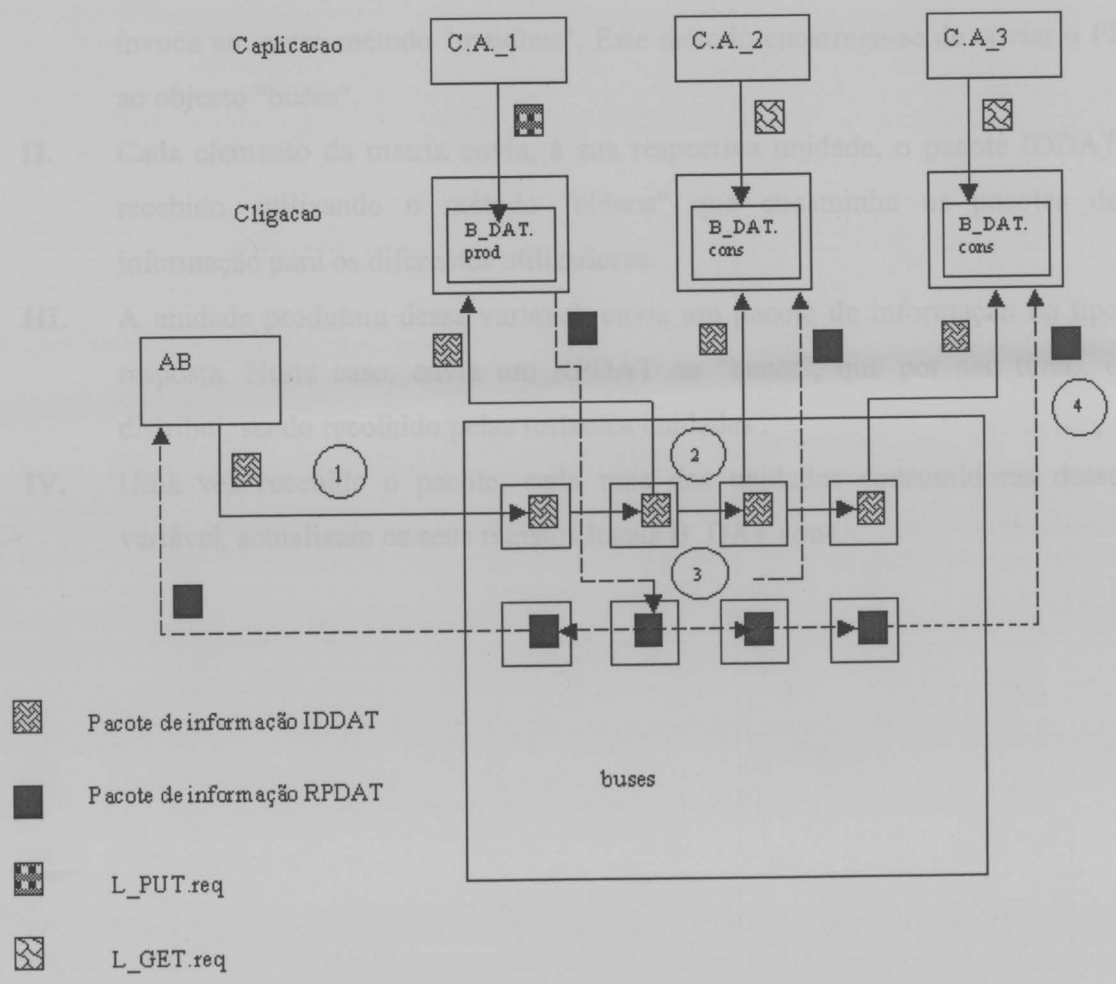


Figura 4.11 - Modelação do serviço de transferência de variáveis

A camada de aplicação utiliza o serviço "L_GET.req" para leitura e "L_PUT.req" para escrita de variáveis nos registos B_DAT.cons e B_DAT.prod da camada de ligação de

dados. Assim, "L_GET.req" e "L_PUT.req" são representados por pacotes de informação que chegam à camada de ligação. Podemos dividir este serviço em várias etapas:

I. "ab", ao ler a tabela de escrutação, envia pacotes de informação IDDAT às diferentes unidades através dos três elementos da 1ª linha da matriz do objecto "buses" (no WorldFIP, as unidades que enviam mensagens também recebem essas mensagens). Com este objectivo o método "criapac" na classe "ab" cria um pacote de informação de acordo com a tabela de escrutação e o coloca na entidade física (buffer)"enviaID". Este ao receber um pacote de informação invoca um outro método "meidbus". Este método encarrega-se de enviar o PI ao objecto "buses".

II. Cada elemento da matriz envia, à sua respectiva unidade, o pacote IDDAT recebido, utilizando o método "eidnos" que encaminha os pacotes de informação para os diferentes utilizadores.

III. A unidade produtora dessa variável, envia um pacote de informação do tipo resposta. Neste caso, envia um RPDAT ao "buses", que por seu turno, o distribui, sendo recolhido pelas restantes unidades .

IV. Uma vez recebido o pacote, cada uma das unidades consumidoras dessa variável, actualizam os seus registos locais B_DAT.cons.

4.2.6 As relações lógicas no serviço de pedido de transferência de variáveis

Como se referiu no capítulo dois, os serviços de pedidos de transferência de variáveis permitem o pedido explicito de difusão de uma ou mais variáveis. Assim, a camada de aplicação envia um pedido aleatório á camada de ligação através dos serviços L_SPEC_UPDATE.pedido(ID,suite) e L_FREE_UPDATE.pedido (SUITE, prioridade) para os pedidos explicitos e para os pedidos livres. Estes serviços são representados, também, por pacotes de informação (ver figura 4.10).

