

UMA METODOLOGIA PARA ALOCAÇÃO DE ARQUIVOS EM  
UM AMBIENTE DE PROCESSAMENTO DE DADOS DISTRIBUIDOS

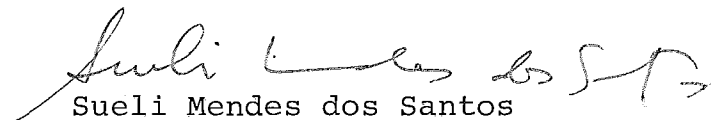
FRANKLIN JOSÉ M. DA TRINDADE

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovado por:



Luiz Antonio C. da C. Couceiro  
Presidente



Sueli Mendes dos Santos



Jayme Luiz Szwarcfiter

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 1984

TRINDADE, FRANKLIN JOSÉ MARIBONDO DA

Uma Metodologia para Alocação de Arquivos em um Sistema de Processamento de Dados Distribuídos. (Rio de Janeiro) 1984.

IX, 179 p. 29,7 cm (COPPE-UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1984).

Tese - Universidade Federal do Rio de Janeiro, Faculdade de Engenharia.

1. Sistemas de Processamentos de Dados Distribuídos.

I. COPPE/UFRJ.      II. Título (série).

As minhas filhas

Renata e Fernanda

A minha mulher

Ilma

Aos meus pais

Cipriano e Maria

AGRADECIMENTOS

Sou muito grato.

Ao meu orientador, Prof. Luiz Antonio C. da C. Couceiro, pela forma segura como conduziu a realização deste trabalho.

A todo corpo docente do curso de Pós-Graduação de Engenharia de Sistemas e Computação, e em particular aos meus Professores: Beatriz Zakimi Miyasato, José Lucas M. Rangel Netto, Sueli Mendes dos Santos, Paulo Mário Bianchi França, Luiz Antonio C. da C. Couceiro, José Fábio Marinho de Araújo, Miguel Jonathan, Estevam De Simone e Leila Eizirik, pela orientação recebida durante todo o curso.

As autoridades da Aeronáutica que, de forma direta ou indireta, contribuíram para que fosse possível a minha participação neste curso e em especial aos: Brig. Eng. Tércio Pacitti, Brig. Av. Leilo Vianna Lobo, Cel. Eng. Paulo Dantas Cabral e o Cel. Int. Aramis Tavares da Silva.

Ao chefe do Centro de Computação da Aeronáutica - C.C.A. RJ, Tcel. Av. Victorio Baptista da Silva, que colocou a minha disposição os recursos do Sistema IBM 4341, onde foram realizados os trabalhos de computação desta tese.

A Eliana Arndt Machado da Silva, pelo carinho e eficiência nos trabalhos de datilografia.

Aos companheiros que muito me incentivaram.

E finalmente a todos aqueles que de alguma forma contribuíram para que este trabalho fosse realizado.



Resumo da Tese Apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.).

Uma Metodologia para Alocação de Arquivos em  
Um Ambiente de Processamento de Dados Distribuídos

Franklin José M. da Trindade

Janeiro de 1984

Orientador: Luiz Antônio C. da C. Couceiro

Programa: Engenharia de Sistemas e Computação

O desenvolvimento tecnológico alcançado pela microeletrônica nestes últimos anos, contribuindo para o baixo custo dos computadores, provocou a disseminação de microcomputadores no âmbito das empresas públicas, privadas e mais recentemente também no campo de ação doméstico, através dos microcomputadores pessoais.

Entretanto esta evolução tecnológica tem influenciado as arquiteturas dos Sistemas de Processamento de Dados das empresas, através da utilização dos conceitos e idéias que descentralizam e distribuem os recursos de computação até então centralizados em um único Centro de Processamento de Dados.

Este trabalho discute uma metodologia para alocação de dados em um ambiente de processamento distribuído, baseada em métodos de programação matemática e heurística.

Inicialmente será feita uma apreciação do binômio Computação-Comunicação, que representam os componentes globais de um Sistema Distribuído, com o objetivo de fazer um posicionamento do contexto do trabalho, e em seguida será feita uma descrição do método proposto.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

A Methodology for File Allocation in a  
Distributed Data Processing Environment

Franklin José M. da Trindade

January, 1984

Chairman: Luiz Antônio C. da C. Couceiro

Department: Engenharia de Sistemas e Computação

The research and development in microelectronics is contributing in reducing the computer cost, stimulating the dissemination of micro computers among public and private companies and, recently, in people's life.

This development is also influencing the design of Data Processing Systems on most companies through the use of Decentralized and Distributed Data Processing concepts and ideas.

This work discusses a methodology for the data allocation on distributed data processing environment. This work is based on mathematic programming and heuristic methods.

Firstly an appreciation of binomial Computation-Communication will be made, which symbolizes the global components in a "Distributed System", having in mind to do placement of the context of the work. Afterwards a description of the method proposed will be presented.

INDICE

	<u>Pág.</u>
I - INTRODUÇÃO .....	01
II - SISTEMAS DE PROCESSAMENTO DE DADOS DISTRIBUÍDOS .....	05
2.1 - Introdução .....	05
2.2 - Histórico .....	05
2.3 - Evolução dos Sistemas de Processamento de Dados ..	09
2.4 - Rede de Comunicação de Computadores .....	11
2.5 - Conceitos de um Sistema de Processamento da Dados Distribuídos .....	19
2.6 - Objetivos de um Sistema de Processamento de Dados Distribuídos .....	21
2.7 - Razões que conduziram ao Sistema de Proces- samento de Dados .....	25
2.8 - Características indispensáveis de um Siste- ma de Processamento de Dados Distribuídos .....	36
2.9 - Tipos de arquiteturas .....	37
III - ALOCAÇÃO DE ARQUIVOS EM UM S.P.D.D. ....	44
3.1 - Considerações gerais sobre Alocação de Arquivos ..	44
3.2 - Formulação de modelos de Alocação de Arquivos ....	53
3.3 - Determinação de uma função ótima de Alocação de Arquivos .....	62
3.4 - Restrições de um modelo de Alocação de Arquivos ..	63
3.5 - Métodos para solução do problema de Alocação de Arquivos .....	64
3.6 - Exemplos de alguns trabalhos sobre Alocação de Arquivos .....	69

	<u>Pág.</u>
IV - CARACTERIZAÇÃO DA METODOLOGIA PROPOSTA PARA ALOCAÇÃO DE ARQUIVOS EM UM S.P.D.D. ....	73
4.1 - Introdução .....	73
4.2 - Fase de concepção de um S.P.D.D. ....	76
4.3 - Fase de elaboração do Projeto .....	78
4.3.1 - Distribuição Lógica .....	80
4.3.1.1. Análise dos Dados .....	81
4.3.1.2. Análise dos Software .....	85
4.3.1.2.1 - Software de Apoio ...	86
4.3.1.2.2 - Software de Aplicação..	87
4.3.1.3. Análise das Localidades .....	90
V - DESCRIÇÃO E FORMULAÇÃO DO MÉTODO DE ALOCAÇÃO PROPOSTO ..	97
5.1 - Descrição do Modelo de Alocação .....	97
5.2 - Fatores Considerados no Método de Alocação .....	100
5.3 - Condições Estabelecidas no Modelo .....	113
5.4 - Formulação do Método de Alocação Proposto .....	115
5.5 - Conceituação do Método de Solução .....	121
5.5.1 - Solução para o Problema de Alocação sem redundância .....	121
5.5.2 - Solução para o Problema de Alocação com Redundância .....	129
5.5.2.1 - Condição Geral para Duplicação de Arquivos .....	133
5.5.2.2 - Procedimentos para duplicação de Arquivos .....	133
5.6 - Considerações Finais da Alocação .....	137

	<u>Pág.</u>
VI - MÉTODO DE AVALIAÇÃO DE DESEMPENHO DE UM S.P.D.D. ....	140
6.1 - Introdução .....	140
6.2 - Medidas de Desempenho .....	142
6.2.1 - Análise do Atraso .....	143
6.2.2 - Throughput .....	145
6.2.3 - Capacidade dos Canais de Comunicações .....	146
6.2.4 - Tráfego na Rede .....	146
6.3 - Formulações para Análise do Desempenho .....	149
6.3.1 - Formulação Considerando Capacidade das Linhas .....	149
6.3.2 - Formulação Considerando o Direcionamento ..	150
6.3.3 - Formulação Considerando a Capacidade e Fluxo .....	150
6.4 - Avaliação de Desempenho em um S.P.D.D. ....	153
6.4.1 - Objetivos .....	153
6.4.2 - Descrição do Método de Avaliação Proposto ..	154
6.5 - Formulação do Método de Avaliação Proposto .....	158
6.5.1 - Os Dados de Entrada .....	158
6.5.2 - Objetivos a Determinar .....	159
6.5.3 - Procedimentos do Método de Avaliação .....	160
6.6 - Considerações Finais da Avaliação .....	163
VII - CONCLUSÃO .....	165
REFERÊNCIAS .....	168
ANEXOS .....	179

## CAPÍTULO I

### I N T R O D U Ç Ã O

O ponto fundamental do desenvolvimento de sistemas, sejam eles ou não, projetados para atuarem com o apoio de computadores, está na solução das questões ligadas à identificação dos seus elementos componentes e na consequente definição de suas propriedades e inter-relacionamentos.

Por outro lado o suporte oferecido pela Teoria Geral de Sistemas à tarefa de analisar e projetar sistemas é muito mais uma filosofia, uma atitude geral diante de problemas complexos, do que um método formal de trabalho. Também os diversos aspectos do conceito de sistemas, enfatizados por diversos autores, chegam até aos analistas muito mais em termos de enganos a serem evitados, do que um programa coerente de ação.

A conclusão a que se pode chegar é que embora com a existência de métodos formais e idéias que assistam os Analistas e Projetistas de Sistemas, e com mais ênfase nos sistemas projetados para atuarem com computadores, a execução de suas atribuições é ainda extremamente dependente de sua experiência profissional e criatividade.

Na caminhada para se alcançar algum objetivo, grande seria o esforço se os nossos pensamentos e atos não fossem regulados por um plano que indicasse os passos a seguir, em segurança, e com um dispêndio mínimo de energias. Esse caminho é o método, que conduzirá ao objetivo com o máximo de eficiência e um consumo

mínimo de esforço. Descartes, que foi o primeiro filósofo que cuidou especificamente do estudo do Método, dizia: "Não é suficiente ter um bom espírito; principal é sabê-lo aplicar bem".

A ciência por si só é fundamentalmente metódica e ela exige em todas as suas divisões que o método seja um meio de se alcançar a eficiência máxima. E a ciência da Computação em todo seu contexto não está destinada a vir a ser uma exceção dentro desta regra.

O trabalho que oferecemos representa, dentro deste espírito, uma contribuição a Engenharia de Sistemas e Computação, na apresentação de uma metodologia para designação de arquivos em um projeto de um Sistema de Processamento de Dados Distribuídos.

O desenvolvimento de um Sistema de Processamento de Dados, seja ele do tipo distribuído ou não, está fundamentado na solução de três importantes questões, que estão relacionadas e configuradas na figura 1 abaixo.

- Capacidade de Processamento
- Dados
- Controle

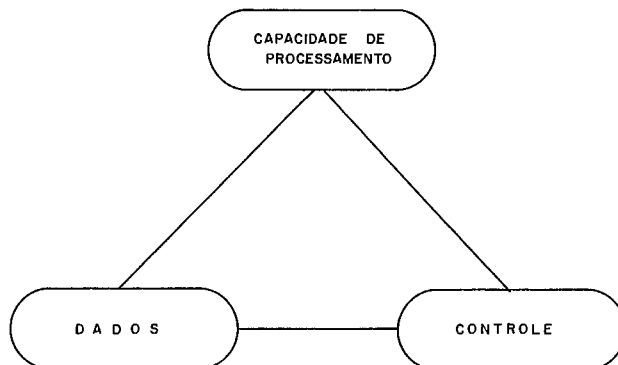


Figura 1

Nos tipos de Sistemas ditos convencionais, estas questões têm sido resolvidos com relativa precisão no que diz respeito ao seu dimensionamento e custos, isto porque em razão de todos estes componentes estarem situados em uma única instalação, eles oferecem apenas uma opção de solução, ou seja a centralização dos dados, controle e processamento, restando apenas um criterioso ba lanceamento entre estes componentes que ofereça uma arquitetura e ficiente em termos de desempenho do sistema de processamento.

Entretanto nos Sistemas Distribuídos que se caracterizam basicamente pela distribuição da capacidade de processamento entre diversos locais, as questões relativas aos dados e ao controle passam a ter diferentes opções de solução tais como:

- Distribuir os dados e manter o controle centralizado em um único local;
- Distribuir o controle e manter os dados centralizados em um único local;
- Distribuir ambos, dados e controle.

Nesta tese que apresentamos nos dedicaremos com mais ênfase aos problemas relativos a distribuição dos dados, não deixando entretanto de fazer abordagens relativas ao controle e a capacidade de processamento quando e na dimensão que se fizerem necessárias.

Convém finalmente ressaltar que a solução de todas estas questões envolvem uma série de técnicas e fases de desenvolvimento, porém o seu número e tipo de técnica a utilizar dependem das características do sistema que queremos projetar.



No próximo capítulo estudaremos todos os aspectos conhecidos de um Sistema de Processamento de Dados Distribuído. Definiremos seus objetivos, estabeleceremos as razões de sua existência e finalmente atribuiremos as suas vantagens e desvantagens de emprego em relação a outros tipos de arquiteturas.

Em seguida dedicaremos um capítulo ao estudo particular do problema de alocação de arquivos em uma rede de computadores.

Nos capítulos seguintes descreveremos um método para desenvolvimento de um projeto de Sistemas de Processamento de Dados Distribuídos, cujo enfoque principal está dirigido à apresentação de um modelo de alocação de arquivos e avaliação de desempenho do Sistema.

## CAPÍTULO II

### SISTEMAS DE PROCESSAMENTO DE DADOS DISTRIBUÍDOS

#### 2.1 - INTRODUÇÃO

Neste capítulo inicial faremos um estudo dos principais componentes de uma rede de computadores, bem como os fatores que condicionam ao Processamento de Dados Distribuídos.

Iniciaremos com um breve histórico sobre a origem e evolução dos computadores através dos tempos e suas respectivas influências nos Sistemas Computacionais das empresas. Em seguida veremos os aspectos principais que conduziram as empresas a distribuírem a sua carga de processamento, e finalmente mostraremos as principais vantagens e desvantagens existentes nas arquiteturas centralizadas e distribuídas.

A apresentação destes fatos neste capítulo inicial tem como objetivo oferecer uma descrição sucinta de todo contexto em que se insere este trabalho, a fim de proporcionar um bom entendimento sobre os assuntos que serão tratados nos capítulos seguintes.

#### 2.2 - HISTÓRICO

Qualquer tentativa no sentido de se estabelecer um histórico sobre a evolução dos computadores eletrônicos de aplicação geral, inicia-se na década de 40, e mais precisamente em 1945,

quando os professores S. P. Eckert e J. W. Mauchly, utilizando as instalações da Moore School of Engineering, na Universidade da Pensilvânia, construíram um grande computador de aplicação geral denominado ENIAC (Eletronic Numerical Integrator and Calculator). Este "cérebro mecânico", como assim era chamado, foi utilizado basicamente para resolver problemas matemáticos ligados as áreas de balística e aeronáutica.

Entretanto a origem do computador digital tem início em 1812 com Charles P. Babbage, um cientista e matemático Inglês, cuja contribuição mais expressiva que deixou foi a chamada "Máquina das Diferenças". Esta máquina projetada por Babbage, utilizava cartões de cartolina contendo buracos que serviam tanto para introduzir as instruções quanto os dados que fossem necessários.

Em 1833 Babbage concebeu uma outra idéia que seria a de construir uma máquina analítica capaz de executar qualquer cálculo. Seria então o primeiro computador digital para todas as finalidades. Ele trabalhou neste projeto pelo resto de sua vida morrendo em 1871 deixando o seu trabalho incompleto. Mas, apesar das limitações tecnológicas da época, Babbage conseguiu introduzir os princípios fundamentais sobre os quais são construídos os computadores modernos.

Quase 100 anos depois as idéias de Babbage passaram a ser aperfeiçoadas de forma que, em maio de 1944, através dos esforços do físico Howard Aiken da Universidade de Harvard, surgiram as calculadoras automáticas de seqüência controlada denominadas Mark-I, Mark-II, ... etc.

O passo seguinte nesta evolução foi o surgimento do ENIAC, o primeiro computador totalmente eletrônico. E a partir de então outras máquinas foram desenvolvidas. O EDVAC (Electronic Discret Variable Automatic Computer), que foi o primeiro computador eletrônico de processamento comercial do mundo. O EDSAC (Electronic Delayed Storage Automatic Computer), o ACE (Automatic Computer Engine), o UNIVAC (Universal Automatic Computer) que tornou-se famoso por ter previsto a vitória de Eisenhower na eleição de 1952.

A partir desta época centenas de pequenos e grandes computadores foram fabricados comercialmente, caracterizando a fase dos computadores de primeira geração. Estes computadores eram volumosos, inflexíveis, utilizavam válvulas eletrônicas, e foram também os primeiros a permitirem programação interna por circuito, o que possibilitava a comparação e tomada de decisões lógicas.

Com a substituição das válvulas pelo transistor, inicia-se a fase dos computadores chamados de segunda geração. A velocidade de processamento aumentou consideravelmente e muitos computadores de médio e grande porte foram projetados. Nesta fase os equipamentos periféricos foram aperfeiçoados, tornando possível a existência de impressoras e leitoras de alta velocidade. Técnicas mais sofisticadas de programação também foram desenvolvidas nesta fase e o tamanho e importância da indústria de computadores cresceu em um ritmo acelerado.

Em 1965 a IBM Corporation introduziu no mercado a sua série 360, que fazia uso intensivo de "circuitos integrados", SSI (Small Scale Integration) nos quais muitos transistores e outros

componentes são fabricados e compactados juntos num único recipiente de pequenas dimensões. Os preços baixos e as altas densidades de integração destes circuitos, adicionados as lições aprendidas nas máquinas anteriores, conduziram a certas diferenças no projeto dos computadores desta época, caracterizando a chamada terceira geração dos computadores eletrônicos digitais. Também a utilização bem mais eficiente dos dispositivos de entrada e saída e de acesso direto, permitem maiores facilidades de armazenar informações operacionais. Além disso, equipamentos de transmissão de dados tornam fácil o envio de informações de qualquer área para a memória do computador conjugando as ações de dois ou mais computadores e permitindo que, de estações remotas, se processem registros de informações ou consultas ao computador.

