

CONSIDERAÇÕES SOBRE O NOVO PARADIGMA DE CONCEPÇÃO DE REDES DE COMPUTADORES

Joberto S. B. Martins
UFPb/DSC/GRC

1.1	<u>O Paradigma</u>	3
1.3	<u>Desafios e Abordagens</u>	7
2.	MODELOS DE REFERÊNCIA	9
2.1	<u>Introdução</u>	9
2.2	<u>Tendências na Evolução do RM-OSI</u>	10
2.3	<u>RM-OSI: Proliferação de Padrões</u>	12
2.4	<u>Um Modelo de Referência para o Processamento Distribuído</u>	15
2.5	<u>RM-OSI: Impactos da Evolução do Transporte de Dados e das Tecnologias de Rede</u>	17
2.6	<u>Tendências e Posicionamento Visando um Modelo Integrador</u>	18
3.	RECURSOS DE IMPLANTAÇÃO DE REDES	22
3.1	<u>Motivação</u>	22
3.2	<u>A Arquitetura do Processador de Rede</u>	23
3.3	<u>Recursos de Implantação - Uma Abordagem para o Núcleo</u>	26
3.3.1	Gerenciamento de Memória (<i>buffers</i>)	26
3.3.2	Gerência de Temporização	28
3.3.3	Mecanismos de Comunicação Interprocesso (MCIs)	29
3.4	<u>Outras Otimizações</u>	31
4.	MECANISMOS DE TRANSPORTE DE DADOS	34
4.1	<u>Transporte de Dados - Estado da Arte e Tendências</u>	34
4.2	<u>Resumo da Funcionalidade e Problemas</u>	38
4.3.	<u>Novas Características para Redes de Alta-Velocidade e Processamento Distribuído</u>	39
4.3.1	Gerência de Conexão	40
4.3.2	Formatação das Mensagens	42
4.3.3	Controle de Fluxo	43
4.3.4	Deteção e Correção de Erros	44
5.	FERRAMENTA DE PROTOTIPAÇÃO E MODELAGEM	46
5.1	<u>Motivação</u>	46
5.2	<u>Ferramentas e Métodos para a Modelagem e Simulação - Abordagens Convencionais</u>	48
5.3	<u>Ferramenta de Prototipação e Emulação de Redes</u>	50
6.	CONCLUSÕES E CONSIDERAÇÕES FINAIS	56
7.	REFERÊNCIAS BIBLIOGRÁFICAS	59

1. PARADIGMA DE CONCEPÇÃO DE REDES

1.1 O Paradigma

Como tem acontecido com diversas áreas da informática, as redes de computadores tiveram uma evolução considerável nos últimos anos.

De maneira geral, esta evolução tem sido alavancada principalmente pelas seguintes fatores:

- O uso de fibras ópticas como meio de transmissão e
- O surgimento de novas tecnologias de redes de alta-velocidade como o FDDI (*Fiber Distributed Data Interface*) [1], o ATM (*Asynchronous Transfer Mode*) [2], o DQDB (*Distributed Queue Dual Bus*) [3] e o SONET (*Synchronous Optical Network*) [4], dentre outras. Na sua maioria, estas tecnologias fazem uso de fibras ópticas como meio de transmissão.

Na década de 70, as redes de computadores operavam tipicamente com taxas de transmissão da ordem de dezenas de milhares de bits por segundo (N Kbps) num contexto de interconexão de computadores de grande porte (*mainframes*).

Na década de 80, as tecnologias de redes locais (*LANs - Local Area Networks*) passaram a operar com taxas de transmissão da ordem de dezenas de milhões de bits por segundo (N Mbps) num contexto de interconexão de computadores de pequeno, médio e grande porte e, também, de interconexão de redes.

Na década de 90, as taxas de transmissão já são da ordem de centenas de milhões de bits por segundo (FDDI e ATM), podendo alcançar a faixa dos *gigabits* por segundo (N Gbps), num contexto de interconexão generalizada e transparente de estações de trabalho (*workstations*), computadores pessoais, computadores de grande porte e redes de computadores com os elementos citados num ambiente local, metropolitano (*MAN - Metropolitan Area Network*) ou de longa-distância (*WAN - Wide Area Network*).

A evolução obviamente continua. Neste sentido, com o advento das fibras ópticas, do processamento óptico e da utilização de técnicas de multiplexação por divisão em comprimento de onda (*WDM - Wavelength Division Multiplexing*) [5] [6], deve-se alcançar taxas de transmissão da ordem de *terabits* (10¹² bits/segundo) nos próximos anos [7]. O uso da técnica WDM associada à necessidade de ter-se múltiplos canais lógicos, capacidade de chaveamento (*switching*) e processamento compatíveis com taxas de velocidade da ordem de *gigabit* ou *terabit* por segundo, ilustra apenas algumas dos muitos desafios impostos pelas novas tecnologias de rede.

Esta evolução faz com que modelos de arquitetura e protocolos tradicionalmente utilizados nas redes e sistemas finais sejam reavaliados de forma a possibilitar a exploração completa da grande faixa de passagem (*bandwidth*) propiciada pelas novas tecnologias de rede de alta-velocidade.

A título de exemplo, alguns dos gargalos que devem ser resolvidos e alguns dos novos aspectos que devem ser considerados na concepção das redes são os seguintes:

- O *overhead* de processamento imposto pela estrutura de camadas das arquiteturas de rede. Na realidade, a vazão (taxa de transmissão de dados) disponibilizada para as aplicações é apenas uma fração da taxa de transmissão da tecnologia de rede utilizada [8] [9];
- A redução do tempo de processamento das informações por camada de protocolo. Na medida em que a taxa de transmissão aumenta, o tempo disponível para o processamento diminui. Por exemplo, um pacote de 100 octetos (*bytes*) consome 12,5 ms para transmissão a uma velocidade de 64 Kbits/s e apenas 8 μ s numa velocidade de 100 Mbits/s. Tal decréscimo impõe novos requisitos para o software de redes;
- A utilização de fibras ópticas como meio de transmissão repercute nas estratégias de controle e de tratamento de erros dado a sua maior imunidade ao ruído.

Em suma, a questão principal a ser considerada na concepção das redes de computadores com o advento das tecnologias de alta-velocidade, consiste em determinar como devem ser desenvolvidas as arquiteturas e os novos protocolos de rede de forma a que os gargalos indicados sejam superados e que a aplicação na rede desfrute de uma vazão comparável à taxa de transmissão da tecnologia sendo utilizada.

O novo paradigma de concepção de redes corresponde então a este conjunto de novas tecnologias, novos meios de transmissão e novos critérios de concepção que influenciam significativamente as ações de pesquisa e desenvolvimento na área.

Não se trata de uma nova metodologia de programação ou de uma nova estruturação da rede, mas sim de um conjunto de ações articuladas que visam:

- Otimizar o desempenho;
- Criar novos recursos de implantação e
- Simplificar a concepção.

O paradigma evocado pode ser visto como uma nova realidade de concepção.

1.2 O Processamento Distribuído

Sendo as redes de computadores um suporte básico para o processamento distribuído tem-se que o novo paradigma de concepção pode ser considerado como parte de um novo paradigma de processamento que visa o processamento distribuído.

Ambos estão interrelacionados através de alguns requisitos comuns de concepção que refletem a necessidade de comunicação no processamento distribuído.

Um sistema distribuído consiste de um conjunto de nós físicos, de suas interconexões (propiciadas por uma rede) e de um conjunto de facilidades de comunicação de alto-nível sobre estes [10].

Uma aplicação distribuída corresponde a um programa de aplicação consistindo de algumas componentes cooperantes executando em diferentes nós físicos. Tal cooperação é realizada através de mecanismos de comunicação de sistemas distribuídos que oferecem um grau razoável de transparência na comunicação.

O processamento distribuído ou a computação distribuída correspondem ao conjunto de elementos que permitem a execução de uma aplicação distribuída num sistema distribuído. Estes elementos incluem modelos estruturais e computacionais, métodos de alocação, ferramentas, linguagens e recursos de distribuição, dentre outros [10].

As novas aplicações e a própria natureza do processamento distribuído levam a um conjunto de requisitos que devem ser satisfeitos pelo suporte de rede e que correspondem em parte ao novo paradigma de concepção. Os principais requisitos a considerar são os seguintes:

- Oferecimento de uma taxa de transmissão de dados elevada para as aplicações;
- Pequenos atrasos na comunicação (baixa latência), especialmente para as comunicações tipo transação, como no caso do uso das facilidades de chamada remota de procedimentos (*RPC - Remote Procedure Call*) [11];
- Suportar várias classes e tipos de serviço (com conexão, sem conexão, datagrama, circuito virtual, etc...);
- Suportar a integração de voz e dados com serviços isócronos e assíncronos [12];

- Facilidades de endereçamento a grupos de usuários (*multicast*) visando a otimização de operações de gerenciamento e controle em sistemas distribuídos;
- Simplificação na concepção e na interface de usuário dos protocolos e
- Solução baixo-custo, na medida em que espera-se que as soluções apresentadas sejam disponibilizadas para o grande parque de computadores pessoais e estações de trabalho.

A distribuição do processamento pode ser vista como uma decorrência natural da evolução da informática. Na realidade, tem-se hoje sobre cada *bureau* sistemas com capacidade de processamento de dezenas de MIPS (Milhões de Instruções por Segundo) que tendem a se conectar cada vez mais através de redes de computadores. Este quadro é alavancado por diversos aspectos como:

- Ganhos inerentes à distribuição e compartilhamento da informação;
- Ganhos operacionais (operação, desenvolvimento, manutenção, ...) resultantes da utilização de redes em relação às soluções centralizadas baseadas tipicamente em computadores de grande porte (*mainframes*).

1.3 Desafios e Abordagens

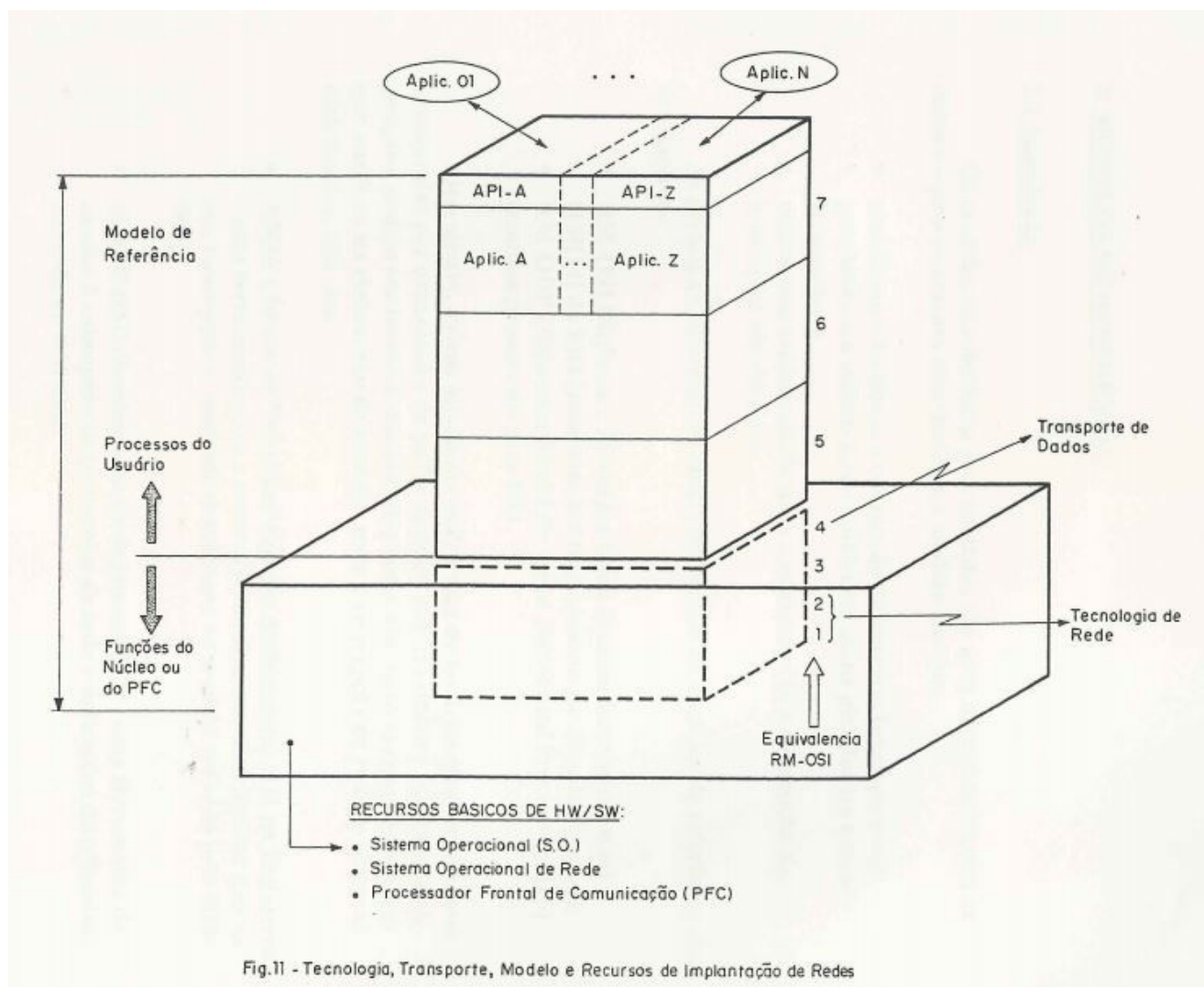
Os sistemas e o processamento distribuído podem ser enfocados sob diferentes pontos de vista. Os pontos de vista refletem diferentes ênfases sobre aspectos de preocupação na concepção do sistema como a comunicação, o suporte às aplicações e outros aspectos mais específicos como segurança e tolerância a falhas. O enfoque utilizado na discussão que segue é basicamente o seguinte (Fig. 1.1):

- A rede de computadores é um suporte de comunicação;
- As aplicações distribuídas utilizam a interface de programação de aplicação (*API - Application Programming Interface*) oferecida pela rede e
- Um modelo de referência é utilizado para a concepção da rede e das aplicações distribuídas.

Assim sendo, nosso enfoque será na evolução e otimização das redes considerando os impactos na sua concepção e na concepção das aplicações distribuídas.

Neste contexto, os focos principais de discussão serão os seguintes (Fig. 1.1):

- O capítulo 2 define um posicionamento quanto à evolução dos modelos de referência e quanto à necessidade de um modelo integrador;
- O capítulo 3 apresenta algumas contribuições propostas e realizadas no sentido de criar recursos básicos de hardware/software (HW/SW) para a concepção de redes;
- No capítulo 4 são discutidos e analisados novos mecanismos de concepção de protocolos e, em particular, do transporte de dados face a evolução das redes de alta-velocidade;
- O capítulo 5 apresenta uma proposta de abordagem sistêmica para a prototipação e emulação de redes através de ferramentas que consideram as alternativas tecnológicas existentes e emergentes a partir dos requisitos das aplicações.



2. MODELOS DE REFERÊNCIA

2.1 Introdução

Os modelos de referência padronizados são uma necessidade para as redes e o processamento distribuído na medida em que:

- viabilizam e facilitam a concepção de sistemas interoperáveis;
- possibilitam a utilização de ambientes multi-plataforma e multi-fornecedor e
- criam uma uniformidade para a estruturação e concepção das aplicações distribuídas.

As principais alternativas atuais em termos de modelos de referência são as seguintes:

- RM-OSI (*Reference Model for Open Systems Interconnection*) [13] [14] da ISO (*International Organisation for Standardization*);
- RM-ODP (*Reference Model for Open Distributed Processing*) [15] também promovido pela ISO.