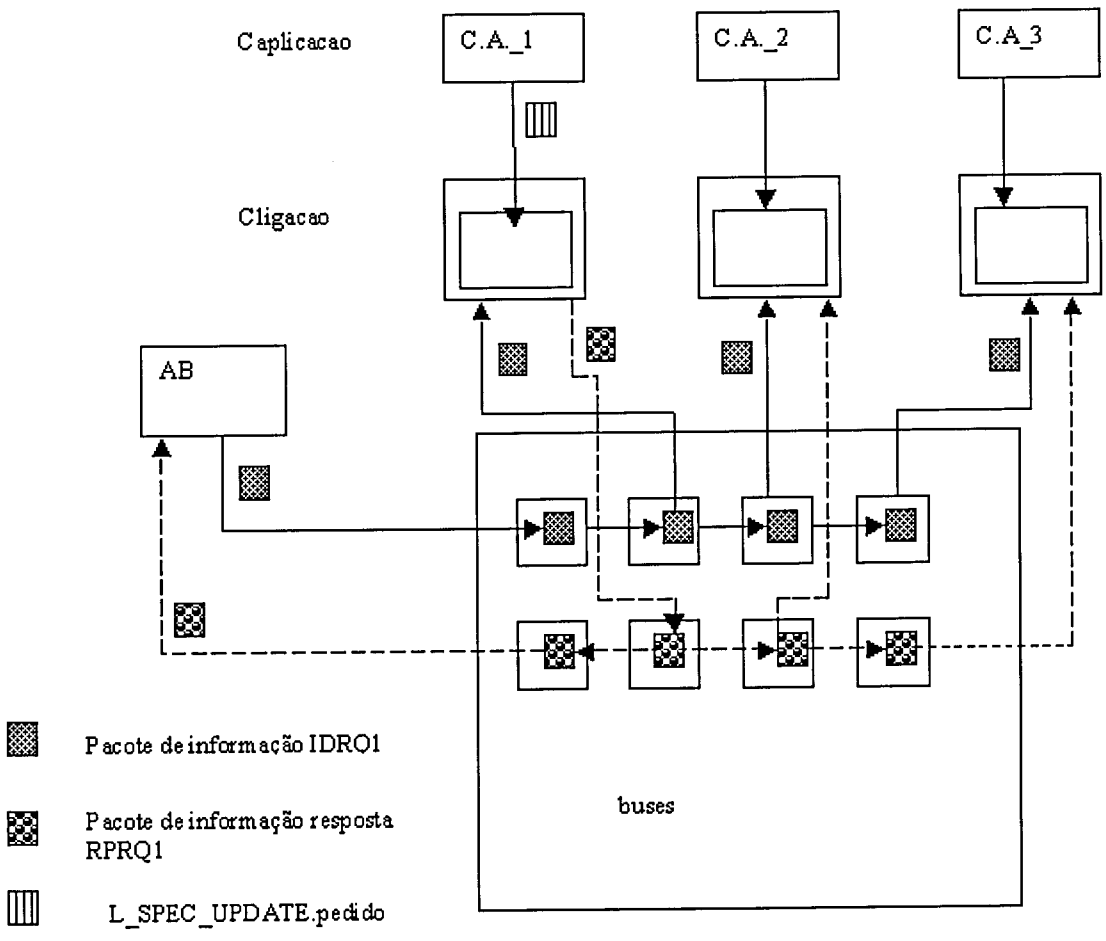


Figura 4.12. Modelação do serviço de pedidos de tranferência de variáveis

Os pedidos chegam de uma forma aleatória e de acordo com as tabelas que foram criadas na camada de aplicação. Assim, quando começa a simulação, cada gerador

gera os respectivos pacotes de informação, sendo cada pedido gerado na camada de ligação. Assim, na classe "camapli", forma criados quatro métodos que têm como função a criação de pedidos na respectiva classe "cligacao".

Suponhamos o caso do pedidos explicito especifico periódico (ver capitulo dois e figura 4.10).

I. O objecto "ab" ao ler a tabela de escrutínio, sabe a que tipo de pedido as variáveis estão associadas. Se Id_k estiver associado ao pedido especifico periódico, envia um pacote de informação IDRQ1(Id_k) às diferentes entidades.

II. A entidade produtora de Id_k, ao receber o pacote de informação IDRQ1(Id_k), verifica se já houve algum pedido para essa variável e consoante a resposta, envia diferentes pacotes de informação. Se já houve pedido, a entidade responde com o pacote RPRQ1, se não, envia RPDAT e o "ab" passa ao elemento seguinte da tabela de escutinio.

III. Ao receber RPRQ1, o árbitro do barramento começa imediatamente a satisfazer o pedido, isto é, envia pacotes IDDAT(ID_j), em que ID_j é a primeira variável a ser transmitida e vai recebendo "RPDAT", utilizando o serviço de tranferência de variáveis.

4.3 A Construção de um Modelo WorldFIP

A construção de um modelo particular para o WorldFIP, tem sempre como principio, a utilização de uma primitiva de programação da classe "ab" e uma da classe "buses". O número de primitivas das classes "cligacao" e "camapli" depende do número de unidades que a rede vai suportar. Este número é um dado introduzido pelo utilizador do modelo. No caso da figura 4.11, o modelo é designado por "modfip1" e é constituido por três unidades P/C.

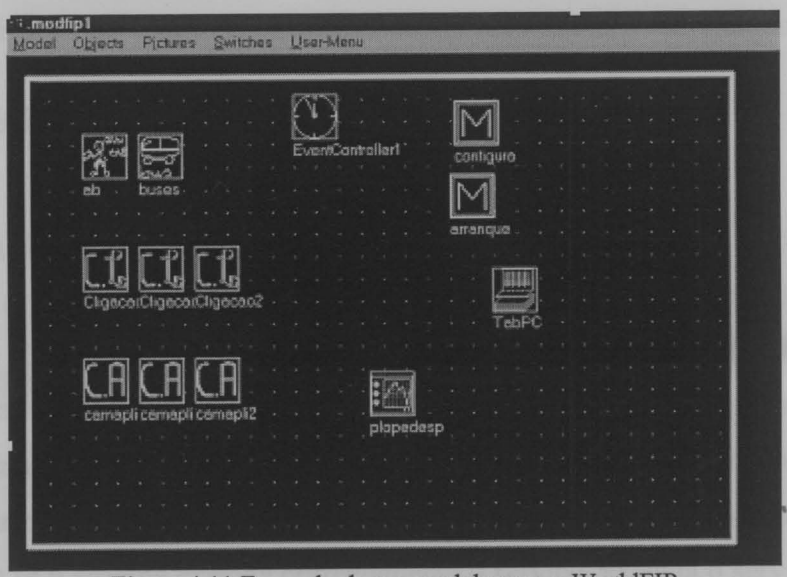


Figura 4.11 Exemplo de um modelo para o WorldFIP

Seguidamente, é necessário configurar a rede, isto é, preencher a tabela que nos indica quais as variáveis utilizadas no barramento e os respectivos periodos de escrutínio, quais os pedidos a que cada variável está associada. O método "configura" tem como função configurar a rede, assim como, criar os recursos necessários para cada "cligação". O método "arranque" prepara a rede para uma nova simulação. Para dar inicio a uma nova simulação, utilizamos o objecto genérico denominado por controlador de eventos que permite o arranque e paragem da simulação. Para visualizar dinâmicamente, a ocorrência de qualquer tipo de evento durante a simulação, utilizou-se uma instância do objecto genérico *plotter,* que se designou por "plopedesp". Este objecto permite não só a visualização dinâmica como a recolha de informação para um ficheiro.