Finalmente, a fabricação dos circuitos integrados tornou-se tão avançada que permite incorporar milhares de componentes ativos em um volume de uma fração de polegada, levando ao que se chama integração em média escala (MSI), integração em larga escala (LSI) e a integração em muito larga escala (VLSI). Este avanço tecnológico caracteriza os computadores chamados de quarta geração, que tornaram-se em sua maioria de pequeno porte, baixo custo, grande memória e ultra rápidos. Os primeiros computadores deste gênero, denominados de minicomputadores apareceram em aplicações aeroespaciais. Em 1962 a Computer Central Corporation e a Digital Equipment Corporation lançaram os primeiros minis para uso civil, voltados para aplicações em laboratórios. Desde então este segmento da indústria dos computadores cresceu em média 30% ao ano versus 7% a 10% ao ano para os grandes computadores que

por sua vez tem-se tornado cada vez mais complexos. Existem atualmente cerca de 50 fabricantes de porte na área de micros e minicomputadores. Os minis e microcomputadores tem hoje uma enorme gama de possíveis aplicações e suas formas de atuação podem ser sem interligação com outros sistemas (STAND-ALONE) ou interligados em uma rede de computadores.

### 2.3 - EVOLUÇÃO DOS SISTEMAS DE PROCESSAMENTO DE DADOS

Desde o aparecimento dos primeiros computadores de uso geral no início da década de 50, até os dias de hoje, a implantação dos Sistemas de Processamento de Dados nas Empresas, passaram por uma série de transformações em razão da necessidade de acompanhar o desenvolvimento tecnológico dos computadores, da diversificação cada vez maior do seu uso, e das formas cada vez mais simples, que são oferecidas, na interação do computador com o elemento humano que o utiliza.

Inicialmente com os computadores de primeira geração os Sistemas de Processamento de Dados eram projetados para serem utilizados na forma de uma pessoa por vez, através de complexos painéis. Eram os chamados Sistemas Simples de Entrada e Saída. Posteriormente, até o fim da década de 50, os trabalhos passaram a serem submetidos ao computador sob forma de lotes (BATCH) e os sistemas passaram a ser multiprogramáveis. As aplicações computacionais, em sua totalidade, eram destinadas a um único usuário. Nesta fase predominava a organização dos sistemas computacionais de forma Centralizada. A tendência dominante era reunir os equipamentos em um único local.

A partir desta época, até os fins dos anos 60, quando os computadores de terceira geração passaram a predominar no mercado, a implementação dos Sistemas de Processamento de Dados nas empresas públicas e privadas pareciam destinados a solucionar definitivamente todos os problemas no tratamento das informações. Estas máquinas, extremamente rápidas, precisas e de custo cada vez mais acessível à maioria das empresas, tornaram-se, durante certo tempo, a solução desejada. Entretanto logo cedo, uma grande parte dos utilizadores de Sistemas Computacionais, perceberam estar diante de um problema que neutralizava uma das maiores virtudes dos seus computadores, qual seja a sua velocidade. Suas informações eram rapidamente processadas pelo computador porém ficavam enfileiradas em algum sistema de expedição, responsável para fazê-las chegarem ao seu destino. A partir de então surgiu a necessidade de que alguns computadores e terminais estivessem interligados sob alguma forma de comunicação, a fim de proporcionar meios para que os processos se comunicassem de uma forma mais eficiente.

A comunicação de dados veio viabilizar o acesso do usuário ao computador através de terminais remotos, utilizando o compartilhamento de tempo (TIME SHARING). Entretanto estas formas operacionais existentes (BATCH e TIME SHARING), que são normalmente empregadas em sistemas centralizados, eram gerenciados pelo computador em todas as suas funções, causando um alto índice de "overhead", pelo software controlador. Estes problemas foram minimizados através da introdução dos processadores de comunicações que liberava o computador das funções de comunicação.

A participação dos processadores de comunicação neste contexto, possibilitou a partir dos anos 70, a introdução de diversas arquiteturas. Inicialmente, de uma forma bem tímida, apenas um computador podia estar ligado a um ou diversos terminais, sendo o único responsável pelo gerenciamento de todo o sistema. Progressivamente, acompanhando o desenvolvimento tecnológico dos computadores, esta comunicação passou a tomar formas mais arrojadas chegando aos dias de hoje a um conceito denominado rede de computadores, sobre a qual se apoia todo um Sistema de Processamento de Dados Distribuído. Atualmente uma rede de computadores consiste de um certo número de computadores e terminais, em geral heterogêneos, interligados por um sistema de comunicação.

#### 2.4 - REDE DE COMUNICAÇÃO DE COMPUTADORES

A estrutura básica de uma Rede de Comunicação de Computadores é, segundo WONG <sup>86</sup> dividida em duas partes principais. A Sub-rede de Comunicação e a Sub-rede de Recursos. A Sub-rede de Comunicação é constituída dos nodos de comutação e dos canais de comunicações, cuja função é enviar mensagens de um nodo de comutação para um outro nodo de comutação. Ao conjunto de terminais e computadores da rede denomina-se a chamada Sub-rede de Recursos. Estes recursos são conectados aos nodos de comutação e comunicam-se com um outro recurso através das vias de comunicações. A figura 2 a seguir ilustra este tipo de ambiente descrito acima.



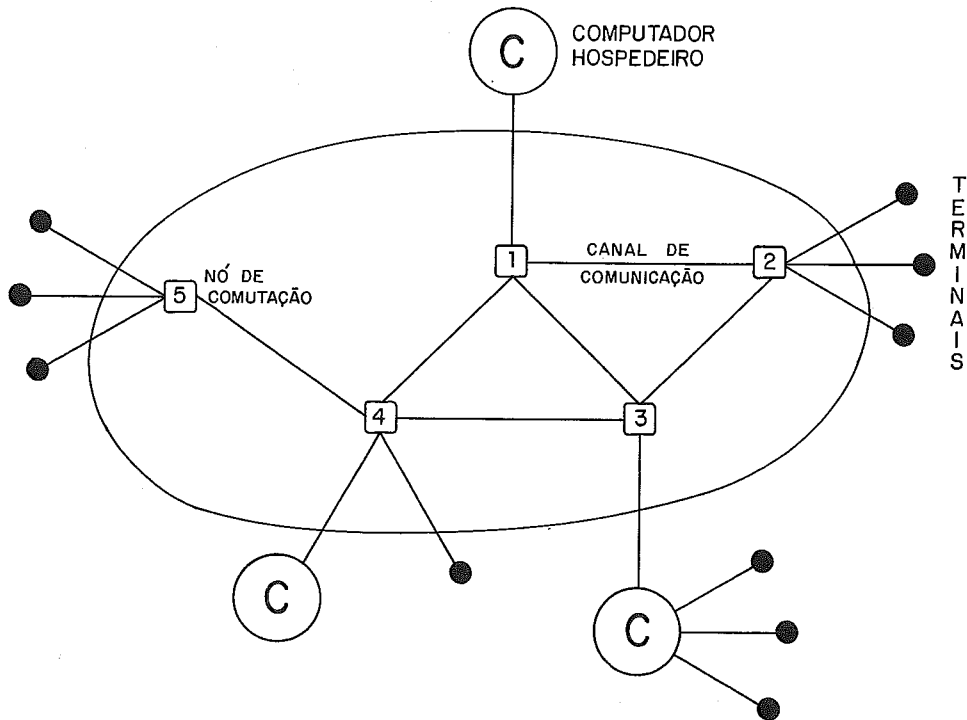


Figura 2

A técnica usualmente utilizada para enviar mensagens em uma Sub-rede de Comunicação é a Comutação de Mensagens. Nesta técnica, a mensagem a ser enviada a algum destino, é inicialmente armazenada em um nodo chamado fonte. A partir deste nodo fonte a mensagem é então dirigida para o canal de comunicação, previamente selecionado de acordo com o seu destino e o algoritmo de direcionamento utilizado, permanecendo na fila correspondente ao canal escolhido pelo processo de direcionamento. Quando o canal estiver disponível para enviar uma mensagem e de conformidade com uma política estabelecida é selecionada uma mensagem que está enfileirada e imediatamente transmitida para o próximo nodo ao longo do seu caminho. Este processo vai-se repetindo até que a mensagem atinja o seu destino.

Existe um tipo particular de rede onde estas mensagens são divididas em segmentos de tamanho fixo denominados de "PACOTES", e cada um deles pode ser direcionado independentemente através da rede para o seu destino. Este processo é conhecido como Comutação de Pacotes. KLEINROCK<sup>55</sup>. Aqui no Brasil a Embratel está trabalhando na implantação da RENPAC - Rede Nacional de Comutação de Pacotes, cuja ativação está prevista para o segundo semestre de 1984. A principal vantagem deste tipo de comunicação entre os diversos locais de uma rede está no fato de que os vários pacotes, que constituem uma mensagem inteira podem ser imediatamente transmitidos para o próximo nodo de comutação de sua rota até o seu destino, em vez de ter que esperar em cada nodo por onde passar, que toda a mensagem seja recebida para prosseguir. Outras motivações que conduziram a este tipo de técnica podem ser citados:

- Flexibilidade;
- Compartilhamento de recursos da rede reduzindo custos de comunicações;
- Tarifas dependem do uso;
- Elimina preocupações com otimização de topologia;
- Viabiliza aplicações de baixo tráfego;
- Viabiliza aplicações de baixo custo.

Uma descrição detalhada da evolução desta técnica, bem como seu processo de funcionamento pode ser encontrado em DAVIES<sup>31</sup>, TANENBAUM<sup>79, 80</sup>.

Os computadores que constituem a Sub-rede de Recursos são geralmente denominados de "Hospedeiros", todos eles desejando produzir ou consumir informações. Para que este intercâmbio torne-se exequível, principalmente em razão da heterogeneidade dos

elementos que constituem o Sistema, são implantados nos sistemas operacionais destes computadores um conjunto de procedimentos necessários para executarem as rotinas de Inicialização, Manutenção e Finalização do Software de Comunicação. A estas regras e procedimentos, que irão cuidar dos diversos aspectos da comunicação, dá-se o nome de Protocolos. Entre estes aspectos podemos enumerar:

- Que tipo de unidade de informação deve ser trocada entre os componentes;
- Que convenções devem ser utilizadas sobre definições de códigos, formatos de mensagens, velocidades de transmissões, etc.
- Estabelecer uma forma de endereçamento que permita as entidades se referenciarem;
- Estabelecer mecanismos de recuperação de erros que garantirão a uma mensagem chegar ao seu destino de forma correta;
- Oferecer um serviço de sequenciação, ou seja, garantir que as mensagens cheguem ao seu destino ordenadamente;
- Exercer um controle de fluxo, controle de prioridade, etc.

A Organização Internacional para a Padronização (ISO) definiu uma arquitetura chamada Interconexão de Sistemas Abertos (OSI) em sete camadas, como mostra a figura 3 a seguir, cuja descrição detalhada de cada camada pode ser encontrada em DAVIES<sup>31</sup>, TANEMBAUM<sup>79,80</sup>.

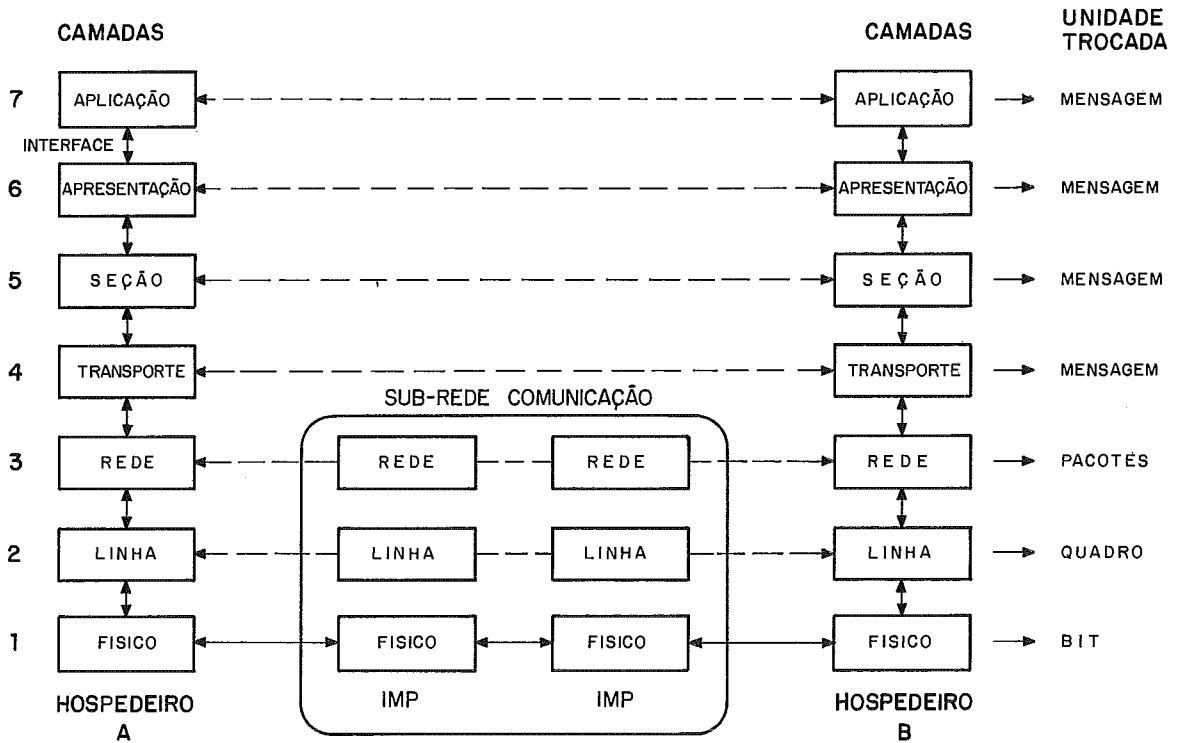


Figura 3

Estes computadores hospedeiros podem aceitar processamento de serviços que são submetidos por usuários denominados Locais e por usuários denominados Remotos. Os serviços remotos, ou sejam, aqueles serviços submetidos por usuários situados em estações remotas, são recebidos como mensagens da Sub-rede de Comunicações. Assim sendo este tipo de serviço requer um tempo adicional de processamento necessário pela utilização dos procedimentos que constituem os protocolos.

## 2.4.1 - Componentes de uma Rede de Computadores

### 2.4.1.1 - Processadores "Host"

De um modo geral os processadores ditos "Host" ou hospedeiros caracterizam-se por possuírem uma memória interna, serem conectados a uma série de equipamentos capazes de armazenar grande quantidade de informações e disporem de vários requisitos de Hardware, especialmente voltados ao atendimento de funções de comunicação e facilidades de controle de transmissão.

### 2.4.1.2 - Processadores "Front-End"

Cada computador da rede, esteja ele atuando como um processador de comunicação ou como terminal, conecta-se à rede por meio de alguma forma de TCU (Transmission Control Unit), cuja função é prover o controle de comunicação dos dados do processador hospedeiro para um sistema externo local.

Quando estas TCUs são dotadas de capacidade adicional de processamento, recebem a denominação de processador Front-End. Desse modo muitas das funções que nas TCUs são executadas por lógica de Hardware, passam a ser executadas por funções de Software, o que propicia maior flexibilidade para interfaciar uma gama maior de transmissões e linhas.

### 2.4.1.3 - Concentradores e Multiplexadores

O alto custo das linhas de comunicações é atualmente um dos maiores problemas na implantação de uma rede de comunicação de dados. Se cada terminal for ligado a um computador central atra

vês de um elo de comunicação independente, a atividade média em cada um desses elos será excessivamente baixa. O modo como os terminais são usados pode variar bastante e algumas linhas podem ficar inativas durante longos períodos, com nenhum ou pouquíssimo fluxo de informação entre o terminal e o computador. Se os períodos ativos das várias linhas nunca coincidem, é possível comutar uma única linha para atender a vários terminais.

Existem duas formas de se fazer a comutação. A primeira delas é quando a capacidade da linha de saída excede a soma das capacidades de todas as linhas de entrada. Nesta modalidade o comutador executa as funções de multiplexador. Esta divisão pode ser feita de duas formas a multiplexação por tempo e a multiplexação por frequência.

A segunda modalidade de comutar a linha envolve o armazenamento das mensagens recebidas dos terminais para posterior envio ao computador central. Esta modalidade denomina-se Concentrador, que é um dispositivo com "buffers" de armazenamento que altera a velocidade de transmissão de uma mensagem. Os concentradores geralmente são dotados de capacidade de processamento local e sua velocidade é suficientemente rápida para que possam aceitar mensagens simultaneamente de vários terminais de baixa velocidade, ou que possuam um fator de demanda também reduzido.

#### 2.4.1.4 - Sistemas de Computação Remota

O termo computação remota refere-se a duas classes de sistemas de processamento localizados remotamente. Um é o tradicional RJE (Remote Job Entry). O outro é o sistema de computação remoto que pode operar tanto no modo "standalone" quanto partici-

pando da rede.

Os RJE não são programáveis e são limitados a uma leitora de cartões como forma de entrada e uma impressora como forma de saída.

Um sistema de computação remota, diferentemente do RJE, é programável e capaz de processar uma série de aplicações que de outro modo teriam que ser feitas pelo "host". Assim ele libera não apenas a carga nas linhas de comunicação, mas também aquela parte do processamento que pode ser feita remotamente, geralmente mais próxima ao ponto de origem.

#### 2.4.1.5 - Terminais

O equipamento terminal é um dos aspectos mais críticos de um sistema de comunicação de dados. Isso se deve ao grande número e variedade dos diferentes tipos básicos dos terminais disponíveis no mercado. Basicamente, um terminal é qualquer estação final em uma rede de comunicação. Sua função principal é de ser fonte ou destino de dados, isto é, um ponto de entrada ou saída para a informação que cruza a rede. Segundo o trabalho de WAIWW-RIGHT <sup>83</sup> existem diferentes classes de terminais, a saber:

##### - Terminais não Controlados

Este é o grupo de terminais mais simples; não possuem "buffers" de armazenamento nem são endereçáveis. A medida que uma tecla é pressionada, os correspondentes "bits" daquele caracter são jogados paralelamente ou serialmente na linha. Só podem ser conectados através de ligação ponto a ponto.

- Terminais com buffer e endereçáveis

Esses tipos de terminais propiciam algumas facilidades de edição e, graças ao buffer de armazenamento, é possível implementar formas mais sofisticadas de detecção de erros. Podem ser utilizados em linhas multiponto uma vez que são endereçáveis.

- Terminais semi-inteligentes

São os terminais geralmente utilizados em sistemas de entrada remota de job e, além das características da classe anterior, possuem alguma capacidade de reconhecimento de mensagens e validação de dados.