Além destes, outras atividades em termos de modelos e/ou ferramentas promovidas por organismos de padronização, universidades e instituições de pesquisa podem ser também classificadas como alternativas complementares que auxiliam na elaboração de modelos para a concepção de redes e sistemas distribuídos. São eles:

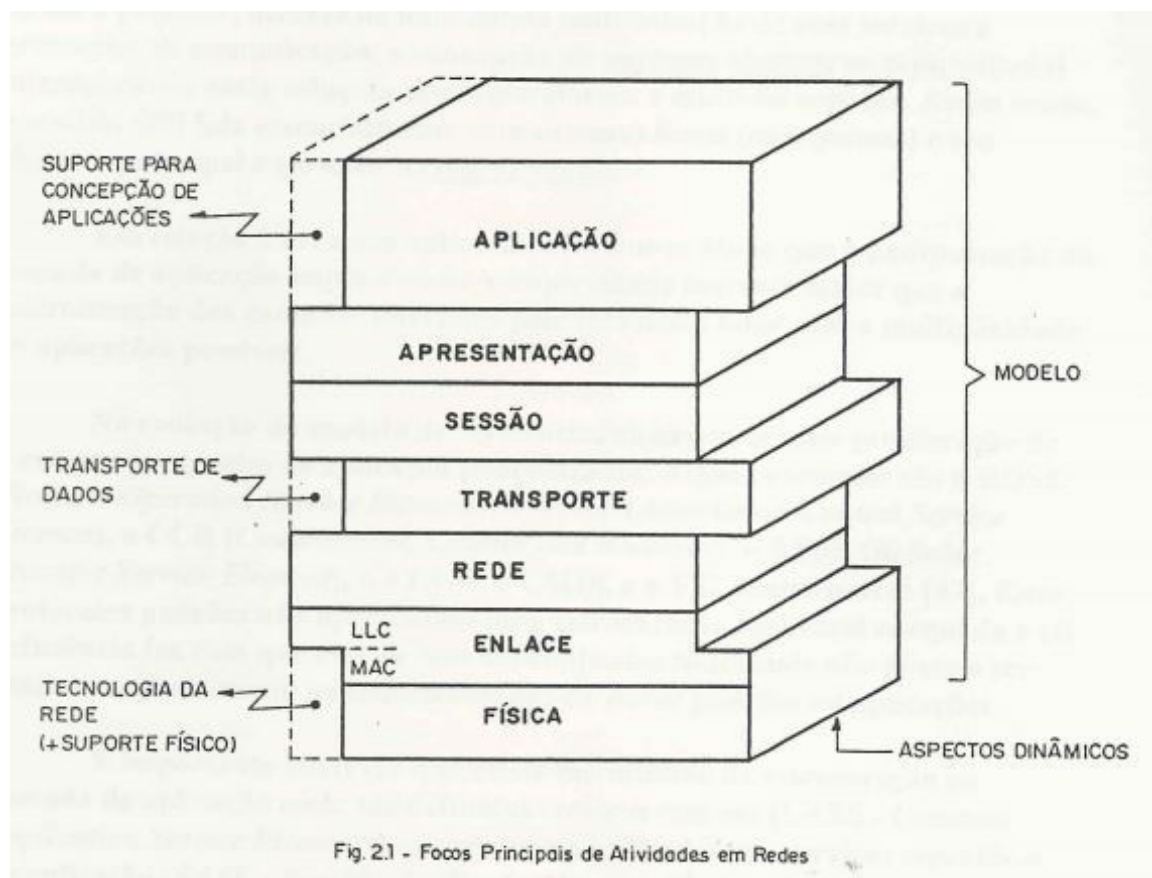
- ANSA (*Advanced Networked Systems Architecture*) [16] na Inglaterra - uma ferramenta para a concepção de sistemas distribuídos que na sua concepção é bastante semelhante ao modelo definido pelo RM-ODP;
- ISODE (*ISO Development Environment*) [17] - uma ferramenta de auxílio à concepção de protocolos de rede e aplicações distribuídas baseada no RM-OSI;
- OSCA [18] do *BellCore* - uma arquitetura de concepção de aplicações distribuídas orientada para as aplicações telemáticas e de telecomunicações.

De maneira geral, tem-se que o RM-OSI e o RM-ODP são as alternativas mais representativas em termos de modelos de referência e, entre eles, tem-se uma evolução que reflete em parte o novo paradigma de redes já evocado. Nas secções seguintes apresentamos uma discussão e um posicionamento sobre esta evolução com enfoque em termos de redes de computadores.

2.2 Tendências na Evolução do RM-OSI

Considerando a estrutura e a concepção da rede em relação à sua implantação e ao suporte oferecido para as aplicações distribuídas, tem-se observado uma evolução no modelo de referência OSI com os seguintes focos principais de atividades (Fig. 2.1):

- **Área das Aplicações** - uma tendência no sentido de expandir o escopo das aplicações e prover uma interface para o usuário (*API - Application Programming Interface*) mais homogênea com uma metodologia associada para a estruturação de suas aplicações [19][20];
- **Transporte de Dados** - uma tendência no sentido de rediscutir aspectos da operação do transporte de dados (modos de conexão, critérios de confiabilidade da transmissão, ...) e a própria concepção dos mecanismos de transporte de dados na rede, ou seja, sua implementação interna [21];
- **Tecnologia da Rede** (incluído o suporte físico) - uma tendência tecnológica muito forte no sentido das redes de alta-velocidade com taxas de transmissão da ordem de vários *gigabits* por segundo [22] e uma reestruturação dos modos e métodos de acesso no sentido de permitir a integração de serviços promovida em particular pela Rede Digital de Serviços Integrados faixa estreita e faixa larga (RDSI-FE e RDSI-FL) [23] [24].



Em seguida apresentamos uma discussão e um posicionamento sobre estes focos principais de evolução.

2.3 RM-OSI: Proliferação de Padrões

O RM-OSI é um padrão internacional (ISO IS 7498) [13] largamente aceito e difundido pela comunidade de informática. A referência [14] tem uma descrição sucinta de sua funcionalidade.

O objetivo básico do RM-OSI foi resolver o problema de heterogeneidade na interconexão dos sistemas de comunicação de dados de forma a permitir, através de uma ampla padronização de seus serviços e protocolos de comunicação, a concepção de sistemas abertos, ou seja, sistemas interoperáveis entre soluções multi-plataforma e multi-fornecedor. Assim sendo, o modelo OSI lida essencialmente com sistemas finais (*end-systems*) e seu objetivo principal é garantir a conectividade.

Em relação à área das aplicações, tornou-se óbvio que a padronização da camada de aplicação implica numa complexidade inerente maior que a padronização das camadas inferiores pelo fato desta lidar com a multiplicidade de aplicações possíveis.

Na evolução do modelo de referência, observou-se uma proliferação de serviços e protocolos de aplicação padronizados. Alguns exemplos são o ROSE (*Reliable Operation Service Element*), o ACSE (*Association Control Service Element*), o CCR (*Commitment, Control and Recovery*), o RTSE (*Reliable Transfer Service Element*), o FTAM, o CMIS, e o VT, dentre outros [47]. Estes protocolos padrões não apresentam uma estruturação funcional adequada e tal deficiência faz com que eles ou seus subconjuntos funcionais não possam ser usados conjuntamente na implementação de novos padrões ou aplicações.

É importante observar que existe um mínimo de estruturação na camada de aplicação onde são definidos serviços comuns (*CASE - Common Application Service Elements*) que são compartilhados por serviços específicos de aplicação (*SASE - Specific Application Service Elements*) [25]. Não existe porém nesta solução uma estruturação funcional adequada sem replicação de funções e de uso genérico para aplicações de informática, telecomunicações, processamento de dados e processamento distribuído. Alguns fatos que validam tal premissa são os seguintes:

- Aspectos funcionais como a confiabilidade de uma operação são replicados entre diferentes protocolos CASE (RTSE, ROSE e CCR);
- A estruturação dos serviços de telecomunicação e serviços telemáticos do CCITT tem um modelo de sinalização nº 7 (CCITT #7) não completamente alinhado com o modelo OSI [15] [26].

Um fato concreto observado em relação à camada de aplicação é que ocorreu uma proliferação de padrões resultante da tentação natural de

estender o RM-OSI para, além da conectividade, atender às necessidades das aplicações e, em particular, das aplicações distribuídas. O termo *middleware* é hoje utilizado para ilustrar as tentativas de equacionar a multiplicidade de serviços padronizados disponíveis de forma a tornar a tarefa de programação das aplicações distribuídas menos complexas e mais eficientes para o usuário de uma rede.

Esta deficiência atual do RM-OSI tem sido abordada no contexto deste modelo pelo seguinte conjunto de ações:

- Uma estruturação da aplicação em perfis básicos de utilização. Tais perfis são os seguintes:
 - .Transferência de arquivos;
 - .Transferência de mensagens;
 - .Operações remotas ou serviços cliente/servidor simples; e
 - .Processamento de transações.
- A definição de perfis arquiteturais (MAP, TOP, MAP/TOP, ...), perfis funcionais e APIs para facilitar a implantação dos protocolos da camada de aplicação e as próprias aplicações;
- A elaboração de um novo modelo de referência mais abrangente.

A primeira abordagem corresponde ao princípio de empilhamento de protocolos (denominado *toolkit*) onde os serviços de aplicação básicos (FTAM, X.400, operações remotas e processamento de transações) são utilizados como suporte para os novos serviços que venham a ser criados ou desenvolvidos. A título de exemplo, o EDI (*Electronic Data Interchange*) [27] pode utilizar o X.400 ou o FTAM como transporte para suas mensagens.

Neste sentido, as duas abordagens mais importantes em relação as aplicações distribuídas são as operações remotas e o processamento de transações (*TP - Transaction Processing*). O serviço de operação remota (*ROS - Remote Operation Service*) é uma facilidade oferecida pelo modelo que corresponde a uma extensão do conceito de chamada remota de procedimentos (*RPC - Remote Procedure Call*) [28] onde as aplicações são estruturadas segundo o modelo cliente/servidor [29]. O serviço de processamento de transações por sua vez provê mecanismos de passagem de mensagem e de sincronização para as aplicações de mais alto nível que façam uso do mesmo. Ambos os serviços são básicos para a concepção de aplicações distribuídas.

A segunda abordagem corresponde a definição de perfis arquiteturais, perfis funcionais e APIs. Os perfis arquiteturais e funcionais são ações complementares à padronização empreendidas por empresas privadas (Boeing, General Motors, ...), instituições governamentais e associações (ABNT, BRISA, CCITT, ISO, SPAG, ...), associações de classe e científicas (IEEE), etc..., visando basicamente:

- Facilitar a implantação do modelo; e
- Facilitar a programação e a portabilidade do software aplicativo (p. ex.: em relação ao código fonte).

Os exemplos mais divulgados de perfis arquiteturais são as arquiteturas TOP (*Technical Office Protocols*) [30], promovida inicialmente pela Boeing, e o MAP (*Manufacturing Automation Protocols*) [30], promovido inicialmente pela General Motors. O TOP e o MAP correspondem a uma definição de arquitetura orientada para as aplicações de automação de escritórios e automação industrial respectivamente [31].

Os perfis funcionais são também documentos complementares aos padrões que visam especificar completamente a implementação de um determinado serviço/protocolo em relação a conjuntos de aplicações ou em relação a uma bateria de teste que deverá ser efetuada para a validação do mesmo. Em se tratando de protocolos de aplicação, especifica-se igualmente a sua interface de programação para garantir a homogeneidade da interface e a portabilidade do código do usuário.

Em resumo, tem-se hoje que o RM-OSI endereça efetivamente o problema da conectividade mas carece de um tratamento mais elaborado no nível da aplicação. A abordagem atual (perfis funcionais e empilhamento de protocolos) não é suficiente para o processamento distribuído ou para o suporte dos serviços de telecomunicações. A elaboração de uma abordagem mais abrangente pela ISO que focaliza mais nos problemas de interação entre aplicações é o objetivo do modelo para o processamento distribuído aberto (RM-ODP) [15], resumido adiante.

2.4 Um Modelo de Referência para o Processamento Distribuído

A elaboração de um modelo de referência para o processamento distribuído aberto (RM-ODP) é uma tarefa de longo-prazo sendo desenvolvida pela ISO com previsão de obtenção de resultados a nível de proposta de padrão internacional (*DIS - Draft International Standard*) em 1995.

O RM-ODP define projeções ou pontos de vista (*viewpoints*) que fornecem descrições ou abstrações do sistema de processamento distribuído [16]. Nesta conceituação, a especificação e o projeto de um sistema distribuído são considerados segundo diferentes pontos de vista que, em suma, enfatizam determinados aspectos do sistema. O sistema distribuído é assim representado por um modelo segundo uma determinada projeção. As projeções ou pontos de vista considerados no RM-ODP são os seguintes:

- Ponto de vista da empresa:
Esta projeção focaliza os requisitos, a estrutura e a organização dos negócios. A título de exemplo, as informações disponíveis para os usuários e a estrutura organizacional da empresa são aspectos considerados nesta projeção.
- Ponto de vista de informação:
Esta projeção focaliza na troca de informação e na estrutura e fluxo dos modelos de informação. Esta projeção descreve basicamente como a informação é distribuída através do sistema.
- Ponto de vista da computação:
Esta projeção focaliza no processamento distribuído, considerando suas componentes (componentes da aplicação) e interações, independentemente dos sistemas operacionais e de comunicação. Alguns dos aspectos considerados nesta projeção são a transparência de acesso, os mecanismos de passagem de parâmetros e a modularidade da aplicação, dentre outros.
- Ponto de vista de engenharia:
Esta projeção focaliza nos problemas de configuração, desempenho, distribuição e infra-estrutura. Alguns dos aspectos considerados nesta projeção são a gerência de processos, os espaços de endereçamento e os mecanismos de compilação e de sistema operacional, dentre outros.

- **Ponto de vista da tecnologia:**
Esta projeção focaliza nos componentes de hardware e software do sistema. Ela corresponde a um mapeamento dos modelos anteriores nas ferramentas operacionais que constituirão o sistema distribuído.

Cada projeção constitui uma abstração do sistema distribuído e as suas saídas são as seguintes [15]:

- **Empresa:** requisitos do usuário;
- **Informação:** especificação e projeto conceitual;
- **Computação:** projeto e desenvolvimento de software;
- **Engenharia:** blocos básicos e
- **Tecnologia:** componentes físicos do sistema concebido.

A modelagem segundo o ODP segue a padronização das redes de computadores preservando assim o RM-OSI.

2.5 RM-OSI: Impactos da Evolução do Transporte de Dados e das Tecnologias de Rede

As evoluções que estão ocorrendo no transporte de dados e nas tecnologias de rede podem ter dois tipos de impacto em relação ao modelo de referência OSI:

1. Podem resultar numa redifinição completa do modelo e de seus princípios de concepção ou
2. Podem forçar uma otimização do mesmo.

A redifinição do modelo de referência pode certamente ser realizada de diferentes maneiras. No entanto, considerando que o principal motivador desta mudança é a necessidade de um maior desempenho global, tem-se que ela deve consistir essencialmente de uma verticalização da funcionalidade do modelo como sugerido em [21] e [32]. Neste caso as ações técnicas tomadas são as seguintes:

- Minimização do número de camadas do modelo;
- Agregação de funções;
- Posicionamento unívoco de funções no modelo (p. ex.: o controle de fluxo seria executado em uma camada e não em várias como ocorre no RM-OSI); e
- compartilhamento eficiente de memória comum.

Outro tipo de impacto que pode ocorrer é a otimização da implantação do modelo e de seus protocolos. Esta estratégia é a mais pragmática na medida em que:

- Preserva a estrutura hierárquica de camadas independentes onde as camadas de nível superior solicitam serviços às camadas de nível inferior;
- Procura otimizar a implementação interna das camadas ou a implantação do conjunto de camadas através de alternativas do tipo:
 - .Concepção de circuitos integrados especializados de rede;
 - .Paralelização na execução de funções;
 - .Otimização dos mecanismos de sistema operacional;
 - .Otimização da arquitetura do hardware de suporte (interface de comunicação).

Na secção seguinte fazemos uma análise global e definimos um posicionamento sobre a evolução dos modelos de referência no contexto do novo paradigma de redes.

2.6 Tendências e Posicionamento Visando um Modelo Integrador

Nossa visão sobre as tendências de evolução dos modelos de referência pode ser sumarizada da seguinte forma:

- A abordagem de otimização do modelo OSI deve prevalecer;
- A adoção do RM-ODP é uma estratégia de longo-prazo que não deve excluir o modelo atual;
- O modelo OSI pode ser utilizado como base de concepção para o processamento distribuído; e
- As novas tecnologias e evoluções de rede deverão ser absorvidas dentro do modelo atual com algum nível de otimização do mesmo.

