

Pontifícia Universidade Católica do Paraná



ROBERT CARLISLE BURNETT

**APLICABILIDADE DA AVALIAÇÃO DE
PERFORMANCE PARA REDES LOCAIS
EM AMBIENTE INDUSTRIAL**

Trabalho científico apresentado como
requisito parcial para o concurso à
classe de professor titular da
Pontifícia Universidade Católica do
Paraná.

Curitiba

Outubro de 1996

Pontifícia Universidade Católica do Paraná

ROBERT CARLISLE BURNETT

APLICABILIDADE DA AVALIAÇÃO DE PERFORMANCE PARA REDES LOCAIS EM AMBIENTE INDUSTRIAL

Trabalho científico aprovado como requisito parcial para o concurso à classe de professor titular da Pontifícia Universidade Católica do Paraná, pela comissão formada pelos professores:

Prof. Dr. Flávio Bortolozzi
Departamento de Informática, PUC-PR

Prof. Dr. Celso Antônio Alves Kaestner
Departamento de Informática, PUC-PR

Prof. Romualdo Wandresen
Departamento de Matemática e Física, PUC-PR

Curitiba, 30 de outubro de 1996

Dedico este trabalho à Ana Lúcia, minha esposa e amiga, e nossa doce filha Eliana, as quais são a razão e dão sentido a minha vida.

Para minha mãe Osmarina e meu pai George (em memória).

*Para aqueles que procuram tornar este mundo
um lugar melhor e mais justo para todos.*

Agradeço à Deus, pois nas muitas ocasiões que tenho recorrido a Ele, sempre tenho encontrado conforto espiritual e, de uma forma ou outra, resposta para minhas dúvidas.

SUMÁRIO

LISTA DE FIGURAS	VII
RESUMO	VIII
ABSTRACT	VIII
1 INTRODUÇÃO	1
2 METODOLOGIAS PARA AVALIAÇÃO DE PERFORMANCE	8
2.1 ANALÍTICAS	8
2.1.1 TEORIA DAS PROBABILIDADES	8
2.1.2 ESTATÍSTICAS	14
2.1.3 TEORIA DAS FILAS	17
2.1.4 REDES DE PETRI	21
2.2 SIMULAÇÃO	31
2.3 MEDIÇÃO	42
3 PARÂMETROS PARA SISTEMAS EM TEMPO REAL	53
4 AMBIENTE PARA AVALIAÇÃO DE LANs	58
5 CONCLUSÕES	64
ANEXO 1 : Simulação de linha multiponto	67
ANEXO 2: Protocolo MMFS	82
GLOSSÁRIO	94
BIBLIOGRAFIA	96

LISTA DE FIGURAS

Fig. 2.1	Modelo simples de protocolo com Redes de Petri	24
Fig. 2.2	Tempos de habilitação e atuação das transições	25
Fig. 2.3	Modelo da figura 2.1 para cálculo de desempenho	28
Fig. 2.4	Fluxo para elaboração de um simulador	34
Fig. 2.5	Comportamento da simulação	41
Fig. 2.6	Ambiente para monitoramento	44

RESUMO

Este trabalho apresenta um estudo sobre as metodologias utilizadas para avaliação de performance de redes locais de computadores, contendo considerações sobre a aplicabilidade para estas redes em ambiente industrial.

Uma proposta de ambiente para avaliação de performance é feita, tendo em vista as várias metodologias analisadas.

ABSTRACT

This work shows a survey on methodologies for performance evaluation of the local area networks, with approach for industrial networks. A proposal for evaluating this kind of network is also included.

Capítulo 1 - Introdução

1.0 - INTRODUÇÃO

Com a aceitação cada vez maior do modelo *Open Systems Interconnection* (OSI) pela comunidade envolvida com comunicação de dados, houve um natural esforço de instituições para uma busca de conhecimento maior sobre este modelo, objetivando a adaptação às suas necessidades.

Como resultado deste esforço, a *General Motors* (GM) propôs o padrão *Manufacturing Automation Protocol* (MAP). MAP é representado por um subconjunto do modelo OSI e padronizações do IEEE e é voltado para a interconexão de equipamentos em um ambiente industrial, contendo aspectos que vão desde a ligação física, até a aplicação do usuário (em [GEN86], tem-se a especificação técnica que deve ser seguida para a implementação do MAP).

Paralelamente com o advento do modelo OSI, ocorreu um incremento na criação de métodos, que possibilitassem uma validação dos protocolos envolvidos, tanto do ponto de vista de correção dos procedimentos, como de performance no ambiente. Como qualquer padrão ou regra, vários itens de controle e verificação foram inseridos com a padronização, acarretando em alguns casos, sobrecarga de mensagens, e um *overhead* para todo o sistema.

Para possibilitar este conhecimento do ambiente, existem medidas de performance, com as quais pretende-se conhecer o comportamento de uma

Local Area Network (LAN) quanto a, p.ex., tempo de resposta, capacidade de tráfego ou *throughput* (vazão do sistema), segurança da rede, taxa de erros na transmissão e sensibilidade a variações no nível de tráfego. Estes tópicos são, normalmente, os pontos que marcam a rede, pois permitem identificar se o ambiente está dentro dos limites desejáveis para o sistema.

As LANs em ambiente industrial possuem características especiais, pois integridade, detecção de erros e considerações sobre os tempos de transmissão, assumem importância fundamental, o que obriga a conhecer-se o comportamento da performance da LAN com precisão. Com a análise destas características pretende-se identificar possíveis parâmetros inadequados para o sistema, e que tipos de ajustes são necessários.

Para um detalhamento um pouco maior na análise de performance, pode-se considerar as medidas como capazes de dar respostas para os seguintes pontos:

a) confiabilidade mínima

- máxima paralização tolerável;
- máximo *Mean Time Between Failure* (MTBF) e mínimo *Mean Time to Repair* (MTTR), que levam ao cálculo da disponibilidade da rede, como sendo a relação entre $MTBF / (MTBF + MTTR)$,

representando o percentual de tempo que o usuário tem a LAN disponível;

b) capacidade e tempo de resposta

- volume de mensagens na hora de maior movimento, tempo de resposta esperado.

c) nível de acesso

- tempo de espera pelo atendimento.

d) taxa de erros tolerável

- *Throughput* dos dados, taxa de bits errados

No presente trabalho são analisadas às metodologias que possibilitam uma avaliação de performance de protocolos, abordando tanto as técnicas puramente analíticas, onde usa-se métodos matemáticos como teoria das filas, para determinar os vários itens de comportamento do sistema, bem como as ferramentas de simulação e medição em uma LAN real. Cabe ressaltar que esta última é a técnica mais segura, pois a situação real está sendo estudada.

Baseado nesta avaliação é feita uma proposta para análise de performance de LANs em ambiente industrial.

Os métodos de cálculo propostos em [KLE75], [MAR82], [REI82] e [GIO86], a técnica de simulação em [GIO86], [JAL94], [STR82] e [SCH78], bem como a criação de um ambiente para medição, descrito em [MIL84], [AME83], e [NAT85], são vistos e analisados para a situação onde existe uma LAN instalada para controle de processos industriais.

A estrutura do trabalho está assim disposta:

- a) no capítulo 2 as ferramentas para avaliação de desempenho são apresentadas, com exemplos e considerações sobre a aplicação em LANs;
- b) no capítulo 3 os parâmetros que são relevantes numa LAN para controle de processos industriais, são discutidos;
- c) no capítulo 4 tem-se uma proposta para avaliação de um ambiente discutido na neste capítulo, com uma diretriz para o projetista seguir, na análise de performance;
- d) o anexo 1 contém a listagem do programa utilizado para a simulação de uma linha de comunicação;

e) o anexo 2 apresenta o protocolo *Manufacturing Messaging Format Standard* (MMFS), que é o de mais alto nível no MAP, sendo utilizado pelo usuário para comunicação com os processos.

Capítulo 2 - Metodologias para Avaliação de Performance

2.0 - METODOLOGIAS PARA AVALIAÇÃO DE PERFORMANCE

Com este capítulo objetiva-se apresentar as ferramentas analíticas ou matemáticas, utilizadas em avaliação de desempenho de redes de computadores, de uma maneira objetiva, servindo de embasamento para os capítulos posteriores.

2.1 ANALÍTICAS

2.1.1 Teoria das probabilidades

Esta teoria trata com eventos aleatórios, que apresentem um comportamento conhecido como "regularidade estatística". Alguns sistemas possuem tal característica, o que permite, através de utilização dos conceitos de teoria das probabilidades, a determinação de tendências e prováveis comportamentos.

A seguir são apresentados conceitos da teoria, que demonstram a potencialidade da metodologia:

a) probabilidade condicional

Para determinação da probabilidade de um determinado evento A acontecer, tendo em vista que um evento B ocorreu, e sendo A e B

dependentes, com $p(AB)$ a probabilidade de A e B acontecerem ao mesmo tempo, tem-se:

$$p(A|B) = \frac{p(AB)}{p(B)}, \text{ sendo } p(B) > 0 \quad (2.1)$$

Se os eventos são independentes, a $p(AB) = p(A) * p(B)$, o que acarreta em (2.1) $p(A|B) = p(A)$, como era de se esperar, pois a $p(A|B)$, será independente se B ocorreu ou não.

b) probabilidade total

Seja um evento B a união de vários eventos A_i , independentes e mutuamente exclusivos, desejando-se calcular a probabilidade total de B , em função da ocorrência dos vários eventos A_i .

$$p(B) = p(B|A_i) * p(A_i) \quad (2.2)$$

c) teorema de Bayes

Com a manipulação de (2.1) e (2.2) chega-se ao teorema de Bayes, que fica então

$$p(A_i|B) = \frac{p(B|A_i) * p(A_i)}{p(B|A_j) * p(A_j)} \quad (2.3)$$

O exemplo citado em [GIO86], páginas 224-225, fornece uma boa ilustração da aplicação de teoria das probabilidades na avaliação de redes, e será reproduzido a seguir.

Tem-se uma LAN na qual são processadas duas aplicações de usuários. Uma das aplicações gera pequenos pacotes para a sub-rede de comunicação (tráfego iterativo ou classe 1), enquanto a outra gera longos pacotes (tráfego proveniente de transferência de arquivos ou classe 2). A mistura do tráfego na sub-rede de comunicação é 2/3 e 1/3 de tráfego classe 1 e 2, respectivamente. Examinando o tráfego na rede, verificou-se que de todos os pacotes da classe 1 enviados, 10% são destinados ao único servidor de arquivos. Para os pacotes de classe 2, 60% são destinados a este servidor.

Pergunta:

Dado que chega um pacote ao servidor de arquivo, qual a probabilidade dele ser classe 2?

Eventos:

A : pacote é classe 1

B : pacote é classe 2

C : chega um pacote ao servidor de arquivo

Do problema tem-se:

$$p(A) = \frac{2}{3} \quad p(B) = \frac{1}{3} \quad p(C|A) = \frac{1}{10} \quad p(C|B) = \frac{6}{10}$$

Pretende-se determinar $p(B|C)$.

Da equação (2.1), tem-se:

$$p(B|C) = \frac{p(BC)}{p(C)} \quad (2.4)$$

Da equação (2.2), tem-se:

$$p(C) = p(AC) + p(BC) \quad (2.5)$$

Manipulando-se (2.4) e (2.5), tem-se o teorema de Bayes, ou seja,

$$p(B|C) = \frac{p(C|B) * p(B)}{p(AC) + p(BC)} \quad (2.6)$$

Aplicando-se novamente (2.1), tem-se:

$$p(AC) = p(C|A) * p(A) = \frac{1}{15}$$

$$p(BC) = p(C|B) * p(B) = \frac{3}{15}$$

Inserindo-se estes 2 valores em (2.6). tem-se o valor para $p(B|C) = 0,75$, ou seja, a probabilidade é de 75% de que seja classe 2, um pacote que chegue ao servidor de arquivos.

O exemplo serve como indicação de utilização da teoria da probabilidade na avaliação, embora considerando que não seja uma ferramenta adequada para medição, pois não possibilita a inclusão de diversos parâmetros, como p. ex, a situação de carga da rede e prioridade de processos, sendo entretanto bastante útil para uma análise preliminar.