- Terminais inteligentes

Esta classe de terminais pode ter conectadas a si diversos vídeos, unidades de armazenamento de massa, leitora de cartões, impressora, unidades de fita cassete, etc. Possui também capacidade de processamento "standalone".

## 2.5 - CONCEITO DE SISTEMAS DISTRIBUÍDOS

A palavra Sistema é amplamente utilizada, no vocabulário contemporâneo, para descrever muitas coisas em uma variedade de campos da atividade humana. Assim nós encontramos os chamados Sistemas de Transporte, Sistemas Educacionais, Sistemas Jurídicos, Sistemas de Computadores, etc.

Todos os diversos tipos de Sistemas, cada um deles executando funções bem distintas, possuem na realidade um conjunto de características comuns, definidas por MATTHEWS <sup>65</sup>, e que podem ser sintetizadas nos seguintes itens:



- Um sistema é composto de uma série de componentes interrelacionados;
- As atividades de um sistema são coordenadas segundo um conjunto de regras pré-estabelecidas;
- O conceito de sistema envolve um ambiente externo e um ambiente interno;
- Um sistema envolve fluxo de informações;
- Um sistema tem um objetivo.

Dentro deste enfoque de sistema descrito acima podemos conceituar um Sistema de Processamento de Dados Distribuído como um conjunto de funções de processamento, implementadas através de um determinado número de dispositivos interrelacionados, tal que cada um deles é responsável pela execução de parte das necessidades totais de processamento.

A essência desta distribuição de recursos está na sua capacidade de permitir que cada local execute independentemente uma significativa parcela de trabalho e que cada local, de alguma forma, confie até certo limite no processamento executado em outro local.

O ambiente adequado para que se desenvolva o tipo de sistema definido acima requer um conjunto de dispositivos, interagindo através de um subsistema de comunicação, cujas funções envolvem processamento e distribuição de informações. Essas funções são executadas a partir de pontos fisicamente remotos entre si, onde o conceito de remoto, neste contexto, pode implicar em distâncias que variam desde alguns metros a milhares de quilômetros. Cada elemento deste sistema deve ser selecionado com base em sua

habilidade para executar a tarefa requerida em determinado lugar.

## 2.6 - OBJETIVOS DE UM SISTEMA DE PROCESSAMENTO DISTRIBUÍDO

Em geral as empresas apresentam seus objetivos de Sistema de Processamento Distribuído, como o de levar a informação certa para o lugar certo, no tempo certo com o menor custo.

Entretanto esta descrição não oferece uma orientação real. Na verdade, a tentativa de produzir um sistema que simultaneamente ofereça a maximização dos serviços e a minimização dos custos, não é uma tarefa fácil e continua sendo, um grande desafio para os projetistas de sistemas.

A minimização dos custos em Sistemas de Processamento Distribuídos implica em uma comunicação de baixo custo, uma estrutura de armazenamento que não envolva a replicação de arquivos, poucos equipamentos e uma reduzida taxa de transmissão. Por sua vez a maximização dos serviços esta mais comprometida com os fatores que conduzirão o sistema, na execução de suas tarefas, a alcançar o mais alto índice de rapidez e eficiência.

Os autores que escrevem sobre Sistemas de Processamento Distribuídos, principalmente no que se refere a formulação matemática do modelo, concentram basicamente a sua atenção na minimização dos custos, sejam de armazenamento, minimizando a replicação de arquivos, sejam de transmissão, com a maximização da replicação. Contudo o processamento distribuído, como parte integrante do planejamento da estratégia da empresa, passa a ter uma relevante importância. Assim sendo aumentando-se um pouco os custos, o nível de serviços prestados pode elevar-se. A minimização dos

custos nem sempre é a melhor resposta, um Sistema de Processamento Distribuído deve ser projetado não para a máxima economia no presente mas sim para a máxima flexibilidade no futuro.

O objetivo deve portanto ser definido mais cuidadosamente, introduzindo-se o conceito de Sistema eficiente onde a eficiência de um Sistema é uma questão da relação entre as suas Entradas e Saídas. Esclarecendo-se quais são as saídas e entradas em um sistema podemos aproximar-mos da definição de um objetivo claro para esse Sistema.

Podemos definir as entradas como sendo o conjunto de objetos fornecidos ao Sistema. Estes recursos são submetidos a um determinado processamento alcançando-se no final, o resultado esperado (saída). Esta caracterização de Sistemas através de um enfoque entrada-saída é bastante utilizada na abordagem de Sistemas.

Uma grande parte dos Sistemas desenvolvidos dentro de uma abordagem Sistêmica, envolve um processo adicional de realimentação. Neste processo as saídas são avaliadas, comparando-se com determinados critérios, cujos resultados servirão de base para se introduzir modificações na entrada do Sistema. A figura 4 a seguir ilustra este tipo de enfoque, aplicado a um Sistema de Processamento de Dados Distribuídos.

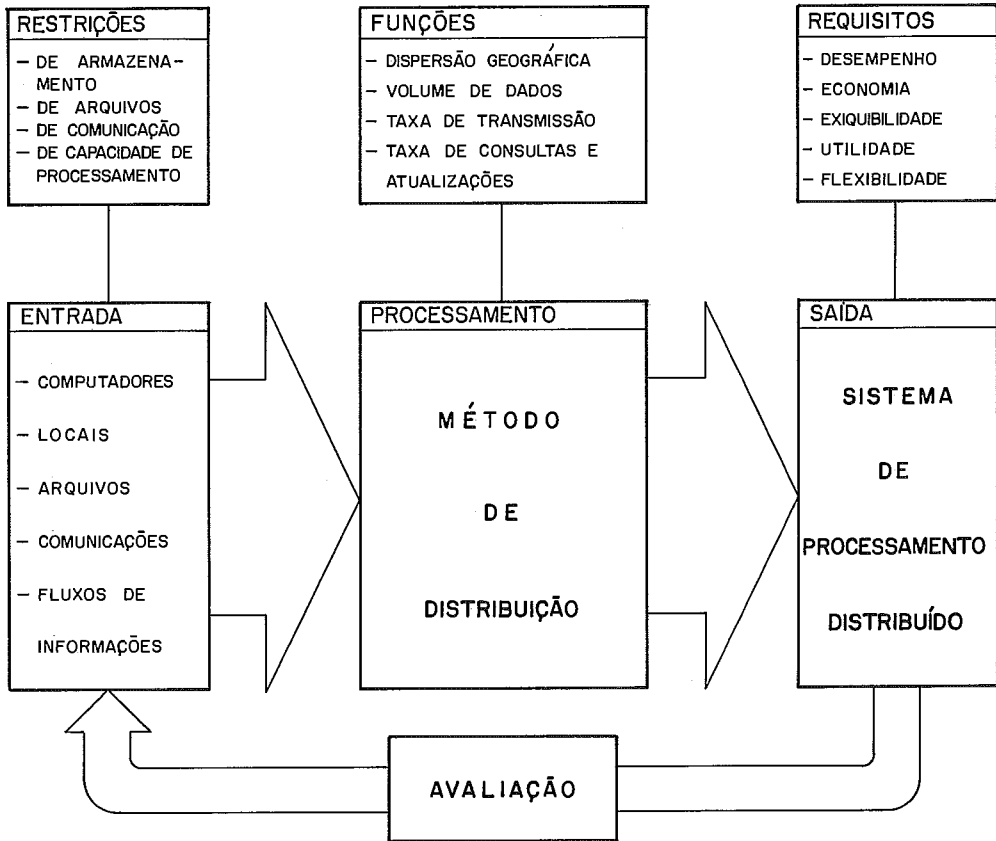


Figura 4

Podemos agora estabelecer uma definição para o objetivo de um Sistema de Processamento Distribuído o qual demonstra em termos gerais uma formulação das metas a serem atingidas.

O objetivo de um Sistema de Processamento Distribuído dentro de uma empresa é oferecer o resultado desejado pelo Sistema de Informações de forma econômica em termos de custos de armazenamento, custos de aplicações e custos de comunicação, na velocidade requerida.

Esses objetivos podem ser atingidos através de um conjunto de decisões, sobre arquivos, seu conteúdo, utilização, localização e armazenamento, sobre comunicação e sua melhor utilização na

interconecção dos computadores, de forma a atingir eficientes índices de:

- Desempenho, como sendo a habilidade de processar transações a uma dada taxa, dentro de um determinado tempo. (throughput-tempo de resposta);
- Economia, como sendo a capacidade para produzir um adequado desempenho a um aceitável custo;
- Exequibilidade, como sendo a capacidade de produzir um sistema possuindo os requisitos de desempenho, dentro de um aceitável custo, com os recursos disponíveis;
- Utilidade, como sendo a capacidade do sistema para produzir o desempenho desejado durante uma aceitável proporção de sua vida operacional;
- Flexibilidade, como sendo a capacidade do sistema para conviver com a mudança no fluxo das transações e com mudança ou inclusão de novas aplicações ou ainda a utilização de novas tecnologias.

Podemos também estabelecer, de acordo com a definição e os objetivos de um Sistema de Processamento Distribuído que sua arquitetura será função:

- da dispersão geográfica da empresa;
- das características das aplicações;
- do volume de dados a ser processado;
- do grau de complexidade das operações;
- da taxa de transmissão;
- da taxa de atualização e consulta aos arquivos.

## 2.7 - RAZÕES QUE CONDUZIRAM AO PROCESSAMENTO DISTRIBUÍDO

Muitas organizações mantêm até o presente momento uma política centralizada no emprego dos seus recursos de computação. Outras, com o passar dos tempos, introduziram novas técnicas em seus modelos de processamento de dados.

O primeiro tipo de comportamento se deve ao fato de que os projetistas de sistemas acreditam que poucos e grandes computadores são capazes de realizar o trabalho de diversos e pequenos computadores por um custo menor. Os que defendem este ponto de vista oferecem também diversas razões para procurar inibir o processo de distribuição. Entre elas incluem-se os da conversão dos sistemas centralizados existentes, a tendência de que os pequenos sistemas venham a se tornar centros de computação, o impedimento de implementar Banco de Dados em pequenos computadores e a dificuldade de uma definição clara dos limites de aplicação em cada local. Além disso acreditam não ser uma tarefa fácil a coordenação das aplicações sobre diversos locais de forma a atender as necessidades de processamento de dados da organização.

A outra forma de se conduzir é consequência de que recentes evidências indicam que a centralização não é necessariamente o melhor caminho. O desenvolvimento das telecomunicações associado ao baixo custo dos componentes de hardware, generalizando a fabricação dos mini e microcomputadores, tem sido primordial à distribuição da capacidade de processamento. A principal vantagem de utilização dos mini e microcomputadores em um Sistema Distribuído é oferecer ao usuário uma adequada potência de computação para as aplicações em questão. Além disso os mini e microcomputadores hoje passaram a não mais se enquadrar na chamada lei de Grosch.

Em 1953 o Dr. Herbert Grosch publicou sob a denominação de "lei de Grosch" um princípio sobre economia de escala em computadores. Este princípio estabelece que o poder de processamento de dados de um computador é proporcional ao quadrado do seu preço. Ou seja, se um computador X custa duas vezes mais do que o computador Y, podemos esperar que X seja quatro vezes mais poderoso que Y. Este princípio descrito na lei de Grosch foi testado por Kenneth Knight em 1966 e posteriormente por Harvey Golub e Paul Stoneman em 1971, que comprovaram a sua validade geral.

Recentemente o advento da tecnologia LSI (Large Scale Integration) na produção de microprocessadores veio a colocar em dúvida a aplicabilidade da lei de Grosch de forma global. Esta tecnologia tem a particularidade que o custo de produção independe da complexidade do circuito LSI em questão. Assim componentes que sejam mais complexos custam mais caros que outros mais simples, unicamente por que no seu preço tem que ser amortizado um montante maior referente a pesquisa e desenvolvimento. No caso em que o mercado para um determinado componente LSI seja muito grande, a parcela do preço responsável pela amortização do desenvolvimento torna-se irrisória, e o preço global do componente bastante reduzido.

Tal é o caso para os microprocessadores que hoje integram vários mini e microcomputadores. No estágio atual da tecnologia LSI, somente os mini e microcomputadores podem se utilizar de um microprocessador como UCP. Vale mencionar que existem configurações de UCP e memória de microprocessadores hoje por US\$400.00 equivalente a do IBM 1130 que valia US\$20,000.00 em 1965.

Assim a lei de Grosch continua a ser válida para os computadores médios e de grande porte, mas sofre uma reversão na categoria dos mini e microcomputadores. Deste modo nesta categoria é mais vantagem descentralizar processamento de dados em vários computadores pequenos.



Entretanto as razões que conduzem a uma política de des-centralização ou distribuição dos recursos e atividades de proces-samento de dados, devem ser analisadas por outros ângulos que não seja exclusivamente o do baixo custo dos elementos eletrônicos que o constituem. Entre estas razões podemos enumerar:

- Segundo a estrutura da organização;
- Os custos de comunicações;
- Capacidade e limitações envolvidas no sistema;
- Relação preço desempenho;
- Tempo de resposta;
- Crescimento modular.

#### 2.7.1 - Segundo a estrutura da Organização

Conceitualmente nos sistemas de organização centralizada, um dos componentes, que é o subsistema principal, representa o pa-pel dominante e que pode deixar em segundo plano os outros compo-nentes. Nestas condições o subsistema principal é o centro das o-perações, tudo é função dele, ficando os subsistemas secundários como satélites em torno das operações centrais. Estas caracterís-ticas fazem, com o passar dos anos, que o tamanho do subsistema principal da organização cresça significativamente, passando a ser o coração da empresa. No caso particular de uma organização que utiliza o processamento de dados de forma centralizada é comum a-contecer que todos os setores da empresa se tornem dependentes do setor responsável pelo computador, como está representado na figu-ra 6 a seguir.

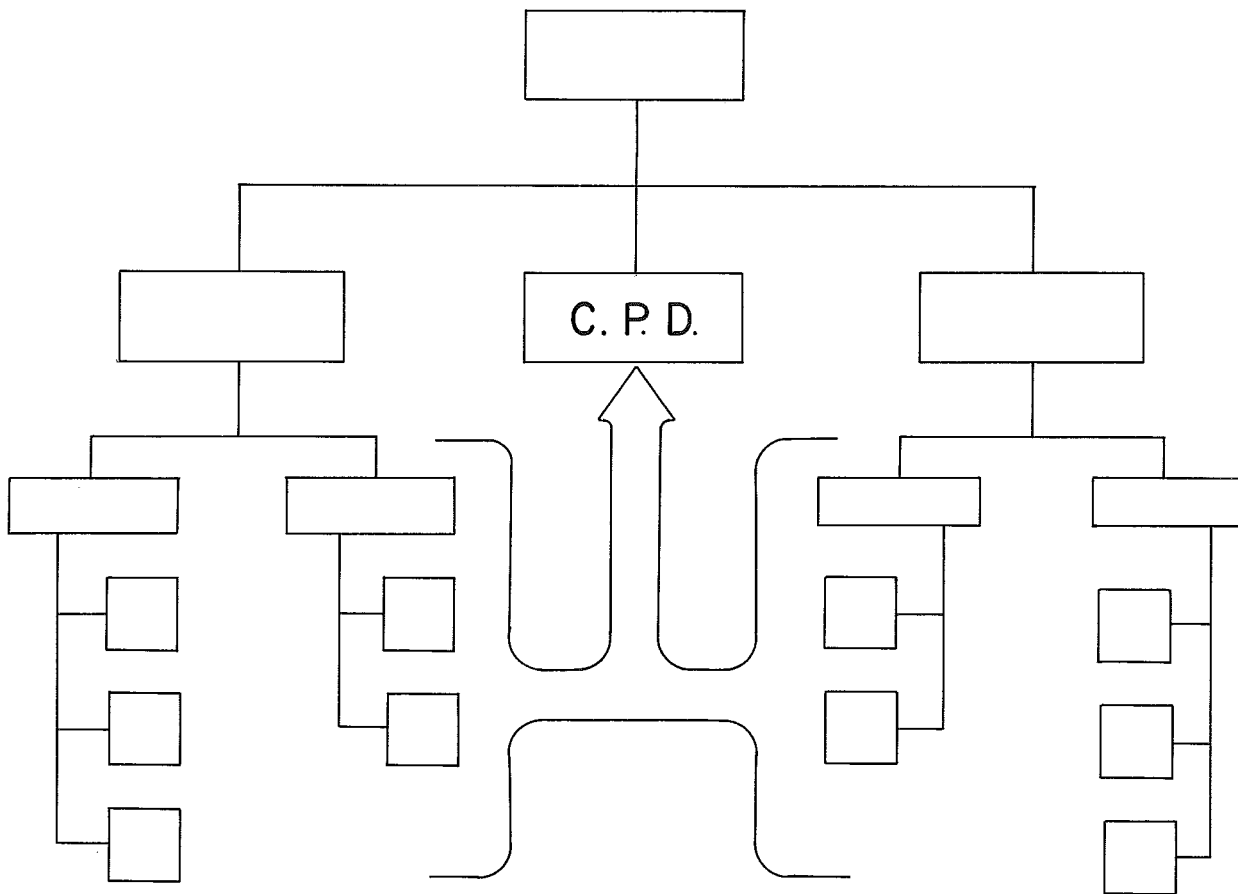


Figura 6

Em muitos casos, onde esse crescimento ocorre, principalmente pela automatização de todos os negócios da empresa, caracteriza-se um gigantismo no órgão responsável pelo processamento de dados e uma consequente perda de controle. Desta forma a conduta da alta administração, tem sido no sentido de dar maior representatividade ao setor de processamento de dados, a fim de garantir o controle sobre seus próprios recursos de computação. Mais e mais são gerados custos de processamento de dados, os problemas da organização

nização parecem resultar sempre de fatores relacionados ao computa\_dor.

Estes mesmos fatos não ocorreria nos Sistemas Descentra\_lizados. Pois mesmo que haja um subsistema principal, este terá praticamente o mesmo valor dos demais em relação às finalidades da empresa.

Entretanto a estrutura da organização é dinâmica, está sempre em evolução, saindo de uma idéia de estrutura para uma de processo. Em vez de visualizar a organização em uma estrutura hi\_e\_rárquica tradicional, com determinadas relações de autoridade, co-meça-se a enxergar a organização como um conjunto de fluxos envol-vendo todos os processos da empresa, relacionando harmonicamente, informações, pessoas e materiais. A figura 7 abaixo mostra uma or-ganização industrial simplificada vista por este prisma.