O primeiro posicionamento é estratégico na medida em que ele considera a implementação do modelo. A abordagem que verticaliza agregando funções e minimizando o número de camadas da rede é, em princípio, mais eficiente do ponto de vista da implementação que a abordagem convencional multi-camadas independentes. No entanto, os seguintes fatores devem ser levados em conta:

- Em relação ao desempenho, não está claro o nível de melhora que pode ser obtido nos modelos convencionais através das técnicas de otimização já evocadas. Este é um tema de pesquisa atual que foi impulsionado especificamente pelo surgimento das tecnologias de rede de alta-velocidade e pela necessidade de tentar manter-se a base instalada de redes. Assim sendo, o critério desempenho não deve ser utilizado para descartar o modelo atual. O critério desempenho certamente descarta algumas implementações atuais mas não o modelo atual.
- A interoperabilidade é um fator decisivo para as redes na medida em que as soluções hardware/software (HW/SW) são majoritariamente multi-plataforma e multi-fornecedor. Este critério torna-se bastante complexo na estratégia de verticalização do modelo na medida em que:
 1. Existe uma compactação funcional no novo modelo vertical semelhante ao proposto em [21];
 2. Existe a necessidade de um consenso quanto à agregação funcional, ao número de camadas e aos protocolos utilizados de forma a garantir a comunicação par-a-par (*peer-to-peer*) e as APIs utilizadas pelo usuário da rede. A título de exemplo, o RM-OSI consumiu aproximadamente 10 anos entre os processos de definição do modelo e disseminação do mesmo que, na prática, corresponde à interoperabilidade.

Consequentemente, o critério interoperabilidade sugere também a preservação do modelo atual.

- A base instalada de redes seguindo a estratégia do modelo multi-camadas independentes (RM-OSI, TCP/IP ou outros modelos com 7 camadas) é muito grande. Alguns números que nos fornecem uma idéia da base instalada são os seguintes:

N> 40.000 redes SNA - *Systems Network Architecture* [33];

N=720.000 nodos (hospedeiros) na rede Internet (em Janeiro/92) com previsão de exaustão da capacidade de endereçamento do TCP/IP em 1995 [34].

Isto representa um investimento enorme em pesquisa, desenvolvimento e em base instalada que dificilmente pode ser mudado a curto-prazo ou médio-prazo.

Assim sendo, a abordagem de otimização do modelo é, na nossa opinião, a que deve prevalecer e algumas das ações técnicas correspondentes a esta abordagem são discutidas nas secções 3 e 4 deste documento.

Considerando que o modelo hierárquico em camadas da ISO prevalece, as questões subsequentes que podem ser discutidas são as seguintes:

1. Como enfocar o problema da replicação funcional e a inflação de padrões discutida anteriormente?
2. Qual a relação entre o RM-OSI e o RM-ODP?
3. O RM-OSI pode ser utilizado para o processamento distribuído?

O problema da replicação funcional tem como solução o princípio do empilhamento de protocolos já discutido.

A inflação de padrões a nível da aplicação e, também, de outras camadas do modelo corresponde a uma ampliação da complexidade do projeto conceitual das rede. Neste caso, as opções quanto aos perfis arquiteturais e funcionais tornam-se mais elaboradas e como solução, vislumbramos a concepção de ferramentas especializadas que auxiliam o usuário nesta fase de projeto. Uma proposição neste sentido é discutida na secção 5 deste documento.

A relação entre o RM-OSI e o RM-ODP é de complementação de soluções e não de concorrência. Alguns dos fatos que apontam nesta direção são os seguintes:

- O RM-ODP corresponde a uma visão mais abrangente e orientada para o processamento distribuído que leva em conta aspectos (projeções ou pontos de vista) que não foram considerados estruturalmente no RM-OSI. Além disso, o RM-

ODP faz uso do RM-OSI quando da especificação dos pontos de vista de computação e engenharia;

- A abordagem RM-ODP é de longo-prazo. No curto-prazo e médio-prazo o RM-OSI deve ser utilizado para suportar as aplicações distribuídas; e
- Conforme discutido em [15], o RM-ODP poderá permitir uma convergência com os serviços de telecomunicações (operadoras, serviços telemáticos, etc...) através da adoção das novas tecnologias de rede de alta-velocidade que oferecem um conjunto de serviços mais adequado para o atendimento das aplicações de telecomunicações e de processamento de dados. Em particular, existem semelhanças conceituais e estruturais entre o RM-ODP e o modelo de redes inteligentes de telecomunicações [15] que podem evoluir para uma convergência numa ótica de sistemas distribuídos.

A utilização do RM-OSI para o processamento distribuído pode ser considerada na medida em que alguns dos mais importantes componentes e requisitos deste tipo de aplicação [10] são atendidos:

- Transparência;
- Desempenho;
- Suporte de comunicação; e
- Facilidades de configuração.

A transparência pode ser obtida na distribuição dos recursos e na comunicação através do uso do serviço de diretório e dos mecanismos implícitos de distribuição existentes nos serviços de operação remota (RPC) e de processamento de transações.

O desempenho é um fator importante que, na abordagem de otimização, resulta numa operação satisfatória para o sistema distribuído conforme resultados discutidos na secção 3 e nas referências [35], [36], [37] e [38].

O suporte de comunicação RM-OSI oferece as alternativas de passagem de mensagem e chamada remota de procedimentos que são as mais utilizadas para o processamento distribuído [10]. Mecanismos orientados a objeto como descrito em [39] são uma terceira alternativa para o suporte de comunicação que não é atendida pelo RM-OSI.

As facilidades de configuração para as aplicações distribuídas são um gargalo no RM-OSI. Linguagens de configuração não são previstas no modelo e, quando desenvolvidas, fazem uso de facilidades de gerenciamento padronizadas que, por sua vez, não são largamente disseminadas na

comunidade de informática [40]. Outras deficiência óbvia do RM-OSI é a ausência de um modelo global que, no caso, deverá ser provido pelo ODP.

Como consideração final é importante considerar a robustez do modelo. Historicamente, o RM-OSI tem sido robusto na medida em que tem sido capaz de absorver novas tecnologias com modificações mínimas [41]. Nada impede que ele preserve esta característica de modelo integrador.

3. RECURSOS DE IMPLANTAÇÃO DE REDES

3.1 Motivação

A taxa de transmissão das redes de comunicação de dados evoluiu na última década de alguns milhares para centenas de milhões de bits por segundo, correspondendo a um acréscimo de aproximadamente 5 vezes na sua ordem de grandeza.

Por sua vez, a velocidade de processamento das UCPs (Unidade Central de Processamento) utilizadas como recurso na implantação dos softwares de rede e dos sistemas operacionais evoluiu aproximadamente de 2 a 3 vezes na sua ordem de grandeza [42]. A título de exemplo, consideremos um sistema conectado a um enlace de 600 Mbps (taxa de acesso do usuário numa rede ATM) que necessita executar aproximadamente 100 instruções para cada quadro recebido. Neste caso, o sistema deve ter uma capacidade computacional de aproximadamente 118 MIPS de forma a poder garantir fluxos arbitrários de quadros de 64 palavras (*bytes*).

Tal discrepância entre as evoluções das tecnologias de rede e da capacidade de processamento das UCPs resulta em novos requisitos para a concepção e implantação das redes. Neste novo contexto tem-se a necessidade de otimizar os recursos de implantação de redes na medida em que a disponibilidade de capacidade computacional torna-se escassa. Além disso, algumas novas aplicações de rede e o processamento distribuído requerem essencialmente um bom desempenho global com:

- Requisitos de tempo de resposta baixos (latência);
- Vazão elevada na transferência de dados; e
- Confiabilidade.

Dentro de uma ótica de melhoria dos suportes e mecanismos de implantação existentes, as principais ações técnicas empreendidas para otimização dos mesmos podem ser as seguintes:

1. Implantação de protocolos e arquiteturas em circuitos integrados;
2. Novas estruturas de concepção de processadores frontais de comunicação (PFCs) (*front-ends*); e
3. Definição ou otimização de recursos de sistema operacional adequados para a implementação de protocolos de redes.

Em seguida apresentaremos uma abordagem em termos das ações técnicas "2" e "3" e discutimos a alternativa "1" juntamente outras ações complementares.

3.2 A Arquitetura do Processador de Rede

A arquitetura utilizada na implementação do processador frontal de rede, também denominado de processador de rede, é fundamental para o desempenho global do sistema.

A arquitetura dos processadores de rede pode variar significativamente entre as alternativas abaixo indicadas:

1. O processamento do software de redes é quase que completamente executado pela UCP principal (hospedeiro). Embora esta seja a solução menos indicada do ponto de vista de concepção, ela é largamente utilizada em boa parte da base instalada atual de redes;
2. Utiliza-se um PFC numa estrutura clássica de arquitetura de redes (camadas independentes e hierárquicas) onde o número de camadas efetivamente integradas no PFC pode variar consideravelmente (de 2 a 7 camadas). Esta solução corresponde ao enfoque de nossa contribuição e, a nosso ver, deve corresponder ao tipo de abordagem de curto e médio prazo para a otimização deste tipo de suporte para as redes; e
3. Utiliza-se estruturas de hardware especializadas fazendo uso de *transputers*, processadores e coprocessadores dedicados, estruturas de processamento paralelo e circuitos integrados de aplicação específica (ASICs) na concepção da interface de rede (PFC). Esta solução começa a ser desenvolvida a nível de pesquisa e deve tornar-se a próxima geração de PFCs.

É importante observar que em alguns casos soluções híbridas entre as alternativas indicadas são realizáveis.

A arquitetura do PFC proposto segundo a abordagem "2" acima para suportar a execução de uma arquitetura de protocolos multi-camada é basicamente a seguinte (fig. 3.1):

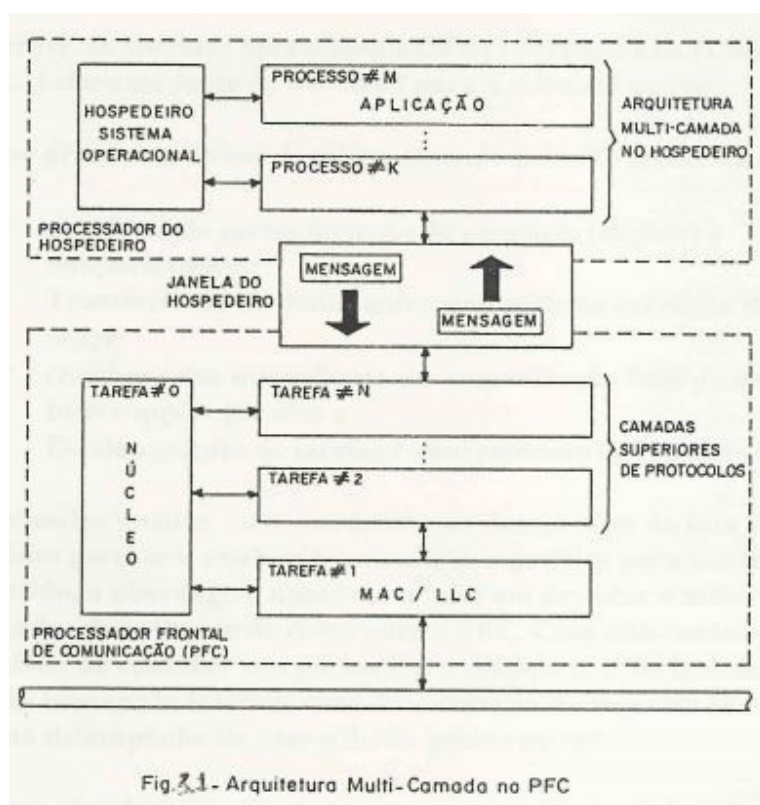
- Um hardware especializado como descrito em [43] que faz uso de um mecanismo de acesso direto à memória (DMA), memória comum e um barramento local que visam minimizar a contenção de barramento entre o PFC e a UCP principal e
- Uma arquitetura multi-camada executando sob o controle de um núcleo especializado que corresponde a uma otimização dos recursos de sistema operacional.

A arquitetura de hardware do PFC inclui circuitos dedicados para o controle e a gerência da tecnologia de rede utilizada e um processador local

com capacidade computacional suficiente para a execução do software multi-tarefa (p. ex.: microprocessadores 286, 386,...).

A arquitetura multi-camada do PFC é composta de um núcleo, um mecanismo de comunicação com o hospedeiro e as camadas de protocolo. O núcleo fornece facilidades de sistema operacional (multi-tarefa, comunicação interprocesso, prioridades, etc...) para suportar a execução concorrente das camadas de protocolo. O mecanismo de comunicação é responsável pela comunicação com o hospedeiro no nível (camada) escolhido pelo projetista do software. A aproximação utilizada na concepção deste PFC consistiu principalmente em otimizar o núcleo de sistema operacional implementado e, também, a estruturação do hardware. Este último ponto é discutido em [43] (Fig. 3.2).

Segue uma apresentação dos gargalos e de uma abordagem para a implementação de um núcleo otimizado e eficiente orientado para a implantação de arquiteturas de redes (software).



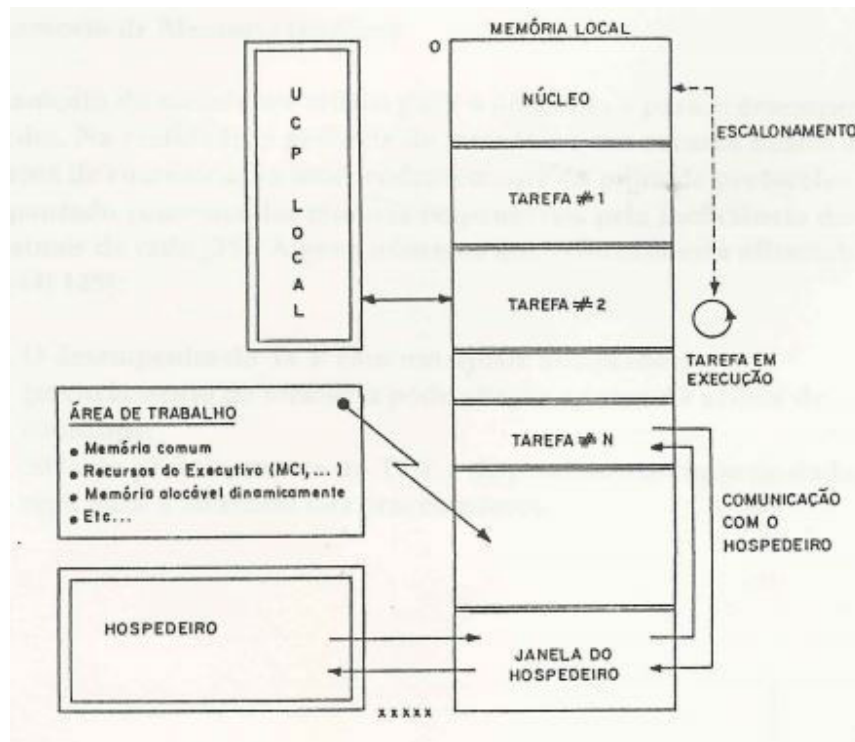


Fig. 3.2 - Alocação de Tarefas e Recursos na Memória

3.3 Recursos de Implantação - Uma Abordagem para o Núcleo

Infelizmente, os sistemas operacionais (SOs) convencionais (Unix e *Unix-likes*, VMS, etc...) são uma fonte de *overhead* para o software de rede

As causas principais do *overhead* introduzido pelo SO são as seguintes:

- *Overhead* do gerenciamento de memória (*buffers*) e temporizadores;
- Transferência de dados entre processos na execução do software de redes;
- *Overhead* dos mecanismos de comunicação interprocesso (MCIs);
- Interrupções geradas; e
- Escalonamento de tarefas e chaveamento de contexto.

Numa primeira análise, estes *overheads* são decorrentes do fato dos SOs serem de propósito geral sem nenhuma orientação específica para o software de redes. Neste sentido, a abordagem imediata consiste em deslocar o maior número possível de camadas do software de redes para o PFC. Com este cenário tem-se então a necessidade de conceber um núcleo especializado e, principalmente, otimizado na sua concepção interna, visando a redução do impacto dos recursos listados acima no desempenho da execução do pacote de redes.