Para a utilização desta ferramenta, é necessária a modelagem cada ambiente, definindo-se as fórmulas, conforme a análise desejada.

Um exemplo de modelagem é apresentada em [FAY78], onde para um protocolo simples de enviar uma mensagem, ou seja, aguardar um *Acknowledgment* - ACK, e então enviar nova mensagem, são feitas análise de probabilidade do comportamento do sistema, podendo-se notar a aplicabilidade desta ferramenta.

Quanto a aplicação da teoria das probabilidades para avaliação de performance de uma LAN em ambiente industrial, não considera-se adequada, pois o acontecimento dos eventos neste ambiente, não obedece necessariamente a regularidade estatística de ocorrência, que é a base de toda a teoria, prejudicando a análise.

Deve-se observar que o conceito de aleatoriedade para os eventos, também não é válido, para este ambiente industrial, pois, na maioria dos casos, tem-se um determinismo de quais eventos ocorrem e em que ordem estes são executados nos processos, o que acarreta uma situação oposta a de eventos aleatórios.

2.1.2 Estatísticas

A mais tradicional das metodologias para avaliação de performance é a extração de dados estatísticos do comportamento dos vários elementos, sendo estes resultados, via de regra, os utilizados pelos usuários finais para análise. A partir dos valores obtidos pode-se determinar, p.ex., o tempo médio de resposta.

Para que se possa dar uma abordagem mais geral a estas análises, necessita-se de medida de dispersão dos valores obtidos, sendo a mais importante destas medidas o desvio padrão, que, juntamente com os valores médios obtidos permite uma avaliação do comportamento de vários itens. Com a utilização destes elementos tem-se outra ferramenta analítica para avaliação de performance.

A seguir é apresentado um exemplo de estatística de tamanho de mensagem, adaptado de [MAR81], onde a aplicabilidade da metodologia pode ser vista.

Considera-se uma LAN contendo equipamento para controle de processos (ECP), com dois tipos de transações, A e B. Para cada mensagem de entrada gerada pelo usuário, existe uma resposta de ECP para o usuário gerador da mensagem. Nenhum outro tráfego ocorre na LAN enquanto os usuários estão ativos solicitando algo para os ECPs. Os percentuais por tipo de mensagem são de 30% para o tipo A e 70% para o tipo B.

Ambas as mensagens de entrada e saída podem variar de tamanho. Uma análise estatística do diálogo entre os usuários e os ECPs, apresenta o seguinte quadro:

número de caracteres	% de transações	
	tipo A	tipo B
entrada		
abaixo de 6	0,0	0,0
6 - 10	39,0	47,3
11 - 15	21,4	28,3
16 - 20	18,6	11,8
21 - 40	14,0	9,9
41 - 60	7,0	2,7
acima de 60	0,0	0,0
saída		
abaixo de 6	0,0	0,0
6 - 20	2,1	5,5
21 - 40	2,9	14,7
41 - 60	3,8	19,3
61 - 80	5,8	27,8
81 - 100	8,2	18,2
101- 120	15,4	7,2
121- 140	29,8	3,4

	tipo A	tipo B
valor esperado de TME	1,65	1,44
valor esperado de TMS	8,88	5,41
valor esperado de TME (A ou B)	1,51	
valor esperado de TMS (A ou B)	6,45	
valor esperado de tempo de resposta na linha	8,96	
desvio padrão de tempo de resposta na linha	3,00	

O exemplo mostra que a partir dos resultados obtidos, refinamentos no projeto podem ser feitos, pois trabalha-se com dados reais. A aplicação da metodologia em LANs para ambiente industrial deve ser vista dentro do contexto de ajuda no refinamento, e não propriamente como uma avaliação de performance.

2.1.3 Teoria das filas

Um sistema de filas é caracterizado como sendo aquele em que os usuários chegam, esperam pelo atendimento, são atendidos e partem do sistema, sendo composto dos seguintes elementos:

- função probabilística entre tempos de chegada de usuários ao sistema;
- função probabilística do tempo de atendimento que é feito ao usuário;

- número de servidores que fazem o atendimento;
- disciplina da fila, p.ex., o primeiro que chega será o primeiro a ser atendido;
- a quantidade de espaço disponível nas filas para "conter" todos os usuários.

A utilização desta teoria, para análise de uma LAN é bastante aplicável, onde um processo de envio de pacotes de informação através de uma LAN, pode ser descrito pela teoria das filas, pois na rede de comunicação, um canal que presta o serviço de levar a informação, é um servidor, e o pacote para o qual o serviço foi prestado, é o cliente.

Os seguintes elementos são relevantes para a análise:

- tempo de chegada dos pacotes ao canal, pode obedecer uma distribuição X;
- o tempo de atendimento pela rede, obedece a uma distribuição Y;
- a ordem que os pacotes são atendidos pela rede, bem como a permanência dos pacotes no sistema, também obedecem a um comportamento específico.

Nesta metodologia o objetivo básico é possibilitar o cálculo do tamanho médio das filas e tempo médio de atendimento para os "clientes" do serviço, proporcionando uma visão global do desempenho da rede.

As fórmulas sobre esta teoria permitem o cálculo dos itens citados acima bem como os relacionados, como p.ex. , quanto tempo o servidor ficará ocupado, qual a probabilidade de um cliente esperar mais que t unidades de tempo para ser atendido, quantos servidores são necessários para manter um determinado tempo de resposta, etc...

Para exemplificar a utilização da teoria das filas na análise de uma LAN, é apresentado a seguir alguns tipos de perguntas que poderiam ser respondidas:

- Qual a quantidade de mensagens que terão um tempo de resposta maior que X segundos?
- Qual a quantidade necessária de servidores?
- Qual a probabilidade de todos os servidores (p.ex. de arquivos), estarem ocupados quando da requisição pelos usuários?
- Qual o tamanho médio das filas?

- Qual o tempo médio nas filas?
- Qual deveria ser o tempo de atendimento do servidor, para que o tempo médio na fila fosse X segundos?

Na prática tem-se alguns desvios dos valores calculados pela teoria, acarretando uma imprecisão dos resultados, devido aos seguintes fatos:

- uma situação real pode não ser totalmente aleatória, sendo desaconselhável a utilização deste método, se o tráfego não seguir uma distribuição de Poisson ([MAR72] e [KLE75]);
- o resultado real também poderá ser fortemente afetado, se ocorrerem "picos" súbito de tráfego, e/ou os tempos de atendimento dos servidores da LAN ultrapassarem de maneira considerável a média;
- a estrutura de disciplina das filas poderá ser mais complexa do que a estabelecida;

Outro ponto a ser considerado, é que em sistemas reais, retardos são inseridos quando tem-se vários servidores e/ou "clientes", fato este que não aparece nas fórmulas.

Apesar destas restrições pode-se utilizar esta teoria para uma análise preliminar da LAN, servindo os elementos obtidos como uma tendência de comportamento, possibilitando também um refinamento e ajuste dos componentes.

2.1.4 Redes de Petri

As Redes de Petri (RdP) representam outra metodologia, através da qual se obtém informações, que analisadas por modelos analíticos, proporcionam também uma consistente avaliação de desempenho.

As RdP apresentam um modelo formal abstrato de fluxo em sistemas, sendo especialmente aplicável ([HEU87] e [SOA84]) quando a realidade a ser modelada apresenta atividades assíncronas e com paralelismo, ou seja, eventos ocorrendo concorrentemente, independentes uns dos outros, que é a situação de uma LAN para controle de processos.

Uma das grandes qualidades das Redes de Petri, senão a principal, é a de possibilitar uma abordagem teórica com aplicabilidade prática, permitindo um rigor e precisão na especificação, ao contrário de outras metodologias de modelagem que possuem um alto grau de praticidade e quase nenhum formalismo, tendo como consequência a geração de especificações imprecisas.

Com base nesta metodologia e associando-se modelos analíticos para avaliação, pode-se calcular, p.ex., "throughput" do sistema, situações de sub-utilização de recursos, fluxo na rede, condições de "deadlock", ou seja, os dados que exprimem o desempenho. Além disso é possível a execução de várias "simulações" do comportamento com a mudança de parâmetros, obtendo-se a vantagem da precisão que a metodologia fornece.

Não está no escopo deste trabalho a discussão sobre os conceitos de Redes de Petri, mas a sua apresentação como uma metodologia de avaliação de desempenho, recomendando-se ao leitor interessado em conhecer a metodologia, a consulta as referências a seguir:

[PET81] e [REI82a] - são livros-texto, contendo as definições de todos os tipos de RdP, bem como as análises teóricas decorrentes;

[PET77] e [HEU87] - apresentam a metodologia, sem muitas considerações teóricas, servindo bem como uma abordagem inicial;

[RAZ84],[AYA84] e [SOA84] - possuem aplicações em análise de desempenho e especificação de protocolos;

[TAZ85] e [TAZ86] - fundamentação teórica para modelo de análise de desempenho

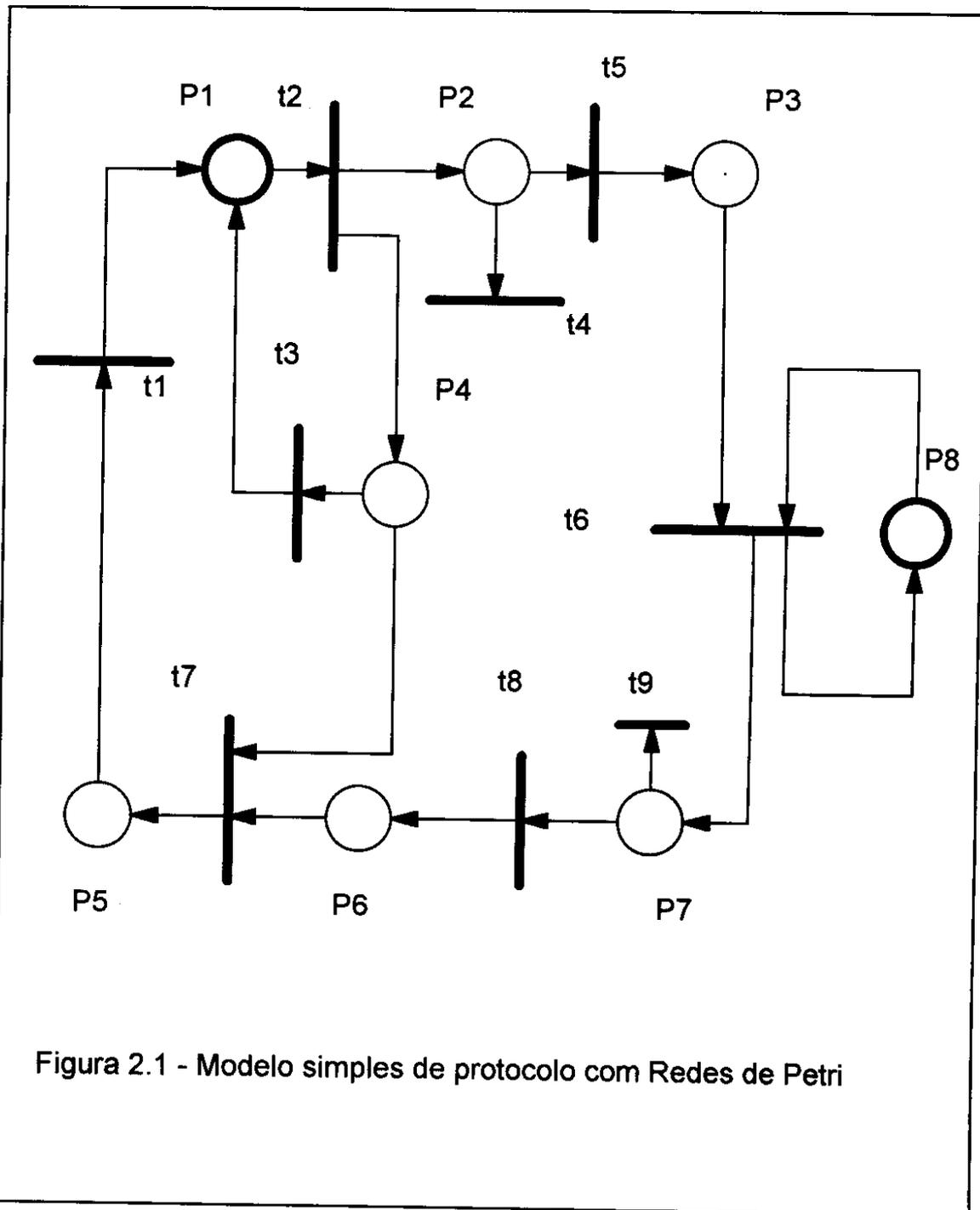
Estas referências cobrem o conhecimento necessário para a análise e avaliação de LANs, possuindo além do mais, uma consistente bibliografia para aprofundamento no assunto.