Capítulo 5 - Resultados

5.1 Introdução

Este capítulo tem como objectivo, verificar o comportamento do modelo "modfip1" no SIMPLE++, avaliar o desempenho da camada de ligação de dados do WorldFIP. Assim, começa-se por fazer uma introdução aos critérios de desempenho de uma rede, e em particular, aos critérios escolhidos para avaliar o desempenho da camada de ligação do WorldFIP. Em seguida, apresentam-se os resultados obtidos.

5.2 Critérios de avaliação de desempenho de uma rede.

Uma das etapas fundamentais, no projecto de uma rede, é a definição dos critérios de avaliação do desempenho. O utilizador, o projectista ou o responsável da manutenção têm objectivos diferentes em relação à concepção e à funcionalidade da rede, solicitando, cada um, diferentes tipos de serviços, o que implica que os critérios de desempenho escolhidos sejam também diferentes. Uma maneira de escolher estes critérios é fazer uma lista de serviços oferecidos pelo sistema e analisar para cada um dos serviços se o sistema executa correctamente o serviço, executa o serviço mas com erros ou simplesmente não executa o serviço(ver figura 5.1).

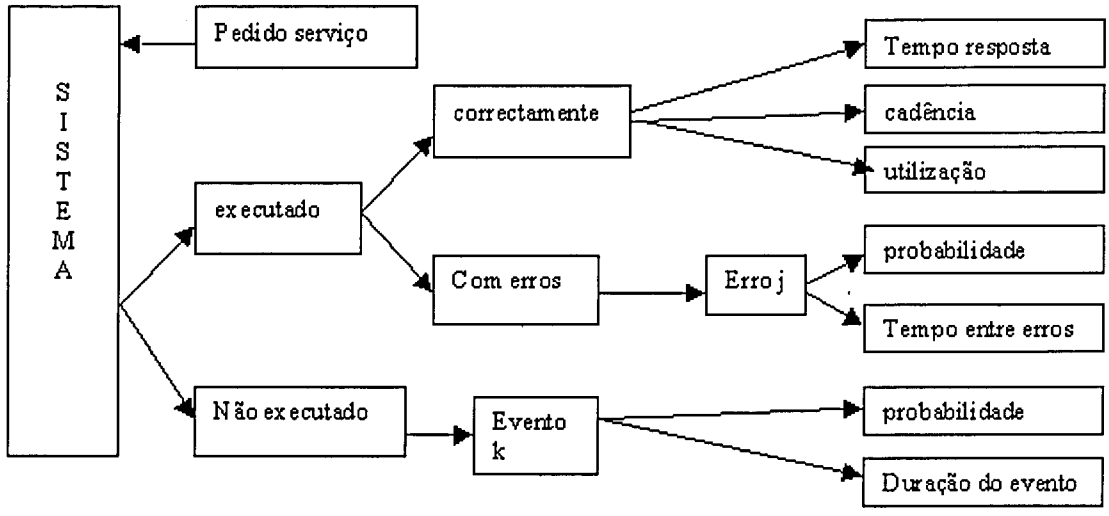


Figura 5.1. Três tipos de respostas do sistema quando solicitado por um serviço[5.2].

Se executar correctamente o serviço, o seu *desempenho* é medido pelo tempo de resposta, pela cadência com que o serviço é executado e a pela utilização dos recursos necessários para executar o serviço. Se executar o serviço com erros convém saber qual a percentagem dos erros e a frequência com que ocorrem. A métrica utilizada é a *fiabilidade.* Se o sistema não executar o serviço, diz-se que está indisponível. A métrica utilizada é a *disponibilidade.* Neste caso convém saber durante quanto tempo o sistema não responde e quais as causas.Assim, pode-se dizer que para a métrica *desempenho* temos:

Tempo de resposta: é definido pelo tempo que decorre entre um pedido de um utilizador e a resposta do sistema. É o tempo de reacção de um sistema a um acontecimento.

Cadência: mede a frequência com que os pedidos podem ser atendidos.Numa rede, a cadência pode ser medida por número de pacotes por segundo ou número de bits por segundo.

Utilização: especifica a porção máxima da capacidade da rede que pode ser utilizada.

A métrica *disponibilidade* indica a estabilidade da comunicação entre utilizadores. É um valor entre 0 e 1 e é dado pela expressão:

$$Disponibilidade = \frac{MTBF}{MTBF + MTTR}$$

Onde o MTBF[1] é o tempo médio entre avarias e MTTR é o tempo médio de reparação de uma avaria.

A *Fiabilidade* é a probabilidade de um sistema não falhar durante algum tempo. É normalmente expressa no tempo médio entre falhas(MTBF). Responde às perguntas:qual a probabilidade de um componente de um sistema falhar, qual o efeito dessa avaria.

[1] MTBF : "Mean Time between Failures" e MTTR: "Mean Time to Repair"

Além destes três critérios, desempenho, fiabilidade e disponibilidade existem outros dois que convém ter em atenção quando se pretende analisar a operacionalidade de um sistema: o custo e a segurança[1].

Custo

Quando se cria um sistema, faz-se um inventário de todo o material necessário, tanto a nivel de hardware como de software. Um sistema tem sempre um custo: custo do equipamento, do software, da manutênção.
A relação custo-desempenho acompanha sempre a escolha de um determinado sistema. Normalmente, para se aumentar a desempenho aumenta-se o custo.

Segurança

Um sistema existe para servir aqueles que o utilizam. Só os utilizadores autorizados devem ter acesso ao sistema. Assim, devem-se criar barreiras para os não autorizados. O critério da segurança indica a capacidade de um sistema reconhecer quem pode aceder ao sistema e como este reage a uma tentativa de utilização não autorizada.

Dependendo da aplicação, estes critérios terão pesos diferentes, no entanto, a relação custo-desempenho acompanha sempre a escolha de um determinado sistema. Por exemplo, para uma empresa competitiva, onde um dos objectivos é manter uma relação custo-produção baixo, concerteza que a rede a utilizar terá também de ter custos baixos. Numa empresa em que a segurança da informação é vital, a rede terá de incorporar o critério de alta segurança.