Segue uma abordagem proposta para a concepção do núcleo de suporte à implantação de software de redes.

3.3.1 Gerenciamento de Memória (*buffers*)

O gerenciamento da memória é crítico para a operação e para o desempenho do software de redes. Na realidade, a gerência de memória é um recurso básico de qualquer subsistema de comunicação independentemente da pilha de protocolos utilizada. Ele é apontado como um dos maiores responsáveis pela ineficiência das implementações atuais de rede [35]. Alguns números que reforçam esta afirmativa são os seguintes [44] [45]:

- O desempenho do TCP com um ajuste adequado no gerenciamento de memória pode chegar a taxas da ordem de 800Mbps; e
- 50% do processamento do TCP é dispendido em cópia de dados da rede para a memória dos processadores.

A funcionalidade básica do módulo de gerenciamento de memória é a seguinte:

- Transferência de dados da rede para a memória de trabalho;
- Transferência de dados da memória de trabalho para a rede ;
- Transferência de dados entre camadas de protocolo; e
- Reordenamento de informações nas camadas onde este tipo de funcionalidade é previsto.

As maiores fontes de ineficiência para o gerenciamento de memória são:

- Os tempos de leitura e escrita, especialmente nas situações onde transfere-se dados entre camadas; e
- A contenção do barramento da máquina e o seu tempo de transferência.

Existem diversos métodos de alocação e liberação de memória baseados em segmentação, blocos de tamanhos fixos e lista de blocos de tamanho variável [46]. A abordagem sugerida utiliza um método baseado numa lista de blocos de tamanho variável com primitivas ("`x_get_memory`" / "`x_return_memory`") que alocam/liberam blocos de memória de até 64K para as camadas do software de rede sem que seja utilizado um esquema de proteção para os mesmos.

A justificativa desta abordagem é basicamente a seguinte:

- Os blocos de memória de tamanho variável tentam atender os diferentes tipos de demandas existentes para as camadas de protocolos executando no PFC. Além disso, tem-se que a alternativa de utilização de filas FIFO com *buffers* físicos de tamanho arbitrado não é a solução mais eficiente. Isto visto que ela necessita de um gerenciamento lógico da fila entre os diversos usuários do sistema de memória; e
- A não implementação de mecanismos de proteção é uma medida de simplificação e otimização do gerenciamento de memória no núcleo. Na realidade, pressupõe-se que o software de rede utilizado neste tipo de hardware é um conjunto fixo e bem testado de tarefas. Em outras palavras, assume-se que as implementações de protocolos são estáveis na sua concepção e execução.

Em relação à otimização do fluxo de dados entre camadas, a solução consistiu na passagem de *buffers* entre camadas através de apontadores com um consequente compartilhamento de memória global entre as diversas execuções das camadas da rede.

No que diz respeito à transferência de dados da rede para a memória de trabalho e vice-versa, a solução consiste em fazer uma transferência direta para a memória comum (DMA) sem a intervenção do processador local. Tal estratégia tem as seguintes vantagens:

- Não interrompe a UCP local;
- Evita mudança de contexto durante a recepção do quadro; e
- Permite um processamento paralelo de um quadro durante a recepção de outro.

Recentemente, Meleis [35] elaborou um modelo e fez uma avaliação deste tipo de solução demonstrando uma otimização de até 50% nos recursos computacionais dispendidos no tratamento da recepção de quadros com esta arquitetura de PFC.

É importante considerar que no projeto utilizou-se um barramento e memórias rápidas para permitir baixa contenção no acesso do mecanismo de DMA e do processador local ao barramento.

3.3.2 Gerência de Temporização

A temporização é uma facilidade essencial para os protocolos na medida em que é uma facilidade largamente utilizada na sua concepção, na sua implementação e, além disso, tem grande influência no seu desempenho.

As características mais importantes para as facilidades de temporização são as seguintes:

- Ter uma boa resolução temporal de maneira a permitir ajustes finos nos protocolos de baixo-nível (camadas inferiores) onde os requisitos de tempo são mais fortes e
- Ter uma implementação eficiente de maneira a minimizar o tempo de execução na manutenção e atualização dos múltiplos temporizadores ativos.

As formas de implementação de temporizações são variadas e, dentre as mais importantes, destacamos [47]:

- Temporizadores implícitos onde cada informação tratada pelas camadas (pacote, quadro, mensagem, etc...) tem um temporizador associado;
- Temporizadores implementados como uma tarefa especial que ao invés de enviar mensagens bloqueia as tarefas de maneira a gerar as respectivas temporizações; e
- Temporizadores como eventos independentes lançados pelas tarefas.

Na nossa abordagem as temporizações são eventos independentes utilizados para a sincronização de tarefas acionados através de um conjunto de primitivas (*set, get, reset, read e kill*) com as seguintes características principais:

- Um algoritmo eficiente de implementação do tipo "atualiza todos pela atualização do primeiro" (*update all by updating the first entry*);
- Múltiplos temporizadores podem ser lançados pela mesma tarefa e provê-se adicionalmente facilidades explícitas de liberação de temporizadores para otimizar a eficiência do algoritmo em si e evitar o envio de sinais sem significação para as tarefas; e
- Os estouros de temporização são sinalizados para as tarefas ou podem ser explicitamente recuperados por uma tarefa ativa.

A temporização é uma função crítica até aproximadamente o nível de transporte. Na maioria dos SOs os mecanismos de temporização criados não permitem uma granularidade pequena da ordem de centenas de microsegundos com um bom grau de precisão [48] [49]. Assim sendo, esta abordagem vem a atender de forma genérica ao novo paradigma de concepção de redes com tecnologias de alta-velocidade onde os tempos de chegada entre pacotes exigem uma boa resolução dos temporizadores.

3.3.3 Mecanismos de Comunicação Interprocesso (MCIs)

Um outro mecanismo básico na concepção do software de rede é o de comunicação entre processos (tarefas). A nível dos SOs os mecanismos mais utilizados são as filas, memória compartilhada, tubos (*pipes*) e caixas postais (*mailbox*), dentre outros [50] [51].

Na implantação de protocolos tem-se uma estrutura hierárquica onde as camadas passam mensagens através de primitivas de serviço [47]. Neste caso, para implantações tanto mono-tarefa como multi-tarefa o mecanismo básico requerido é a passagem de mensagem (*message passing*) [52]. Do ponto de vista

das tarefas/processos tem-se duas alternativas básicas para a implementação de um mecanismo de passagem de mensagens:

- Uma mensagem estruturada é passada e
- Um apontador de mensagem é passado.

Nossa abordagem consiste na utilização da segunda alternativa. O mecanismo de mensagens proposto não bloqueia primitivas do tipo envia/recebe (*send/receive*) seguindo um ordenamento FIFO (*First-In First-Out*) para cada tarefa em questão. Como nos casos anteriores, a implementação no núcleo é otimizada visando uma melhoria do desempenho global do software da rede.

As demais facilidades incluídas no núcleo correspondem às facilidades básicas de SO e não têm um relacionamento direto com a estrutura dos protocolos. Na sua essência, a estratégia de otimização foi diretamente ligada ao desempenho e otimização da implementação do núcleo. A título de exemplo, listamos as facilidades adicionais embutidas no núcleo:

- Escalonamento de tarefas;
- Prioridades;
- Sincronização de tarefas; e
- Comunicação com o hospedeiro (processador principal).

A comunicação com o hospedeiro é um caso particular de passagem de mensagens com uma forte dependência do hardware instalado na medida em que passa pelo barramento da máquina.

3.4 Outras Otimizações

Um outro tipo de otimização pode ser obtido na implantação das redes através da paralelização.

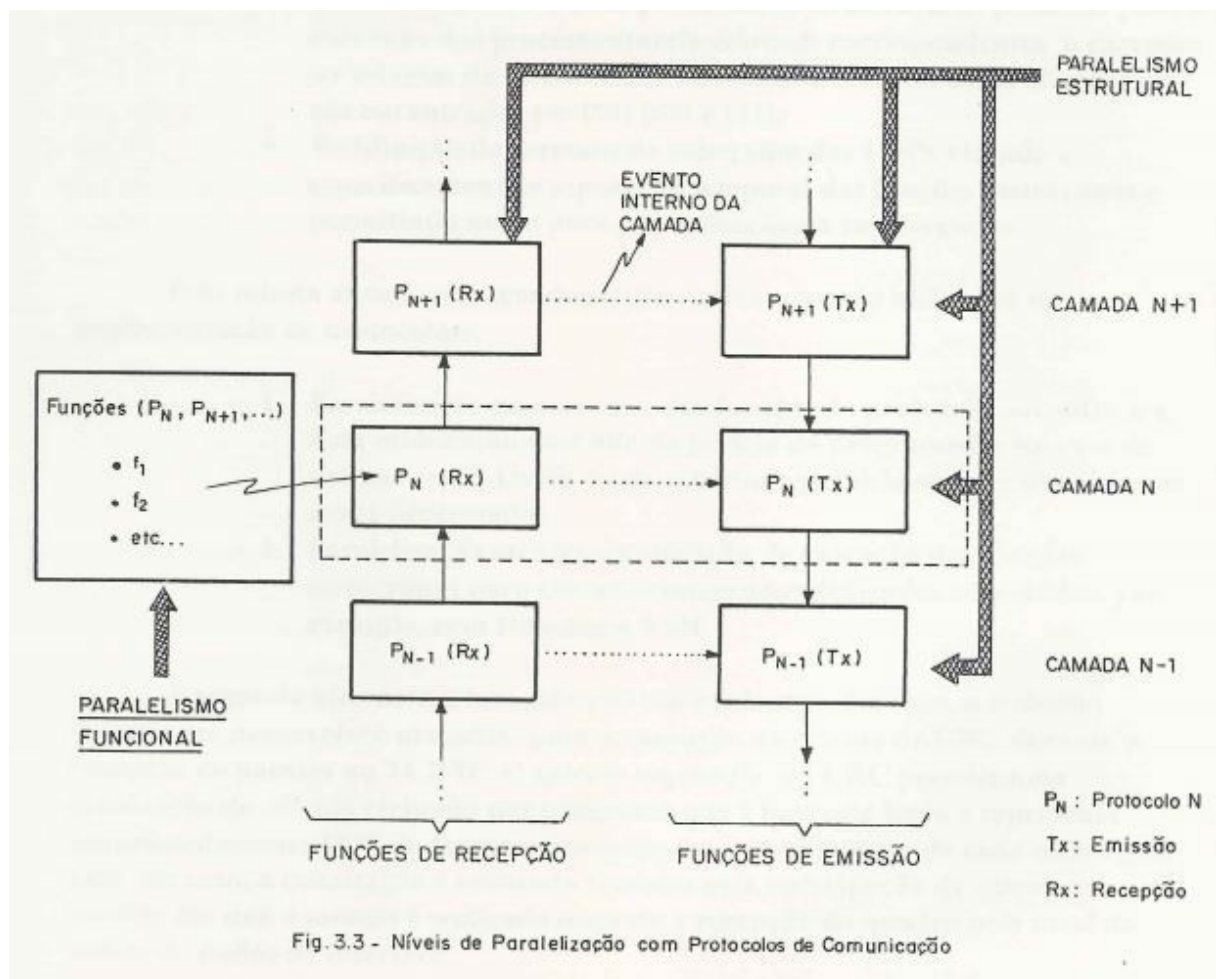
A estruturação típica dos sistemas de comunicação é em camadas de protocolos hierárquicas. Os níveis de paralelização possíveis neste tipo de estrutura são basicamente os seguintes:

- Entre camadas de protocolos e
- Na execução da máquina de protocolo em cada camada ou subcamada.

Uma outra distinção pode ser feita em relação aos níveis de paralelismo que podem ser obtidos (Fig. 3.3):

- Tem-se o paralelismo estrutural que pode ser obtido a partir da estrutura de camadas e da dicotomia entre as fases de emissão e recepção de informações para cada camada de protocolo;
- Tem-se o paralelismo funcional que é baseado na configuração e execução de funções do protocolo. Tal nível de paralelismo depende da sintaxe de concepção do protocolo, ou seja, da maneira como o protocolo tem seu quadro formatado e da sequência de execução das funções de sua máquina de estados (mecanismo de controle de fluxo, controle de erros, etc...).

As funções paralelizáveis neste tipo de estrutura podem ser de dois tipos. Algumas funções são sequenciais. Por exemplo, a segmentação de uma informação só pode ser realizada após a definição do roteamento. Outras funções são independentes ou assíncronas na medida em que podem ser executadas independentemente de outras funções. Exemplos deste tipo de função são o roteamento, o controle de erros na recepção da informação ou funções disparadas por eventos como temporização.



As abordagens que visam a otimização na implementação dos protocolos através da paralelização funcional e arquitetural adotam as seguintes estratégias:

- Utilização de HWs multiprocessados ou estruturas paralelas para a execução dos processos/tarefas/*threads* correspondentes a camadas ou estágios da estrutura de camadas. Exemplos desta abordagem são encontrados em [53] [54] e [21]; e
- Redefinição do formato do cabeçalho das UDPs visando especificamente a separação temporal das funções assíncronas e permitindo assim uma paralelização na sua execução.

Esta última abordagem tem duas alternativas que são utilizadas na implementação de protocolos:

1. Paralelização na execução das funções do protocolo por software. Esta otimização depende da perícia do programador no caso de sistema multi-tarefa ou da estrutura paralela no caso dos sistemas multi-processados.
2. Paralelização através da alocação da execução das funções assíncronas para circuitos integrados dedicados concebidos, por exemplo, com tecnologia ASIC.

A segunda alternativa tem sido por nós explorada. No caso, o trabalho consiste em desenvolver um ASIC para a execução do cálculo de CRC durante a recepção de pacotes no TCP/IP. O cálculo *on-the-fly* do CRC permite uma otimização do cálculo efetuado por programa que é bastante lento e representa aproximadamente 60% do tempo consumido no processamento de cada mensagem [36]. No caso, a otimização é realizada também pela antecipação de cálculo na medida em que o mesmo é realizado durante a recepção do quadro pelo nível de enlace de dados da interface.

A referência [55] apresenta um outro exemplo desta abordagem para a RDSI. De maneira geral, os coprocessadores de rede existentes hoje (*ethernet, FDDI, token-ring, token-bus, etc...*) são um outro exemplo desta alternativa. A implementação adequada destes circuitos num PFC como proposto na secções 3.2 e 3.3 corresponde efetivamente a uma otimização pela paralelização de funções de HW/SW.

4. MECANISMOS DE TRANSPORTE DE DADOS

4.1 Transporte de Dados - Estado da Arte e Tendências

Na medida em que as comunicações de dados avançam na direção das redes de alta-velocidade, faz-se necessário uma redifinição de alguns conceitos e estruturas atualmente utilizadas em redes. Um das áreas de enfoque e atenção da comunidade de pesquisa tem sido os mecanismos de transporte de dados para redes locais e de longa-distância [56] [57] que, do ponto de vista do modelo de referência RM-OSI, corresponde aproximadamente à funcionalidade da camada de transporte.

Em relação a este tema, existe uma forte discussão e uma tendência no sentido de propor alternativas para os protocolos de transporte padronizados ou não que estão atualmente em uso como o TCP/IP (*Transmission Control Protocol/ Internet Protocol*) [58] e o OSI TP4 (Transporte - Classe 4) da ISO [59] dentre outros.

Assim sendo, tem-se que o transporte de dados tem de evoluir no sentido de ser mais otimizado nas suas implementações e na sua estruturação interna. É importante ressaltar que, por exemplo, protocolos como o TCP/IP foram desenvolvidos na década de 70. Nesta época os requisitos impostos pelas aplicações eram basicamente de transmissão de dados em redes tipicamente de longa-distância com taxas de velocidade da ordem de alguns milhares de bits por segundo (9 a 19.2 Kbps). Neste contexto, a capacidade computacional existente era capaz de decodificar quadros, pacotes e mensagens numa taxa superior à taxa de fornecimento de novas informações para processamento pela rede. Consequentemente, as implementações hoje existentes do TCP/IP não são otimizadas para os requisitos das redes de alta-velocidade e do processamento distribuído.