Para a apresentação da aplicabilidade da metodologia, descreve-se a seguir o exemplo citado em [RAZ84], que por especificar uma situação simples, torna-se menos complexa a modelagem e análise.

Descrição do exemplo:

A análise será feita no modelo simples de protocolo da fig. 2.1, que mostra uma Rede de Petri temporizada, com um particular modelo de rede, baseado num protocolo de mensagens não-numeradas e acknowledgements (ACK).

Neste protocolo um nodo remetente envia pacotes (transição t2) e espera por um ACK. Um "timeout" (transição t3) é usado para recuperar pacotes perdidos na rede. Se t3 chegar a ser acionada, significa que o pacote foi perdido, e um token será colocado em P1, habilitando t2 para enviar novamente na rede. O receptor espera por uma mensagem e envia um ACK imediatamente (transição t6). A rede pode perder pacotes (se transição t4 é executada) e/ou ACKs (se transição t9 é executada). Neste modelo, o protocolo assume que "timeouts" são somente disparados se o pacote ou ACK foi perdido. Também é assumido que o receptor pode detectar mensagens duplicadas, mas o remetente não pode detectar um ACK duplicado.



A figura 2.2 a seguir, contém os tempos de habilitação e atuação de cada transição:

transição	tempo de habilitação (em ms)	tempo de atuação (em ms)
t1	0	1
t2	0	1
t3	1000	1
t4	0	100
t5	0	100
t6	0	10
t7	0	100
t8	0	100
t9	0	100

Fig. 2.2 - Tempos de habilitação e atuação das transições

A rede contém 3 conflitos , ou seja, situações em que mais de uma transição está habilitada, concorrendo para o mesmo recurso (token, que neste exemplo, pode ser considerado uma mensagem). No modelo básico de RdP, a escolha é não determinística, necessitando então para a avaliação de desempenho, a determinação da probabilidade de cada uma das transições conflitantes atuarem. Observar que nesta rede, como existe somente um token em cada lugar (P_i), somente uma das transições conflitantes poderá atuar.

No exemplo existem os seguintes conflitos, com as probabilidades de ocorrência respectivas:

1. $\{t4:0,05 , t5:0,95\}$, significando 5% de chance de haver perda de pacotes;
2. $\{t3:0 , t7:1\}$, significando o fato que t7 tem prioridade sobre t3, sempre que ambos são habilitados para atuação;
3. $\{t8:0,95 , t9:0,05\}$, significando 5% de chance de haver perda de ACKs

As seguintes restrições são feitas:

1. Tempo de habilitação de t3 > Tempo de atuação de t5 + t6 + t8, significando que o token só permanecerá em P4 se efetivamente a informação (pacote ou ACK) foi perdida;
2. Só a transição t3 é temporizada (vide fig. 2), possuindo então um tempo de transição diferente de zero;
3. A perda da mensagem não requer mais tempo que a transmissão com sucesso, ou seja, tempo de atuação da transição t4 é igual ao

tempo de atuação da transição t5, e o tempo de atuação da transição t9 é igual ao tempo de atuação da transição t8.

A análise de desempenho será feita segundo os modelos definidos em [TAZ85] e [TAZ86], onde existe um tempo associado a lugares e não a transições, ficando então a fig.2.1 representada da maneira mostrada na fig.2.3.

Para o cálculo os seguintes elementos são considerados:

P0(N-PE) período básico de N processadores de envio de mensagem;

P0(N-PR) período básico de N processadores de recebimento de mensagens;

M0(PE) número de processadores de envio;

M0(PR) número de processadores de recepção;

t(i) tempos envolvidos no envio;

t(j) tempos envolvidos na recepção;

D0(N-PE) desempenho do processador de envio

D0(N-PR) desempenho do processador de recepção

$P0(N-PE) = t_2 + t_5 + t_6 + t_7 + t_8 + t_1 = 1 + 100 + 10 + 100 + 100 + 1 = 312 \text{ ms}$

$D0(N-PE) = \frac{M0(PE)}{P0(N-PE)} = \frac{1}{312} = 3,2 \text{ pacotes por segundo}$

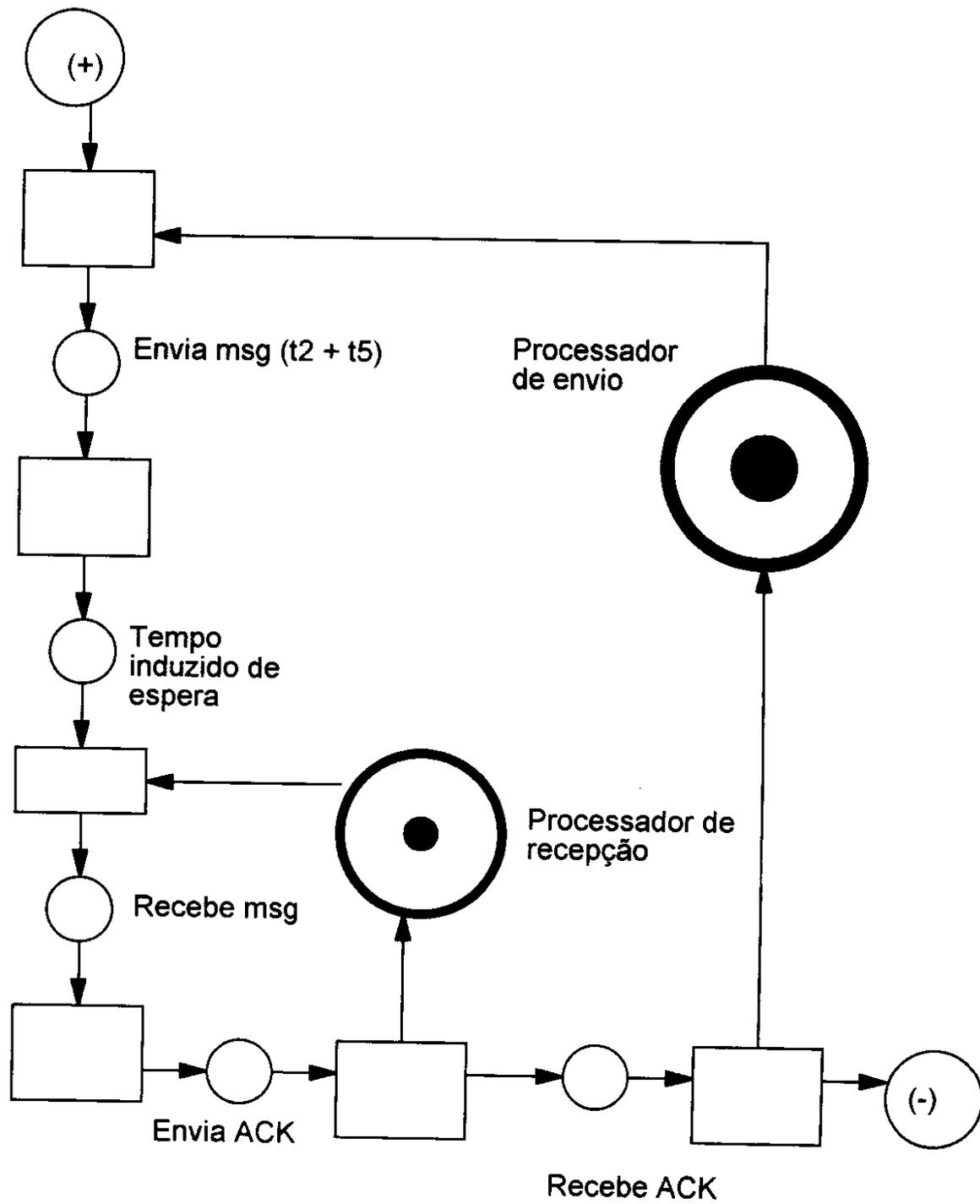


Figura 2.3 - Modelo da figura 2.1 para cálculo de desempenho

$$P0(N-PR) = t6 + t7 + t8 = 210 \text{ ms}$$

$$D0(N-PR) = \frac{1}{210} = 4,7 \text{ pacotes por segundo}$$

O desempenho geral será o mínimo entre os 2 desempenhos calculados, mas levando-se também em consideração as probabilidades de t5 e t8 ocorrerem simultaneamente, pois o acontecimento de t4 ou t9, acarreta desempenho zero.

Logo:

$$D_{\text{final}} = 3,2 \times 0,95 \times 0,95 = \underline{2,88 \text{ pacotes por segundo}}$$

Esta é a vazão máxima teórica possível do sistema, significando que na prática deve-se obter um resultado pior.

Outro ponto que pode ser visto imediatamente, é uma forte sub-utilização dos recursos de recepção.

Neste modelo simples poderia-se simular outros comportamentos, como a alteração para que fosse liberado o processador de envio, logo após a transmissão do pacote, e não no recebimento do ACK.

Outras medidas podem ser feitas, utilizando-se a formulação teórica de [TAZ85] e [TAZ86], mas neste trabalho não objetiva-se esgotar a análise, mas sim mostrar a utilização da metodologia.

A aplicação da metodologia em conjunto com os modelos analíticos, mostra-se bastante útil quando tem-se sistemas não complexos, sendo esta a questão principal para sua utilização.

Os protocolos envolvendo uma LAN para controle de processos, são bem representados nesta metodologia, existindo um certo grau de complexidade na transposição dos modelos físicos, para os critérios exigidos pelas Redes de Petri.

A utilização na avaliação de desempenho, requer um completo domínio por parte do analista, tanto da técnica de modelagem do sistema, como da teoria para tratamento analítico do modelo, sendo pois, para os iniciantes, de maior complexidade que as outras metodologias estudadas.

Para a obtenção de ganhos reais com esta análise, recomenda-se sua utilização somente a partir do momento que o sistema e seus processos forem de amplo domínio do analista, evitando-se assim a modelagem incorreta, e, conseqüentemente, conclusões inadequadas.

Outras ferramentas analíticas para avaliação são disponíveis, sendo algumas delas uma conjugação das estudadas anteriormente.

Uma interessante manipulação de formulas empíricas pode ser vista em [MOT82], onde são propostas fórmulas para cálculo da taxa efetiva de transmissão, tempo de trânsito de uma mensagem na rede e situações de sobrecarga. Esta aplicação apresenta não uma maneira nova de tratamento da avaliação, mas complementar de análise dos problemas, servindo pois, para o projetista como subsídio adicional, para compreensão melhor da tendência do comportamento. Conforme sugerido pelo autor do artigo, são bastantes úteis quando uma comparação rápida de diferentes implementações de protocolo é necessária.

2.2 SIMULAÇÃO

2.2.1 Procedimento

Conforme pode ser visto na seção anterior, a aplicação das ferramentas analíticas depende de algumas restrições e premissas, não sendo aconselháveis para sistemas complexos.

A ferramenta de simulação permite que se estude o comportamento do sistema, através de um modelo, sem utilizar o sistema real, propriamente dito. Este modelo deverá "reagir" de maneira idêntica ao sistema real, contendo as mesmas formas de entrada, para que se obtenha as tendências de performance.

Com a simulação pretende-se obter uma projeção dos resultados, análise da reação quando submetido a situações variadas de carga e/ou investigação de diagnóstico, para solução de problemas observados em sistemas reais.

Alguns pontos que podem indicar a aplicabilidade da simulação ou não, conforme proposto em [STR84]:

- a) não há uma formulação matemática completa para o problema e/ou habilidade pessoal para a resolução do modelo;
- b) não há método analítico para a resolução do modelo matemático;
- c) obter os resultados por simulação são mais fáceis do que através de ferramentas analíticas. Deve-se entender por "fácil", a relação custo x benefício, bem como a segurança dos dados obtidos;
- d) deseja-se estudar o comportamento por longos períodos de tempo;

e) não é possível ou é de alto custo, a experimentação em sistemas reais.