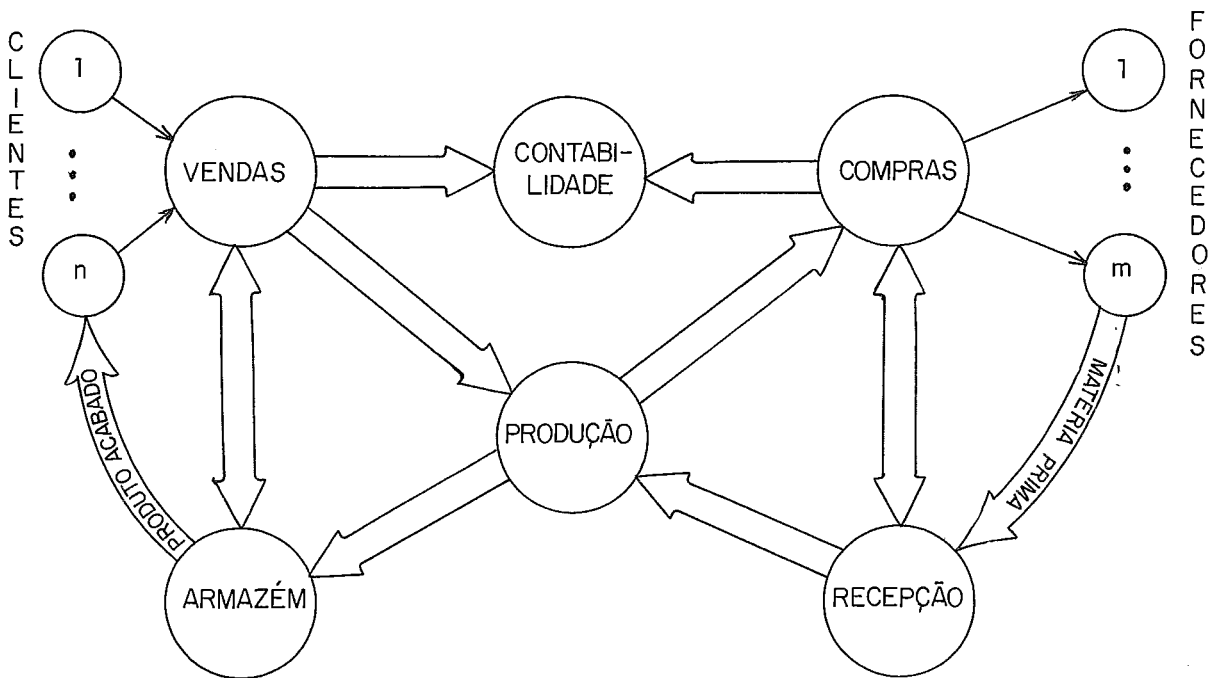


Figura 7

Nestas circunstâncias, quando a organização é concebida a partir de um enfoque sistêmico é possível utilizar com mais vantagens as técnicas de processamento distribuído, onde se torna necessário a divisão de controle e de recursos de computação entre as diversas cadeias de comando da empresa. Este procedimento permitirá um certo grau de independência aos diversos setores envolvidos na distribuição, devido a capacidade de processamento local oferecida pelos mini e micros, enquanto ao mesmo tempo prover uma essencial comunicação entre eles.

### 2.7.2 - Custos de Comunicações

Através dos anos o custo de comunicação e o custo de computação por transação vem diminuindo consideravelmente. Entretanto nos dias atuais a taxa de declínio já não possui os mesmos índices do passado. O gráfico da figura 8 abaixo mostra o comportamento do mercado internacional em relação a estes custos.

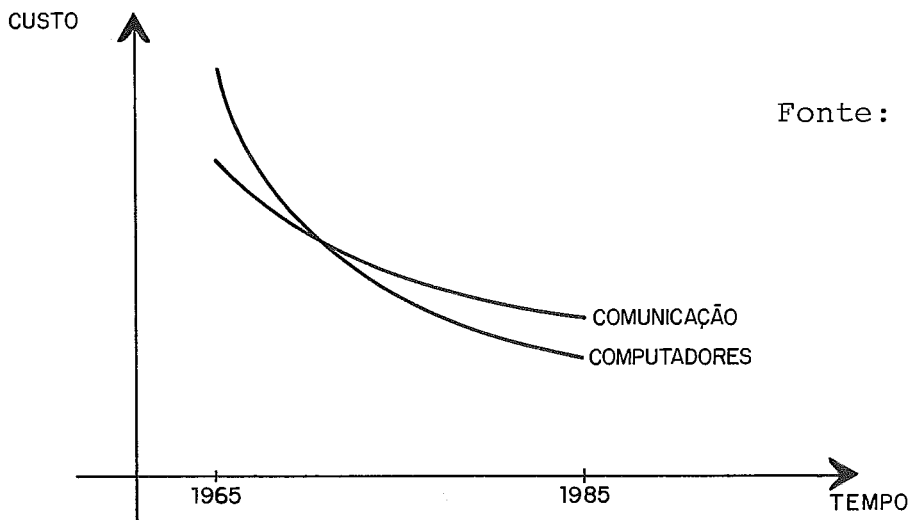


Figura 8

Estas indicações conduzem a uma tendência geral de colocar os recursos de computação o mais próximo possível do usuário final da informação, oferecendo a possibilidade de eliminar ou reduzir os custos de comunicação.

Uma outra forma de reduzir os custos de comunicação é incluir o uso de computadores remotos para concentrar a comunicação de um certo número de terminais remotos, de baixa velocidade, em uma simples linha de alta velocidade. E a utilização da capacidade de processamento local para executar partes de aplicações que não requisitem interação com uma máquina central ou qualquer sistema remoto.

### 2.7.3 - Capacidade e Limitações Envolvidas no Sistema

Uma das fortes razões que conduzem ao Processamento de Dados Distribuídos ou pelo menos a descentralização do processamento é o fato de que um simples sistema de computador com a adequada capacidade não existe. Embora a performance dos sistemas do tipo "High-End" tenham crescido bastante nas últimas décadas, este crescimento ainda é inadequado ao grande número de alternativas de emprego do computador. Muitas empresas que utilizam a computação como parte integrante dos seus negócios acham também que a capacidade de uma simples máquina tem sido sempre sobrepujada por um aumento nos volumes dos seus negócios. Assim duplicando-se este volume, requereria proporcionalmente um aumento na capacidade do computador.

Um outro problema que existe é aquele que está diretamente ligado a necessidade de instalar novos "software" ao sistema

operacional existente, para suportar requisitos técnicos exigidos pela implementação de novas ou complexas aplicações. Ou ainda velhas aplicações que necessitam de ser convertidas pelas exigências de novos "software".

Em qualquer dos casos especificados, a distribuição da carga de trabalho entre dois ou mais nodos, contribuirá para que este tipo de problemas, provocados por fatores ligados a limitação e capacidade, sejam reduzidos permitindo uma melhor adequação do crescimento do sistema.

#### 2.7.4 - Relação Preço-desempenho

Um fator de grande importância e que conduz as empresas em direção a distribuição das aplicações entre diversos computadores é a crescente possibilidade do uso específico dos mini e micro computadores. Tais máquinas podem ter uma vantagem na razão preço desempenho em relação aos grandes computadores de uso geral. SCHERR <sup>75</sup> mostra através da figura 9 a seguir este tipo de relação para uma variedade de arquiteturas e graus de complexidade das aplicações. A variação desta relação preço-desempenho entre um microcomputador e um computador de grande porte atinge uma ordem de grandeza de três vezes mais e entre os mini e microcomputadores esta ordem de grandeza é de uma ou duas vezes mais. Observamos também na figura a seguir que esta relação preço-desempenho se torna ótima quando a complexidade da aplicação se nivela a complexidade da arquitetura utilizada.

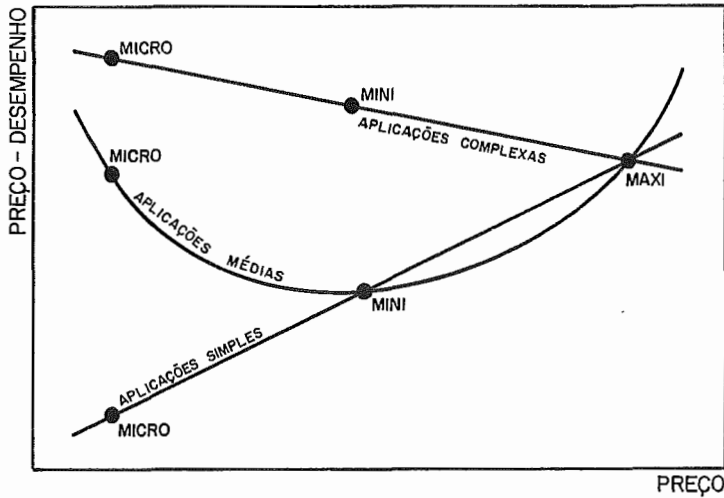


Figura 9

Desde que a relação preço-desempenho é dependente da natureza do programa que está sendo executado, a conclusão que se pode tirar é que o software do sistema operacional dos microcomputadores é correspondentemente simples, caracterizando o que se costuma chamar de eficiência através da especialização. Isto porque é inconcebível que uma boa relação preço-desempenho seja alcançada com um sistema operacional cuja complexidade exceda a capacidade de hardware.

#### 2.7.5 - Tempo de Resposta

No período entre 1960 e 1970, quando os grandes computadores dominavam o mercado de processamento de dados, impondo assim uma filosofia centralizadora, era comum ocorrer "engarrafamentos"

nas filas de entrada e saída do sistema, provocando situações em que o retorno das informações, necessárias para tomadas de decisões, acontecia com grande atraso. Assim a visão de processamento distribuído não é somente a de dar certa autonomia a determinados locais, oferecendo-lhes um computador, mais também acabar com esta facilidade de engarrafamento comum, em um ambiente centralizado, tendo como imediata consequência um substancial aumento na velocidade do tempo de resposta.

#### 2.7.6 - Confiança e eficácia

A confiabilidade e eficácia são dois aspectos de positiva importância em Sistemas de Processamento de Dados. No caso particular de um Sistema de Processamento de Dados Distribuídos, onde existe o particionamento da carga de trabalho entre os diversos locais do sistema bem como o provimento de redundância de dados, a extensão do impacto das falhas que possam ser provocadas por tais fatores devem ser consideravelmente reduzidas.

Entendemos portanto por confiabilidade em um Sistema de Processamento Distribuído como a acessibilidade de um arquivo no caso de falha de um nó ou no canal de comunicação. Esta confiabilidade pode ser atingida através de vias alternativas para se atingir o nodo ou ainda através de uma cópia do arquivo situada em outro nodo.

E por eficácia entendemos como sendo a capacidade de a rede continuar a executar a suas tarefas de comunicações entre os nodos da rede em caso de ocorrer tais falhas.



Em um sistema do tipo centralizado fica fácil observar que no caso de ocorrer uma falha no computador central do sistema todos os usuários estarão comprometidos.

### 2.7.7 - Crescimento Modular

Um Sistema de Processamento de Dados Distribuídos pode crescer mais facilmente do que um Sistema Centralizado. Se é necessário expandir o Sistema porque o volume de processamento aumentou torna-se mais fácil acrescentar novo nó à rede de computadores do que substituir um Sistema Centralizado por outro maior. Estas adições de novos nodos ao sistema podem ser feitas em pequenos incrementos, sem causar interrupções no funcionamento geral de todo o Sistema.

## 2.8 - CARACTERÍSTICAS INDISPENSÁVEIS DE UM SISTEMA DISTRIBUÍDO

Um Sistema de Processamento de Dados Distribuídos deve possuir diversas características fundamentais e necessárias ao atendimento dos diversos tipos de aplicações e modalidades de processamento, geralmente existentes e indispensáveis dentro de uma organização. Dentre estas características listamos algumas que THIERAUF<sup>82</sup> achou essencial dentro de um Sistema Distribuído.

- Tanto a descentralização quanto o caso particular da distribuição, requerem um bom planejamento em função das necessidades da organização;
- Suportar atividades de entrada de dados, para fins de coleta, armazenamento, validação e processamento de relatórios gerenciais;

- Prover uma rede de processamento que reúna as necessidades de usuários locais, regionais e até mesmo um ní-  
vel de utilização doméstica (home office);
- Dar ao usuário a opção de processamento em modalidade  
interativa e em "batch" ou a combinação das duas;
- Oferecer a possibilidade de conexão com terminais dos  
tipos não controlados, endereçáveis, semi-inteligentes  
e inteligentes;
- Prover uma eficiente alocação de arquivos de dados bem  
como oferecer uma boa estratégia de processamento de  
consultas e controle de concorrência;
- Implementar uma política de programação e métodos de  
operação simplificadas.

## 2.9 - TIPOS DE ARQUITETURAS

Todos os fatores estudados no item 2.7 e que conduzem a uma política distribuída são amplamente pesquisados por diversos autores entre os quais encontramos DAVENPORT<sup>29,30</sup>, SCHERR<sup>75</sup> e ZIEGLER<sup>89</sup> cujos estudos indicam uma série de vantagens e desvantagens em cada um dos tipos de sistemas.

### 2.9.1 - Arquiteturas Centralizadas

As arquiteturas de redes centralizadas se caracterizam por possuírem um sistema de computador de grande porte, um conjunto de usuários que interagem com o computador através de terminais locais e um sistema de comunicação que faz a interligação destes componentes. A figura 10 a seguir dá uma visão simplificada deste tipo de sistema.

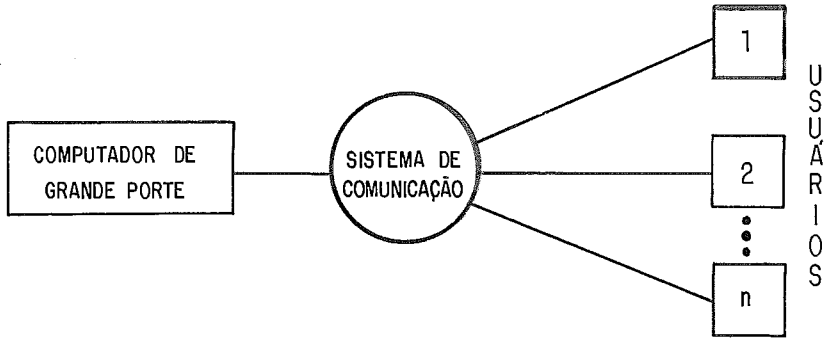


Figura 10

2.9.2 - Arquiteturas Distribuídas

Nas arquiteturas de redes distribuídas a caracterização ocorre pela existência de dois ou mais sistemas de computação, podendo ser de qualquer tipo ou porte, onde os usuários podem interagir com apenas um dos computadores, através das facilidades de telecomunicação local, ou podem estar conectados diretamente no sistema de comunicação da rede e interagir com qualquer um dos outros computadores. A figura 11 abaixo dá uma visão simplificada deste tipo de sistema.

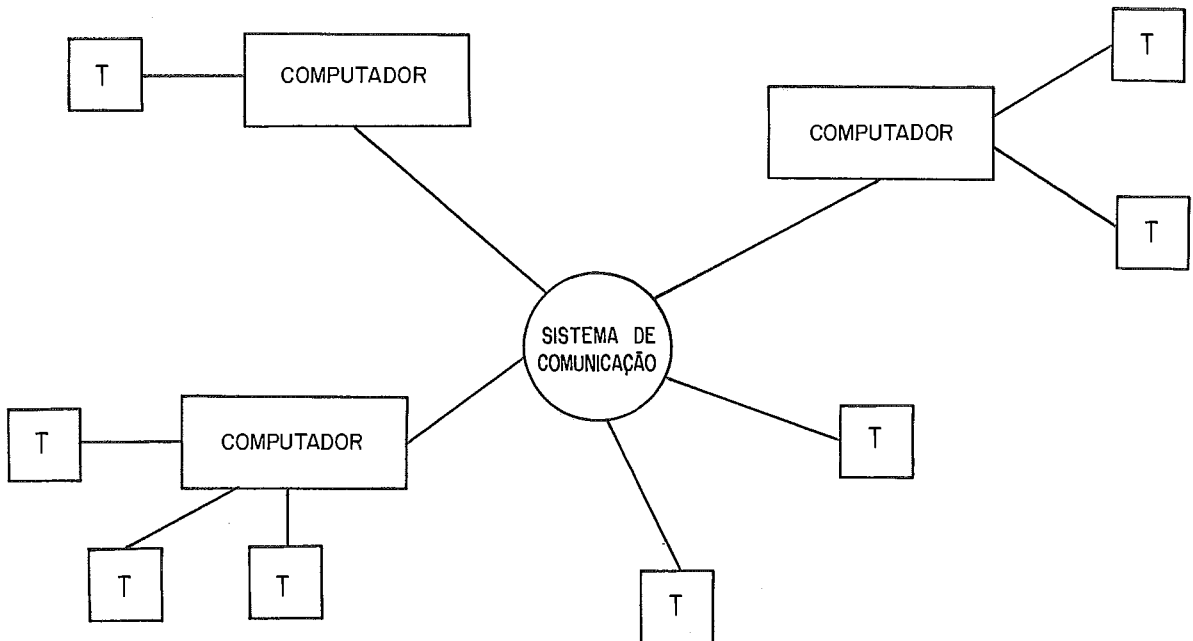


Figura 11

Dentre este tipo de arquitetura existem duas formas de distribuição dos recursos de computação. A distribuição horizontal e a distribuição vertical.

### 2.9.2.1 - Distribuição Vertical

Na distribuição vertical a interconexão dos computadores estabelece uma hierarquia, compartilhando tarefas em uma estrutura de caminho onde os componentes de um nível são controlados por membros do sistema situado em um nível maior da hierarquia. Isto possibilita um crescimento com reduzidos custos de comunicação, entretanto fornece poucas alternativas de emprego, particularmente por membros localizados em níveis mais baixo. Este tipo de Distribuição se aplica e é recomendado para casos em que uma mesma aplicação atua em todos os nodos da rede, como por exemplo no caso de aplicações bancárias. Esses tipos de sistemas exigem um controle centralizado, o que corresponde a arquitetura hierárquica de um Sistema de Informação. Teoricamente não existe limite na quantidade de níveis na hierarquia, porém atualmente em um modelo típico, como apresentado na figura 12 abaixo, o número de níveis situa-se na ordem de três.