5.3 Critérios de desempenho de uma RLI

Numa RLI, o serviço mais solicitado é o de transporte de pacotes de informação para um determinado destino. A informação é transmitida através de um meio de transmissão, que pode ser em forma de barramento ou anel. Como já se disse anteriormente, terá de haver um protocolo entre as várias entidades ligadas ao meio, que estabeleça as regras de acesso ao meio de transmissão. Muito dificilmente, um protocolo consegue fazer a gestão da rede de uma forma ideal, isto é, não é possivel utilizar a capacidade da rede só para transmissão de informação útil entre as entidades. Um protocolo de acesso ao meio, precisa sempre de introduzir informação, para seu próprio uso, nos pacotes de informação que circulam na rede, introduz sempre tempos de acesso ao meio o que implica atrasos nas chegadas dos pacotes. Interessa pois, no projecto de uma RLI, fazer o estudo de desempenho do protocolo de acesso ao meio de forma a poder fazer uma estimativa de quais as modificações a fazer na rede: velocidade de transmissão, comprimento da rede, comprimento das mensagens.
Numa RLI, os principais critérios de desempenho são: o tempo de resposta, a taxa de utilização e o rendimento da rede[3].

Se um pacote chega correctamente ao destino, interessa-nos saber qual o **atraso na transferencia** (ou o tempo de resposta) que exprime o tempo que vai desde que uma estação recebe um pacote e a recepção da resposta desta pela estação destinatária.
A taxa de utilização da rede exprime a relação entre a quantidade de informação que circula no meio(por segundo) e o débito da rede. Se tivermos o caso de uma taxa de utilização baixa pode significar que ou a rede é pouco solicitada pelas estações ou que a técnica de acesso à rede utiliza mal a capacidade da rede.
Um pacote que é transmitido numa RLI é constituido por uma informação útil e uma informação adicional. Esta informação adicional pretende controlar se houve ou não erros na transmissão ou se o pacote chegou ao destino correcto. Assim, um outro parâmetro utilizado numa RLI é **o rendimento** do protocolo utilizado, que exprime a relação entre a quantidade de informação útil (sem ter em atenção os pacotes de

informação de serviço) e a quantidade total de informação veiculado pela rede. Este critério é também utilizado para estimar a eficácia da estrutura dos pacotes de um protocolo de acesso.

5.4 Resultados obtidos

Como já se disse anteriormente, consoante o tipo de aplicação e os objectivos a atingir, escolhem-se os critérios mais convenientes. Neste trabalho e dado o objectivo proposto, isto é, analisar como a camada de ligação responde às solicitações da camada de aplicação, escolheram-se os seguintes critérios:

- Taxa de utilização da rede por cada ciclo elementar.
- Atraso na transferência, isto é, o tempo que demora um pedido realizado pela camada de aplicação, a ser atendido pela camada de ligação de dados.
- E a probabilidade de um pedido não ser atendido, isto é, um pedido ser anulado por um outro.

Assim, foram criados oito cenários diferentes, a partir dos quais se executaram oito simulações. Cada cenário corresponde a uma tabela de "prodcons" e à respectiva tabela de escrutação. A tabela "prodcons" indica-nos quais são as variaveis que circulam no barramento, e para cada variável, qual o seu tempo de escrutação e a que tipo de pedido está associada. A tabela de escrutação é calculada a partir da tabela anterior e indica quantos ciclos elementares existem num macro ciclo e por cada ciclo elementar (EC), quais são as variáveis que são escrutinadas. O gráfico de simulação resultante representa a taxa de utilização do barramento durante um macro ciclo. O número de variáveis considerado é sempre igual a quatro, apresentando-se em seguida os resultados das oito simulações.

Simulação nº1

Para primeiro cenário escolheu-se a seguinte configuração para a rede de acordo com a tabela "prodcons" (tabela 5.1) e a respectiva tabela de escrutação (tabela 5.2) que se apresentam em seguida. Neste caso considerou-se que nenhuma das quatro variáveis está associada a um pedido.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	RP_DAT	1	134	rien	1	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	rien	2	1	p
4	RP_DAT	5	166	rien	1	3	p

Tabela 5.1 - Tabela "prodcons1"

	EC1	EC2	EC3	EC4
Identificador variável	1	1	1	1
	4	4	4	4
		3		3
				2

Tabela 5.2 - Tabela de escrutação 1

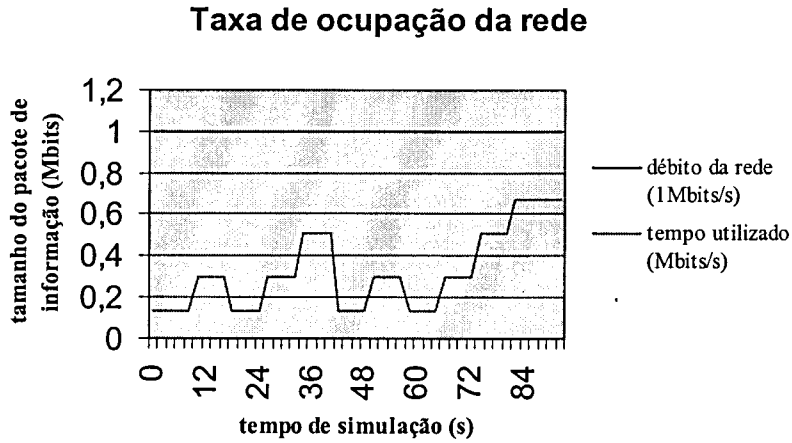


Gráfico 5.1 - Taxa de ocupação da rede durante um macro ciclo

A simulação demorou cerca de 100s e a taxa de ocupação da rede foi de 62%.

Simulação nº2

A segunda simulação foi realizada com base nas tabelas 5.3 e 5.4. Em relação ao caso anterior, alterou-se o tempo de escrutação das variáveis, resultando, por isso, uma tabela de escrutação diferente.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	RP_DAT	1	134	rien	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	rien	2	1	p
4	RP_DAT	5	166	rien	1	3	p

Tabela 5.3 - tabela "prodcons2"

	EC1	EC2	EC3	EC4
Identificador variável	4	4	4	4
		1		1
		3		3
				2

Tabela 5.4 - tabela de escrutação 2

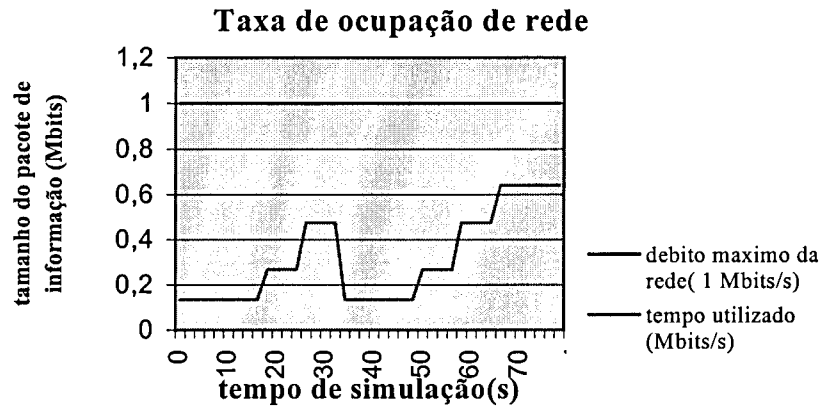


Gráfico 5.2 - Taxa de ocupação da rede durante um macro ciclo

Neste caso a simulação demorou menos tempo relativamente ao caso anterior, cerca de 30 s, pois tem menos variáveis para escrutinar. A taxa de utilização do

barramento foi de 67,2%.