A principal dificuldade para a utilização da ferramenta de simulação, é o entendimento do funcionamento do sistema, para extração das características fundamentais de modelagem, especialmente para sistemas onde tem-se muitas variáveis que marcam o comportamento do processo.

A lógica geral de um estudo de simulação é semelhante ao projeto de um sistema, ou seja, estudo inicial, especificação, desenvolvimento e análise dos resultados, onde existe o seguinte fluxo lógico (figura 2.4), conforme proposto em [STR84] e [GIO86]:

Para a elaboração do simulador utiliza-se basicamente três formas de procedimentos:

a) utilização de linguagem de simulação, como GPSS, SIMULA, SIMSCRIPT ([STR84], [SCH74], [SHI75] ; em [STR84] a linguagem GPSS está descrita em sua sintaxe e semântica, inclusive com exemplos);

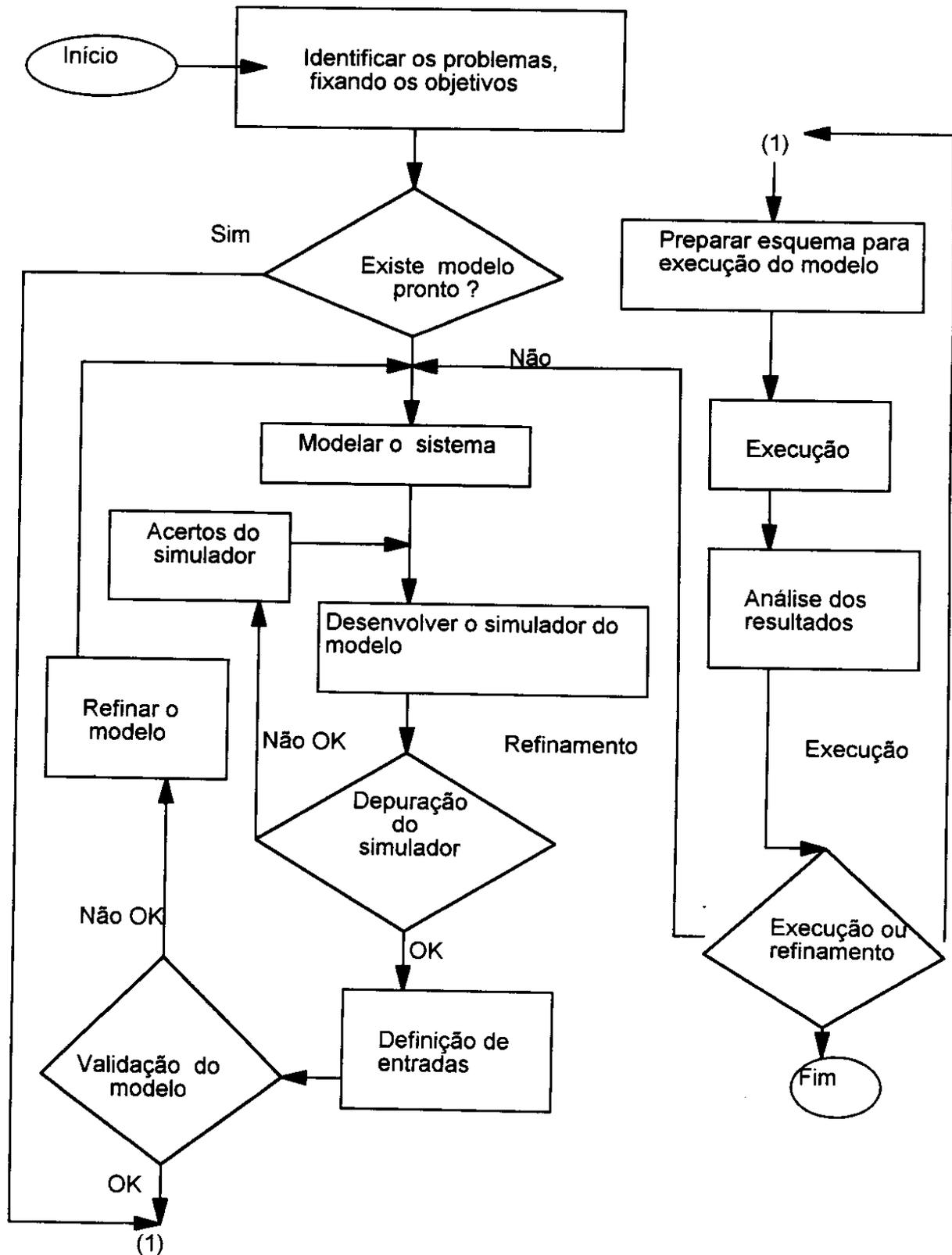


Figura 2.4 - Fluxo para elaboração de um simulador

b) utilização de rotinas ou pacotes escritos em linguagens de alto nível, contendo soluções para problemas específicos, como p.ex geração de números aleatórios e curvas de resposta;

c) elaboração de um modelo próprio, a partir da confecção de programa em linguagem de alto nível, como C. Para sistemas complexos normalmente esta é a técnica mais usada, pois permite uma modelagem mais próxima do sistema em estudo. Deve-se observar que não existe um padrão para simulação, pois a partir da sugestão do fluxo básico proposto, existem diversas formas de chegar-se ao resultado final, dependendo, fundamentalmente, da habilidade do modelador.

Como toda ferramenta, esta também deve ser utilizada com certos cuidados, tendo em vista que:

a) os resultados obtidos do simulador são, via de regra, uma amostragem da tendência e não os resultados reais;

b) a elaboração de um simulador só tem sentido em sistemas grandes e complexos, pois conforme [JON82], em sistemas pequenos e/ou simples, o custo de simulação poderá corresponder ao orçamento total do projeto;

c) ter sempre em mente, que o simulador é um modelo com simplificações feitas, e que para qualquer inferência, este fato deve ser levado em consideração;

d) para aumentar a precisão dos resultados, tendo em vista a obtenção de vários dados a partir de variáveis aleatórias, necessita-se de uma grande quantidade de execuções do simulador, o que pode tornar caro o processo.

2.2.2 Simulação híbrida

A junção das ferramentas de simulação pura e ferramentas analíticas, é definida como de simulação híbrida, sendo bastante utilizada para o aumento de eficiência dos modelos analisados, pois os valores calculados através dos modelos, são comparados com os obtidos da simulação, servindo então ambos os resultados para o refinamento do modelo geral do sistema, aproximando-se assim da situação real da rede.

O que observa-se como resultado desta comparação, é que os resultados analíticos são mais "otimistas" do que os obtidos pela simulação, devendo-se este fato, principalmente, a interdependência entre eventos, que é sentida na simulação.

Uma interessante aplicação desta metodologia é vista em [MAN77], para uma LAN de baixa complexidade, onde os resultados dos dois métodos foram depurados, tornando possível a simulação de um modelo realista.

Para o projeto de um simulador, objetivando-se modelar uma LAN para ambiente industrial, existem condições diferentes das envolvidas em p.ex., automação de escritórios. Neste ambiente industrial deve-se dar prioridade ao estudo de itens como estabilidade, integridade, "throughput" e retardo, sendo os parâmetros de capacidade, facilidade para conexão e desconexão de estações, tamanho das mensagens, determinismo na recepção das mensagens e tolerância a falhas, fundamentais no simulador.

Uma proposta de metodologia para descrever os modelos de simulação é apresentada em [CHI85] e [BUX84], onde são descritas as operações da LAN em termos de eventos individuais, e elementos da rede. O interrelacionamento dos eventos são então construídos para o modelo, sendo os pontos a seguir, alguns dos considerados:

- os dados de entrada são obtidos por estatística;
- o programa de simulação tem mecanismo de tempo para a rede simulada;

- cada evento é identificado pelo tempo em que ocorre, e pela ação que gera;
- o programa de simulação mantém uma lista de eventos, ou passos, ordenado pelo tempo de ocorrência, e faz:
 - seleção do evento com tempo mais cedo;
 - coloca o relógio simulado para este tempo executa a ação;

Como um exemplo da aplicação da ferramenta, o autor desenvolveu a simulação de um linha multiponto, half-duplex, com o protocolo poll-select, com o objetivo de determinar o tempo de resposta para diferentes velocidades de transmissão e número de terminais, estando a listagem do programa no anexo 1.

As mensagens de entrada foram definidas com 21 caracteres e as de saída com 33 caracteres, com a utilização de transmissão síncrona.

Considerações sobre a simulação:

- a) o protocolo simulado foi o pool/select da Burroughs, obedecendo as seguintes características básicas:

- foi utilizado um processo para polling, onde cada terminal só envia uma mensagem de cada vez (independente de haver mais mensagens na fila e prontas para envio ao "mainframe"), e o "mainframe" faz uma varredura sequencial do terminal 1 ao n, e depois retornando ao 1, só interrompendo o processo, se no "mainframe" tiver alguma resposta para enviar, mas não quebrando a sequência, nem dando prioridade para nenhum terminal.

- cada terminal possuía 100 mensagens para transmitir ao Host e obter resposta, onde cada mensagem tinha "hora de nascimento" aleatória;

- neste protocolo o tempo de resposta para atendimento de uma mensagem, é dado por:

$TCH + TAH + TRT$, onde

TCH - tempo para a mensagem chegar ao Host, ou seja, desde a hora que esta nasceu até ser aceita no Host;

TAH - tempo que o Host demora atendendo a mensagem;

TRT - tempo de retorno da mensagem ao terminal, após ter sido concluído o atendimento pelo Host;

Em [TAR83] pode-se obter maiores detalhes sobre o protocolo.

b) um baixo número de mensagens por terminal, p.ex., 10 por terminal, distorce os resultados, não gerando dados confiáveis, gerando-se um comportamento mais linear do modelo, quando na realidade é exponencial. A partir de 100 mensagens por terminal consegue-se uma melhor precisão, mas o tempo de processamento do simulador aumenta consideravelmente, com o aumento do número de mensagens.

O ideal seria ter-se um número da ordem de 500 mensagens por terminal, pois teríamos uma distribuição bastante aleatória do nascimento das mensagens, e, portanto, uma simulação mais precisa da interdependência entre processos.

c) o aumento do número de terminais afeta também a execução do processo, mas em menor grau do que o crescimento do número de mensagens.

d) as curvas resultantes da simulação feita, são apresentadas na fig. 2.5, onde as considerações feitas, podem ser melhor visualizadas:

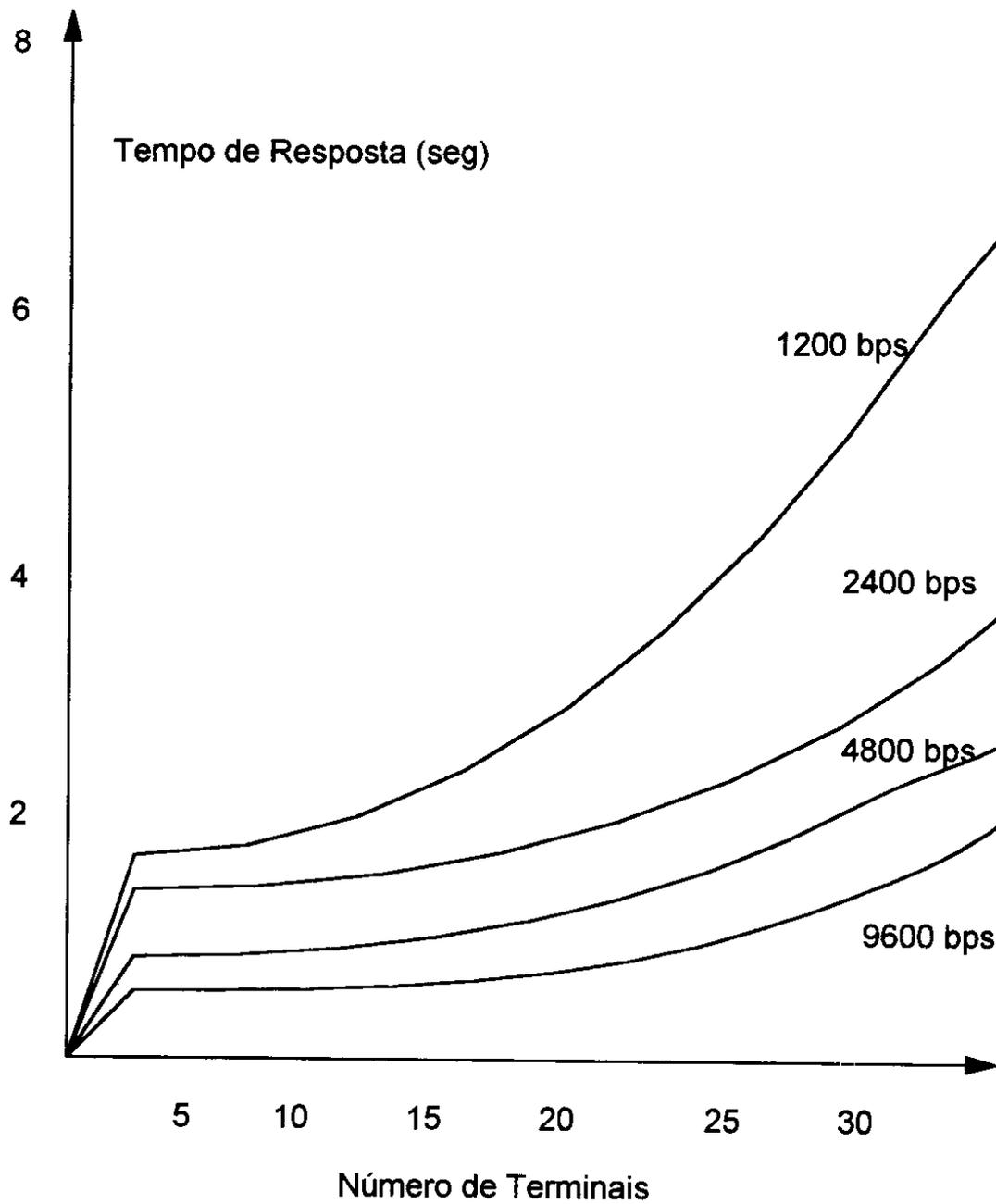


Figura 2.5 - Comportamento da Simulação

e) a geração das variáveis aleatórias, como p.ex., o nascimento das mensagens nos terminais, afeta significativamente os resultados. Deve-se escolher uma função que randomize o máximo possível, ou seja, que tenha uma distribuição bem "espalhada" dos valores, pois do contrário acontece um comportamento muito linear de todos os terminais, e não pode-se assim simular, adequadamente, o fluxo entre "mainframe" e terminais.

f) para a diminuição do tempo de execução do simulador, é aconselhável que o avanço do clock de controle não seja feito por incrementos da unidade de clock, mas pelos tempos envolvidos no processo (transmissão da mensagem de entrada, saída), evitando-se laços desnecessários só para incremento de clock.

2.3 MEDIÇÃO

As ferramentas até aqui estudadas não trabalham com a rede real, mas com modelos matemáticos ou simulação, sendo pois os resultados obtidos, uma indicação de tendências, não garantindo-se que aquele seja o resultado a ser obtido na prática. O método que efetivamente permite uma análise concreta, é a medição na rede real, a partir de software e hardware que possibilitem coletar os dados e interpretar os vários parâmetros, sendo este fato o grande

diferencial em relação aos demais métodos. Como desvantagem tem-se o custo, pois como será visto adiante, requer software e hardware, com um ambiente próprio.

Para a obtenção dos resultados da rede real, necessita-se de medidas dos vários parâmetros envolvidos, o que pode ser feito através de um Centro de Medição ou CM, que, conforme [MAD86], deve ser capaz de:

- a) coletar os dados de tráfego;
- b) ser transparente para o usuário, isto é, não afetar as condições de operação da rede;
- c) não incrementar a carga da LAN;
- d) não afetar os parâmetros sob medição;
- e) manipular os dados, para a apresentação de relatórios do comportamento.

Esta técnica parte do princípio de que informações sobre o ambiente a ser monitorado, serão captadas e enviadas a um CM para análise, onde tem-se um esquema como o da figura 2.6 a seguir:

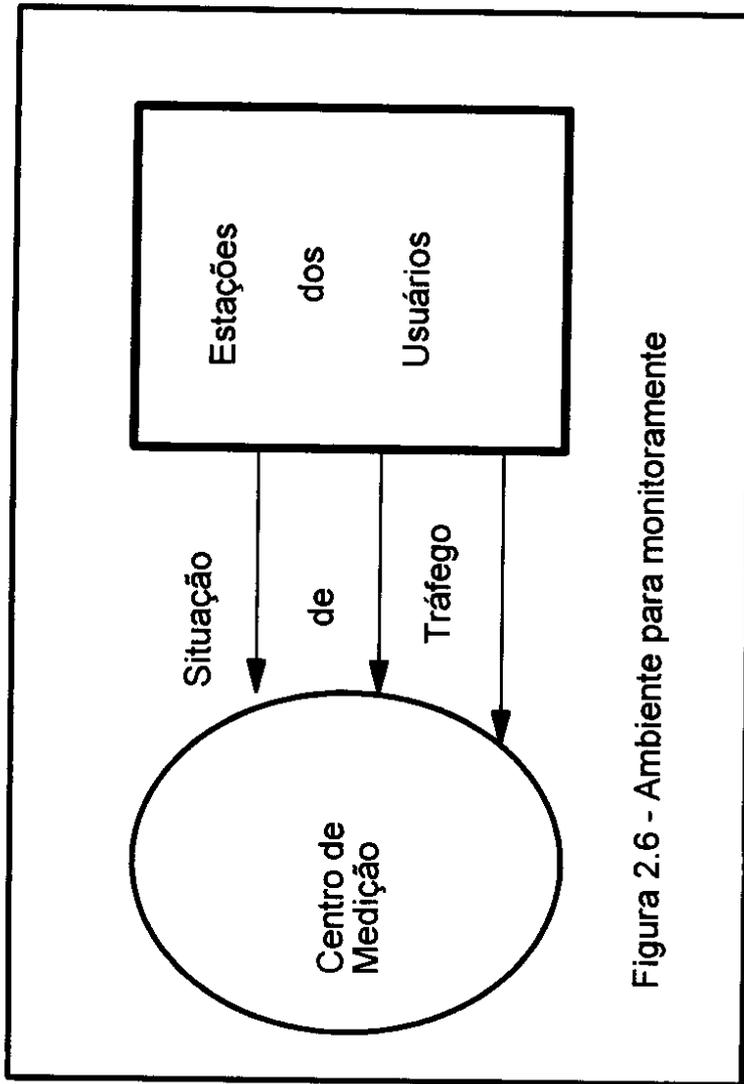


Figura 2.6 - Ambiente para monitoramento

Neste contexto a performance de uma LAN pode ser medida, de acordo com os seguintes tipos de topologias:

a) Centralizada: um único nodo ou CM efetua as medições

Este método é o de menor custo, pois somente um nodo possui software e hardware mais elaborado, o que não permite um monitoramento completo de todas as informações desejadas, como p.ex.: pode-se saber sobre uma colisão, mas não quais ou quantas estações foram envolvidos.

b) Descentralizada: existe coleta de dados local em cada nodo da rede. Periodicamente os dados são transmitidos para um nodo central, para que este faça a consolidação. Esta transmissão periódica, acarreta um overhead na rede, contrariando os objetivos b) e c) anteriormente citados. Uma alternativa para envio dos dados monitorados seria o uso de um canal adicional, o que, por outro lado incrementaria o custo.

Além disso, precisa-se mais memória e capacidade de processamento nos nodos, para que tenha-se um monitoramento preciso, o que significa, também, custo.

c) Híbrida: utiliza uma combinação dos dois métodos, sendo coletado centralizadamente, o máximo de informações possíveis.

As desvantagens deste método, são os problemas para coordenação de medidas obtidas de maneira centralizada e descentralizada, ao mesmo tempo.

Em [AME82] pode ser visto que o processo escolhido para avaliar a performance da LAN do NBS, foi o híbrido. Esta LAN tem o protocolo CSMA_CD, com cabo coaxial de 1MBPS, com múltiplas portas, onde cada porta possui um equipamento de interface, que é um microcomputador. Nesta avaliação foram gerados relatórios das medidas obtidas, que permitiram um completo conhecimento da rede, sendo descrito a seguir, os principais tópicos:

- Matriz de comunicação: indica o fluxo de tráfego entre usuários, onde tem-se tabulado para cada par fonte(f)-destino(d), o número de pacotes e quantidades de bytes, para um dado período, transmitidos de f para d , servindo a medida para a caracterização do tráfego.
- Histograma dos tipos de pacotes: indica a distribuição de cada tipo de pacote transmitido. Na rede da NBS existem 12 tipos de pacotes, e o CM conta cada tipo por uma indicação em um campo de controle, servindo a medida para a caracterização do tráfego.
- Histograma do tamanho dos pacote de dados: registra o número e proporção de pacotes de dados, numa particular classe de tamanho, ou seja, conta quantos pacotes possuem tamanho compreendido, p.ex.,

entre 8 e 15 bytes. Esta medida complementa a análise sobre a caracterização do tráfego.

- *Throughput*: indica o fluxo de bytes na rede, com a medição do total de bytes transmitidos e os de informação. Os dados obtidos mostram o uso do canal de comunicação, sendo esta medida caracterizada como de análise de performance.

- Histograma do retardo na aquisição do canal: registra o tempo gasto pelos usuários para conseguir adquirir a permissão para transmissão no canal, sendo esta uma medida caracterizada como de análise de performance.

- Histograma do retardo de comunicação: indica os retardos que ocorrem na comunicação, entre fonte e o destino, sendo a medida de retardo considerada, desde o momento que o pacote está pronto para ser transmitido na fonte, até o instante que é recebido no destino. Esta medida foi realizada somente para os pacotes originados do gerador de tráfego artificial, sendo caracterizada como de análise de performance.

- Histograma do contador de colisões: tabula o número de colisões encontradas para um pacote, antes de ser efetivamente transmitido, possibilitando outra análise de performance.

- Histograma do contador de transmissão: indica o número de vezes que um pacote é transmitido (transmissão original mais retransmissões), antes da comunicação definitiva para o destino, possibilitando análise de performance.

Pela análise da NBS pode-se notar a necessidade de componentes específicos no CM, como p. ex. para variação do tráfego, sendo imprescindível que o CM, para atender os requisitos de controle, tenha um ambiente com:

a) Gerador de tráfego artificial: dispositivo que permitirá análise da LAN nas mais variadas cargas de operação, principalmente nos casos extremos de picos ou sobrecargas

b) Monitor: responsável pelas medições, ou seja, reunião das informações para posterior análise.

c) Software para análise dos dados: sumariza as informações obtidas pelo sistema monitor, fornecendo as estatísticas do comportamento da LAN, possivelmente apresentando os dados sob forma gráfica.

Um problema que observa-se nas medições feitas em LANs, é exatamente o que medir. Em [AME82] chega-se a sugerir para medir-se tudo, o que na prática é inviável, pois necessitaria a criação de uma base de dados

contendo todo o tráfego original, para que a qualquer momento, pudesse ser reconstruído.

Uma maneira natural para o desenvolvimento, é determinar-se primeiramente os resultados que deseja-se obter através dos relatórios, e a partir daí, obtem-se como decorrência, que medidas devem ser feitas e com qual simulação de tráfego.

Para a rede da NBS ([MIL84], [AME82], [AME83]), um conjunto de grupos de análise foram selecionados, como:

- fluxo de tráfego entre nodos

- histograma do tamanho dos pacotes

- fluxo de bytes na rede (throughput de utilização)

- histograma dos retardos de transmissão

- histogramas dos retardos para aquisição do canal

O que medir está diretamente vinculado ao equipamento disponível para medição, devendo-se estabelecer os parâmetros, dentro de uma realidade viável para o CM da instalação.

De uma maneira geral pode-se dizer que o Centro de Medição deve prover medidas de estudo do tráfego, retardos, integridade, tolerância a falhas, colisão e estabilidade.