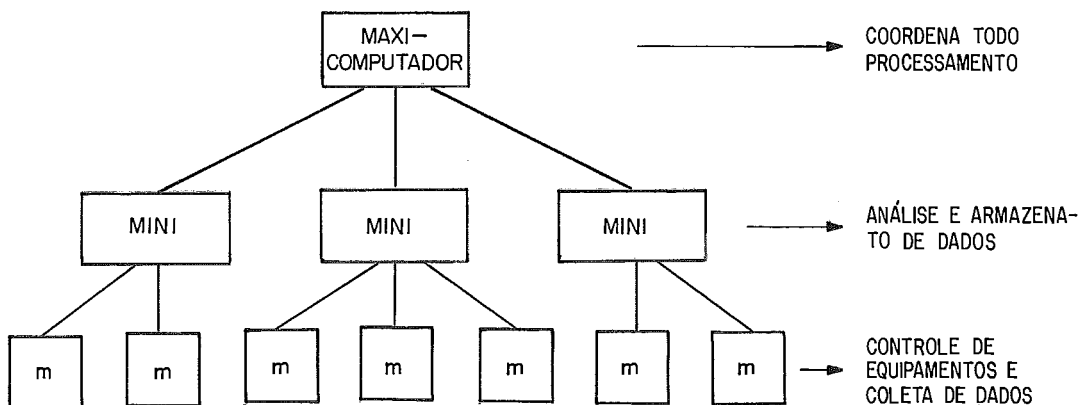


Figura 12

### 2.9.2.2 - Distribuição Horizontal

Na distribuição horizontal a interconexão dos dispositivos estabelece um nível lógico de relacionamento em igualdade de condições, para executar um conjunto de tarefas. Este tipo de distribuição oferece uma grande capacidade de múltiplos empregos e um desempenho maior em relação à distribuição vertical. Esta arquitetura implica na distribuição dos tempos de controle do Sistema. Com isto uma sobrecarga maior é introduzido no Sistema. Por outro lado esta sobrecarga é superado pela flexibilidade introduzida no Sistema em um ambiente de aplicações heterogêneas. A figura 13 abaixo ilustra este tipo de arquitetura.

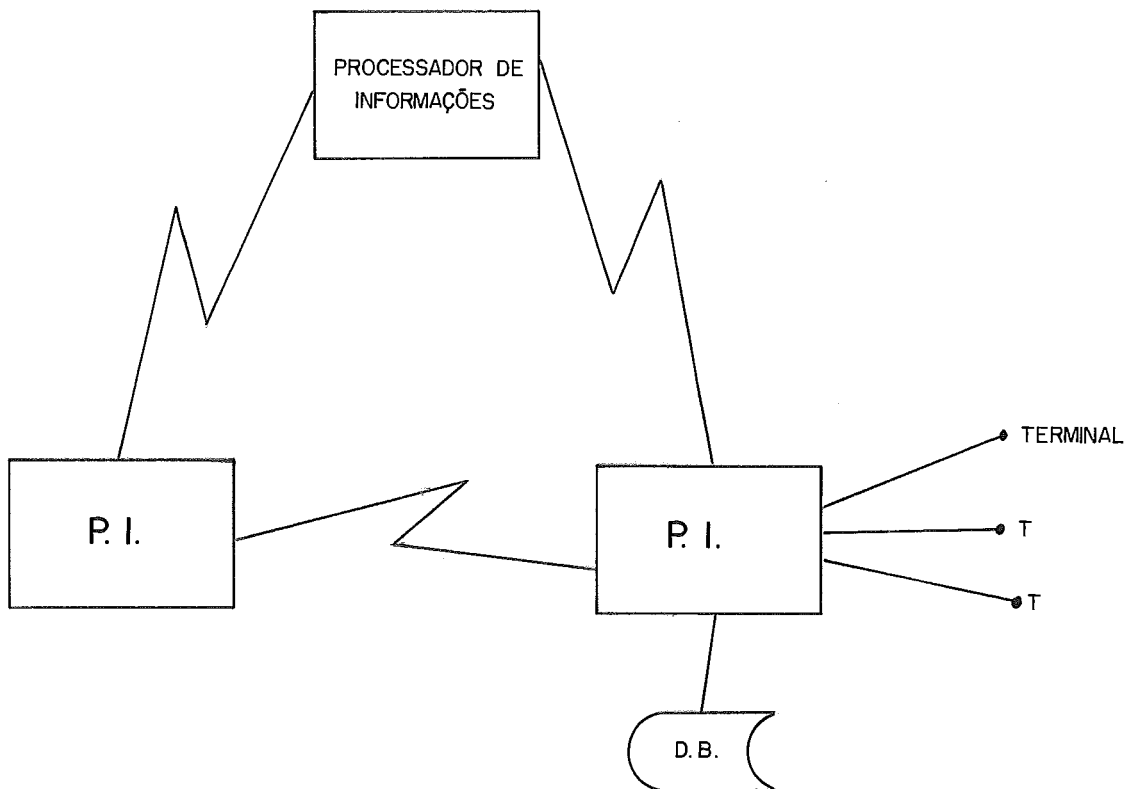


Figura 13

### 2.9.3 - Vantagens e Desvantagens

Dentre os diversos estudos efetuados por ENSLOW<sup>36</sup>, BOOTH<sup>8</sup>, ZIEGLER<sup>89</sup>, SHERR<sup>75</sup>, DAVENPORT<sup>29,30</sup> pesquisadores sobre este assunto, selecionamos uma série de vantagens e desvantagens de cada tipo de sistema.

#### 2.9.3.1 - Vantagens do Sistema Centralizado

- Economia de escala em termos de hardware e software;
- Relativa facilidade em manter a consistência dos dados;
- Um melhor controle e padronização das operações pela gerência do centro de processamento;
- Capacidade para justificar economicamente um mais elegante "system software";
- Reduz a duplicação de esforços em termos de custo de computador, custo de pessoal para operação e suporte do Sistema;
- Aperfeiçoamento dos sistemas de desenvolvimento e programação.

#### 2.9.3.2 - Desvantagens do Sistema Centralizado

- Menos flexibilidade em relação às necessidades do usuário;
- Custo total do sistema com comunicação e transmissão muito alto;
- Possui uma alta sobrecarga devido o seu uso geral;
- Vulnerabilidade as falhas;
- Incapacidade para responder rapidamente à troca de tecnologia (conversão de software, treinamento de pessoal);

- Alto custo com operações de "Backup";
- Baixa integridade do sistema como um todo;
- Aumento substancial nos custos pela necessidade de uma expansão do sistema;
- Limitado e lento acesso aos arquivos centrais.

#### 2.9.3.3 - Vantagens do Sistema Distribuído

- Maior capacidade "standalone" para um processador;
- Capacidade para oferecer múltiplos empregos;
- Capacidade para responder rapidamente à troca de tecnologia;
- Alto desempenho do sistema (devido à especialização);
- Implementação modular;
- Fácil e rápido acesso para arquivos locais;
- Baixo custo em operações de Backup;
- Pequeno incremento nos custos pela necessidade de expansão do Sistema;
- Baixa sobrecarga;
- Alta integridade pela dispersão dos sistemas de computadores;
- Baixo custo de comunicação do sistema como um todo.

#### 2.9.3.4 - Desvantagens do Sistema Distribuído

- Alto custo devido à duplicação em termos de hardware, software, dados, espaço e pessoal;
- Duplicidade de procedimentos de operação e processamento;

- Maior dificuldade de controle sobre o desenvolvimento, programação, padronização e Banco de Dados;
- Crescimento limitado em termos de CPU e capacidade de armazenamento;
- Limitação do tamanho e complexidade das aplicações.



### CAPÍTULO III

#### ALOCAÇÃO DE ARQUIVOS EM UM SISTEMA DISTRIBUÍDO

##### 3.1 - CONSIDERAÇÕES GERAIS SOBRE ALOCAÇÃO DE ARQUIVOS

O problema de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos consiste, de uma forma geral, em determinar os locais de uma rede de computadores onde deverá ser designado um conjunto de arquivos, de maneira que uma função objetivo que tenha sido estabelecida, seja minimizada sem prejudicar contudo o desempenho do sistema.

Podemos caracterizar diversos enfoques para solucionar este tipo de problema. Entretanto duas formas bem distintas de alocação são facilmente percebidas no tratamento dado pelos diversos pesquisadores, que estudam sobre o assunto, entre os quais relacionamos os trabalhos de AKOKA <sup>1</sup>, CASEY <sup>12</sup>, CARRARESI <sup>14</sup>, CERI <sup>15</sup>, CHEN <sup>22</sup>, CHU <sup>24</sup>, MAHMOUD <sup>62</sup>, MORGAN <sup>67</sup>, GRAPA <sup>46</sup>.

A primeira delas seria aquela em que cada um dos arquivos deve ser designado apenas para um único local do sistema, isto é, não haverá duplicação de arquivos nos nodos da rede. Esta forma de alocação foi denominada por diversos autores, tais como GHOSH <sup>45</sup>, CERI <sup>15</sup>, de Alocação sem Redundância. Neste tipo de alocação o problema de direcionamento das ações de acesso ao arquivo, seja para efetuar uma recuperação de informação ou para atualizar um dado, resume-se em enviar o pedido para o local onde o arquivo está armazenado.

A segunda forma de alocação é aquela na qual a duplicação de arquivos no sistema é permitida, ou seja, um arquivo pode ser atribuído a um ou mais nodos da rede. De forma similar esta modalidade de designação de arquivos em uma rede de computadores denomina-se de Alocação com Redundância. Nestas condições de organização as ações de acesso ao arquivo possuem dois tipos de tratamentos. No primeiro, que se refere ao acesso para simples consulta de dados, o problema do direcionamento se passa de forma semelhante ao caso sem redundância. No segundo, quando o acesso ao arquivo implica em atualização de dados no arquivo, o problema de direcionamento do pedido se torna mais complexo devido principalmente à necessidade de se manter a consistência e a integridade dos dados do Sistema. Assim sendo, para cada ação de acesso ao arquivo, que implique em atualização de dados, deve ser considerado que os pedidos devem ser enviados para todos os locais onde existem armazenado uma cópia do arquivo que está sendo atualizado.

Neste contexto, podemos ainda caracterizar uma forma intermediária de alocação de arquivos. Seria aquela em que apenas determinados arquivos tem permissão para serem duplicados.

### 3.1.1 - FATORES RELACIONADOS COM A ALOCAÇÃO DE ARQUIVOS

Muitos fatores são considerados na fase de alocação de arquivos em um projeto de Sistema de Processamento de Dados Distribuídos. O objetivo, como já vimos, é conseguir um bom desempenho no tempo de resposta, economia nos custos de armazenamentos e nos custos de transmissão, flexibilidade às mudanças, etc. Os fatores principais que são considerados pelos diversos autores que pesquisam sobre alocação de arquivos são:

- Frequência de acesso, ou seja, a taxa de acesso a um arquivo por parte de um determinado usuário. Em geral os autores fazem uma distinção entre o acesso relativo às consultas e o acesso relativo às ações de atualização;
- Custos de transmissão, ou seja, o custo de transmitir ou receber mensagens de consulta ou manutenção de um determinado arquivo;
- Custos de armazenamento, ou seja, o custo de armazenar o arquivo em um nodo da rede. Alguns autores como por exemplo CHEN <sup>22</sup> e MORGAN <sup>67</sup> consideram os custos de armazenamento de arquivos de dados e programas separadamente. Isto porque em suas formulações eles consideram a existência de uma dependência entre programas e arquivos. Outro enfoque é considerar o custo de armazenamento somente quando os dados a serem armazenados exceder a capacidade de armazenamento do nodo;
- Volume de dados, ou seja a quantidade representativa de informações acessadas em determinado arquivo;
- Custos das linhas de comunicações, ou seja, o custo a ser pago pela linha de comunicação de diferentes capacidades (Bandwidths);
- Outros fatores, tais como: custos de software, custos de equipamento, capacidade de processamento, capacidade das linhas de comunicações, frequência de manuseio local de determinado arquivo, etc., são também considerados em alguns modelos de alocação.

Entretanto para simplificar a formulação dos modelos os autores em geral fazem considerações do tipo que relacionamos abaixo:

- A topologia da rede é fixa;
- A capacidade de armazenamento em cada nodo é considerada ilimitada;
- Não existe dependência entre arquivos e programas;
- As capacidades das linhas de comunicações não são consideradas;
- O retorno do fluxo de informação é negligenciado.

### 3.1.2 - Classificação dos Arquivos em um Sistema Distribuído

Um Sistema de Processamento de Dados Distribuídos pode ter os seus arquivos distribuídos através dos nodos da rede de três formas:

- Arquivos Individuais;
- Arquivos Replicados;
- Arquivos Particionados.

#### 3.1.2.1 - Arquivos Individuais

Estes arquivos não requerem esforços especiais para serem implementados. Cada usuário cria seus próprios arquivos e torna-se responsável pela sua manutenção no seu processador. Entretanto esses arquivos podem ser acessados, para efeito de consulta, por outros nodos do sistema. A figura 14 a seguir ilustra uma distribuição deste tipo de arquivo.

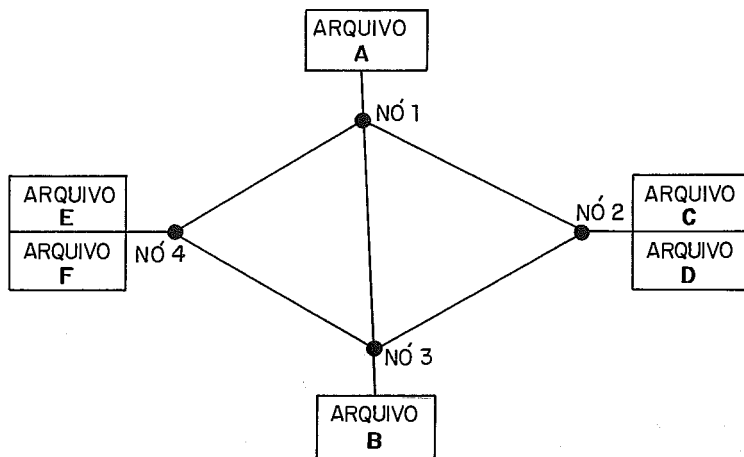


Figura 14

### 3.1.2.2 - Arquivos Replicados

No estudo para distribuir os arquivos nos nodos de uma rede, existem alguns fatores que conduzem à necessidade de duplicar determinados arquivos ou parte deles em vários nodos do Sistema. Estes fatores poderiam até levar ao que se denomina de Replicação Extrema, onde em cada nodo existiria cópia de todos os arquivos. Os fatores que podem conduzir a replicação dos arquivos são:

- Aumento da acessibilidade do Arquivo, isto é, aumentando o número de cópias do arquivo, a concorrência a um arquivo diminuirá. Com isto pode-se obter respostas mais rápidas devido a maior disponibilidade, tornando possível um aumento no número de usuários que acessam o arquivo;
- Diminuição no volume de comunicação e custos de transmissão;

- A taxa de acesso ao arquivo é muito grande de forma que não possa ser suportada em um único nodo do sistema;
- Facilidade de um rápido "backup" em caso de falhas em um nodo do sistema que contenha cópia do arquivo.

A figura 15 abaixo ilustra uma distribuição com redundância.

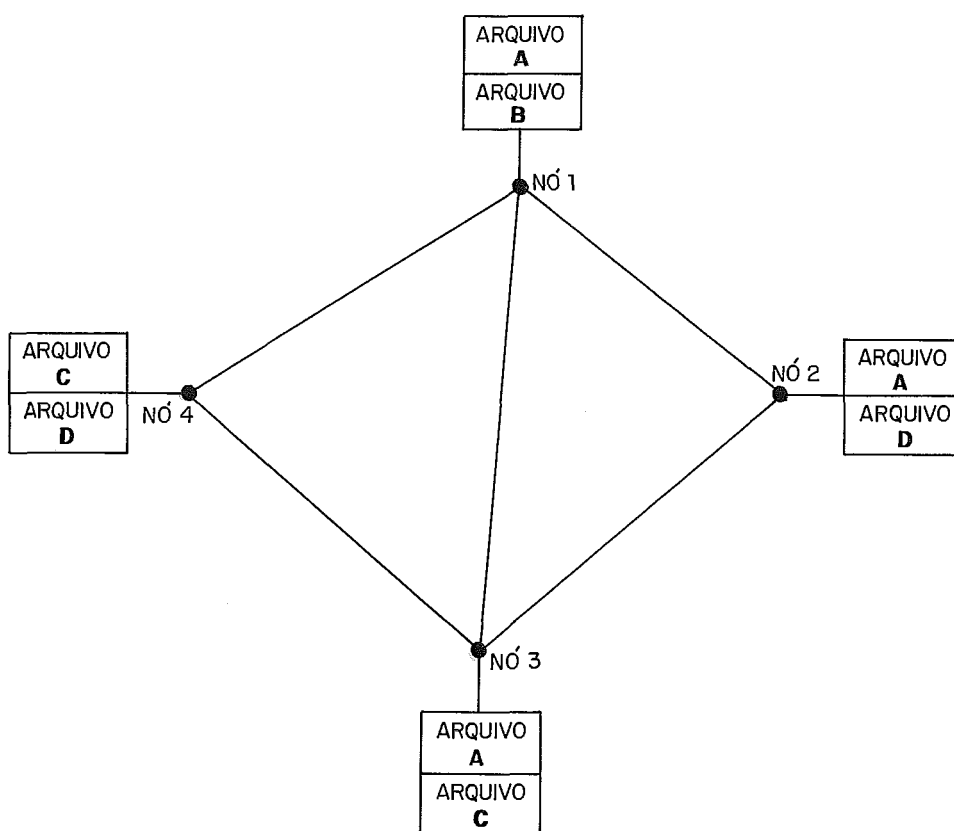


Figura 15

Entretanto a replicação de arquivos traz consigo o problema de redundância, que poderia provocar um impacto na integridade e desempenho do sistema. Ela torna-se aceitável se o índice de

atualizações dos arquivos for muito baixa.

Várias soluções têm sido propostas para resolver o problema da atualização das cópias dos arquivos. Podemos dividi-las em dois grupos principais. Aquele que propõe que a atualização das cópias sejam feitas de forma imediata e aquele que propõe que as atualizações sejam feitas a posteriori.

A solução imediata seria aquela em que qualquer atualização em um arquivo que tenha cópia, implica na imediata atualização de suas cópias. Esta solução torna-se viável quando apenas uma pequena proporção das transações requer atualizações de um dado em um terminal remoto. Assim sendo a maioria das transações necessária somente ler o dado do nodo remoto.

A solução a posteriori seria aquela em que um nodo qualquer seria eleito para conter um arquivo "mestre", com todas as cópias replicadas do sistema. Qualquer transação que envolvesse modificações do dado seria dirigida a esse arquivo mestre, e todos os outros nodos operariam na forma "read-only". A atualização definitiva dos arquivos que estiveram envolvidos em modificações seria realizada em um horário mais conveniente.