Simulação nº3

A terceira simulação já pressupõe que a variável 3 está associada a um pedido específico periódico. Mantém-se, por isso, a tabela de escrutação e a geração de pedidos está programada para começar aos 20s com intervalos de 60s.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	RP_DAT	1	134	rien	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	B_req	10	206	rien	2	1	p
4	RP_DAT	5	166	rien	1	3	p

Tabela 5.5 - tabela "prodcons3"

	EC1	EC2	EC3	EC4
Identificador variável	4	4	4	4
		1		1
		3		3
				2

Tabela 5.6 - tabela de escrutação 3

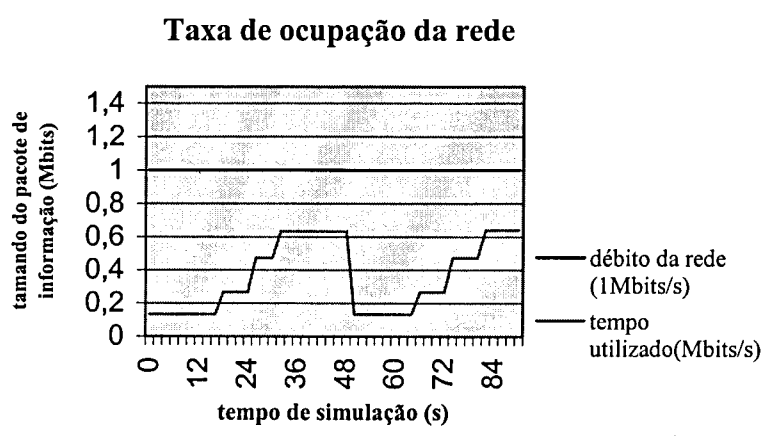


Gráfico 5.3 - Taxa de ocupação da rede durante um macro ciclo

Neste caso, o primeiro pedido associado à variável 3 surge aos 20s. quando o AB

interroga a variável (3) no EC2, é informado que já houve um pedido que tem de ser atendido. Demora cerca de 16s atender o pedido (32-48), voltando depois à tabela de escrutação. O segundo pedido surge aos 80s, já depois da variável (3) ter sido interrogada (EC4 - 74s) o que implica que este pedido não seja atendido.

Simulação nº4

Para a quarta simulação, a variável (3) ficou também associada ao pedido especifico periódico, alterando-se a sua periocidade. O primeiro pedido é gerado aos 20s, o segundo aos 50s e assim sucessivamente. As duas tabelas, "prodcons" e escrutação" permanecem as idênticas ao caso anterior, e o gráfico resultante é o seguinte:

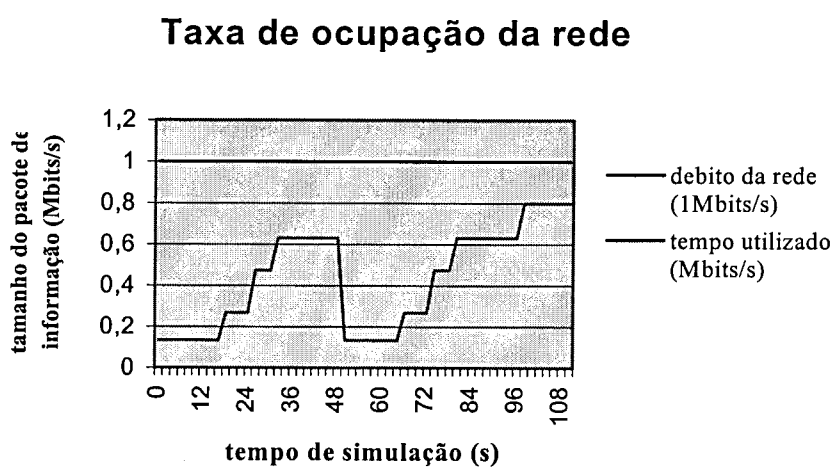


Gráfico 5.4 - Taxa de ocupação da rede durante um macro ciclo

Relativamente ao caso anterior, temos intervalos de tempos para a geração de pedidos de 30s. Assim, até ao ciclo elementar 3 o comportamento da rede é idêntico ao anterior. Entretanto, aos 50s, surge um outro pedido, quando o AB está a interrogar a variável (4) no ciclo elementar 4 (EC4). Este fica em lista de espera até o AB interrogar nocvamente a variável (3). Quando isto acontece, aos 82s, o pedido que vai ser atendido é o que surgiu aos 80s e não aos 50s, anulando-se assim este pedido. O AB volta à tabela de escrutação, para interrogar a variável (2).

Simulação nº5

Na quinta simulação, a variável (3) ficou associada a um pedido especifico aperiódico que surge aos 20s e tem um intervalo de 1mn. Teremos, então, a seguinte configuração dada pela tabela 5.8, mantendo-se a mesma tabela de escrutação.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	RP_DAT	1	134	rien	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	spec	2	1	p
4	RP_DAT	5	166	rien	1	3	p

Tabela 5.8 - tabela "prodcons4"

Resultando o seguinte gráfico:

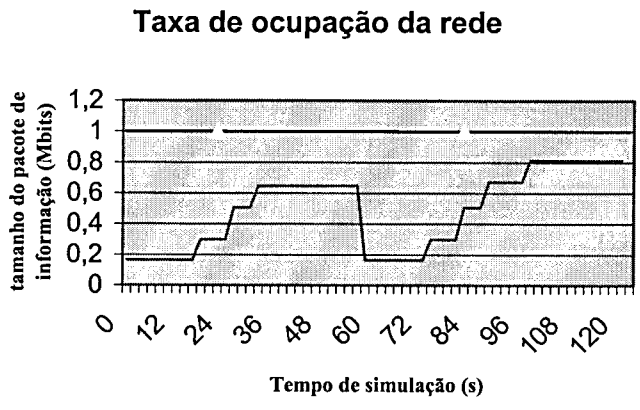


Gráfico 5 - taxa de ocupação da rede durante um macro ciclo

O primeiro pedido é gerado aos 20s (assinalado na figura pelo triângulo amarelo) e será atendido no ciclo elementar EC2 demorando cerca de 16s (32-58). O segundo pedido surge aos 80s quando o AB está a interrogar a variável (4). Só no fim do ciclo elementar 4 é que o pedido vai ser atendido (102-120s).