As ações técnicas empreendidas nesta área são basicamente de dois tipos:

- No sentido de propor novos recursos de implantação mais adequados para os protocolos de transporte que, por sua vez, podem ser estendidos para os protocolos das demais camadas [43] [60] [61]; e
- No sentido de propor uma nova estruturação da própria funcionalidade dos protocolos de transporte numa ótica diferente da utilizada atualmente [57] [56]. Esta estruturação tem seus princípios aplicados para outras camadas de protocolo e, em particular, para o nível de enlace de dados.

A lógica desta evolução é decorrente de dois fatos principais:

- Tem-se, com o advento das fibras ópticas como meio de transmissão, do processamento óptico e da utilização de multiplexação por divisão em comprimento de onda (*WDM - Wavelength Division Multiplexing*) [5] [6], a possibilidade de alcançar-se taxas de transmissão da ordem de *terabits* (10e12 bits/s) nos próximos anos [7];
- Paralelamente, as novas aplicações que surgem (processamento distribuído, vídeoconferência, vídeo sob demanda, multimídia, ...) necessitam de um bom desempenho da rede de suporte tendendo a fazer uso da grande banda de passagem provida pelas tecnologias de rede de alta velocidade [62].

O estado da arte da pesquisa em relação aos mecanismos de transporte de dados tem dois eixos de atividades principais:

- a. A concepção de novos protocolos de transporte de dados para redes de alta-velocidade mais "leves" e eficientes, também conhecidos como protocolos *lite* ou *lightweight*;
- b. O refinamento das implementações de protocolos de transporte de dados existentes, ou seja, que seguem o modelo clássico estruturado em camadas hierárquicas independentes segundo modelos iguais ou semelhantes ao RM-OSI da ISO. Dentro desta linha específica de atuação tem-se dois tipos de ações técnicas que podem ser empreendidas:
 - b1. Implementação dos protocolos em hardware através de técnicas de integração e concepção de circuitos integrados (CIs) ou concepção de processadores frontais de comunicação (PFCs) otimizados a base de transputers [21], microcontroladores dedicados, etc.;
 - b2. Implementação dos protocolos utilizando-se novos recursos ou recursos otimizados de sistema operacional (SO) visando a obtenção de implementações mais eficientes.

As soluções principais já empreendidas visando a concepção de novos protocolos de transporte de dados mais orientados ao uso com tecnologias de alta-velocidade são as seguintes:

- **VMTP (*Versatile Message Transaction Protocol*)** da Universidade de Stanford [63]:
O VMTP foi desenvolvido na década de 80 visando suportar comunicações baseadas em transações no estilo RPC com previsão de utilização como suporte para sistemas operacionais distribuídos.
- **XTP (*Express Transfer Protocol*)** da *Protocol Engines* [57]:
O XTP foi também desenvolvido na década de 80 visando uma implementação otimizada em VLSI e, neste sentido, equivale funcionalmente às camadas 3 e 4 do modelo de referência da ISO.
- **Delta-t** do Lawrence Livermore Laboratory [64]:
O Delta-t é um protocolo da década de 70 que inova pelo fato de gerenciar mais adequadamente o estabelecimento e o encerramento de conexões (gerenciamento de conexões otimizado - *lightweight*) com mecanismos implícitos de inicialização e término de conexões. Consequentemente, o Delta-t suporta melhor as aplicações orientadas a transações onde o fluxo de dados é pequeno.
- **NETBLT (*Network Block Transfer Protocol*)** do MIT (*Massachusetts Institute of Technology*) [57]:
O NETBLT foi desenvolvido visando a otimização da vazão de grandes fluxos de dados (*bulk data transfers*) através da separação dos mecanismos de controle de fluxo e controle de sequenciamento e erros.

Algumas das ações principais que fazem uso de técnicas de otimização na implementação de protocolos clássicos de transporte de dados são as seguintes:

- a. **Técnicas VLSI (*Very Large Scale Integration*):**
 - .Compilador de silício PSi desenvolvido pela IBM Yorktown [53];
 - .Compilador de especificações formais de protocolos para implementações VLSI da AT&T - Bell Labs [53];

b.Implementações baseadas em PFCs, *transputers*, circuitos especializados ou otimizações de sistemas operacionais, núcleos ou executivos:

.NAB (*Network Adapter Board*) da Universidade de Stanford [53] que foi concebido e orientado para rodar o VMTP;

.Implementação baseada em *transputers* da Universidade de Karlsruhe [54];

.PFC - Processador Frontal de Comunicação da UFPb/GRC (Grupo de Redes de Computadores) [65].

4.2 Resumo da Funcionalidade e Problemas

Inicialmente, faz-se necessário diferenciar os mecanismos de transporte de dados dos protocolos de transporte de dados que correspondem ao nível 4 do modelo de referência RM-OSI da ISO.

Os mecanismos de transporte de dados correspondem a mecanismos de transferência de informação (mensagens) para os usuários os quais podem fazer uso diretamente ou não de algum tipo de protocolo de transporte. A diferenciação entre estes termos fica mais aparente quando consideramos as redes de longa-distância onde o *frame relay* [66] e o próprio X.25 [67] são exemplos de mecanismos de transporte de dados sem que sejam protocolos do nível de transporte (nível 4). De maneira geral, tem-se hoje em dia uma tendência a considerar também determinados serviços como o RPC como um mecanismo de transporte de dados numa rede confiável. Consequentemente, tem-se que os mecanismos de transporte de dados são diversos e genéricos e, do ponto de vista deste trabalho, serão enfocados aqueles mecanismos que fazem uso diretamente de protocolos de transporte para efeito de análise de suas características em relação às redes de alta-velocidade.

Os protocolos de transporte podem prover dois tipos de serviços básicos que são utilizados segundo o grau de confiabilidade da rede de comunicação de dados existente (local ou longa-distância) e o grau de confiabilidade requerida pela aplicação [59]:

- Transporte de dados confiável através de uma mecanismo de circuito virtual (CV);
- Transporte de dados não confiável através de uma mecanismo de datagrama.

Na maioria dos casos, o objetivo principal deste protocolo é prover um transporte ou serviço confiável onde a informação é entregue em sequência, sem erros e sem duplicação.

De maneira geral os protocolos de transporte têm uma arquitetura que corresponde as definições dos seguintes aspectos de sua funcionalidade:

- Conexões;
- Transferência de dados;
- Formatação de mensagens;
- Controle de fluxo;
- Controle de erros; e
- Interface com o usuário (alguns protocolos dispõem de uma API com, por exemplo, *TLI - Transport Layer Interface* no Unix, TCP/IP, etc...).

4.3. Novas Características para Redes de Alta-Velocidade e Processamento Distribuído

As novas aplicações de rede e as necessidades do processamento distribuído levam a um conjunto de requisitos que devem ser satisfeitos pelos novos mecanismos de transporte de dados. Alguns destes requisitos são os seguintes:

- Taxa de vazão de dados elevada;
- Atraso reduzido na gerência da comunicação;
- Otimização no processamento dos protocolos; e
- Oferecimento de alternativas de serviço para o usuário (circuitos confiáveis, não confiáveis, comunicação com e sem conexão, etc...).

Na implantação de novos mecanismos de transporte de dados leva-se em conta alguns princípios indicados abaixo, que são hoje válidos com o advento das redes de alta-velocidade e das fibras ópticas.

As redes são intrínsecamente confiáveis, com taxas de erro muito baixas da ordem de 10^{-15} para as fibras ópticas [68] e com MTBFs (*Mean Time Before Failure*) cada vez maiores [22]. O aumento do MTBF é decorrente da evolução do nível de integração propiciado pela evolução da microeletrônica e pelas facilidades de concepção de circuitos integrados de aplicação específica (*ASICs - Application Specific Integrated Circuits*) [69].

Em função do fato das taxas de velocidade no meio tenderem a ser muito altas, requerendo uma boa capacidade computacional nas máquinas da rede, toma-se por princípio que os novos protocolos devem ser de processamento simplificado e sem muita preocupação com o *overhead* causado pela inserção da mensagem no meio.

Algumas decorrências destes princípios básicos são as seguintes:

- Volta-se a utilizar o princípio fim-a-fim (*end-to-end principle*) largamente utilizado na década de setenta e discutido por pesquisadores do MIT na referência [70];
- O processamento da informação dos protocolos tende a ser orientado a palavra (*byte oriented*) (4, 8, 16, ... bits) evitando assim um *overhead* de processamento na decodificação de subcampos com bits. Exemplos concretos desta tendência são alguns novos protocolos de enlace de dados:

- .O protocolo FDDI que processa símbolos de 4 bits (*nibbles*) [71];
- .O DQDB que processa informações com 4 e 8 bits;
- .O ATM;

.outros

- As unidades de informação manipuladas pelos protocolos (quadros, pacotes, células - *cells*, ...) tendem a ser de tamanho fixo no sentido de permitir uma maior facilidade de integração do protocolo em silício ou o processamento em hardware. Esta tendência é observada em praticamente todas as novas tecnologias e serviços de rede de alta-velocidade como o DQDB, o ATM e o SONET. A título de exemplo, o ATM usa células de tamanho fixo comutadas por hardware especializado (*fast packing*).

Em seguida, fazemos uma discussão sobre a arquitetura dos protocolos de transporte visando identificar as novas soluções que devem ser adotadas na concepção de protocolos eficientes e adaptados às novas tecnologias de rede de alta-velocidade.

4.3.1 Gerência de Conexão

O gerenciamento de conexões corresponde à representação do estado das entidades pares de transporte (*peer transport entities*). O estado das entidades muda com a transmissão e recepção de dados e reconhecimentos no decorrer do processo de comunicação. A abertura de uma conexão inicializa a tabela de estados da conexão e o encerramento da conexão libera a tabela de estados para uso subsequente.

A troca das informações de controle (sinalização- no jargão das telecomunicações) entre as entidades pares pode ser multiplexada com os dados transmitidos (*in-band signaling*). Esta é a solução adotada na maioria dos protocolos como o TCP, TP4-OSI, XTP, Netblt, Delta-t e VMTP, dentre outros. Outra solução consiste em passar as informações de controle através de um canal ou associação independente (*out-of-band signaling*). Esta solução é utilizada pelo APPN (*Advanced Peer-to-Peer Networking*) [72] da IBM e na RDSI-FE (Rede Digital de Serviços Integrados - Faixa Estreita) através de seu canal D de sinalização [73].

O estabelecimento e a liberação de uma conexão pode ocorrer de diferentes formas:

- Usando um procedimento explícito com primitivas de reconhecimento para o estabelecimento da conexão (*handshake*) ou
- Usando um mecanismo implícito baseado no uso de temporizadores.

Os protocolos mais tradicionais como o TP4-OSI e o TCP usam mecanismos de reconhecimento para o estabelecimento da conexão que

implicam num grande *overhead* para conexões de curta-duração bastante comuns nas aplicações distribuídas. Nestas aplicações necessita-se de conexões rápidas com baixa latência pois o volume de dados envolvido na transação é tipicamente pequeno. Transações num modelo cliente/servidor [74] são um exemplo típico de processamento onde o *overhead* da gerência de conexão deve ser baixo.

A alternativa dos mecanismos implícitos de estabelecimento de conexão é a mais recomendada para os protocolos de transporte redes de alta-velocidade. Os protocolos VMTP e XTP utilizam esta solução. Neste caso, a conexão é estabelecida implicitamente com a primeira mensagem e fechada confiavelmente com a ocorrência de uma estouro de temporização gerado por um temporizador (Delta-t e VMTP). Além disso, dados podem também ser transmitidos junto com a informação de sinalização na primeira mensagem. Esta solução é particularmente atrativa para aplicações distribuídas, acesso a banco de dados e solicitações de transferência de arquivos de servidores (*downloading*). A figura 4.1 ilustra o efeito da utilização de um mecanismo implícito em relação a solução clássica numa rede longa-distância. Neste caso particular, percebe-se a importância do mecanismo implícito.

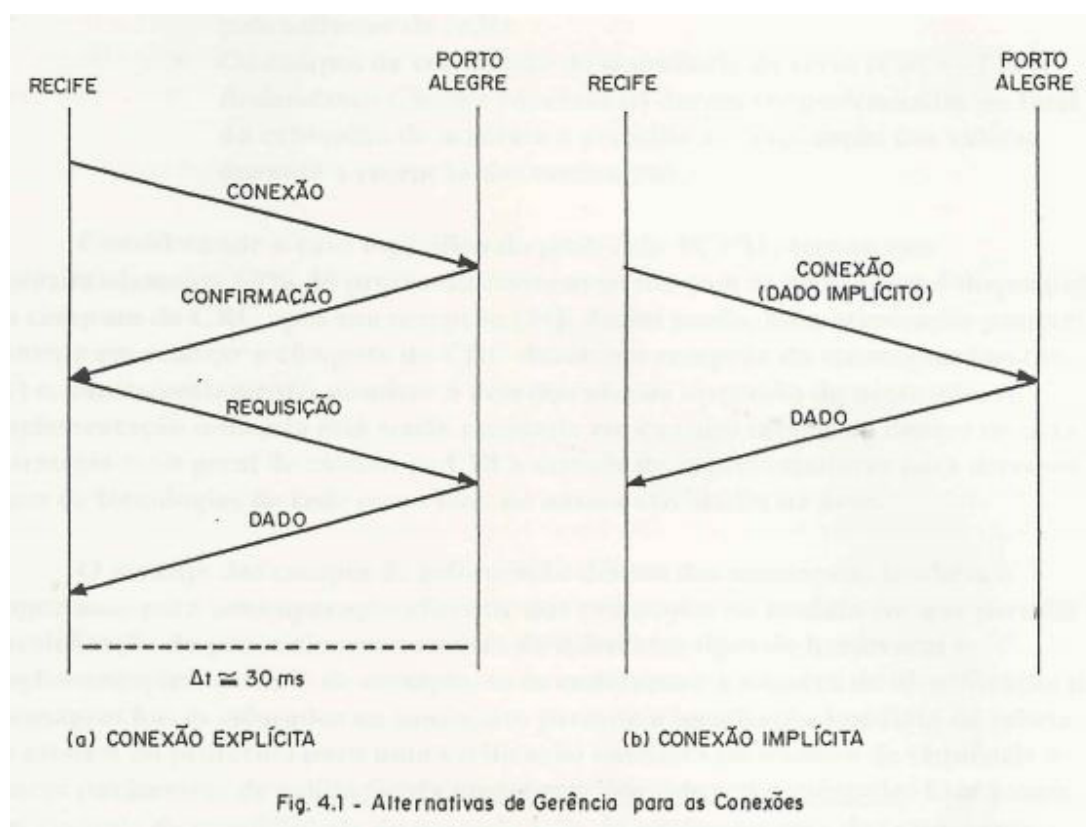


Fig. 4.1 - Alternativas de Gerência para as Conexões

4.3.2 Formatação das Mensagens

A formatação das mensagens é crítica para a simplificação do projeto do receptor numa rede de alta-velocidade. A concepção de um receptor simples é uma das premissas dos novos protocolos (transporte, rede e enlace) associados a tecnologias e meios onde as taxas de velocidade são muito altas.

A formatação de mensagens visa basicamente:

- Facilitar a implementação do(s) protocolo(s) em VLSI e
- A utilização de técnicas de processamento paralelo no tratamento das mensagens recebidas da rede.

Os princípios básicos a serem seguidos na formatação das mensagens são os seguintes:

- Os campos de informação devem ter uma localização fixa e tamanho fixo nas mensagens;
- Os campos devem ser alinhados em múltiplos de palavras (8 bits) de forma a permitir uma verificação e decodificação mais eficiente pelo software da rede; e
- Os campos de verificação de ocorrência de erros (*CRCs - Cyclic Redundancy Check* e *checksums*) devem ser posicionados no final do cabeçalho de maneira a permitir a computação dos valores durante a recepção das mensagens.