Todas as soluções propostas para resolver eficientemente o problema da atualização em arquivos replicados, encontram uma série de dificuldades que parecem torná-las sempre soluções incompletas. Estas dificuldades surgem por exemplo quando um nodo que contém cópias de arquivos, fica desativado por um certo período de tempo. Uma das formas propostas para solucionar este tipo de problema, seria a existência de um procedimento automático que enviaria a todos os nodos solicitações das atualizações realizadas nos ar-

quívos replicados deste nó durante o período em que ele estava desativado. Uma outra forma seria existir um nó que servisse de "backup" a um outro. Desta forma as solicitações seriam dirigidas apenas ao seu nó "backup".

### 3.1.2.3 - Arquivos Particionados

Este terceiro tipo de arquivo é caracterizado pelo particionamento de um arquivo entre vários nós. Isto é, parte dos dados do arquivo estarão localizados em um nó, uma outra fração do arquivo estará localizada em outro nó, e assim por diante de acordo com o critério de divisão escolhido. Algumas razões que conduzem a este tipo de distribuição são:

- O tamanho do arquivo é muito grande para ser armazenado em um único nó;
- Um determinado arquivo é muito ativo, de tal forma que sua taxa de acesso não é suportado por um único nó do sistema;
- A frequência de acesso as diferentes partes do arquivo pode ser muito grande para vários nós diferentes.

O particionamento de um arquivo como podemos observar está ligado a dois fatores. O tamanho do arquivo e a sua frequência de acesso. Atualmente o problema de tamanho é um fator de importância secundária pois existem dispositivos de grande capacidade de armazenamento. Entretanto esse aumento da capacidade de armazenamento não ocorre de maneira proporcional ao aumento da velocidade de acesso dos mesmos dispositivos. Assim persiste ainda como um fator importante no critério de particionar o arquivo, a



frequência de acesso a este arquivo.

O maior problema que se enfrenta na utilização deste particionamento de arquivos está ligado à manipulação de arquivos que possuem índices. Desta forma alguns problemas têm que ser resolvidos como por exemplo: Onde localizar o índice? Como endereços idênticos em diferentes nodos são resolvidos pelo índice? Finalmente a catalogação destes arquivos requer cuidados também especiais devido à necessidade de se fornecer nomes diferente a cada segmento do arquivo dividido ou ainda como uma opção alternativa podemos fornecer um único nome para o arquivo como um todo desde que contenha indicações referentes a cada segmento. A figura 16 abaixo ilustra uma distribuição com este tipo de arquivo.

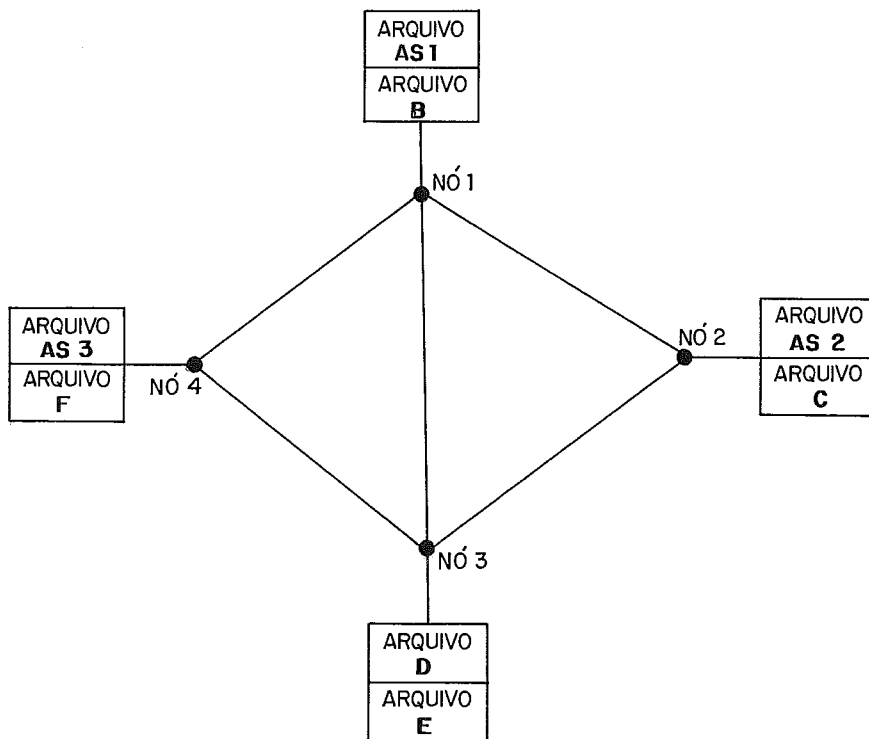


Figura 16

Finalmente alguns pesquisadores fazem referência ao tipo de arquivo "centralizado", onde todas as informações relativas ao arquivo estariam armazenadas em um único local. É evidente que essa configuração só se torna viável quando a rede é formada por um único computador com terminais a ele conectado (sistema centralizado). No caso de um Sistema Distribuído esse esquema fica impraticável, já que a taxa de acesso ao nó central atingirá índices elevados, com conseqüências imprevisíveis em termos de desempenho.

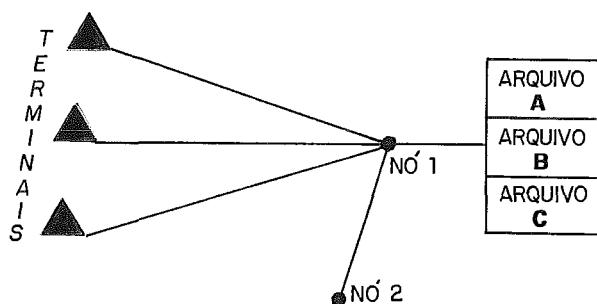


Figura 17

### 3.2 - FORMULAÇÃO DOS MODELOS DE ALOCAÇÃO DE ARQUIVOS

Em geral, os pesquisadores dos problemas relativos ao projeto de um Sistema de Processamento de Dados Distribuídos, procuram dar maior ênfase para o problema específico da alocação de arquivos, desenvolvendo uma formulação matemática do modelo e um método de solução como por exemplo em AKOKA<sup>1</sup>, CARRARESI<sup>14</sup>, CASSEY<sup>13</sup>, CHU<sup>24</sup>.

Entretanto a formulação de um modelo para designação de arquivos, para os diversos locais de um Sistema de Processamento de Dados Distribuídos, não é tarefa tão simples como a princípio parece. Isto se deve ao fato de que uma formulação estritamente matemática de um modelo de alocação, deverá levar em conta uma série de fatores, que além de não representarem a situação real do Sistema, conduzirão a uma solução extremamente custosa, em termos de tempo de processamento, na busca da solução ótima. Já por sua vez a utilização isolada de critérios meramente intuitivos para a solução do problema de alocação de arquivos, podem conduzir a resultados impróprios.

CHEN e AKOKA <sup>22</sup> afirmam que poucos modelos projetados podem evoluir para situações práticas, devido também as dificuldades de se obterem dados de um ambiente real. Assim por exemplo MORGAN e LEVIN <sup>67</sup> que muito embora tenham desenvolvido um modelo de alocação de arquivos bem mais abrangente que o método apresentado por CASSEY <sup>12</sup>, testaram o seu modelo com os próprios dados do modelo de CASSEY.

Relacionamos abaixo algumas das circunstâncias encontradas em um ambiente real e que por diversas razões não são consideradas nos métodos de alocação de arquivos, para um Sistema de Processamento de Dados Distribuídos.

- A maioria dos algoritmos existentes leva em consideração o número de mensagens existentes entre os diversos nodos da rede, na tentativa de minimizar o custo total do sistema. Quando na realidade o custo da mensagem depende da distância a ser percorrida e do tamanho da

mensagem. Existem mensagens curtas, que praticamente não custam quase nada, como por exemplo as mensagens referentes aos "Acknowledgement". E existem mensagens que transferem arquivos inteiros de um nodo para outro;

- A tentativa de otimizar o tráfego de mensagens na rede nem sempre conduz à otimização do desempenho geral do sistema. Isto é, concentrar demasiado serviço em um nó, poderá transformá-lo em um nó gargalo para o Sistema tendo pouca importância a otimização do número de mensagens, pois estaria caracterizado uma situação de "node-bound";
- Quando se opta pela possibilidade de duplicar arquivos em diferentes nodos da rede, deve-se levar em conta que as operações de atualizações são mais complicadas, necessitando de mecanismos de controle de concorrência mais complexos;
- Deve-se considerar também que arquivos podem não ser necessariamente as unidades de acesso, e que a localização no nó não precisa ser de um arquivo inteiro e sim de parte de um arquivo, onde a divisão está baseada em predicados e atributos;
- Para atingir a máxima eficiência o algoritmo para determinar a localização dos arquivos em uma rede de computadores deveria levar em conta o problema de "otimização de consultas", CHU<sup>25</sup>. Isto porque colocar dois

arquivos no mesmo nodo pode ter um custo menor para uma mensagem que venha de outro nodo e que utilize os dois arquivos. Porém colocá-los em nodos diferentes implica que o acesso pode ser feito em paralelo, diminuindo o tempo de resposta. Isto em resumo significa que colocar arquivos no mesmo nó envolve serializar processos enquanto que distribuí-los acarreta em paralelismo.

Se for levado em conta todos os componentes que influenciam e constituem a situação real de um Sistema, para formulação de um modelo analítico, seria extremamente penoso, e até mesmo inviável à procura da solução ótima. CERI<sup>15</sup> afirma em seu trabalho que diversos modelos têm sido propostos para solucionar este tipo de problema, entretanto a maioria deles não pode ser utilizados em um ambiente da vida real, devido à sua complexidade de processamento e o grande número de parâmetros difíceis de serem estimados. Assim por exemplo levando em conta apenas o problema da localização dos arquivos em uma rede de computadores contendo apenas 6 nodos e um conjunto de 10 arquivos que devem ser distribuídos através destes nodos, o número possível de designações de arquivos, onde cada nodo pode ter apenas uma cópia de cada arquivo, é de  $2^{60}$ . Sendo  $2^6$  a possibilidade para cada arquivo e  $(2^6)^{10}$  para todos os arquivos.

CASSEY<sup>12</sup> também observou que um programa codificado em FORTRAN e submetido a um Sistema IBM/360-91 consumiu um tempo de aproximadamente 10 segundos, estando incluídas neste tempo a compilação e execução do programa, para solucionar 6 diferentes problemas de alocação de arquivos para uma rede de 19 nodos. GRAPA<sup>46</sup> por sua vez complementou esta observação mostrando que para cada

adição de um nodo na rede implica em multiplicar o tempo de processamento por um fator 2. Desta forma se considerarmos que a alocação ótima de um arquivo em uma rede de 19 nodos consome aproximadamente 2 segundos de processamento, temos que para uma rede de 30 nodos esse tempo cresce para cerca de uma hora, atinge a marca de 48 dias para uma rede de 40 nodos e finalmente para uma rede de 50 nodos esse tempo será cerca de 136 anos de processamento.

Finalmente em relação à topologia da rede podemos afirmar que muito embora algumas das técnicas existentes para otimização de topologia sejam baseadas em métodos empíricos, na prática elas se mostram bastante satisfatórias aos projetistas de Sistemas, e afortunadamente os resultados destes métodos mostram-se bastante próximos dos valores ótimos, determinados pelos métodos essencialmente matemáticos. Isto porque em muitos casos o processamento do valor ótimo é bastante custoso em termos de tempo de processamento alcançando índices de inviabilidade na busca de uma solução ótima. De forma similar ao problema de alocação de arquivos mencionados anteriormente podemos mostrar que se existem  $N$  vértices em uma rede, existirão  $N(N-1)/2$  possíveis arcos. Desde que cada arco pode estar presente ou não, o número possível de virtuais configurações que podem ser realizadas é de  $2^{N(N-1)/2}$ . E para termos uma idéia mais concreta da dimensão do problema, temos que exatamente para uma rede com somente 10 nodos existem  $2^{45}$  possibilidades de topologias para se considerar. Se um programa gastar 1 mseg para analisar cada estrutura, o que seria um tempo razoavelmente bom, seriam necessários cerca de 10 séculos para concluir o trabalho. Isto evidentemente inviabiliza a busca essencialmente matemática do objetivo.

No intuito de reduzir a dimensão destes problemas muitos autores introduzem na formulação de seus modelos alguns critérios, possuindo muitas vezes um caráter meramente intuitivo, que ajudam a determinar uma solução, se não ótima, pelo menos uma aceitável resposta para questão de alocação de arquivos com um limitado esforço de pesquisa. Dentre estes diversos critérios utilizados podemos enumerar:

- Localização Geográfica;
- Tipo de Arquivo;
- Frequência de acesso;
- Volume de dados.

### 3.2.1 - Critério da Localização Geográfica

O princípio de alocar arquivos segundo a localização geográfica dos usuários que acessam os arquivos indica que um determinado arquivo deva ser designado para o local que mais o acessa. Contudo este critério sendo considerado isoladamente pode conduzir a distorções no desempenho do sistema. Este fato fica bem caracterizado através do seguinte exemplo. Se 10 usuários em uma localidade X acessam o arquivo K, e se apenas 2 usuários em uma localidade Y também acessam este arquivo, seria natural que, segundo o critério da localização geográfica do usuário que mais acessa o arquivo, o arquivo K seja designado para X. Entretanto se a frequência de acesso ao arquivo K dos usuários localizados em Y for muito maior do que a frequência de acesso dos usuários situados em X, o procedimento estabelecido pelo critério torna-se imperfeito.

### 3.2.2 - Critério do Tipo de Arquivo

Este princípio indica que a alocação do arquivo deva ser baseada no tipo de arquivo. Assim por exemplo os arquivos de empregados, os arquivos de produtos etc., devem ser designados de acordo com a localização dos empregados e produtos respectivamente. Este princípio também apresenta as mesmas restrições do critério anterior, a sua aplicação isolada de outros fatores conduzem a uma alocação deficiente.

### 3.2.3 - Critério da Frequência de Acesso

Este critério afirma que a distância de um arquivo a um nodo é inversamente proporcional a frequência de acesso desse nodo ao arquivo. Desta forma quanto maior a frequência de acesso de um nodo a um arquivo, mais próximo deste nodo deve estar localizado o arquivo.

### 3.2.4 - Critério do Volume dos Dados

O volume de dados que circulará entre os diversos locais da rede, constitui também um importante critério a ser observado a fim de determinar uma alocação ótima para o sistema. Isto porque, apesar de se ter uma reduzida frequência de acesso a um determinado arquivo, o volume de dados a ser transmitido pode ser suficientemente alto que pode ser considerado para alocação de arquivos como correspondendo a uma maior frequência de acesso. O seguinte exemplo apresentado por OHANA<sup>69</sup> caracteriza este tipo de raciocínio:



- O primeiro nodo acessa 1000 itens de 4 bytes, ou seja, 4000 bytes de informações;
- O segundo nodo acessa 100 registros onde o tamanho do registro é 20 vezes o tamanho do item, ou seja, 8000 bytes de informações.

Com esses dados podemos afirmar que o maior volume de informações do segundo nodo corresponderá a uma frequência de 2000 itens de 4 bytes que também é superior ao primeiro nodo.

Todavia a introdução de critérios meramente intuitivos para a solução do problema de alocação de arquivos podem conduzir a resultados impróprios. Assim por exemplo no problema apresentado por DOWDY e FOSTER <sup>35</sup> onde os autores supõe um Sistema de Processamento Distribuído com três nodos e um conjunto de 10 arquivos para ser designados através dos nodos, de forma a maximizar a soma das vazões dos nodos do sistema, a conclusão que se chega é que neste enfoque de atribuição a utilização de critérios heurísticos podem conduzir a resultados enganosos. A figura 18 a seguir ilustra os dados do modelo proposto por FOSTER <sup>35</sup>, e como podemos observar seria natural, por exemplo, atribuir o arquivo número 10 para o nodo do Sistema-1, que apresenta segundo a tabela de frequência de acessos uma taxa mais elevada. Contudo solução ótima determinada pelo seu método depois de exaustiva pesquisas conduz ao seguinte resultado.

- Sistema-1 ... designados os arquivos ( 1,4 )
- Sistema-2 ... designados os arquivos (3,6,7,8,9)
- Sistema-3 ... designados os arquivos (2,5,10)

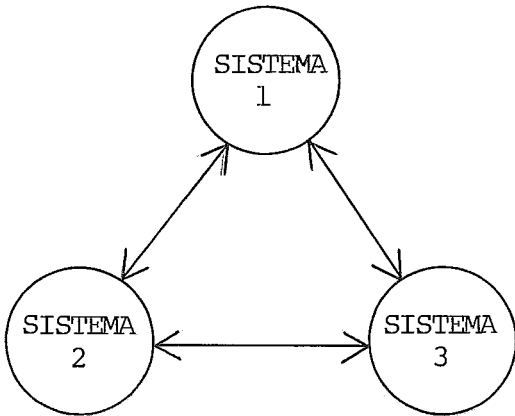


Figura 18

ARQUIVOS	S I S T E M A S		
	1	2	3
1	.170	.111	.012
2	.040	.041	.032
3	.060	.141	.162
4	.150	.051	.082
5	.210	.151	.242
6	.080	.061	.072
7	.070	.101	.022
8	.020	.181	.202
9	.110	.091	.142
10	.090	.071	.032
	1.000	1.000	1.000

Todas estas considerações gerais sobre a questão de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos, bem como os critérios utilizados na formulação de um modelo nos indicam três importantes conclusões. A primeira delas nos mostra que o problema de alocação de arquivos não é uma tarefa trivial. A segunda conclusão que tiramos é que a interação entre os parâmetros envolvidos é complexa. E finalmente podemos admitir que soluções intuitivas podem conduzir a resultados infelizes.

### 3.3 - DETERMINAÇÃO DE UMA FUNÇÃO ÓTIMA

A função ótima de uma alocação de arquivos em uma rede de computadores, objetiva primordialmente uma minimização dos custos ou uma maximização do desempenho.

A minimização dos custos consiste tipicamente em incluir, em uma função objetivo, os custos de armazenamento dos arquivos, custos de consultas, custos de atualizações e custos de comunicações.

O estudo de todos estes custos de forma isolada poderia simplificar a discussão do problema. Contudo deve ser dada atenção para o sistema com todos os seus aspectos possíveis, isto porque pode ser interessante que maiores custos de comunicação possam ser mais do que compensados por menores custos de armazenamento. O custo total da função objetivo seria portanto dado pela solução da expressão:

$$\text{CUSTO-TOTAL} = \sum_{i=1}^N \text{CUSTO}_i$$

onde  $n$  representa todos os custos considerados no modelo do sistema escolhido, sejam custos diretos ou indiretos, fixos ou variáveis.