Simulação nº 6

Para esta simulação, considerou-se o caso de duas variáveis (1) e (4) estarem associadas a pedidos especificos periódicos. A tabela de escrutação mantém-se, alterando-se a tabela "prodcons". Os primeiros pedidos são feitos aos 20s e têm um intervalo de 1mn.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	B_req	1	134	rien	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	rien	2	1	p
4	B_req	5	166	rien	1	3	p

Tabela 5.12 - tabela "prodcons6"

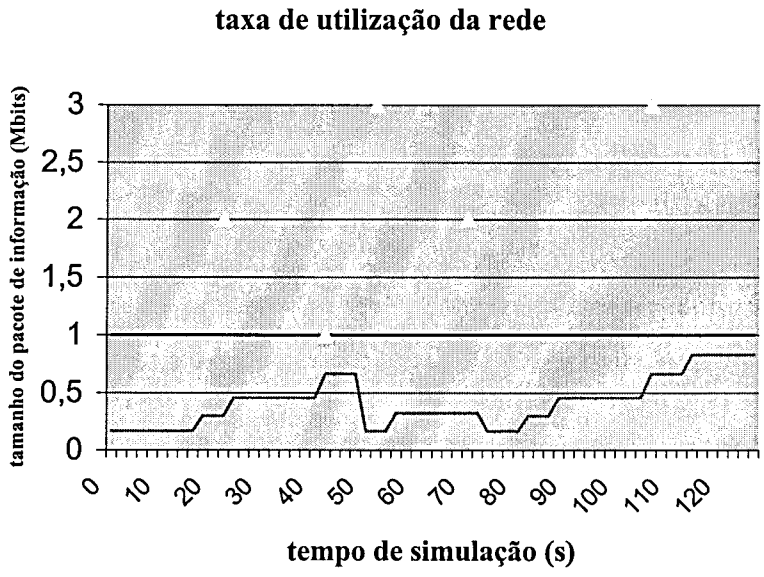


Grafico 5.6 - Taxa de utilização da rede durante um macro ciclo

São gerados dois pedidos aos 20s (assinalados no gráfico por um triângulo), um associado à variável (1) e outra à variável (4). Aos 42s este número de pedidos decresce de uma unidade quando o pedido associado à variável (1) acaba de ser atendido, aumentando para três o número de pedidos aos 60s. No fim de atender o pedido relativo à variável 4 (76s) diminui para dois. Neste caso, também há pedidos

anulados: o primeiro pedido associado à variável (4) é anulado pelo segundo.

Simulação nº7

Para esta simulação considerou-se a seguinte configuração:

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	nºestacao	p/c
1	RP_DAT	1	134	rien	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	spec	2	1	p
4	RP_DAT	5	166	spec	1	3	p

Table 5.13 - tabela "prodcons7"

Os pedidos são gerados aos 20s e têm um intervalo de 1mn. A tabela de escrutação permanece inalterada.

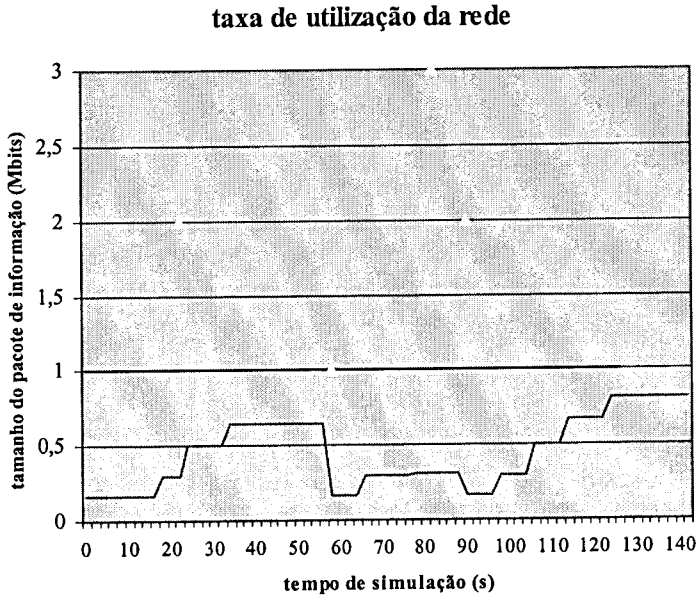


Gráfico 7 - taxa de ocupação da rede durante um macro ciclo

Relativamente ao caso anterior, verifica-se que os pedidos são gerados com a mesma frequência alterando-se a sequência de atendimento dos pedidos dado que estes pedidos só são atendidos no fim de cada ciclo elementar. A taxa de utilzação da rede aumentou para 82%.

Simulação n°8

A configuração utilizada é dada na tabela 5.15, em que a variável (3) está associada a um pedido especifico aperiódico e a variável (4) a um pedido específico periódico.

var	tipo_resp.	tamanho (n)	Tcyc per.	tipo pedido	tempescr	n°estacao	p/c
1	RP_DAT	1	134	spec	2	2	p
2	RP_DAT	5	166	rien	4	2	p
3	RP_DAT	10	206	rien	2	1	p
4	B_req	5	166	rien	1	3	p

Tabela 5.14 - tabela "prodcons8"

Os pedidos, associados à variável (1) começam a ser gerados aos 20s e têm uma periocidade de 1mn e os pedidos associados à variável (4) começam aos 20s e têm uma periocidade de 30s, resultando o seguinte gráfico:

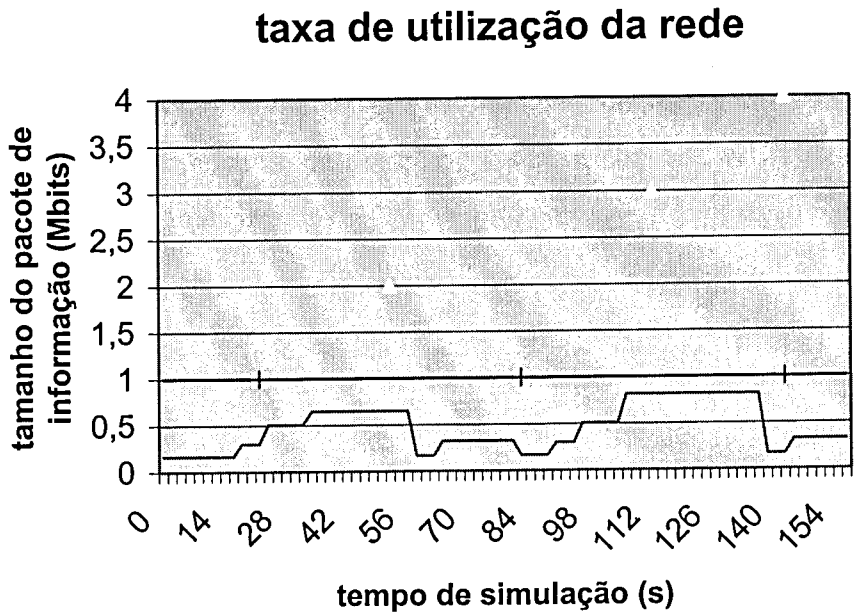


Gráfico 5.8 - Taxa de ocupação da rede durante um macro ciclo

O primeiro pedido a ser atendido é o associado à variável (1) que começa aos

36s e acaba aos 58s (Os pedidos periódicos são assinalados na figura por um triângulo e os aperiódicos por uma cruz e em alguns pontos estão sobrepostos). O primeiro pedido para a variável (4) que surgiu aos 20s é anulado pelo que aparece aos 60s que, é atendido aos 64s demorando 12s. O tempo de simulação foi 156s e a taxa de utilização foi de 82%.

Conclusão

A partir dos resultados obtidos destas simulações, chega-se à conclusão que este barramento pode suportar um maior número de variáveis, dado que, a taxa de utilização da rede foi variando entre 60% a 80%.

Outra conclusão, é que a frequência de geração de pedidos é um factor que contribui para o aumento da taxa de utilização da rede. Quanto maior for o número de pedidos gerados maior é o tamanho da informação que circula na rede. No exemplo 3 a taxa de ocupação do barramento era de 62% passando para 80% no exemplo 4. Outra consequência do aumento da frequência dos pedidos gerados é o aumento do número de pedidos que são anulados. No exemplo 3, a frequência de geração de pedidos era de 1mn e não houve nenhum pedido anulado; passou-se para 30s e houve um pedido anulado (ver exemplos 3 e 4;8).

Outra conclusão a que se chega é que o tempo de espera de um pedido para ser atendido, depende essencialmente da posição onde se encontra o AB na tabela de escrutação, quando surgem os pedidos. Por exemplo, no caso da simulação 3 o pedido que surge aos 20s é atendido aos 32s, tendo esperado 12s para ser atendido.

Os factores principais que influenciam a taxa de utilização da rede são, por um lado, o número de variáveis e o seu tempo de escrutação, e por outro, os tipos de pedidos associados a cada variável e a sua frequência. Quando, para uma dada configuração da rede se obtém uma tabela de escrutação com alguns ciclos elementares completos, corre-se o risco de ultrapassar o débito tolerado pela rede. Para evitar que isto aconteça, transforma-se a tabela de escrutação, distribuindo-se as variáveis pelos ciclos elementares por forma a torná-la mais uniforme.

5.5 Referências Bibliográficas

[1] Harrell Van Norman; LAN/WAN Optimization Techniques; 1992 Artech House, INC.

[2]Raj Jain; The Art of Computer Systems Performance Analysis; 1991 by John Wiley & Sons, INC.

[3] Song, Jé-Qiong; Évaluation de Performances, Chapitre 5; Communication Industrielle, Cours INPL 13/14 Decembre, C.R.I.N.

Capítulo 6 - Conclusões e Trabalho Futuro

6.1 Conclusões

O objectivo do trabalho era de demonstrar a possibilidade de utilizar uma linguagem de simulação geral, o Simple++, para modelar e estudar o comportamento de barramentos de campo.

A utilização de uma linguagem Simple++ proporciona todas as vantagens inerentes a uma linguagem de programação por objectos que se sobrepõem às que uma linguagem tradicional pode proporcionar. Uma das vantagens, é o facto de toda a estrutura subjacente à simulação estar escondida do modelador de tal modo que este só tem que se preocupar com a criação do modelo e com a escrita do programa. Além disso, qualquer alteração que tenha de se introduzir no código é executado com relativa facilidade. Relativamente aos pacotes de simulação, a linguagem Simple++ tem a vantagem de permitir a criação de qualquer tipo de modelo, correpondendo ao desejo do modelador.

O Simple++ é uma linguagem de simulação orientada ao objecto que permite a criação de modelos e respectiva simulação, sendo frequentemente aplicada em linhas de produção ou em sistemas de fabrico completo.

Pareceu assim interessante recorrer às potencialidades do Simple++ para as utilizar na modelição e simulação de um barramento de campo, por forma a evitar a recorrência a simuladores de redes.

O Simple++ é uma linguagem de simulação que permite a construção, por parte do modelador, de classes de objectos que juntamente com os objectos pertencentes à bibiloteca do Simple++, possibilita a construção de um modelo particular.

O barramento de campo modelizado neste trabalho foi o WorldFIP. Assim, construíu-se uma classe de objectos no Simple++, denominada FIP, que pode ser utilizada por qualquer pessoa interessada no estudo do comportamento do WorldFIP. O FIP mostrou ser um modelo fácil de implementar, permitindo a qualquer utilizador, configurar o barramento da forma que desejar e, através de repetidas simulações, estudar o seu comportamento.

A visualização dinâmica durante a simulação do modelo e os resultados obtidos demostraram que o modelo criado corresponde às especificações do funcionamento do WorldFIP, podendo concluir-se que o objectivo deste trabalho foi alcançado.

Além do WorldFIP, o Simple ++ permite a modelização de outros tipos de barramentos, como por exemplo o CAN, pois a filosofia de construção do modelo é idêntica, havendo no entanto, a necessidade de definição das relações lógicas existentes neste barramento.

6.2 Trabalho Futuro

As propostas para a continuação deste trabalho são várias e que passo a descrever.