Considerando o caso específico do protocolo TCP/IP, tem-se que aproximadamente 60% do processamento envolvido com as mensagens é dispendido no cômputo do CRC após sua recepção [36]. Assim sendo, uma otimização possível consiste em realizar o cômputo do CRC durante a recepção da mensagem (*on-the-fly*) e, consequentemente, otimizar o desempenho da operação do protocolo. A implementação indicada está sendo realizada em circuito integrado dentro de uma estratégia mais geral de cálculo de CRCs através de coprocessadores para diversos tipos de tecnologias de rede como uma de nossas atividades na área.

O arranjo dos campos de informação dentro das mensagens também é importante para uma operação eficiente dos protocolos na medida em que permite a paralelização do processamento através de diferentes tipos de hardwares e implementações. A título de exemplo, se os endereços e o número de identificação da mensagem forem colocados no início, isto permite a localização imediata da tabela de estados do protocolo para uma verificação imediata do número de sequência e outros parâmetros de validação da mensagem durante a sua recepção. Este é mais um exemplo de possibilidade de paralelização do processamento das mensagens.

4.3.3 Controle de Fluxo

O controle de fluxo é um mecanismo que visa basicamente a proteção dos recursos do receptor de mensagens. A idéia básica consiste em dispor de um mecanismo que evite a inundação de informações no receptor quando este não dispõe de recursos para o seu processamento, evitando portanto a perda de informação e a necessidade de retransmissões [47].

De início uma das premissas que são consideradas para a concepção de protocolos eficientes é separar o controle de fluxo do controle de erros. Na maioria dos protocolos clássicos são operações executadas através de mecanismos comuns de sequenciamento e controle de janela. Eventualmente, estas operações são redundantes entre as diferentes camadas que compõem a arquitetura de protocolos da rede [56] [59]. Na realidade, o controle de fluxo e o controle de erros têm diferentes objetivos em diferentes tipos de rede. A título de exemplo, numa rede longa-distância com um tempo de propagação elevado (p. ex.: 30 mseg) a janela de controle de fluxo deve ser a maior possível para permitir o fluxo contínuo de informações entre emissor e receptor. Por outro lado, o controle de erros deve poder reagir o mais rapidamente possível a uma situação de erros, requerendo portanto uma janela pequena.

Os mecanismos básicos utilizados para o controle de fluxo são os seguintes:

- Janela de Controle de Fluxo (*windowing*) e
- Controle de Vazão e Maximização de Crédito.

Em relação ao mecanismo de janelas as alternativas básicas de implementação são as seguintes:

- Janela de tamanho variável e
- Janela de tamanho fixo.

O mecanismo de janela com tamanho fixo é o preferível na medida em que é aquele que apresenta o menor percentual de *overhead* no processamento e, além disso, pode ser facilmente implementado segundo a disponibilidade de memória (*buffers*) no receptor. Esta é a solução adotada nos protocolos NETBLT E VMTP. A solução janela deslizante (*sliding window*) é usada nos protocolos clássicos como o TCP e o TP-4 OSI e, neste caso, a atualização da janela é realizada através da acumulação de reconhecimentos no emissor que apresenta um grande *overhead* de processamento.

Em relação ao tamanho da janela, tem-se que a opção mais eficiente em termos de desempenho do protocolo é aquela em que aloca-se a maior janela possível para o emissor (crédito máximo). Neste caso, evita-se que o emissor perca tempo em procedimentos de solicitação e liberação de créditos para

transmissão em termos de unidades de protocolos (UDPs) ou palavras a transmitir (*bytes*). Além disso, pressupõe-se que o emissor envia suas mensagens de forma impulsiva (*bursty*) em blocos, visando também maximizar a execução do protocolo.

Um problema decorrente da alocação de janelas fixas com crédito máximo para o emissor é a possibilidade do receptor ser inundado por emissões impulsivas oriundas simultaneamente de diferentes conexões. Para minimizar este tipo de problema, os protocolos mais eficientes utilizam o princípio de controle de vazão com maximização de crédito (p. ex.: XTP e Delta-t). O controle de vazão pode ser implementado de duas maneiras:

1. No primeiro caso, é fixado para o emissor o tamanho do bloco de informações que pode ser transmitido impulsivamente na sua vazão máxima e o intervalo entre os blocos de transmissão impulsiva. Isto permite ao receptor acomodar as vazões oriundas de diferentes conexões;
2. No segundo caso, o emissor envia informações até o limite de seu crédito, mas faz uma parada entre a emissão de cada mensagem por um período estipulado pelo receptor (*interpacket delay*).

De maneira geral, a implementação de controle de fluxo através da combinação das técnicas de crédito máximo e controle de vazão permite a obtenção de um desempenho melhor em relação ao mecanismo de janela deslizante [44]. As vantagens principais destas novas técnicas são as seguintes:

- Evita inundação da rede;
- Ajusta o fluxo de mensagens à velocidade do receptor;
- Permite a utilização de janelas muito grandes; e
- É mais eficiente em redes com grandes atrasos de propagação.

As desvantagens principais destas novas técnicas são as seguintes:

- Processamento mais elaborado em software e
- Necessita de informações de desempenho da rede para parametrização eficiente das conexões.

4.3.4 Detecção e Correção de Erros

Os mecanismos de detecção e recuperação de erros apresentam os seguintes aspectos que podem ser analisados à luz dos novos requisitos de redes de alta-velocidade:

- A detecção do erro;

- A recuperação dos erros e a política de geração de reconhecimentos ou notificação de erros.

Um aspecto importante a considerar na discussão destes pontos é que as novas redes de alta-velocidade são intrinsecamente confiáveis e, assim sendo, a maioria dos erros serão resultantes de descartes de mensagens ocorridos durante congestionamentos na sub-rede ou no receptor.

A detecção de erros pode ser implementada de diferentes maneiras:

Em relação a entrega de mensagens sem erro:

- Verificação de erros apenas no cabeçalho com verificação opcional de erros nos dados (Delta-t, VMTP, XTP e NETBLT). Este tipo de solução é particularmente importante para aplicações tipo multimídia e voz onde faz mais sentido garantir a sequencialidade dos dados que a retransmissão de dados incorretos;
- Verificação de erros na mensagem completa (TCP e TP-4 OSI). Esta é a solução clássica que apresenta os problemas de desempenho já mencionados no processamento dos CRCs e paridades.

Em relação às mensagens com erro, as ações tomada pelo receptor podem ser as seguintes:

- Rejeita com algoritmo de retransmissão do tipo volta-N (*go-back-N*) (Delta-t) onde todas as mensagens subsequentes ao erro são descartadas. As vantagens nesta caso são a simplicidade da implementação e da gerência de memória.
- Rejeita com algoritmo de retransmissão seletiva (XTP, VMTP e NETBLT). Neste caso, as retransmissões são solicitadas apenas para as mensagens com erro.

Com as redes de alta-velocidade a repetição seletiva é a mais eficiente na medida em que acomoda melhor as situações de congestionamento onde mensagens são descartadas. Neste mesmo contexto, o receptor não deve solicitar explicitamente a retransmissão de uma mensagem de maneira a reduzir o número de retransmissões e simplificar o mecanismo de recuperação de erros.

Como último ponto de discussão, ressaltamos que os princípios discutidos para o protocolo de transporte são aplicáveis para protocolos em outras camadas da arquitetura da rede.

5. FERRAMENTA DE PROTOTIPAÇÃO E MODELAGEM

5.1 Motivação

Durante esta década as redes de computadores experimentaram uma grande evolução em relação a vários aspectos técnicos. Dentre os principais avanços, temos os seguintes:

- Em relação às tecnologias de rede;
- Em relação aos meios de transmissão utilizados;
- Em relação aos serviços de aplicação e suas APIs; e
- Em relação às opções entre arquiteturas de redes e aos subconjuntos funcionais de protocolos dentro de arquiteturas específicas.

A título de exemplo, tem-se hoje que a opção quanto à tecnologia utilizada na rede é quase que completamente independente da opção do meio de transmissão utilizado. Por exemplo, pode-se hoje utilizar par trançado de diferentes tipos (*UTP - Unshielded Twisted Pair* e *STP - Shielded Twisted Pair*) com opções tecnológicas como os padrões IEEE 802.3 (*ethernet*) [75], IEEE 802.5 (anel com ficha - *Token- Ring*) [76] e FDDI [1] que, no caso, é comumente denominado de CDDI (*Copper Distributed Data Interface*) [77].

Outro fato novo no universo de alternativas e soluções para as redes de computadores é a constatação que a dicotomia existente anteriormente entre as redes locais (LANs) e as redes de longa-distância (WANs) esta gradativamente desaparecendo. Alguns exemplos concretos deste fato são os seguintes:

- Tecnologias antes exclusivamente utilizadas em ambiente longa-distância como o X.25 são hoje também utilizadas em ambiente local ou privado;
- Novas tecnologias como o ATM (*Asynchronous Transfer Mode*) [2] sinalizam com a possibilidade de serem utilizadas em redes locais com a denominação LATM ("Local ATM") [78];
- Tecnologias de redes metropolitanas (*MAN - Metropolitan Area Network*) como o FDDI são hoje largamente utilizadas em redes locais de computadores.

No que diz respeito aos serviços de aplicação disponíveis para o usuário da rede, tem-se hoje uma grande disponibilidade entre as alternativas padronizadas, padrões de uso (*de facto*) e soluções proprietárias. Alguns dos exemplos mais usuais e conhecidos do programador ou integrador de redes são os seguintes:

- Protocolos de aplicação padronizados segundo o modelo RM-OSI da ISO como o FTAM (*File Transfer Access and Management*) [79], o X.400 [80] e o X.500 [81], dentre outros;
- Protocolos de aplicação da família TCP/IP como o FTP (*File Transfer Protocol*), o Telnet, e o SNMP (*Simple Network Management Protocol*) dentre outros [58];
- Serviços de chamada remota de procedimentos - RPC (*Remote Procedure Call*) [82];
- Serviço de acesso a banco de dados como o RDA (*Remote Database Access*) [83] e SQL (*Structured Query Language*) [84];
- Etc...

De maneira geral, o somatório dos fatos acima citados faz com que a especificação, prototipação e projeto de uma rede de computadores para um determinado tipo de aplicação seja uma tarefa relativamente complexa que exige um grau de conhecimento e uma experiência prática de implantação muito bons. Se nós considerarmos adicionalmente a influência dos recursos de implantação e os novos avanços propostos para os protocolos (transporte e outros) (secções 3 e 4), percebe-se mais fortemente a necessidade de ter-se algum tipo de metodologia que possa auxiliar no processo de definição de uma rede de computadores para suportar, por exemplo, o processamento distribuído.

Em seguida, propomos uma abordagem que responde parcialmente ao tipo de necessidade evocada na discussão acima a partir de uma ferramenta de prototipação e emulação de redes.

5.2 Ferramentas e Métodos para a Modelagem e Simulação - Abordagens Convencionais

As soluções tradicionalmente utilizadas nas ferramentas e métodos de modelagem e simulação orientadas para as redes de computadores e sistemas distribuídos são basicamente as seguintes:

- Baseada numa linguagem (p. ex.: Simula, Simescript);
- Baseada em modelos; e
- Baseada num ambiente computacional.

A solução baseada numa linguagem como o Simula, oferece ao usuário recursos (*constructs*) de linguagem de programação para suportar a modelagem e a simulação. A vantagem principal desta solução é a sua generalidade quanto ao escopo das aplicações da mesma. A desvantagem principal desta solução é a sua limitação quanto ao estudo de sistemas distribuídos complexos [85]. Os principais inconvenientes são:

- A separação das fases de modelagem e simulação das fases de projeto e desenvolvimento. A título de exemplo, o projetista do protocolo de transporte desenvolve seu código em um ambiente (p. ex.: Unix, linguagem C) e, ao mesmo tempo, um modelo de simulação consistente numa outra linguagem (p. ex.: Simula).
- A necessidade do modelador gerenciar eficientemente a complexidade do escalonamento dos modelos de sistemas distribuídos.

A solução baseada em modelos oferece normalmente ao usuário uma coleção de ferramentas suportando uma ou mais técnicas de modelagem. Exemplos típicos desta solução são ferramentas de simulação baseadas em teoria das filas como as descritas em [86] [87]. A principal vantagem deste tipo de solução é a sua eficiência na execução de grandes simulações através de técnicas algorítmicas de otimização [85]. A sua principal desvantagem é o escopo limitado de sua aplicação. Além disso, esta solução também separa a fase de modelagem/simulação da fase de projeto/desenvolvimento e exige do modelador um conhecimento apurado da técnica de modelagem sendo utilizada.

A solução baseada num ambiente computacional oferece ao usuário a possibilidade de *link-editar* o código executável do sistema a ser simulado com os diferentes recursos de um ambiente computacional como, por exemplo, o sistema operacional Unix e suas ferramentas. Neste caso, obtém-se uma simulação direta do código de implementação. Esta simulação direta é a principal vantagem deste tipo de solução pois, como discutido em [85], oferece ao simulador a possibilidade de aglutinar as fases de modelagem/ simulação e

projeto/ desenvolvimento. A principal desvantagem deste tipo de solução é decorrente da utilização direta do ambiente operacional. Na realidade, a solução amarra os resultados da simulação a um determinado ambiente e impede qualquer estudo ou prototipação mais genérica onde deseja-se alterar os recursos de implantação (SO) e o próprio ambiente da prototipação.

5.3 Ferramenta de Prototipação e Emulação de Redes

As fases básicas envolvidas no processo de projeto (*design*) de um sistema de comunicação, sistema distribuído ou rede de propósito geral são basicamente as seguintes:

- Edição de requisitos;
- Projeto conceitual; e
- Projeto detalhado.

A ferramenta proposta é orientada para a fase de realização do projeto conceitual, fornecendo também um suporte na fase de definição de requisitos através de uma metodologia de projeto cooperativo como o usuário.

Os objetivos globais do conjunto de ferramentas proposto são os seguintes:

- Prototipação rápida de sistemas de comunicação e, em particular, sistemas distribuídos, redes de computadores e protocolos de comunicação;
- Simulação de desempenho do sistema prototipado; e
- Simulação de características funcionais.

A abordagem técnica utilizada na concepção do conjunto de ferramentas utiliza o conceito de modelagem por emulação (Ferramenta OONET - *Object-Oriented Network Evaluation Toolkit*) [88] [89] associado a uma metodologia de derivação de especificações conceituais de projeto (Ferramenta NETSET - *Network System Evaluation Toolkit*) (Fig. 5.1). Nesta abordagem um pseudo-código de implementação é criado ou disponibilizado sob a forma de uma biblioteca. Este código é executado de forma a simular o desempenho e as características funcionais principais do sistema prototipado.

As vantagens principais deste tipo de abordagem para a implementação da ferramenta são as seguintes:

- Prototipação rápida. Esta é uma característica resultante do fato de termos flexibilidade de emular diversos ambientes, protocolos, etc...
- Independência do ambiente operacional. Na realidade, esta é uma característica que tem sido apontada como uma necessidade pelos usuários [90]. Ocorre que não estão disponibilizadas ferramentas para determinadas linguagens ou ambientes que, nesta abordagem, podem ser emulados.
- Metodologia de projeto *top-down*. A ferramenta é intrinsecamente orientada para uma abordagem que gera a especificação a partir dos requisitos da aplicação. Embora esta

seja a estratégia correta e óbvia, segundo nossa experiência na área de redes, muitos conceptores de sistemas usam uma metodologia *down-top*. Isto ocorre principalmente pela diversidade de alternativas tecnológicas para as redes que acaba dificultando uma abordagem mais sistêmica.

- Projeto interativo tanto a nível da fase de especificação de requisitos como a nível da fase de emulação da rede.