A outra forma de medir uma ótima alocação de arquivos, inclui em sua função objetivo o tempo mínimo de resposta e a máxima vazão do sistema. Estes tipos de atribuições levam em conta os problemas de atraso e direcionamento, sendo o atraso função da velocidade dos dispositivos e da taxa de acesso aos dispo-

sitivos e sendo o problema de direcionamento função da habilidade de requisitar arquivos via caminhos alternativos. Estas duas formas de estabelecer uma função objetivo para o problema de alocação de arquivos são desenvolvidas de formas distintas.

A função objetivo que busca a minimização dos custos é desenvolvida durante a fase de projeto, onde as variáveis de custos tais como: custos de armazenamento, comunicação e acesso influenciam diretamente na arquitetura inicial do sistema. Os problemas relativos ao desempenho são tratados de forma secundária neste enfoque. Este fato deve-se ao pouco conhecimento, em tempo de projeto, sobre o comportamento do sistema quando submetidos à sua carga de trabalho.

A maximização do desempenho é por sua vez aplicada em tempo de produção. Muitas vezes, neste tipo de enfoque, os custos são considerados fixos e a ênfase do problema é dirigida à carga de trabalho que em tempo de produção pode ser perfeitamente medida.

Uma verdadeira função ótima para o problema de alocação deveria considerar o desempenho e custos, como elementos a serem avaliados em tempo de projeto e em tempo de produção.

### 3.4 - RESTRIÇÕES DOS MODELOS DE ALOCAÇÕES

Nenhum modelo de sistema pode ser projetado em condições ilimitadas. Em geral eles operam com limitações de recursos. Por isto devem ser formuladas restrições ou condições de limite que devem ser observadas para alcançar os objetivos desejados.

CYPSE<sup>28</sup> identifica três conjuntos de restrições. A primeira delas são aquelas que limitam o "exterior" da organização, além de controlar o seu gerenciamento. São as chamadas restrições "ambientais". No segundo conjunto encontramos as restrições "organizacionais" que estão sobre controle interno, tais como: custos operacionais, treinamento de pessoal etc. E finalmente o terceiro grupo de restrições são aquelas denominadas "operacionais", e consiste dos padrões que o sistema deve satisfazer para se tornar aceitável. Por exemplo o número de unidades de processamento diário.

Entre as restrições operacionais mais comuns que são consideradas nos métodos de alocação, encontramos:

- A topologia da rede é dada;
- A capacidade de armazenamento que pode ser instalada em um nodo é limitada;
- A rede é completamente conectada;
- Não existe dependência entre arquivos e programas;
- O número de cópias de arquivos é fixo;
- Não são considerados os fluxos de retorno das informações.

### 3.5 - TÉCNICAS PARA SOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE ARQUIVOS

Uma das condições a ser satisfeita para o aperfeiçoamento de Sistemas de Computação é o conhecimento adequado das ferramentas disponíveis capaz de aumentar o seu desempenho. No caso particular da designação de arquivos para os diversos nodos de um

Sistema de Processamento Distribuído a principal fonte de análise científica, para a solução do problema, é a Pesquisa Operacional. Ela, como muitos outros campos da ciência, está particularmente interessada em fabricar decisões para situações com origem no mundo real e que se caracterizam pela necessidade de alocação limitada de recursos. Para essas situações a análise científica proporcionada pela pesquisa operacional oferece meios e uma considerável perspicácia para a solução deste tipo de problema.

O seu processamento começa com uma cuidadosa observação e formulação do problema e a partir desta formulação, uma construção científica (tipicamente matemática) do modelo, para representar o sistema sob estudo. A próxima fase da pesquisa operacional será deduzir uma solução para este modelo e finalmente as fases relativas a teste, controle e implementação do modelo.

Entre os diversos métodos de programação que solucionam este tipo de questão nos encontramos:

- A Simulação Matemática;
- A programação Linear;
- A Programação Inteira;
- A Programação Heurística;
- A Teoria dos Grafos.

### 3.5.1 - Simulação Matemática

Na Simulação Matemática os componentes essenciais do sistema podem ser reduzidos a um modelo matemático e, através do uso de computador, podem ser usados para testar diversas configura-

ções que permitam reduzir os custos envolvidos no sistema. Desta forma, através da comparação dos resultados simulados, para cada uma das alternativas propostas poder-se-á então melhor escolher a opção da distribuição que mais se aproxima dos objetivos pretendidos.

### 3.5.2 - Programação Linear

A Programação Linear por sua vez pode ser empregada para alocação de arquivos através do planejamento do tráfego de dados entre os diferentes nodos da rede de forma a minimizar os custos de transmissão. Assim conhecendo-se a demanda do nodo consumidor pelo dado, que pode estar armazenado em diversos locais, e o volume a ser transmitido é possível através da Programação Linear, determinar que conjunto de locais deverão enviar dados de forma a minimizar os custos totais de transmissão.

### 3.5.3 - Programação Inteira

A técnica de Programação Inteira é uma variante da Programação Linear. Consiste simplesmente em determinar respostas que apareçam sempre sob a forma de número inteiro. A principal vantagem em relação à programação linear é a sua capacidade para determinar a solução ideal dentre considerável número de alternativas mais do que a programação linear, e sem o dispêndio de considerável tempo e esforço.

#### 3.5.4 - Programação Heurística

O instrumento heurístico é qualquer tipo de método utilizado para reduzir o esforço de solucionar problemas - ou seja é o ovo de colombo, empregado para resolver uma específica questão. Embora esta técnica apresente resultados relativamente rápidos, elas apresentam como principal desvantagem o fato de estar voltada para o encontro de uma boa resposta para o problema e não para o encontro da melhor solução.

#### 3.5.5 - Teoria dos Grafos

Ao contrário de muitos ramos da matemática, nascidos de especulações puramente teóricas, a Teoria dos Grafos tem sua origem no confronto de problemas práticos de diversas procedências e se deriva no estudo das características comuns desses problemas.

Muitos dos problemas estudados em Grafos dizem respeito ao puro e simples inter-relacionamento de vértices, especialmente nos casos em que apenas os vértices têm significado quando materialmente ligado aos elementos do problema. Nestes tipos de casos as arestas, ou arcos, têm valor apenas por indicarem o inter-relacionamento existente entre os vértices. São os problemas de características booleana: Há ou não há relação.

Em outros casos, como por exemplo em problemas de rede de transporte ou de computadores, os arcos têm um significado material bem definido ao qual estão, em geral, associadas determinadas propriedades, tais como: custos, distâncias, capacidades, etc. Encontramos também nestes tipos de redes, dados relacionados com a utilização das estruturas que deverão ser modeladas, assim como



por exemplo fluxos de tráfegos, cargas elétricas transmitidas, "bits", etc. Nestas situações teremos então que definir funções associadas aos vértices e aos arcos para tornar possível a modelagem do problema e que descrevam o funcionamento da estrutura.

Desta forma a teoria dos grafos é um importante instrumento para o projeto de rede de computadores, principalmente nos problemas relativos a conexidade e atraso. Os algoritmos para determinar o menor caminho entre dois nodos de um grafo ponderado e também os algoritmos para determinar o fluxo máximo possível entre dois nodos são exemplos destas técnicas.

WHITNEY<sup>84</sup> investigou o problema de atribuição de cópias de arquivos para os vértices de um grafo linear, enquanto minimizava o tráfego induzido ao longo dos arcos deste grafo.

Todos estes tipos de métodos apresentados, que na realidade representam em sua formulação algoritmos de otimização, podem ser expressos de forma estática e de forma dinâmica.

A otimização estática consiste no desenvolvimento de uma função objetivo que será minimizada ou maximizada em um simples instante de tempo, sujeito a um conjunto de restrições. Já a idéia de otimização dinâmica é minimizar ou maximizar uma função objetivo sobre um intervalo de tempo também sujeito a algumas restrições. A escolha entre o uso de algoritmos estáticos ou dinâmicos é dependente do sistema. Se o sistema permite uma determinação precisa de todos os parâmetros então os algoritmos estáticos são bons. Em caso contrário, onde a determinação de alguns parâmetros torna-se imprecisa, os algoritmos dinâmicos apresentam um melhor comportamento.

Dentro deste contexto apresentado citaremos no item seguinte, alguns autores que pesquisaram sobre o problema de alocação de arquivos.

### 3.6 - ALGUNS TRABALHOS REALIZADOS SOBRE ALOCAÇÃO DE ARQUIVOS

Um dos primeiros investigadores sobre o problema de alocação de arquivos foi CHU <sup>24</sup> . Ele analisa o problema da distribuição baseado no modelo relacional de dados, desenvolvendo um método para obter custos mínimos de operação, na qual o custo de transmissão estava associado à alocação do arquivo. Entretanto o modelo oferecia uma série de restrições tais como: exige uma rede completamente conectada, um número fixo de cópias de arquivos, e uma fixa disponibilidade de armazenamento em cada nodo. A técnica utilizada para encontrar a solução é a da programação linear inteira zero-um.

Já CASSEY<sup>12,13</sup> , considerou o problema de alocação de arquivos separadamente do custo de transmissão. O número de cópias de arquivos em cada nodo não foi considerado fixo por CASSEY <sup>12</sup> . O seu modelo diferenciava os custos de recuperação de informação e os custos de atualização, fazendo uso de uma rede completamente conectada. O principal objetivo do modelo é determinar em que nodos da rede cópias de um único arquivo serão armazenadas. Esta alocação se presume seja feita de tal maneira que possa minimizar o custo total de comunicação entre usuários e arquivos. A técnica utilizada para atingir a solução do problema se baseia na programação inteira ou na teoria dos grafos (busca de uma rota ótima num grafo de custos) para determinar o limite máximo do núme-

ro de cópias do arquivo que estarão presentes na rede, levando em conta as proporções do tráfego de atualização em relação ao tráfego de consultas gerados pelos usuários de um arquivo da rede.

GHOSH <sup>45</sup> propôs a pesquisa do problema de distribuir um banco de dados que contém associações lógicas entre tipos de segmentos, onde um segmento é uma agrupação de informações, sobre uma rede de computadores, de tal forma que os tipos múltiplos de segmentos satisfaçam uma consulta que possa ser recuperada em paralelo de diferentes nodos. O processamento em paralelo de diferentes subconjuntos de dados pertinentes a uma consulta é uma característica importante pela redução do tempo de resposta. A técnica utilizada no modelo é baseada em uma busca combinatória através de várias soluções possíveis. A contribuição do algoritmo é a de formular um modelo que leva em conta as associações lógicas entre subconjuntos de dados que possam ser recuperados em paralelo, minimizando o tempo de resposta.

IRANI e KHABDAS <sup>51</sup> examinaram e combinaram no seu modelo de alocação de arquivos o projeto da rede de comunicação. Eles estabeleceram um procedimento determinístico para atingir em subsequentes passos a alocação de arquivos e a alocação da capacidade dos canais de comunicações.

MORGAN e LEVIN <sup>67</sup> estudaram o problema de alocação de arquivos em uma rede de computadores levando em conta o fato de que o acesso e requisições sobre arquivos de dados são executados por programas, e assim sendo o modelo proposto por eles considera a localização dos programas e arquivos simultaneamente no processo de otimização de sua função objetivo. Uma transação é encami-

nhada inicialmente a seu programa pertinente e desse programa a consulta é transmitida à cópia do arquivo mais próxima, enquanto as mensagens de atualizações são transmitidas a cada uma das cópias dos arquivos. O problema formulado pertence à classe de problemas de programação não linear zero-um.

MAHMOUD e RIORD <sup>62</sup> desenvolveram um modelo onde a capacidade do canal de comunicação e o problema de alocação de arquivos são examinados simultaneamente, no intuito de alcançar um custo mínimo. O modelo fica sujeito às restrições de disponibilidade de armazenamento e o atraso na rede. Seu objetivo é alocar cópias de arquivos e capacidade dos canais de comunicações de modo que um custo mínimo seja atingido com base nas demoras na fila de espera da rede e nas restrições de disponibilidade do arquivo. A solução é baseada em métodos heurísticos com duas rotinas principais, a rotina de Inicialização e a rotina de otimização. A rotina de inicialização gera um número de soluções iniciais possíveis onde uma solução possível é uma alocação de arquivos e capacidade dos canais de comunicações que satisfaçam as restrições de demora e disponibilidade. Começando com uma solução inicial, a rotina de otimização tenta otimizar o custo total por sucessivas somas e diminuições de cópias de arquivos. Quando uma solução possível com menores custos é achada, ela é adotada como uma solução inicial e o processo continua. Cada execução da rotina de otimização, para determinar os custos mínimos correspondentes a uma solução inicial, é chamada de "execução heurística". O resultado da execução heurística que corresponder ao custo mínimo de todas as soluções achadas é adotada como solução final.

Muitos outros pesquisadores estudaram o problema de alocação de arquivos assim como AKOKA <sup>1</sup> , CARRAESI <sup>14</sup> , KOLLIAS <sup>57</sup> , LANING <sup>59</sup> , OHANA <sup>69</sup> , RAMAMOORTHY <sup>71,72</sup>. DOWDY e FOSTER <sup>35</sup> realizaram um estudo sobre diversos modelos de alocação em termos de suas funções objetivos e restrições, fazendo considerações sobre as características envolvidas em cada um deles. A tabela da figura 19 abaixo é uma consolidação das características destes modelos sugerindo uma comparação e avaliação dos diversos enfoques apresentados. Eles também consideram que os modelos em cujas formulações apresentam características tais como: uma topologia generalizada, capacidade dos canais de comunicações, duplicação de arquivos, eficácia, conduziram provavelmente a uma melhor distribuição e desempenho.

CARACTERÍSTICAS		M O D E L O S													
		CASSEY <sup>12</sup>	CHANDY <sup>18</sup>	MORGAN <sup>67</sup>	CHU <sup>24</sup>	MAHMOUD <sup>62</sup>	RAMAMOORTHY	ARORA <sup>2</sup>	BUZEN <sup>11</sup>	CHEN <sup>21</sup>	KLEINROCK <sup>55</sup>	FOSTER <sup>39</sup>	FOSTER <sup>36</sup>	FOSTER <sup>38</sup>	DOWDY <sup>34</sup>
1	MÚLTIPLOS ARQUIVOS				•	•	•	•	•	•	•	•	•	•	•
2	CÓPIAS DE ARQUIVOS	•	•	•	•	•								•	
3	TRÁFEGO DE CONSULTA E ATUALIZAÇÃO	•	•	•	•	•								•	
4	CUSTO DE ARMAZENAMENTO	•	•	•	•	•	•								
5	FILAS DE ATRASO				•	•		•	•	•	•	•	•	•	•
6	TEMPO DE ACESSO, RESPOSTA, THROUGHPUT				•	•	•	•	•	•	•	•	•	•	•
7	SOLUÇÕES LIMITADAS		•					•	•	•	•	•	•	•	•
8	CAPACIDADE DE ARMAZENAMENTO				•		•				•	•	•	•	•
9	CAPACIDADE DOS CANAIS DE COMUNICAÇÕES					•				•					
10	DIRECIONAMENTO NÃO FIXADO			•					•	•	•	•	•	•	•
11	DEPENDENCIA ENTRE PROGRAMAS/ARQUIVOS			•			•								
12	DESIGNAÇÃO INTEGRAL DO ARQUIVO	•		•	•	•		•			•	•	•	•	•
13	DEPENDENCIA ENTRE LOCAIS/ARQUIVOS			•			•								
14	TOPOLOGIA GENERALIZADA	•	•	•		•				•					•
15	ÓTIMA ATRIBUIÇÃO	•		•	•		•	•	•						
16	TRATAMENTO HEURÍSTICO		•			•				•	•	•	•	•	•
17	EFICÁCIA ENTRE ARQUIVOS/LOCAIS				•	•									
18	FUNÇÃO OBJETIVO LINEAR E RESTRIÇÕES	•	•				•	•							

CAPÍTULO IV

CARACTERIZAÇÃO DA METODOLOGIA

4.1 - INTRODUÇÃO

O método discutido nesta tese procura ser o mais geral possível, suficientemente flexível para permitir adaptá-lo as necessidades de cada organização. Assim a metodologia tende a ser aplicável às variações do tipo de organização, pois não existe nenhuma razão aparente para que a metodologia exposta não possa incluir organizações atuantes nas mais diversas áreas da atividade pública e privada, sejam aplicações bancárias, médicas, industriais, transporte, etc. O método pode também ser adequado aos vários tipos de topologias de Rede de Computadores, sejam estruturas em anel, estrela, distribuída, etc.

Tem-se convencionado que a técnica de desenvolvimento de projetos, de uma maneira geral, envolve pelo menos três aspectos principais. A concepção, a elaboração propriamente dita e a avaliação. Estes três aspectos constituem o que muitos autores costumam chamar de "ciclo de vida" de sistemas de informações apoiados em computador, como procura mostrar a figura 20 abaixo.