Uma das propostas é a de considerar a criação de uma nova classe de objectos, a que se chamaria "PC", que englobaria as classes "cligacao" e "camapli". Esta alteração tornaria o modelo FIP mais próximo da realidade, dado que os componentes do barramento Wordfip são o árbitro de barramento, o próprio barramento e as unidades produtoras/consumidoras. Embora o funcionamento não se altere, gráficamente há uma representação mais realista.

Outra proposta é a de inserir no modelo actual os serviços de pedidos livres e, eventualmente, o serviço de mensagens que o WordFIP suporta.

Uma outra proposta, para a qual este trabalho se pode direcionar, é a de colocar o Simple++ a comunicar com uma carta WorldFIP dado que existem cartas electrónicas WorldFIP que, quando inseridas num PC, permitem o envio e recepção de mensagens. Nesta situação, o PC ou o Simple++ podem ficar a funcionar como árbitro de barramento, gerindo as outras unidades, que funcionarão como unidades produtoras/consumidoras de informação tais como sensores e unidades de comando.

Para finalizar, outra hipótese é de responsabilizar o Simple++ pela monitorização de um barramento de campo WorldFIP.

7 Referências Bibliográficas

Para o primeiro capítulo

[1.1] Harrell Van Norman.1992. *LAN/WAN Optimization Techniques*. Artech House.

[1.2] Daniel T.Miklovic. 1993. *Real Time Control Networks*. Instrument Society of America.

[1.3] Pierre Rolin. 1993. *Réseaux locaux, normes et protocoles*. Hermes.

[1.4] Luís Manuel Gouveia. "Comunicação de dados". Versão 1.1. http://www.ufp.pt/staf/lmbg/textos/norma_osi.html. Outubro de 1997,

[1.5] Zoubir Mammeri et Jean Pierre Thomesse. 1994. *Réseaux locaux industriels, FIP et MAP dans les systémes automatisés*. Eyrolles.

[1.6] W.Gustschke and K.Mertins CIM. 1987 "Competitive Edge in Manufacturing". Science & Computer Integrated Manufacturing, Vol 3, N°1.

[1.7] Pascal Lorenz. 1994. "Le temps dans les arquitectures de communication: apllication au réseau FIP". Tese de doutoramento.

[1.8] L.Almeida, M.L.Chavez, J.A. Fonseca e J.P.Thomesse. 1999. "Real time communications in manufacturing".

[1.9] Jean Pierre Thomesse; "Le Réseau de Terrain FIP"; Ensem.

[1.10] "FIP Technical Description".

[1.11] "Distributed Real-Time Systems". http://www.mecel.se/html/body_distributed_real_time_systems.html.

[1.12] Jean Luc Delcuvellerie. "Ingeniere des systemes d'automatisation de production, Place des RLI dans les Arquitectures SAP".

[1.13] "FIP demo System". http://icat.snu.ac.kr:3333/tecnology_e/fielddem.html

[1.14] "Une solution rentable dès la phase d'étude". Le guide dáchat - Les résaux de terrain.

Para o segundo capítulo

[2.1] "Réseaux généralistes: l'utilisateur doit choisir entre deus mondes" - Guide d'achat- Lés réseaux de terrain.

[2.2] L.Almeida, M.L.Chavez, J.A. Fonseca e J.P.Thomesse. 1999. "Real Time Communications In Manufacturing".

[2.3] Ghassan SABA.1996. "Protocoles Multipoint et interconexion de réseaux FIP". Tese de Doutoramento.

[2.4] Jean Pierre Thomesse. "Le Réseau de terrain FIP".Ensem. Nancy.

[2.5] Thierry LAINE."Le Bus de Terrain FIP". CEGELEC.

[2.6] Philippe Leterrier. "Le FIP Protocole". Centre de Compétence FIP.

[2.7] Normalisation française - C46-604, 1988.

Para o terceiro capítulo

[3.1] Raj Jai. 1991.*The Art of Computer Systems Performance Analysis Tecniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley & Scons,INC.

[3.2] Cristoph Lindemann. 1998. *Performance Modelling with Deterministic and Stochastic Petri Nets*. John Wiley.

[3.3] Harrell Van Norman. 1992. *LAN/WAN Optimization Techniques*. Artech House, INC.

[3.4] Cristian N.Madu and Chu-hua Kuei. 1993. *Experimental Statistical designs and Analysis in Simulation Modeling*. Quorum Books.

[3.5] Yé Quiong Song."Évaluation de Performances". Communication Industrielle Cours INPL.

[3.6] Jerry Banks, John S. Carson,II. 1984. *Discret Event System Simulation*. Prantice-Hall, INC.

[3.7] William E. Biles. *Discret Event Systems*. University of Louisville, Louisville, Kentucky.

[3.8] Peter Ball. 1996. "Introduction to Discrete Event Simulation". http://www.strach.ac.uk/Departements/DMEM/MSGR/simulate .html.

[3.9] René David and Hassane Alla. 1992. *Petri Nets &Grafcet Tolls for Modelling Discrete Event Systems*. Prentice Hall.

[3.10] Nathalie berge et Guy Juanole. Janeiro 94. "Refraichissement et Promptitude dans le reseau FIP: Modelisation et Evaluation au moyen de reseaux de Petri Temporisés Stochastiques". Actes des Conferénces.

[3.11] Queieng Theory. http://mscmga.ms.ic.ac.uk/jeb/ou/queue.html

[3.12] Yé Quiong Song. "Étude de Performance de FIP; aide au dimensionnemet d'applications".

[3.13] Ian Graham. 1993. *Object Oriented Methods*. Addison, Wesley Publishing Company.

[3.14] David R.C. Hill. 1996. "Object Oriented Analysis and Simulation".

[3.15] Sklenar. J. "Simulation". http://staff.um.edu.mt/jsk11/simul.html.

[3.16] "Introduction - What is Object Oriented Programming?". http://203.230.73.22

[3.17] José Mendonça. 1994. "apontamentos da cadeira de automação industrial do curso de mestrado em informática industrial".

Para o capítulo 4

[4.1] Tecnomatix. "Simple++ Plant and Line Plannig".http://194.90.80.3Products/Simple++.asp;

[4.2] J. J. Pinto Ferreira. "Suporte à Gestão de Operações fabris através de Modelos Executáveis". Tese de Doutoramento.

Para o quinto capítulo

[5.1] Harrell Van Norman. 1992. *LAN/WAN Optimization Techniques*. Artech House, INC.

[5.2]Raj Jain. 1991. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, INC.

[5.3] Song, Jé-Qiong. "Évaluation de Performances, Chapitre 5; Communication Industrielle". Cours INPL, C.R.I.N.