As principais desvantagens da abordagem proposta são as seguintes:

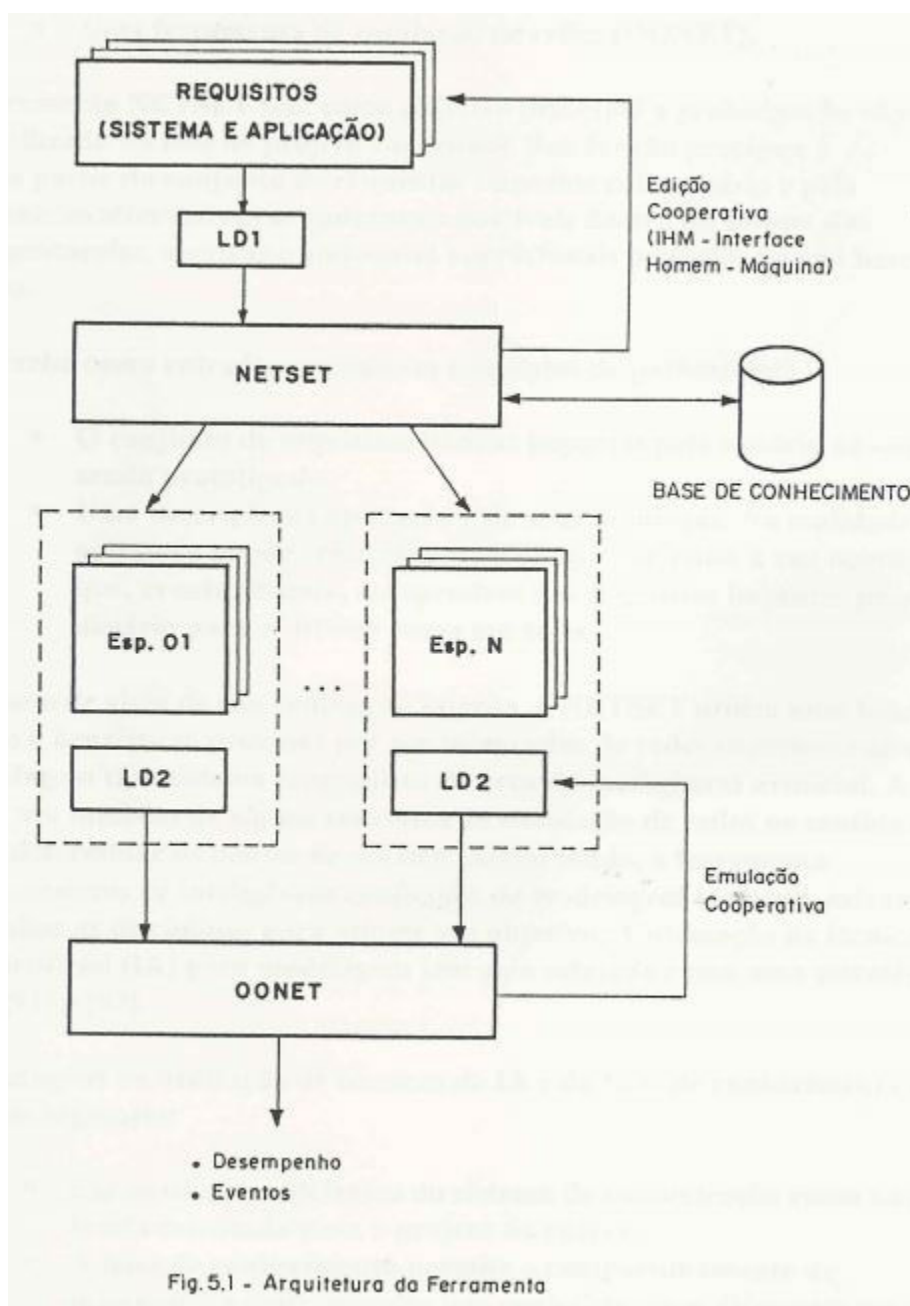
- O pseudo-código utilizado na emulação não pode ser utilizado diretamente, por exemplo, para a implantação de um protocolo de comunicação com novos requisitos funcionais. Assim sendo, a ferramenta separa num certo nível a fase de modelagem/simulação da fase de projeto/desenvolvimento. Por outro lado, é importante considerar que o pseudo-código utilizado é certamente uma referência bastante elaborada do seu código executável correspondente.
- O escopo das aplicações suportadas pela ferramenta é principalmente dependente da quantidade de módulos oferecidos para o usuário.

Esta última desvantagem corresponde à situação onde o usuário não deseja desenvolver módulos de emulação. Do ponto de vista da utilização da ferramenta, esta situação corresponde aos usuários que querem simular a operação da rede com determinados tipos de aplicações em ambientes convencionais (protocolos conhecidos). Os usuários que desejarem simular novos protocolos ou ambientes operacionais específicos terão necessariamente de desenvolver módulos de pseudo-código a serem acoplados na ferramenta.

5.4 Arquitetura da Ferramenta

A arquitetura geral do conjunto de ferramentas proposto é apresentada resumidamente na figura 5.1. Ela é composta de duas ferramentas principais:

- Uma ferramenta de derivação de requisitos de rede e especificação de arquiteturas (NETSET) e
- Uma ferramenta de emulação de redes (OONET).



A ferramenta NETSET tem como objetivo principal a prototipação rápida do sistema e é utilizada na fase de projeto conceitual. Sua função precípua é determinar, a partir do conjunto de requisitos impostos pelo usuário e pela aplicação, quais as alternativas arquiteturais possíveis dentro do escopo das tecnologias, protocolos, serviços e ambientes operacionais previstos na sua base de conhecimento.

Ela recebe como entrada os seguintes conjuntos de parâmetros:

- O conjunto de requisitos básicos impostos pelo usuário ao sistema sendo prototipado;
- Uma descrição da aplicação e de seus requisitos. Na realidade, a aplicação impõe requisitos específicos e inerentes à sua operação que, eventualmente, independem dos requisitos impostos pelo usuário para o sistema como um todo.

Do ponto de vista de sua concepção interna, a NETSET utiliza uma base de conhecimento e heurísticas providas por um integrador de redes experiente através de uma abordagem tipo sistema especialista da área de inteligência artificial. A NETSET faz uso também de alguns conceitos de simulação de redes no sentido de minimizar e determinar os pontos de decisão. Assim sendo, a ferramenta compartilha conceitos de inteligência artificial e de modelagem tentando extrair o melhor de ambas as disciplinas para atingir seu objetivo. A utilização de técnicas de inteligência artificial (IA) para modelagem tem sido relatada como uma estratégia eficiente em [91] e [92].

As vantagens na utilização de técnicas de IA e da base de conhecimento são basicamente as seguintes:

- Ela auxilia a modelagem do sistema de comunicação como uma tarefa orientada para o projeto do sistema;
- A base de conhecimento permite o compartilhamento de informações especializadas por projetistas com diferentes graus de especialização em redes de computadores. Assim sendo, diminui-se a dependência de indivíduos altamente especializados para a concepção de tais sistemas; e
- As modificações e inserções de conhecimento podem ser facilmente realizadas, permitindo assim uma maior facilidade na absorção de novas tecnologias, serviços, protocolos, etc..., que se renovam, evoluem e surgem com uma frequência cada vez maior.

A base de conhecimento é certamente um aspecto importante da ferramenta pois armazena a essência da informação e da experiência envolvida

na prototipação do sistema. Alguns dos requisitos impostos para a concepção desta base de conhecimento são os seguintes:

- Utilizar orientação a objetos na representação das informações de forma a garantir as vantagens inerentes a este paradigma de programação; e
- Utilização de uma estrutura em árvore semelhante a estrutura utilizada na representação dos objetos de gerenciamento da base de dados (*MIB - Management Information Database*) do CMIS/CMIP (*Common Management Information Service and Protocol*) [40] e do serviço de diretório X.500 do RM-OSI.

Em resumo, a ferramenta NETSET auxilia inicialmente na transformação de uma lista de requisitos abstrata ou modelo abstrato do usuário num conjunto de requisitos formais. Isto é obtido interativamente com o usuário através de um editor de especificações que efetivamente implementa o que denominamos de edição cooperativa. Neste processo são usadas heurísticas e a base de conhecimento para gerar uma descrição dos requisitos na linguagem LD1. A partir da descrição em LD1 o NETSET gera as possíveis especificações usando novamente a base de conhecimento e heurísticas. Estas especificações formam o conjunto de parâmetros de entrada para a ferramenta OONET.

A ferramenta OONET tem como objetivo principal a emulação do sistema especificado e obtido a partir dos requisitos da aplicação e do usuário através da ferramenta NETSET. Assim sendo, a OONET implementa efetivamente a estratégia de modelagem por emulação onde um pseudo-código da rede especificada é executado de forma a simular seu comportamento, analisar sua validação funcional e testar diferentes aspectos funcionais.

A ferramenta recebe como parâmetro de entrada uma descrição do sistema a ser emulado na linguagem LD2. Esta linguagem permite a configuração e parametrização da ferramenta segundo as opções indicadas abaixo:

- Seleciona as opções de configuração dos módulos internos de rede da OONET (tecnologias, protocolos, classes de protocolos, etc...);
- Parametriza as opções de módulos segundo os requisitos da aplicação e do usuário;
- Seleciona o ambiente operacional requisitado com sua respectiva parametrização; e
- Define o tipo de emulação, validação ou avaliação de desempenho requerida pelo usuário da ferramenta.

Como no caso da ferramenta NETSET, a OONET também permite uma emulação cooperativa onde o usuário pode agir diretamente no meta-código LD2 de forma a alterar sua especificação básica segundo necessidades detectadas durante o processo de emulação.

No que diz respeito à sua concepção interna, as características básicas da ferramenta OONET são as seguintes [88] [89]:

- Implementação orientada a objeto e
- Baseada no conceito de eventos (*event-driven*).

A orientação a objeto é utilizada tanto na implementação da ferramenta através do uso de C++ [93] como na representação dos componentes do modelo (protocolos, sistema operacional, redes locais, etc...). Os eventos são uma classe de objeto processados por um *shell* que integra efetivamente os vários módulos correspondentes ao sistema sendo emulado.

Os resultados obtidos pela emulação na OONET são, a título de exemplo, figuras de desempenho e relatório de eventos tais como perdas excessivas, congestionamento e relatórios de parâmetros funcionais dos protocolos e entidades componentes do sistema.

6. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Como consideração final relativa ao novo paradigma de concepção das redes de computadores, ressaltamos nossa visão para a próxima geração destes sistemas:

- Faz uso intensivo das fibras ópticas como meio de transmissão;
- Faz uso das tecnologias de rede de alta-velocidade;
- É otimizada na sua concepção e implementação e
- É um suporte básico para o processamento distribuído (computação distribuída).

A otimização é uma necessidade que visa:

- A preservação da vazão para as aplicações. Em outras palavras, minimiza-se os gargalos de processamento do software de rede fazendo com que seja possível a obtenção de uma vazão elevada para novos serviços e aplicações como multimídia, vídeoconferência, vídeo sob demanda, computação gráfica de alta-resolução, dentre outras;
- A adaptação da capacidade computacional dos computadores de propósito geral à taxa de transmissão das novas tecnologias de rede de alta-velocidade. A alta taxa de transmissão das redes exige uma capacidade computacional elevada com interfaces especializadas e otimizadas na sua concepção.

É importante observar que a necessidade de preservação da vazão não é tão premente quando se utiliza as redes de alta-velocidade como dorsais (*backbone*) para a interconexão de redes ou sistemas de baixa-velocidade. Neste caso, as aplicações são convencionais com baixa demanda de volume de transmissão de dados e sem requisitos críticos de tempo de resposta para a rede. A abordagem que enfocamos consiste em otimizar em todos os níveis permitindo:

- Melhoria da vazão na camada de aplicação para tecnologias convencionais como as da família IEEE 802.X;
- Utilização das redes de alta-velocidade como redes locais ou de longa-distância através da conexão direta de estações de trabalho (*workstations*) com aplicações críticas ou não.

As ações propostas neste documento visando a concepção de redes neste novo paradigma foram consideradas segundo os modelos de referência, os recursos básicos de implantação, os mecanismos de transporte de dados e uma ferramenta de prototipação rápida e emulação de redes.

Em relação aos modelos de referência a abordagem é a seguinte:

- Adotar um novo modelo (otimizado) mais abrangente para o processamento distribuído. Até o momento, o RM-ODP é a alternativa mais qualificada no processo de estabelecimento deste novo modelo;
- Preservar a estrutura multi-camadas atual, otimizando-a no sentido de garantir a interoperabilidade entre soluções multi-plataforma e multi-fornecedor.

No que diz respeito aos recursos de implantação a abordagem adotada é a seguinte:

- Otimização da arquitetura dos PFCs;
- Implementação de recursos de sistema operacional orientados e otimizados para as redes que façam movimentos mínimos de dados na memória e façam o menor número possível de chamadas ao sistema operacional nativo do ambiente computacional;
- Implementar o maior número possível de camadas em HW (no PFC) evitando o diálogo da interface com a UCP principal;
- Utilizar recursos de paralelização na execução de funções.

Para os mecanismos de transporte de dados e protocolos de maneira geral a abordagem é basicamente a seguinte:

- Remoção da redundância de funções a nível da arquitetura de redes;
- Redução da complexidade dos protocolos, particularmente no que diz respeito à sua máquina de estados, visando soluções *lite*;
- Reorganização do formato e estruturação das UDPs visando a otimização do processamento das funções de rede, por exemplo, através de recursos como execução paralela;
- Redefinição das funções nas camadas da rede (controle de erro, fluxo, endereçamento, ...) visando o aproveitamento da alta-velocidade e da grande imunidade ao ruído das fibras ópticas.

Em relação à ferramenta de prototipação e emulação de redes a abordagem pretendida é o auxílio ao usuário na concepção de sistemas distribuídos, redes e protocolos considerando a multiplicidade de soluções e alternativas existentes.

Obviamente, o conjunto de ações relacionadas com as redes de alta-velocidade não se restringe à abordagem discutida para o novo paradigma de concepção. Segue uma lista não-exaustiva de outras importantes ações sendo empreendidas:

- Otimização de redes longa-distância com tecnologia *frame relay* [66];
- Evolução do protocolo FDDI nas suas versões FDDI-2 e FFOI (FDDI Follow-on Lan) [94] [95];
- Desenvolvimento de interfaces paralelas de alta-velocidade para a computação de alto-desempenho e supercomputadores [7];
- A rede digital de serviços integrados de banda larga (BISDN) [96];
- As redes metropolitanas [3];
- Serviços públicos de transporte de dados em alta-velocidade como o SONET [4] e o SMDS (*Switched Multimegabit Data Service*) [97];

Em termos de projetos experimentais de redes de alta-velocidade destacamos:

- O projeto NREN (*National Research and Education Network*) desenvolvido pelo NSF/DARPA nos EUA para a experimentação de tecnologias emergentes (05 redes *gigabit*) [98] e
- O projeto MULTICOM - "Comunicações Modais para os Anos 90" com objetivo semelhante sendo desenvolvido principalmente pela Unicamp.

REFERENCES

- [1] McCOOL, J. F. "The Emerging FDDI Standard", *Telecommunications*, v. 21, nº 5, Maio, 1987.
- [2] CHEUNG, N. K. "The Infrastructure for Gigabit Computer Networks", *IEEE Communications Magazine*, v. 30, nº 04, Abril, 1992, p. 60-68.
- [3] VALOVIC, T. "Metropolitan Area Networks: A Status Report", *Telecommunications*, v. 32, nº 7, Julho, 1989, p. 25-32.
- [4] GRIFFIN, J. - "Next-Generation Transport for WANs", *Telecommunications*, Abril, 1991, p. 48- 54.
- [5] KUNG, H. T. "Gigabit Local Area Networks: A System Perspective", *IEEE Communications Magazine*, v. 30, nº 04, Abril, 1992, p. 79-89.
- [6] FRASER, S. "Switching Trends in Integrated Optics", *Telecommunications*, v. 25, nº 7, Julho, 1991, p. S38-S40.
- [7] CATLETT, C. E. "Balancing Resources", *IEEE Spectrum*, v. 29, nº 9, Set., 1992, p. 48-55.
- [8] MARTINS, JOBERTO S. B. and NOGUEZ, G. "A Unix Distributed Application Support suitable for Mini and Microcomputers based Systems", *Proceedings of the 13º EUROMICRO - Symposium on Microprocessing and Micro-Programming*, Portsmouth, England, 1987, p. 205-210.
- [9] MARTINS, JOBERTO S. B. and NOGUEZ, G. "Considerações sobre a Arquitetura e o Desempenho de uma Implantação do Protocolo Ethernet em Ambiente Unix", *Anais do 4º SBT - Simpósio Brasileiro de Telecomunicações*, SBT, Rio, RJ, 1986, p.157-160.
- [10] SCHILL, A. "Distributed Application Development: Problems and Solutions", *Proceedings of the Information Network and Data Communication - III*, IFIP, 1990, p. 303-317.
- [11] BIRRELL, A. D. e NELSON, B. J. "Implementing Remote Procedure Calls", *ACM Transactions on Computer Systems*, Fev., 1984.
- [12] MUKHERJEE, B. "Integrated Voice-Data Communication over High-Speed Fiber Optic Networks", *IEEE Computer*, v. 24, nº 2, Fev., 1991, p. 49 - 58.
- [13] ISO - *Information Processing Systems - Open Systems Interconnection - Basic Reference Model*, ISO IS 7498, 1984, 40p.