Em [TOB78] e [AME82] são propostas perguntas que devem ser respondidas, para auxílio do projetista na identificação dos parâmetros relevantes para sua análise, como:

- Qual é a capacidade máxima do canal sob condições normais de operação? Quantos usuários ativos são necessários para atingimento deste máximo?

- Como o tamanho dos pacotes afeta a utilização e retardo?

Em uma rede com topologia descentralizada, como a da NBS, tem-se uma arquitetura básica de medição, onde:

- a) Os monitores constantes de cada estação do usuário, devem enviar periodicamente para o analisador de dados, o resultado das medições locais, preferencialmente, por canais alternativos, para que não afete o tráfego;

- b) O monitor central executa medidas não efetuadas nos monitores locais, como p.ex. a utilização do canal, enviando para o analisador de dados.
- c) O gerador de tráfego deve injetar na rede os vários pacotes, com configurações diferentes, a partir de seleção feita pelo operador, informando ao analisador de dados o que foi enviado.
- d) O analisador de dados coleta as informações mantendo registro estatístico, e fornecendo os resultados do monitoramento por uma impressora local, ou em tempo-real, através do terminal de supervisão.

Os equipamentos para geração de tráfego, monitoramento central e analisador de dados, devem ser dedicados para estas tarefas, sendo os softwares de monitoramento desenvolvidos tendo como princípio a eficiência, sugerindo-se que os mesmos sejam programados em C, buscando-se portabilidade para o CM.

Para o contexto de uma LAN em ambiente industrial, esta metodologia apresenta-se como bastante adequada, pois permitirá uma avaliação de parâmetros críticos, como p.ex., retardos, tempos de transmissão e "throughput". O principal problema para este ambiente, é ter-se um conhecimento bastante preciso dos equipamentos para controle de processos,

pois precisa-se simular um tráfego que represente, o mais próximo possível, as condições da LAN real.

Para a obtenção de um maior grau de análise, recomenda-se associar a metodologia de medição com a de simulação, sendo esta última utilizada para a modelagem dos equipamentos para controle de processos. No capítulo 4 é proposto um esquema para avaliação de LANs em ambiente industrial, onde tem-se um detalhamento maior da utilização desta metodologia.

Uma interessante aplicação da análise de comportamento em uma rede, pode ser visto em [HAU87], onde é apresentado um pacote desenvolvido para funcionar como analisador de protocolos, nível por nível do modelo OSI, mas com características que permitem um conhecimento amplo do tráfego. Este pacote apresenta de forma compacta e integrada, um ambiente de medição contendo os conceitos anteriormente discutidos, e que foram implementados.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

Capítulo 3 - Parâmetros para Sistemas em Tempo Real

3.0 - PARÂMETROS PARA SISTEMAS EM TEMPO REAL

Os sistemas para controle de processos, objeto de LANs em ambiente industrial, são classificados como de tempo real, pois tem-se necessidade rigorosa de tempo entre a geração do comando e a ação nos processos, sendo suas atividades caracterizadas como de coleta dos dados (local ou remota), processamento das informações, supervisão e atuação nos processos, no desenrolar dos acontecimentos.

Os componentes destes sistemas são:

- a) sensores digitais e/ou analógicos para efetivação das medições;
- b) conversores, que transformam os sinais captados pelos sensores, em sinais passíveis de manipulação por computadores e vice-versa;
- c) LAN para suporte da comunicação entre os diversos componentes;
- d) controladores, para a execução dos comandos aos processos;
- e) facilidades de comunicação entre operadores e os equipamentos;

De acordo com [AYA82] as características fundamentais em um sistema de controle são:

- modularidade: módulos de tamanho limitado, com execução integral de uma função;
- expansibilidade: possibilidade de inclusão de novos componentes, sem necessidade de reconfiguração total dos processos;
- dependência entre processos;
- tolerância a falhas: um sistema computacional atuando em tempo real, deve apresentar alto grau de confiabilidade, pois uma falha ignorada, pode significar perdas materiais e/ou humanas.

As principais características do sistema computacional, segundo [STR84], para atendimento dos requisitos citados são:

- a) resposta rápida para os eventos do processo;
- b) dispositivo de E/S compatíveis com as operações do processo;
- c) paralelismo na execução de programas;

d) capacidade de obter informações do processo assim que fiquem disponíveis, e enviar sinais de controle, caso sejam necessários;

Nestes sistemas em tempo real, o objeto de controle e o sistema controlador são ligados via sensores e atuadores, baseados em interfaces.

O sistema de controle aceita os dados dos sensores, a intervalos regulares, processa os dados, e envia os resultados para os objetos de controle, utilizando-se os atuadores, sendo que, na maioria dos sistemas em estudo, tem-se um comportamento regular, com a mesma sequência de ações executadas periodicamente, com ciclos entre as ações com tempos curtos e previsíveis.

O sistema deve possuir software para atuar na modalidade de tempo real, onde, para o caso de controle de processos, o sistema operacional deve permitir programação concorrente.

Quando da modelagem do sistema (em [AND82] tem-se uma proposta com uso de grafos), a escolha de que parâmetros serão mais enfatizados dependerá dos objetos a serem controlados, e, conseqüentemente, do nível de precisão desejado, sendo o tempo de resposta e tolerância a falhas mandatórios em qualquer análise.

Em sistemas de tempo real, a principal condicionante que influencia a metodologia para avaliação de performance, está ligada ao fator tempo, pois deseja-se saber neste casos qual probabilidade de que uma estação da rede (um equipamento para controle de processos- ECP), fique sem condições de comunicação por um tempo intolerável, em vista do processo por ela controlado.

Adicionalmente, deve-se levar em consideração, que as aplicações em ambiente industrial tem peculiaridades, onde os protocolos implicam um padrão de iteração particular, com a inserção, em alguns casos, de retardos que tornam-se críticos para os controles. Como um exemplo de protocolo envolvido em sistemas de tempo real, tem-se no anexo 2 o protocolo MMFS, através do qual o usuário fornece os comandos para os processos, sendo este uma linguagem de mensagens para o sistema.

Capítulo 4 - Ambiente para Avaliação de LANs

4.0 - AMBIENTE PARA AVALIAÇÃO LANs

Neste capítulo é apresentada uma proposta para um ambiente integrado de hardware e software, que possibilite a avaliação de performance de LANs em ambientes industriais, com utilização das metodologias analisadas nas seções anteriores.

Para o estabelecimento do ambiente, como visto na seção 2.3, precisa-se da identificação dos relatórios desejados, sendo propostos:

a) Controle de retardos de comunicação: indicação do tempo gasto na transmissão do pacote, desde a sua criação, até a aceitação pelo nodo destino. Este levantamento deverá ser feito a nível de tipo de pacote, devendo este tipo ser identificado pelo comando de atuação do ECP, como p.ex., READ, WRITE e LOAD (ver anexo 2).

A análise destes resultados indicará a performance da LAN para cada comando, o que permitirá conhecer quão rápido é cada um, possíveis tempos inaceitáveis e comandos críticos.

b) Indicação dos tipos de pacotes: distribuição de cada tipo de pacote, com o objetivo de caracterizar o tráfego, com conhecimento de

quais comandos são mais executados, em cada tipo de controle de processos.

Para cada tipo de pacote (que são os comandos) deve-se separar os resultados por tamanho, o que facilitará uma análise de quais parâmetros em cada comando são mais onerosos para os processos.

c) *Throughput*: obter o fluxo de pacotes no sistema, para, através de gerador de tráfego artificial, identificar o ponto de carga máxima aceitável, a partir do qual os tempos de resposta tornam-se críticos.

Nesta análise deve-se obter a quantidade de bytes sendo transmitidos, o que permitirá a avaliação e adequação do melhor meio físico para a transmissão.

A escolha dos itens acima, deve-se a necessidade de concentrar-se a análise no comportamento da LAN, para execução dos comandos que atuam nos ECPs.

Para a definição de um ambiente que possibilite a obtenção das informações definidas anteriormente, será utilizada uma combinação das metodologias de medição e simulação, onde esta última será aplicada na modelagem dos ECPs, como p.ex., um controlador lógico programável, permitindo assim variar-se toda a faixa de comandos, a um custo mais baixo.

Neste esquema, os seguintes elementos farão parte do ambiente:

a) Gerador de tráfego artificial: deve ter capacidade de gerar todos os tipos de pacotes, que são os comandos para os processos, com controles separados para os diversos tipos de equipamentos que podem ser conectados a LAN (p.ex: CLPs). Este gerador será um software a ser processado em PC, onde através de parâmetros, um operador possa monitorar os pacotes, indicando quais, como e quando serão gerados.

O algoritmo para geração dos pacotes, deverá seguir rigorosamente a lógica de cada equipamento para atuar em um processo, sendo esta ordem obtida nas especificações técnicas de cada um.

b) Monitor: responsável pela medição dos parâmetros nos nodos fonte e destino dos pacotes. Deverá ser um software capaz de identificar os tipos de pacotes transmitidos na rede, com os respectivos tempos envolvidos, com a transmissão periódica para o analisador de dados.

Neste ponto aparece um problema importante a resolver, que é o de ter-se um relógio único para o sistema, pois do contrário, não pode-se calcular os retardos envolvidos. O leitor pode ver no clássico artigo de

Lamport [LAM78], um algoritmo que permite a sincronização de relógios físicos e lógicos, ficando aqui o registro que este fato é capital para o sucesso da medição.

c) Analizador de dados: responsável pela compilação dos dados obtidos dos monitores e dos que foram injetados na rede, pelo gerador de tráfego artificial. A partir destes dados, deve ser capaz de gerar os relatórios especificados anteriormente, devendo, preferencialmente, apresentar os mesmos numa console central para monitoramento, e de forma gráfica.

d) Simulador de equipamento: software que realizará a simulação dos equipamentos para controle de processos, onde o comportamento deve ser o mesmo que o do modelo físico, como p.ex. de um CLP.

Observar que este simulador deve reagir nos 2 sentidos, ou seja, como destino ao receber pacotes do gerador de tráfego artificial ou estação de usuário, e como fonte para dar resposta as perguntas solicitadas.

Neste ambiente proposto deve-se ter uma adaptação do MAP, para simplificação dos testes, com a utilização dos níveis 1, 2 e 7 do MAP, com a geração dos comandos segundo o MMFS, conforme descrito no anexo 2. Observar que com esta simplificação, deve-se obter um resultado melhor do que seria obtido se fosse utilizado todos os níveis.

Com este ambiente ter-se-a uma análise de performance de uma LAN, trabalhando em condições semelhantes a de controles de processos industriais, o que servirá, após um conjunto de medidas, para os ajustes e refinamentos devidos, possibilitando assim, uma medição real com a substituição dos simuladores dos ECPs, pelos próprios modelos físicos, e o gerador de tráfego artificial passando a atuar somente para simular os momentos de "pico".

Capítulo 5 - Conclusões

5.0 - CONCLUSÕES

Neste trabalho foram analisadas algumas metodologias para avaliação de performance, onde, para cada uma, considera-se a sua aplicabilidade para LANs em ambiente industrial. As avaliações sob a forma de modelos analíticos são discutidas, e exemplos mostram a potencialidade da ferramenta.

Como consequência das análises apresentadas, uma proposta de um ambiente para avaliação de performance de LANs em ambiente industrial é feita, onde, através das metodologias de medição e simulação, pretende-se obter o comportamento da rede, para atendimento dos comandos que fazem a atuação dos ECPs, como p.ex., os controladores lógicos programáveis.

Apesar de ter-se uma variedade de metodologias para análise, tem-se necessidade no ambiente estudado, de requerimentos rigorosos de controle, razão pela qual a metodologia de medição apresenta-se como a mais recomendada, pois permite a avaliação dos parâmetros na configuração real, ficando as demais como auxiliares para uma abordagem inicial, para os primeiros ajustes da rede.

A proposta feita no capítulo 4, contém os conceitos e elementos básicos para avaliação, sendo uma continuação natural deste trabalho, a implementação do ambiente proposto, cujos resultados poderão servir para

avaliação de ambientes, onde tem-se LANs utilizando o protocolo MAP, para controle de processos industriais.

Anexo 1 - Listagem do Programa de Simulação

ANEXO 1

```
program redes2(input,output);
```

```
{*****
```

```
* simulacao de tempo de resposta,numa linha multiponto *
```

```
* e half-duplex,com protocolo poll/select *
```

```
*****}
```

```
const
```

```
tr          = 0.052; {tempo de propagação + turnaround}
```

```
tah         = 0.418; {tempo de atendimento do host}
```

```
qtd_mens    = 100;
```

```
qtd_term    = 30;
```

```
qtd_prontas = 30;
```

```
type
```

```
mensagem    = record
```

```
    t:real; {hora de nascimento no terminal}
```

```
    mt:real; {hora de chegada no terminal}
```

```
    t:integer; {indica se a mensagem ja foi enviada
```

```
para o host...0->nao...1->sim}
```

```

        end;
filafmt      = array [1..qtd_term] of array[1..qtd_mens] of mensagem;
prontas      = record
                num_terminal:integer;
                num_mensagem:integer;
                hch:real; {hora de conclusao no host}
            end;
filaftp      = array [1..qtd_prontas] of prontas;
saida        = record
                vel_tran:integer;
                tempo_resp:real;
                num_t:integer;
            end;
filasaida    = array [1..200] of saida;
var
    FMT:filafmt; {fila de mensagens dos terminais}
    FTP:filaftp; {fila de transacoes prontas no host}
    SIMULA:filasaida; {dados estatisticos da simulacao}
    FTERM:array [1..qtd_term] of integer;
    cps, velocidade, nummens, numterm, i, j, m, iterm, imens,
    indice_simula:integer;
    num_mens_ftp, numterm_carga, nummens_carga, k, l, x, y, cor:integer;
    x_ant, y_ant:integer;
    ch:char;

```

```

hnmt_aux,random_aux,ttms,ttmi,tpv,ttp,ic,clock,
tempo_resposta:real;
fim_pesq,tem_mensagem,ftp_vazio,x_logico:boolean;

```

```
begin {1}
```

```

{*****
* carga inicial da fila de mensagens dos terminais *
*****}

```

```
for numterm_carga := 1 to qtd_term do
```

```
begin {2}
```

```
hnmt_aux := 0;
```

```
for nummens_carga := 1 to qtd_mens do
```

```
begin {3}
```

```
random_aux := random;
```

```
if random_aux = 0.0
```

```
then
```

```
random_aux := random_aux + 0.0000001;
```

```
hnmt_aux := hnmt_aux - (70 * ln(random_aux));
```

```
FMT[numterm_carga,nummens_carga].hnmt := hnmt_aux;
```

```
FMT[numterm_carga,nummens_carga].sit := 0;
```

```
end; {3}
```

```
end; {2}
```

```
{*****
*início do loop de velocidade de transmissao *
*****}
```

```
velocidade := 1200;
```

```
indice_simula := 0;
```

```
while velocidade <= 9600 do
```

```
begin {4}
```

```
    cps := velocidade div 8;
```

```
    tpv := (2 * tr) + (10.0 / cps); {tempo de poll vazio}
```

```
    ic := tpv / 50.0;
```

```
    ttp := (7 / cps) + tr; {tempo de poll}
```

```
    ttms := (56 / cps) + (4 * tr);
```

```
        {tempo de transmissao de saida}
```

```
    ttmi := (44 / cps) + (4 * tr);
```

```
        {tempo de transmissao de entrada}
```

```
    limpar fila de mensagens dos terminais
```

```
}
```

```
for numterm_carga := 1 to qtd_term do
```

```
begin {2}
```

```
    for nummens_carga := 1 to qtd_mens do
```

```

begin {3}
    FMT[numterm_carga,nummens_carga].hcmt := 0;
    FMT[numterm_carga,nummens_carga].sit := 0;
end; {3}

FTERM[numterm_carga] := 1;
end; {2}

```

```
{ loop do numero de terminais }
```

```

for numterm      := 1 to qtd_term do
begin {6}
    clock          := 0.0;
    tem_mensagem  := true;
    tempo_resposta := 0;
    ftp_vazio     := false;
    num_mens_ftp  := numterm * qtd_mens;

```

```

{
    limpar fila de mensagens dos terminais
}

```

```

for numterm_carga := 1 to qtd_term do
begin {2}
    for nummens_carga := 1 to qtd_mens do

```

```
begin {3}
    FMT[numterm_carga,nummens_carga].hcmt := 0;
    FMT[numterm_carga,nummens_carga].sit := 0;
end; {3}
FTERM[numterm_carga] := 1;
end; {2}

{ inicializar a fila de transacoes prontas }

for i := 1 to qtd_prontas do
    begin {5}
        FTP[i].num_terminal := 0;
        FTP[i].hch := 0;
    end; {5}
{ fim da inicializacao }

while tem_mensagem do
    begin {7}

        for i := 1 to numterm do
            begin {8}

{retirar mensagem do FTP}
```

```
for j := 1 to qtd_prontas do
begin {9}
  if FTP[j].hch <> 0.0
  then
    begin {10}
      if FTP[j].hch <= clock
      then
        begin {11}
          iterm                :=
            FTP[j].num_terminal;

          imens                :=
            FTP[j].num_mensagem;

          clock                :=
            clock + ttms;

          FMT[iterm,imens].hcmt := clock;
          FMT[iterm,imens].sit  := 1;

          tempo_resposta       :=
            tempo_resposta +
            FMT[iterm,imens].hcmt -
            FMT[iterm,imens].hnmt;

          FTP[j].num_terminal  := 0;
```

```

        FTP[j].hch          := 0;

        num_mens_ftp       :=
        num_mens_ftp - 1;
    end; {11}
end {10}
else
    if num_mens_ftp = 0
    then
        ftp_vazio := true;
    end; {9}

```

{ verificar se tem mensagem no terminal i }

```

    imens := FTERM[i];
    if FTERM[i] > 0
    then
        begin {12}
            if FMT[i,imens].hnmt < (clock + ttp)
            then
                begin {13}
                    FMT[i,imens].sit := 1;
                    if imens = qtd_mens
                    then

```

```

    FTERM[i] := 0
else
    FTERM[i] := FTERM[i] + 1;

```

{inclusão de mensagem na fila de saída FTP... }

```

m := 0;
x_logico := false;
while not x_logico do
begin {14}
    m := m + 1;
    if m > qtd_prontas
    then
        begin {18}
            writeln('....FTP CHEIO.....');
            writeln('....FTP CHEIO.....');
            writeln('....FTP CHEIO.....');
        end {18}
    else
        if FTP[m].num_terminal = 0
        then
            begin {19}
                clock :=
                clock + ttmi;

```

```
        FTP[m].num_terminal      := i;
        FTP[m].num_mensagem     := imens;

        FTP[m].hch              :=
        clock + tah;
        x_logico                 := true;
        end; {19}
    end; {14}
end {13}
else
    begin {15}
        clock := clock + tpv;
    end; {15}
end {12}
else
    clock := clock + tpv;

end; {8}

if ftp_vazio
then
    begin {17}
        tem_mensagem := false;
```

```

tempo_resposta :=
tempo_resposta / (numterm * qtd_mens);

indice_simula := indice_simula + 1;
SIMULA[indice_simula].vel_tran :=
velocidade;
SIMULA[indice_simula].tempo_resp :=
tempo_resposta;
SIMULA[indice_simula].num_t := numterm;

writeln(1st,'velocidade = ',velocidade,'... tempo = ',
tempo_resposta:8:4,'..qtd de terminais = ',
numterm,
'...qtd de mensagens = ',qtd_mens);

end {17}
else
tem_mensagem := true;
end; {7}
end; {6}
velocidade := velocidade * 2;
end; {4}

```

```
{ mostrar graficamente os resultados que estao em SIMULA }
```

```
hires;
```

```
{ montar eixo x}
```

```
for x := 0 to 639 do
```

```
begin {26}
```

```
plot(x,179,7);
```

```
end; {26}
```

```
{ montar eixo y}
```

```
for y :=0 to 179 do
```

```
begin {27}
```

```
plot(0,y,7);
```

```
end; {27}
```

```
{ plotar o resultado que esta em simula }
```

```
velocidade := 1200;
```

```
while velocidade <= 9600 do
```

```
begin {29}
```

```
x_ant := 0;
```

```
y_ant := 179;
for k := 1 to indice_simula do
begin {28}
if SIMULA[k].vel_tran = velocidade
then
begin {30}
case SIMULA[k].vel_tran of
1200 : cor := 7;
2400 : cor := 7;
4800 : cor := 7;
9600 : cor := 7;
end;
x := SIMULA[k].num_t * 20;
y := 179 - trunc(SIMULA[k].tempo_resp * 16.9);
draw(x_ant,y_ant,x,y,cor);
x_ant := x;
y_ant := y;
end; {30}
end; {28}
velocidade := velocidade * 2;
end; {29}
```

{colocar escala no eixo dos x: esta assinalada a cada 2 terminais}

```
k := 1;
i := -2;
while k <= 80 do
  begin {35}
    i := i + 2;
    gotoxy(k,23);
    write(i);
    k := k + 5;
  end; {35}
```

{colocar escala no eixo dos y:esta assinalada a cada 1 segundo}

```
k := 23;
i := 0;
while k >= 2 do
  begin {36}
    gotoxy(1,k);
    write(i);
    i := i + 1;
    k := k - 2;
  end; {36}
read(ch);
end. {1}
```

Anexo II - O Protocolo MMFS

ANEXO 2

O PROTOCOLO MMFS

O objetivo do MAP é estabelecer um conjunto de padrões para uma LAN em ambiente industrial, para suportar comunicação entre computadores e dispositivos com capacidade de processamento, de fabricantes diversos, podendo ser implementado em qualquer tipo de computador, terminal ou dispositivo programável (a referência bibliográfica básica para este anexo é [GEN86]).

A descrição do modelo OSI pode ser vista em [TAR86], onde os protocolos de comunicação e funções de cada nível são descritos, sendo apresentado a seguir uma visão geral dos vários níveis existentes:

- a) Programas do usuário - são os programas aplicativos e não fazem parte do modelo OSI

- b) Nível 7 (aplicação) - providencia os serviços para os programas do usuário, funcionando como uma "janela", através da qual o usuário comunica-se com o sistema;

- c) Nível 6 (apresentação) - formatação e reestruturação de dados;

- d) Nível 5 (sessão) - sincronização e gerência do diálogo;
- e) Nível 4 (transporte) - providencia a transparência e segurança na transferência de dados entre os nodos;
- f) Nível 3 (rede) - executa o roteamento da mensagem pelos nodos da rede;
- g) Nível 2 (enlace de dados) - controla a transferência de unidades de informação através de uma ligação do nível físico;
- h) Nível 1 (físico) - providência as características funcionais e procedimentos para transferência de bits entre nodos.

Para o MAP tem-se os seguinte protocolos especificados, em relação aos sete níveis do modelo OSI:

- programas do usuário - MMFS
- nível 7 - ISO CASE KERNEL
- nível 6 - sem padrão até a versão 2.2

- nível 5 - ISO SESSION KERNEL

- nível 4 - ISO TRANSPORT CLASS 4

- nível 3 - ISO CLNS

- nível 2 - IEEE 802.2 LINK LEVEL CONTROL CLASS 1

- nível 1 - IEEE 802.4 TOKEN ACCESS BROADBAND ou CARRIERBAND ou MODULATION ON COAXIAL MEDIA

Nesta seção é discutido o MMFS, cujo propósito é facilitar a transferência de informação codificada entre os vários dispositivos, providenciando uma maneira comum, independente de máquina, de troca de informações entre processos.

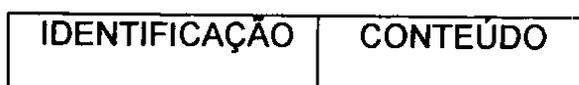
O MMFS pode ser considerado como uma linguagem de mensagens, possuindo sintaxe e gramática, bem como semântica e vocabulário dos vários campos. Estes itens serão abordados de forma compacta, para ter-se um noção da potencialidade do protocolo, sendo que para maiores referências, consultar apêndice 6 de [GEN86].