- [14] MODIRI, N. "The ISO Reference Model Entities", *IEEE Network*, v.5, nº 4, Jul., 1991, p. 24-33
- [15] VILAIN, B. e CHABERNAUD, C. "Telecommunication Services and Distributed Applications", v. 4 , nº 6, Nov., 1990, p. 10-13.
- [16] OLIVEIRA, A. M. B. et al. "Uma Plataforma de Computação para Administração de ICBNs", *10º Simpósio Brasileiro de Redes de Computadores*, Recife, 1992, p. 483-496.
- [17] ROSE, M. T. *ISODE - The ISO Development Environment User's Manuals*, 5a ed., Wollongong Group, 1989.
- [18] NG, P. A. e ROSSAK, W. "Some Thoughts on Systems Integration: A Conceptual Framework", *Journal of Systems Integration*, v. 1, nº 1, Jul., 1991, p. 97-114.
- [19] VACCA, J. "Modest Progress", *LAN Magazine*, v. 7, nº 7, Jul., 1992, p. 77-83.
- [20] KING, S. S. "Middleware - Making the Network Safe for Application Software", *Data Communications*, v. 21, nº 4, Mar., 1992, p. 58-67.
- [21] HAAS, Z. "A Communication Architecture for High-Speed Networking", *IEEE Infocom - Conference on Computer Communications*; San Francisco, Junho, 1990, p. 433 - 441.
- [22] CHLAMTAC, I. and FRANTA, W. R. "Rationale, Directions and Issues Surrounding High Speed Networks", *Proceedings of the IEEE*, v. 78, nº 1, Jan., 1990, p. 94-120.
- [23] EIGEN, D. J. "Narrowband and Broadband ISDN CPE Directions", *IEEE Communications Magazine*, v. 28, nº 4, Abril, 1990, p. 39-46.
- [24] TAUBE, C. A. "Uma Introdução à RDSI de Faixa Larga e aos Sistemas Assíncronos de Comutação (ATM)", *Revista Telebrás - Edição Tecnologia*, v. 15, nº 54, Dezembro, 1991, p. 5-9.
- [25] MOURA, J. A. B. *Redes Locais de Computadores: Protocolos de Alto-Nível e Avaliação de Desempenho*, McGraw-Hill, 1986.
- [26] SCHLANGER, G. G. "An Overview of Signaling System Nº 7", *IEEE Journal of Selected Areas in Communications*, v. SAC-4, nº 3, Maio, 1986.
- [27] HUTCHISON, G e DESMOND, C. L. "Electronic Data Interchange", *IEEE Network*, v. 1, nº 4, Out., 1987, p. 16-20.

- [28] LEVY, H. M. and TEMPERO, E. D. "Modules, Objects and Distributed Programming: Issues in RPC and Remote Object Invocation", *Software - Practice and Experience*, v. 21, n° 1, Jan., 1991, p. 77-90.
- [29] ANDREWS, G. R. "Paradigms for Process Interaction in Distributed Programs", *ACM Computing Surveys*, v. 23, n° 1, Março, 1991, p. 49-90.
- [30] JONES, V. C. *MAP/TOP Networking*, McGraw-Hill, 1988.
- [31] MARTINS, JOBERTO S. B. "Redes Industriais: O Estado da Arte da Tecnologia", *1º Seminário sobre Redes de Comunicação Industrial*, São Paulo, SP, 1990, pp. 1-10.
- [32] GREEN, P. E. "The Future of Fiber-Optic Computer Networks", *IEEE Computer*, v. 24, n° 9, Set., 1991, p. 78-87.
- [33] MARTIN, J. *SNA: A Solução IBM para Redes*, Editora Campus, 1990.
- [34] JOHNSON, J. T. "NREN: Turning the Clock Ahead on Tomorrow's Networks", *Data Communications*, v. 21, n° 12, Setembro, 1992, p. 43-62.
- [35] MELEIS, H. E. and SERPANOS, D. N. "Designing Communication Subsystems for High-Speed Networks", *IEEE Network*, v. 6, n° 4, Julho, 1992, p. 40-46.
- [36] CHANSON, S. T. et all. "The Design of a Transport Protocol for Local Area Networks", *IEEE Infocom*, 1988, p. 631-640.
- [37] MEISTER, B. W. "A Performance Study of the ISO Transport Protocol", *IEEE Transactions on Computers*, v.40, n° 3, Mar., 1991, p. 253-262.
- [38] SVOBODOVA, L. "Measured Performance of Transport Services in LANs", *Computer networks and ISDN Systems*, v. 18, n° 1, 1989, p. 31-45.
- [39] CHIN, R. S. e CHANSON, S. T. "Distributed Object-Based Programming Systems", *ACM Computing Surveys*, v. 23, n° 1, Março, 1991, p. 91-124.
- [40] CARUSO, R. E. "Network Management; A Tutorial Overview", *IEEE Communications Magazine*, Março, 1990, p. 20-25.
- [41] DANTHINE, A. A. S. "Communication Support for Distributed Systems - OSI versus Special Protocols", *Proceedings of the Information Processing - IFIP*, 1989, p. 181-190.
- [42] HAAS, Z. "A Protocol Structure for High-Speed Communication over Broadband ISDN", *IEEE Network Magazine*, Janeiro, 1991, p. 64 - 70.

- [43] MARTINS, JOBERTO S. B. and TURNELL, D. J. "Considerations About the Implementation of Mini-MAP Architectures on a Front-End Processor", *Proc. IFAC - 4th Latin American Congress on Automatics*, Puebla, México, Nov., 1990, P. 1-12.
- [44] LA PORTA, T. and SCHWARTZ, M. "Architectures, Features and Implementation of High-Speed Transport Protocols", *IEEE Network Magazine*, v. 4, n° 2, Fev., 1991, p. 14-22.
- [45] CLARK, D. D. et al. "An Analysis of TCP Processing Overhead", *IEEE Communications Magazine*, Junho, 1989.
- [46] FURTH, B. and MILUTINOVIC, V. "Survey of Microprocessor Architectures for Memory Management", *IEEE Computer*, Março, 1987, p. 48-67.
- [47] HALSALL, F. *Data Communications, Computer Networks and OSI*, 2 ed. Addison-Wesley, 1988, 549 p.
- [48] MARTINS, JOBERTO S. B. "Un Support de Communication pour le Système Unix/Ethernet", These de Doctorat de L'Université Paris VI, Paris, França, 1986, 288 p.
- [49] MARTINS, JOBERTO S. B. and NOGUEZ, G. "Virtualisation des Ressources Matérielles à l'Aide d'un Réseau Local Ethernet sous Unix", Congrès des Nouvelles Architectures pour les Communications, Paris, França, Set., 1984, p. 227-233.
- [50] DEITEL *An Introduction to Operating Systems with Case Studies*, 1a. ed., Addison-Wesley, 1984.
- [51] ROCHKIND, M. J. *Advanced Unix Programming*, 1a. ed., Prentice-Hall, 1985.
- [52] REISS, S. P. "Connecting Tools Using Message Passing in the Field Environment", *IEEE Software*, v.7, n° 4, Jul., 1990, p. 57-66.
- [53] ZITTERBART, M. "High-Speed Transport Components", *IEEE Network Magazine*, Janeiro, 1991, p. 54 - 63.
- [54] ZITTERBART, M. "A Multiprocessor Architecture for High-Speed Network Interconnections", *Proceedings of the Infocom - Conference on Computer Communication*, Ottawa, Canada, Abril, 1989.
- [55] HAAS, Z. "A Protocol Structure for High-Speed Communication over Broadband ISDN", *IEEE Network Magazine*, Janeiro, 1991, p. 64 - 70.

- [56] LAI, W. S. "Protocols for High-Speed Networking", *Proceedings of the Infocom - Conference on Computer Communications*, San Francisco, Junho, 1990, p. 1268-1269.
- [57] KAISERSWERTH, M. "Lightweight Transport Protocols for High-Speed Networks", *Internal Report*, IBM Research Laboratory, Zurich, 1990.
- [58] COMER, D. *Internetworking with TCP/IP - Principles, Protocols and Architecture*, Prentice-Hall, 1988.
- [59] TANENBAUM, A. S. *Computer Networks*, 2 ed., Prentice-Hall, 1988.
- [60] TURNELL, D. J. and MARTINS, JOBERTO S. B. "Uma Implementação das Camadas MAC/LLC para a Interface Mini-MAP", *Anais do 8° SBRC - Simpósio Brasileiro de Redes de Computadores*, Campinas, SP, 1990, p. 285-297.
- [61] TURNELL, D. and MARTINS, JOBERTO S. B. "Executivo Multi-Tarefa para Processador Frontal de Rede", *Anais do 7° SBRC - Simpósio Brasileiro de Redes de Computadores*, Porto Alegre, RS, 1989. p. 721-730.
- [62] FRAME, M. "Broadband Service Needs", *IEEE Communications Magazine*, v.28, n° 4, Abril, 1990, p. 59-62.
- [63] CHERITON, D. R. "VMTP as the Transport Layer for High-Performance Distributed Systems", *IEEE Communication Magazine*, Junho, 1989.
- [64] WATSON, R. W. et al. "The Delta-t Transport Protocol", *Protocols for High Speed Networks*, North-Holland, 1989.
- [65] SPHOR, E. M., AGUIAR, M. and MARTINS, JOBERTO S. B. "Aspectos da Interface entre as Camadas MMS e LLC-3 em uma Arquitetura Mini-MAP", *11° SEMISH - Congresso Nacional da Sociedade Brasileira de Computação*, Santos, SP, 1991, p.157-171.
- [66] RAHNEMA, M. "Frame Relaying and the Fast Packet Switching Concepts and Issues", *IEEE Network Magazine*, v. 5, n° 4, Julho, 1991, p. 18-23.
- [67] HEINANEN, J. "Review of Backbone Technologies", *Computer Networks and ISDN Systems*, v. 21, n° 4, Junho, 1991, p. 239-245.
- [68] GIOZZA, W. F. e outros - *Fibras Ópticas: Tecnologia e Projeto de Sistemas*, McGraw-Hill, 1991.
- [69] SMITH, M. J. S. "More Logic Synthesis for ASICs", *IEEE Spectrum*, v. 29, n° 11, Nov., 1992, p. 44-48.

[70] SALTZER, J. H. et al. "End-to-End Arguments in System Design", *Proceedings of the 2° International Conference on Computer Communications*, Abril, 1981.

[71] McCOOL, J. F. "FDDI - Getting to Know the Inside of the Ring", *Data Communications*, Março, 1988, p. 185-192.

[72] POZEFSKY, D. P. et al. "SNA's Design for Networking", *IEEE Network*, 6, Novembro, 1992, p. 18-31. vii

[73] PANDHI, S. N. "The Universal Data Connection", *IEEE Spectrum*, Julho, 1987, p. 31-37.

[74] NUTT, G. J. *Centralized and Distributed Operating Systems*, Prentice-Hall, 1992, 418p.

[75] BOGGS, R. and METCALFE, R. M. "Ethernet: Distributed Packet Switching for Local Computer Networks", *Communications of the ACM*, v. 19, n° 7, Jul., 1976, p. 395-403.

[76] MCMULEEN, M. "Token Ring Tales", *LAN Magazine*, v. 7, n° 9, Setembro, 1992, p. 87-106.

[77] SCHNAIDT, P. "Twisted-Pair FDDI", *LAN Magazine*, v. 7, n° 9, Setembro, 1992, p. 19-20.

[78] LYLES, J. B. e SWINEHART, D. C. "The Emerging Gigabit Environment and the Role of Local ATM", *IEEE Communications Magazine*, v. 30, n° 4, Abril, 1992, p. 52-58.

[79] CLIZENIT, P. A. L. and MARTINS, JOBERTO S. B. "Estruturação e Implementação do Protocolo FTAM", *8° SBRC - Simpósio Brasileiro de Redes de Computadores*, Campinas, SP, 1990, p. 1-14.

[80] HENKEN, G. "Development and Interconnection of X.400 Message Handling Systems", *Computer Standards and Interfaces*, n° 7, Set., 1988, p. 17-22.

[81] CHADWICK D. W. "Tutorial on X.500 - The Directory", *Proceedings of the IFIP Informations Network and Data Communication III*, North-Holland, 1990, p. 35-51.

[82] LEVY, H. M. e TEMPERO, E. D. "Modules, Objects and Distributed Programming: Issues in RPC and Remote Object Invocation", *Software - Practice and Experience*, v. 21, n° 1, Jan., 1991, p. 77-90.

[83] SABHARWAL, H. et all. "SQL Access: An Implementation of the ISO Remote Database Access Standard", *IEEE Computer*, v. 24, n° 12, Dez., 1991, p. 74-78.

- [84] SEO, J. and KIM, W. "Classifying Schematic and Data Heterogeneity in Multidatabase Systems", *IEEE Computer*, v. 24, n° 12, Dez., 1991, p. 12-18.
- [85] DUPUY, A. et all. "NEST : A Network Simulation and Prototyping Testbed", *Communications of the ACM*, v. 33, n° 10, Out., 1990, p. 63-74
- [86] VALENZANO, A. et all. "Performance Analysis of Type 3 LLC in Industrial 802.5 Networks", *IEEE Network*, v. 2, n° 3, Maio, 1988, p. 90-96.
- [87] VAN ZIJL, L. e MITTON, D. "A Tool For Graphical Network Modeling and Analysis", *IEEE Software*, v. 9, n° 1, Jan., 1992, p. 47-54.
- [88] TURNELL, D. J. "A Tool Kit for Event-Driven Network Modeling", *Relatório Técnico do Grupo de Redes de Computadores*, n° 01, 1992, 08 p.
- [89] TURNELL, D. J. OONET - Uma Ferramenta para a Modelagem de Redes de Computadores, *Relatório Interno - Proposta de Tese de Doutorado*, UFPB, 1992.
- [90] FORTE, G. "Tools Fair: Out of the Lab, Onto the Shelf", *IEEE Software*, v. 9, n° 3, Maio, 1992, pp 70-79.
- [91] JAIN, S. et all. "Expert Simulation for On-Line Scheduling", *Communications of the ACM*, v. 33, n° 10, Out., 1990, p. 54-60.
- [92] SHIRATORI, N. e TAKAHASHI, K. "Using Artificial Intelligence in Communications System Design", *IEEE Software*, v. 9, n° 1, Jan., 1992, p. 38-46.
- [93] BERRY, J. *The Waite Group's C++ Programming*, Howard W. Sams & Company, 1988, 381 p.
- [94] PESSOA, P. M. C. e CARNEIRO, M. C. C. Evolução da Rede FDDI para a Integração de Serviços, *Revista Telebrás - Edição Tecnologia*, v. 15, n° 54, Dezembro, 1991, p.10-20.
- [95] FINK, R. L. e ROSS, F. E. "Following the Fiber Distributed Data Interface", *IEEE Network*, Março, 1992, p. 50-55.
- [96] BOCKE, P. e ARMBRUSTER, H. "Broadband Services: An Overview", *Telecommunications*, v. 25, n° 12, Dez., 1991, p. 24-32.
- [97] BOTTOMLEY, J. F. "Any-to-Any Networking: Getting There From Here", *Data Communications*, Setembr, 1992, p. 69-80.
- [98] BIRSACK, E. W. e outros "Gigabit Networking Research at Bellcore", *IEEE Network*, Março, 1992, p. 42-48.