Todas as comunicações entre aplicações devem fazer uso do MMFS e CASE do nível 7 (em [TAR86] tem-se a descrição do CASE), sendo que os dispositivos programáveis constantes da rede (CLPs, CNCs, robots, etc..), devem possuir as funções de LOAD, armazenamento e recuperação de dados, informação do status dos processos e diagnósticos remotos.

1 Estrutura de sintaxe do MMFS

a) Existem duas construções básicas de bloco, dentro da mensagem: o FIELD e o DATA STREAM, ambos contendo um número inteiro de octetos (cada octeto = 8 bits).

- O FIELD é composto de 2 sub-campos:



IDENTIFICAÇÃO: indica que informação está sendo transportada no sub-campo CONTEÚDO, através de um código de diagnóstico ou nome simbólico

CONTEÚDO: a interpretação deste campo depende do valor de IDENTIFICAÇÃO.

Ex: o sub-campo IDENTIFICAÇÃO pode indicar que em CONTEÚDO tem-se um endereço

O FIELD é representado por <FIELD>. Quando representado os dois sub-campos, tem-se <IDENTIFICAÇÃO:CONTEÚDO>.

Esta notação é usada somente na especificação MMFS, não sendo transmitidos na rede os caracteres < e > da mensagem.

- DATA STREAM pode ser usado para dados longos ou para múltiplas unidades de dados contíguas do mesmo tipo. Diferem dos FIELDS por não possuírem nenhuma identificação do seu conteúdo de dados, e que os dados podem ter uma quantidade de octetos muito maior que CONTEÚDO.

A notação utilizada é <<DATA STREAM>>

b) Existe uma construção chamada GROUP para organizar e delimitar o conjunto de FIELDS e DATA STREAM dentro do MMFS, consistindo de

um conjunto de campos específicos contíguos ou outros grupos dentro da mensagem. Uma comparação que se faz, é que FIELDS e DATA STREAM podem ser considerados "átomos" da estrutura da mensagem e os grupos são as "moléculas".

A sintaxe da mensagem então pode ser vista como uma sequência de FIELDS e DATA STREAM, sendo sempre iniciada pelo GROUP, que indica o tamanho da mensagem, escrevendo-se da esquerda para a direita os componentes da mensagem, na ordem que devem ser transmitidos na rede.

2 Semântica do MMFS

a) A semântica indica o significado das mensagens do MMFS.

Cada FIELD é representado por 2 ou 3 caracteres mnemônicos (utilizados por conveniência dentro da especificação) que através de tabela obtem-se o valor de IDENTIFICAÇÃO.

ex: Mnemônico - LA

IDENTIFICAÇÃO - 08

Significado em CONTEÛDO tem-se um endereço lógico

A semântica do MMFS foi projetada para suportar dispositivos programáveis em um ambiente industrial, e, portanto, os vários códigos (como, p.ex., os mnemônicos), tentam contemplar todo este universo de equipamentos.

b) Transações

O termo transação é usado para a indicação de um controle de procedure. Representa uma sequência de campos do mesmo tipo, na qual associamos um número de transação, para ser usado em uma associação, quando vier uma resposta, relativa a uma solicitação.

Os tipos de transações disponíveis são:

- 1) command/request (resposta requerida)
- 2) command/request (resposta não permitida)
- 3) resposta (continuação)
- 4) resposta (final)
- 5) resposta negativa
- 6) abort
- 7) não solicitado indicação de dados
- 8) não solicitado resposta de dados

Cada transação começa com um campo de número de transação (<TN>) e outro de controle de procedure (<PC>), como:

<TN> <PC> <←→>.....<←→>

c) Procedures de comunicação

São regras que governam a sequência de troca de informações entre duas entidades que se comunicam.

Ex:

```
SEND REQUEST TO DOWNLOAD    →
RECEIVE RESPONSE INDICATING READINESS
TO RECEIVE DOWNLOAD        →
DOWNLOAD PROGRAM           →
```

d) A tabela com as várias funções disponíveis, tem os seguintes itens:

- definição de dados

ex: mnemônico - LA

significado - endereço lógico

- tipos de dados

ex: mnemônico - BO

significado - booleano

- controle de mensagem

ex: mnemônico - MN

significado - número de sequência de mensagem

- dispositivos

ex: mnemônico - DDR

significado - disk drive

- qualificador de arquivos

ex: mnemônico - NAM

significado - nome de arquivo

- controle de procedure

ex: mnemônico - ABO

significado - abort

- manipulação de erros

ex: mnemônico - DI

significado - data not found

- operações com arquivos

ex: mnemônico - SEL

significado - selecionar um arquivo

- funções especiais

ex: mnemônico - SYN

significado - sincronizar clocks

- instruções para máquina de controle numerico

ex: mnemônico - ROT

significado - rotacionar

- instruções para CLPs

ex: mnemônico - REA

significado - read

e) Na prática é conveniente e eficiente combinar-se múltiplas funções para uma mesma mensagem, sendo que as transações devem ser enviadas na ordem de prioridade desejada:



onde mensagem 1 é a de mais alta prioridade

4.3 Os outros itens constantes de [GEN86], apêndice 6, representam um detalhamento de formatos e interpretação para cada sub-campo de IDENTIFICAÇÃO, com exemplos para aumentar-se a eficiência, bem como uma extensão semântica para atendimento de robots, como dispositivos programáveis na rede.

Glossário

GLOSSÁRIO

ACK	Acknowledgment
CASE	Common Application Service Elements
CLP	Controlador Lógico Programável
CM	Centro de Medição
ECP	Equipamento para Controle de Processos
GM	General Motors
IEEE	Instituto dos Engenheiros Elétricos e Eletrônicos
ISO	International Standard Organization
LAN	Local Area Network
MAP	Manufacturing Automation Protocol
MBPS	Mega Bits por Segundo
MTBF	Mean Time Between Failure
MMFS	Manufacturing Messaging Format Standard
MTTR	Mean Time to Repair
NBS	National Bureau of Standards U.S. Department of Commerce
OSI	Open Systems Interconnection
PC	Personal Computer
RdP	Redes de Petri

Bibliografia

BIBLIOGRAFIA

- [AME82] AMER, P.D. A measurement center for the NBS local area computer network *IEEE Transaction on Computers*, New York c-31(8):723-29, Aug. 1982
- [AME83] AMER, P.D. Measuring a local network's performance *Data Communications*, New York. 12(4):173-82, Apr 1983
- [AND83] ANDERSON, T. et alii A framework for software fault tolerance in real-time systems *IEEE Transactions on Software Engineering*, New York se-9(3):355-64, Aug. 1983
- [AND90] ANDERSON, T. and LEE, P. *Fault tolerance principles and practice*, second edition, , Spring-Verlag 1990
- [AYA82] AYACHE, J.M. et alii REBUS, A fault tolerant distributed systems for industrial real-time control . *IEEE Transactions on Computers*, New York. c-31(7):637-47, Jul 1982
- [AYA84] AYACHE, J.M. et alii Petri nets are good for protocols *Computer Communications Review* , New York 14(2):66-74, 1984
- [BUX84] BUX, W. Performance issues in local area network *IBM Systems Journal*, Armonk 23(4):351-74, Oct. 1984
- [CHI85] CHIEN, J.Y. Detailed tests show how well industrial local network performs *Data Communications*, New York 14(9):119-31, Aug. 1985
- [FAY78] FAYOLLE, G. et alii An analytic evaluation of the performance of the "send and wait" protocol *IEEE Transactions on Communications*, New York com.26(3):313-19, Mar. 1978
- [GEN86] GENERAL MOTORS TECHNICAL CENTER *MAP Specification*, Warner - Aug, 1986
- [GIO86] GIOZZA, W.F. et alii *Redes locais de computadores: protocolos de alto nível e avaliação de desempenho* São Paulo, McGraw Hill 1986
- [HAU87] HAUGDAHL, J.C. Analyzing Network Traffic *PC Tech Journal* 48-62 Oct 87
- [HEU87] HEUSER, C.A. *Modelagem de sistemas com redes de petri* Porto Alegre CPGCC/UFRGS, 1987 Notas de aula

- [JAL94] JALOTE, P. *Fault tolerance in distributed systems*, Prentice-Hall, 1994
- [JON82] JONES, T.W. et alii Prototyping and small scale software projects *Software Engineering Notes*, New York 7(5):169-70, Dec. 1982
- [KLE75] KLEINROCK, L. *Queueing systems, theory* New York, John Wiley, 1975 v.1
- [KLE76] KLEINROCK, L. *Queueing systems: computer applications* New York, John Wiley, 1976 v.2
- [LAM78] LAMPORT, L. Time, clocks, and the ordering of events in a distributed systems *Communication of the ACM* 21(7):558-65 July 1978
- [MAD86] MADAN, M.S. Performance monitor for an operational LAN *Computer Communications*, Surrey, 9(6):283-86, Dec. 1986
- [MAN77] MANNING, E.G. et alii A homogeneous computer network-analysis and simulation *Computer Networks*, Amsterdam, 1(4):225-40, May 1977
- [MAR82] MARTIN, J. *Systems analysis for data transmission* Englewood Cliffs, Prentice Hall 1982
- [MAY81] MAYER, P.L. *Probabilidade, aplicações a estatística* Rio de Janeiro, Livros Tecnicos E Cientificos, 1981
- [MIL84] MILLS, K.L. Testing OSI protocols: NBS advances the state of the art *Data Communications*, New York 13(3):277-85, Mar 1984
- [MOS93] MOSER, L.E. et al. *Asynchronous Fault-Tolerant Total Ordering Algorithms*, SIAM Journal of Computing, 22(4):727-750, Ago 1993
- [MOT82] MOTUS, L. & VAIN, J. A set of tools for designing and evaluating communication protocols in industrial computer networks *IFAC Software for Computer Control*, Madrid Spain 1982
- [NAT85] NATIONAL BUREAU of STANDARDS *Performance measurement of OSI class 4- transport implementation*, Washington US Department of Commerce Jan 1985 (PB85-177657-NBSIR 85-3104)

- [PET77] PETERSON, J. *Petri nets Computing Surveys*, New York 9(3):223-52, Sep 1977
- [PET81] PETERSON, J. *Petri nets: theory and the modelling systems* Englewood Cliffs, Prentice Hall 1981
- [RAZ84] RAZOUK, R.R. The derivation of performance expressions for communications protocols from timed Petri Net models *Computer Communicatios Review*, New York, 14(2):210-17 1984
- [REI82a] REISIG, W. *Petri Nets* Berlin, Springer Verlag 1985
- [REI82] REISER, M. Performance evaluation of data communication systems *Proceeding of the IEEE*, New York 70(2):171-96, Feb 1982
- [SCH74] SCHRIBER, T. *Simulation using GPSS* New York, John Wiley 1974
- [SCH78] SCWETMAN, H.D. Hybrid simulation models of computer systems *Communications of the ACM*, New York 21(9):718-23, Sep 1978
- [SHI75] SHIMIZU, T. *Simulação em computador* Edgard Blucher, São Paulo, 1975
- [SOA84] SOARES, L.F.G. Especificação de protocolos através de Redes de Petri com temporização - o protocolo de acesso à barra da rede local - REDPUC, *Revista Brasileira de Computação - RBC - 3*:(3) 171-93 1983/1984 Rio de Janeiro
- [STR84] STRACK, J. *GPSS: modelagem e simulação de sistemas* Rio de Janeiro, LTC, 1984
- [TAR83] TAROUCO, L.M.R. *Projeto de redes de teleprocessamento* Porto Alegre, CPD da UFRGS, 1983 (apostila)
- [TAR86] TAROUCO, L.M.R. *Redes de computadores locais e longa distância* São Paulo, McGraw Hill, 1986
- [TAZ85] TAZZA, M. *Performance evaluation using a net theory based model* Porto Alegre, CPGCC/UFRGS, 1985 (RT n.23)
- [TAZ86] TAZZA, M. *Fundamentals of a net theory based performance evaluation model* Porto Alegre, CPGCC/UFRGS, 1986 (RT n.56)

- [TOB78] TOBAGI, F.A. et alii Modeling and measurement techniques in packet communications networks *Proceeding of the IEEE*, New York, 66(11):1423-47, Nov. 1978