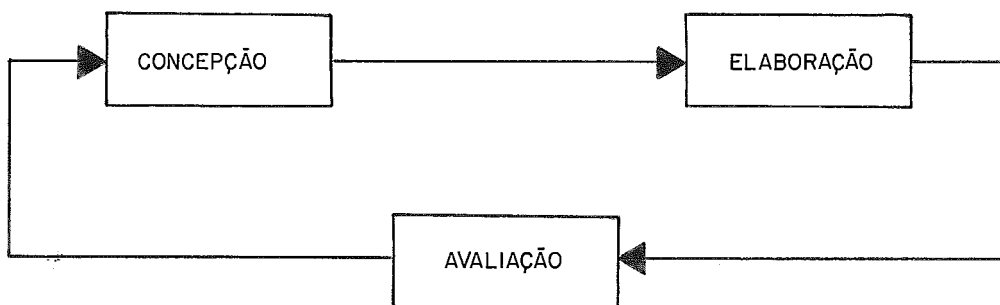


Figura 20

A metodologia apresentada se propõe dentro destes três aspectos, concepção, elaboração e avaliação, a desenvolver um conjunto de técnicas, baseadas em uma extensão de alguns princípios fundamentais de concepção de projetos de Sistemas de Banco de Dados convencionais, para o projeto de Sistemas de Processamento de Dados Distribuídos, procurando dar uma maior ênfase ao problema de alocação de arquivos. Os objetivos principais da metodologia são:

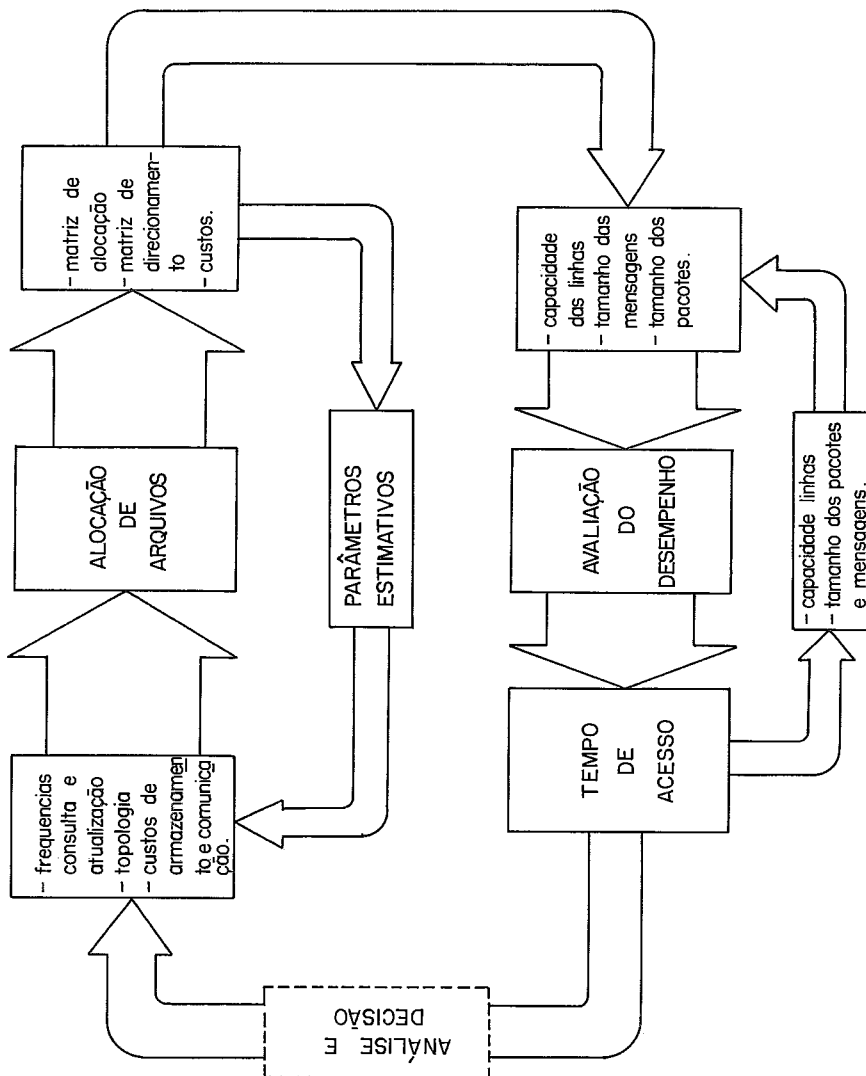
1º) Identificar o fluxo de informações do Sistema por meio do relacionamento existente entre os componentes PROCESSOS, SETORES, APLICAÇÕES e DADOS da organização. Estes componentes serão respectivamente os geradores de:

- Um conjunto de ATIVIDADES (A), necessárias a administração dos recursos e operação de uma empresa;
- Um conjunto de LOCAIS (N), onde são realizadas estas atividades, que representarão virtualmente os nodos de uma rede de computadores;
- Um conjunto de PROGRAMAS DE APLICAÇÕES (P), responsáveis pelas transações a serem submetidas pelos usuários;
- Um conjunto de ARQUIVOS (F), que constituirão o Banco de Dados a ser compartilhado pelos diversos usuários do Sistema.

2º) Determinar a alocação ótima dos arquivos para os diversos locais do Sistema, que melhor atenda os interesses da organização em termos de custos de armazenamento e custos de comunicação.

39) Executar uma análise formal do modelo de alocação projetado, a fim de determinar se a alternativa selecionada, tem um desempenho em termos de tempo de acesso ao dado tão bom como o esperado. Caso isto não aconteça, o método permite um processo de realimentação que determinará novas estruturas a serem analisadas.

A figura 21 abaixo ilustra esta caracterização que foi descrita nesta introdução.





#### 4.2 - FASE DE CONCEPÇÃO DE UM S.P.D.D.

A fase de concepção está fundamentada em três estudos principais:

- O da percepção da necessidade, que procura identificar precisamente os problemas que determinarão o desenvolvimento dos estudos de um novo sistema ou a reformulação do sistema atual, bem como a fixação dos objetivos que deverão ser alcançados com a introdução de novos procedimentos;
- O estudo de viabilidade, que deve atentar para as vantagens e desvantagens existentes nas diversas alternativas propostas para solucionar os problemas da organização;
- E o processo de decisão, que consiste da seleção entre as diversas alternativas possíveis aquela mais conveniente em termos econômicos, técnicos e operacionais.

O primeiro movimento a ser executado no sentido de verificar se um Sistema de Processamento de Dados Distribuídos seria a melhor solução é o estudo de viabilidade. Este estudo tem a finalidade de determinar as necessidades do usuário no que diz respeito:

- a sua distribuição geográfica;
- ao fluxo de informações;
- ao tempo de resposta necessário;
- a carga de trabalho;

- ao uso dos recursos e suas disponibilidades;
- a frequência de utilização das informações.

Podemos ainda acrescentar que um outro fator importante neste estudo é determinar se existe uma perspectiva de expansão do raio de ação das aplicações em um futuro próximo.

Após uma análise consciente de todos estes aspectos, algumas questões podem ser levantadas no intuito de estabelecer se o projeto de um Sistema de Processamento de Dados Distribuído é realmente essencial a organização.

CYPSE<sup>28</sup>, estabelece dois critérios básicos na determinação das funções que podem ser distribuídas em um projeto de sistemas.

- A mesma função pode ser executada em mais de um local na rede;
- A função não é completamente executada em um único nó. Parte desta função é executada cooperativamente em separados nodos.

Entretanto podemos ainda estabelecer outros critérios que podem contribuir a definir necessidades e possibilidades para um Sistema de Processamento Distribuído, suas vantagens e desvantagens em relação as aplicações.

- Um dos critérios a adotar seria verificar se as aplicações para o qual o sistema está sendo projetado tem uma reduzida necessidade de interrelacionamento. Este fator sendo mínimo, dificilmente justificará os custos envolvidos no processo de distribuição;

- A disposição física das várias localidades que constituirão os nodos da rede, é um outro fator importante quando se trata de necessidades e custos de comunicação;
- O volume e a importância das informações trocadas é também um critério que deve ser considerado, já que, se a quantidade de informações trocadas for mínima, ou se as informações não forem relevantes, não seriam justificados os custos de comunicação;
- E finalmente todas as abordagens feitas no item 2.7 do capítulo II, são instrumentos valiosos na tomada de decisão, para o desenvolvimento de um projeto de um Sistema de Processamento de Dados Distribuído.

O caminho a ser seguido para a descoberta de que um Sistema de Processamento Distribuído é a solução ideal, poderá consumir mais tempo do que o usualmente gasto ao planejar um sistema centralizado. Também não devemos esquecer que embora seja mais fácil lidar com partes do sistema do que com o todo, os problemas em um Sistema de Processamento de Dados Distribuídos tendem a distribuir-se também, e que a eficiência do Sistema de Informações é mais importante do que o uso eficiente do computador. O Sistema que se está desenvolvendo deve ser orientado para o utilizador, ao invés de ser orientado para a máquina.

#### 4.3 - FASE DE ELABORAÇÃO

Dividimos a fase de elaboração em duas etapas principais. A primeira delas denominada Distribuição Lógica está funda

mentada no conceito de que o plano de um Sistema de Processamento de Dados Distribuídos de uma instituição deve ser compatível e concordante com os objetivos da organização como um todo e não apenas com os interesses de órgãos ou setores isolados.

Esta visão abrangente da organização e de suas necessidades de informações, normalmente conduz a um plano de sistema de informação integrado através de Banco de Dados. Porém não estudaremos neste trabalho as características dos software sobre Sistemas de Gerência de Banco de Dados Distribuídos existentes. Também os problemas relativos ao processamento de consultas e controle de concorrência não serão abordados nesta tese. As referências LANN <sup>60</sup> , COURLOIS <sup>27</sup> , BERSTEIN <sup>4,5,6,7</sup> , CELLARY <sup>16</sup> , ROSENKUANTZ <sup>73</sup> , MENASCE <sup>66</sup> , WONG <sup>87</sup> , HEVNER <sup>48</sup> , BRACHI <sup>9</sup> , tratam exaustivamente destes assuntos. O Banco de Dados nesta metodologia é representado pelos arquivos disjuntos  $\{F_1\}$  ,  $\{F_2\}$  , ..  $\{F_n\}$  que unidos constituem um virtual Banco de Dados  $F = \{F_1, F_2, \dots, F_n\}$  .

Portanto um Sistema de Banco de Dados Distribuídos pode ser encarado como uma forma de Sistema de Processamento Distribuído, existindo quando cada processador do Sistema possui arquivos permanentes a ele ligado. Desta forma um Banco de Dados Distribuídos existe quando um Banco de Dados logicamente integrado encontra-se fisicamente distribuído por vários processadores interconectados entre si.

#### 4.3.1 - Distribuição Lógica

A fase do projeto lógico baseia-se quase sempre no sistema existente na organização. Nesta fase são examinados os processos que são apoiados pelo sistema atual, são analisadas as informações que lhe dão suporte, e como produto final obtém-se um novo sistema de informações proposto em termos lógicos, com indicações precisas de objetivos, requisitos, componentes funcionais, entradas, saídas, pontas de coletas de dados, fluxos gerais de informações e os diversos relacionamentos existentes entre os processos e entre os diversos componentes identificados.

Nesta primeira fase são também identificados e quantificados os diversos componentes que servirão de entrada ao processo de alocação de arquivos.

- Os processos da instituição e suas aplicações;
- Os dados que constituirão o conjunto de arquivos;
- A estrutura dos arquivos;
- As frequências de recuperação e atualização por transação em relação a um arquivo e a frequência com que cada transação é empregada em cada nodo;
- Os diversos setores envolvidos nestes processos;
- Distâncias entre os locais;
- Tipos e capacidades dos equipamentos.

Para execução desta primeira fase são utilizadas as ferramentas de Análise e Projeto de Sistemas, direcionadas especificamente ao Projeto de um Sistema de Processamento Distribuído, cujos passos são assim discriminados:

- Análise dos Dados;
- Análise dos Software;
- Análise das Localidades;
- Análise das Atividades;
- Projeto da Rede do Sistema Distribuído.

#### 4.3.1.1 - Análise dos Dados

Em sistemas de computação do tipo convencional, as estruturas de dados são geralmente projetadas em função de um particular emprego. Desta forma um arquivo de dados está intimamente associado a um ou mais programas de aplicações. Para estes tipos de sistemas a análise dos dados torna-se uma tarefa extremamente simples. Na verdade é a aplicação que determina que tipo de dados o arquivo deve conter, sacrificando até mesmo o grau de redundância e consistência dos dados e a integridade do sistema como um todo.

O aparecimento das técnicas de Banco de Dados provocou uma migração crescente das aplicações para este novo método de gerência de dados, como mostra KING <sup>54</sup>, no gráfico da figura 22 a seguir, e uma conseqüente mudança substancial neste tipo de análise. Isto porque as informações que irão compor a base de dados, devem possuir uma estrutura independente dos requisitos de qualquer programa responsável pela solução de uma aplicação do sistema.

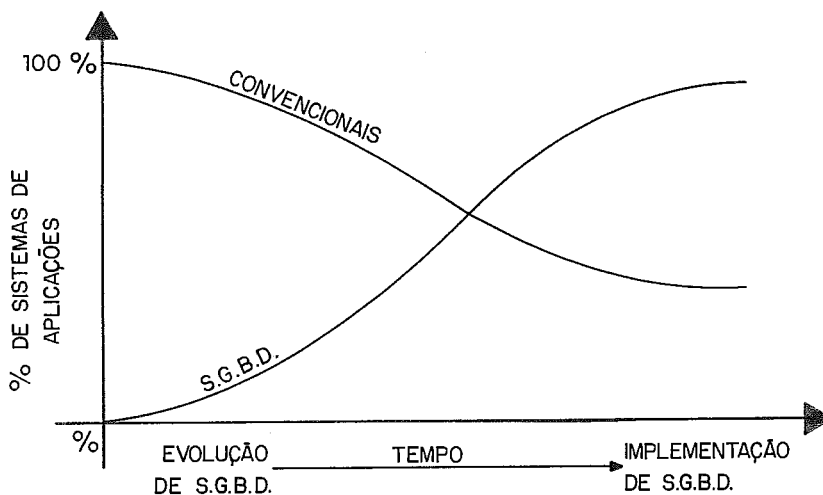


Figura 22

Considerando este novo enfoque, a análise dos dados, em um Sistema de Gerência de Banco de Dados, apresenta-se com objetivos de grande importância, entre os quais podemos citar a identificação dos recursos de dados de um organização, sua natureza, seus interrelacionamentos, e a organização de um processo de documentação de todos os fatos inerentes ao dado. Estas tarefas de análise de dados adquiriram formas de tal importância, no contexto de um S.G.B.D., que as empresas normalmente adotam em sua estrutura funcional uma específica função denominada "Administrador de Dados".

No caso particular de um projeto de Sistema de Processamento de Dados Distribuídos sob a gerência de um software para Banco de Dados Distribuído, o problema da análise de dados torna-se ainda mais complexo. Esta complexidade é devido aos problemas adicionais encontrados em sistemas deste tipo, que requerem um tratamento mais apurado na análise e estruturação dos arquivos de dados.

Em primeiro plano encontramos a questão relacionada com as necessidades comuns de informações nas diversas localidades.

Este fato pode conduzir à replicação de arquivos por toda a rede, se a relação custo-benefício por favorável à alternativa de armazenar cópias de arquivos onde for necessário ao invés de procurar e acessar a informação desejada, onde ela estiver armazenada. Esta escolha tem por sua vez a inconveniência de se ter que pagar um preço adicional pela atualização de todas as cópias dos arquivos existentes a fim de se manter a integridade da Base de Dados.

Em segundo lugar, mais em igualdade de condições quanto ao grau de complexidade, encontramos o problema da necessidade de particionar o arquivo entre vários nodos. Este fato pode ocorrer em função de três fatores principais:

- A necessidade de diferentes partes de um arquivo serem manipuladas por mais de um nodo;
- O arquivo é muito grande não podendo ser armazenado em um único local;
- A taxa de acesso a um arquivo é muito elevada, de tal forma que não possa ser suportado por um único nodo.

Para que estes objetivos estabelecidos na Análise dos Dados sejam atingidos são necessários o desenvolvimento de alguns passos, os quais poderiam ser empregados indiferentemente, quer para um enfoque centralizado ou distribuído. Entre esses procedimentos encontramos alguns dos sugeridos por DAVENPORT<sup>29,30</sup>, quando definiu este tipo de análise.

1º) Inicialmente devem ser definidas as áreas de aplicações a serem analisadas. Isto pode ser feito identificando-se as principais áreas de aplicações do Sistema, tanto para as existentes como para aquelas projetadas para o futuro.



29) O próximo passo a ser executado é identificar todos os dados necessários a cada área de aplicação e determinar os principais tipos de "entidades" para cada uma dessas áreas de aplicações. Cada tipo de entidade deve ser capaz de ser definida precisamente.

39) Sendo identificadas as entidades, o próximo passo será determinar o relacionamento existente entre as entidades. O resultado deste relacionamento, que pode ser expresso através de diagramas, como por exemplo o modelo entidade-relacionamento proposto por CHEN <sup>20</sup>, tem como objetivo identificar o modelo de dados de cada área de aplicação.

49) As propriedades ou descrição dos valores associados às entidades devem também ser considerados durante a Análise dos Dados. Estes valores são denominados "atributos" e um deles deve ser escolhido para identificar unicamente uma entidade (chave).

59) O processo de normalização de CODD <sup>26</sup> pode ser utilizado para examinar os atributos associados com uma entidade, para permitir que não exista nenhuma dependência oculta entre os atributos que possa ser representada diretamente no modelo de dados.

69) Em cada área de aplicação é necessário descrever e identificar o número de TRANSAÇÕES que serão executadas. Normalmente uma transação estará ligada por um evento, onde um evento pode ser definido como sendo um estímulo para a organização e transação pode ser definida como sendo a tarefa que deve ser executada como resultado imediato de um evento.

Os procedimentos descritos acima tem como objetivo final a estruturação dos diversos arquivos necessários ao atendimento das múltiplas aplicações e que constituirão o Banco de Dados do Sistema. Durante o desenvolvimento destes procedimentos deverão ser registradas para cada transação a sua frequência de uso, tempo de resposta necessário, utilidade e também uma estimativa do número de entradas e saídas (pares de mensagens do tipo diálogo) necessários para a transação. Em relação às entidades deverão ser registradas as ocorrências, taxa de crescimento, propriedades de acessos e domínios, formas de consultas e atualizações, frequências, volume de informações. E finalmente em relação aos atributos informações adicionais a respeito do seu emprego, criação, atualização e deleção.

Entretanto para ajudar o processo de seleção do tipo de arquitetura a ser utilizada informações adicionais serão necessárias. Estas informações são concernentes à localização das transações e à localização dos dados que aquelas transações necessitam.

#### 4.3.1.2 - Análise dos Software

Em um projeto de Sistema de Processamento Distribuído, a análise do Software tem uma grande importância para o desempenho de todo o sistema. Isto porque é exatamente neste contexto que nos defrontamos com os problemas prático existentes. YEH<sup>88</sup> conclui nesse seu trabalho que na verdade são os problemas práticos que oferecem as maiores dificuldades em um projeto de Sistemas de Processamentos Distribuídos e não aqueles problemas que necessitam de um tratamento matemático como se poderia pensar. Para

ele o mais difícil problema em um projeto de sistema e aquele concernente dã obtenção de dados sobre o proposto modelo de tráfego tal que as entradas para o algoritmo de solução, sejam representativos da realidade.

Consideramos nesta metodologia a necessidade de uma anãlise dos software sobre dois aspectos principais:

- Software de Apoio;
- Software de Aplicação.

#### 4.3.1.2.1 - Software de Apoio

Na anãlise deste grupo deve ser dada uma atenção nao sã para o conjunto de programas necessários ao sistema operacional de um específico computador, mas também para um software que permita uma confiável cooperação dos processos distribuídos.

Em um sistema distribuído podemos distinguir quatro componentes de software de apoio.

- O primeiro componente é o Sistema Operacional de cada computador;
- O segundo componente é o software de comunicação disponível que permita a troca de informações;
- O terceiro componente é um software para gerência do Banco de Dados LOCAL;
- O quarto componente é o software de controle responsãvel pelo manuseio dos aspectos da distribuição dos Dados tais como: onde os dados estão armazenados, problemas de integridade, problemas de consistência, etc, ou seja, o S.G.B.D.D. A figura 23 a seguir apresenta

a estrutura destes componentes do Software de Apoio.

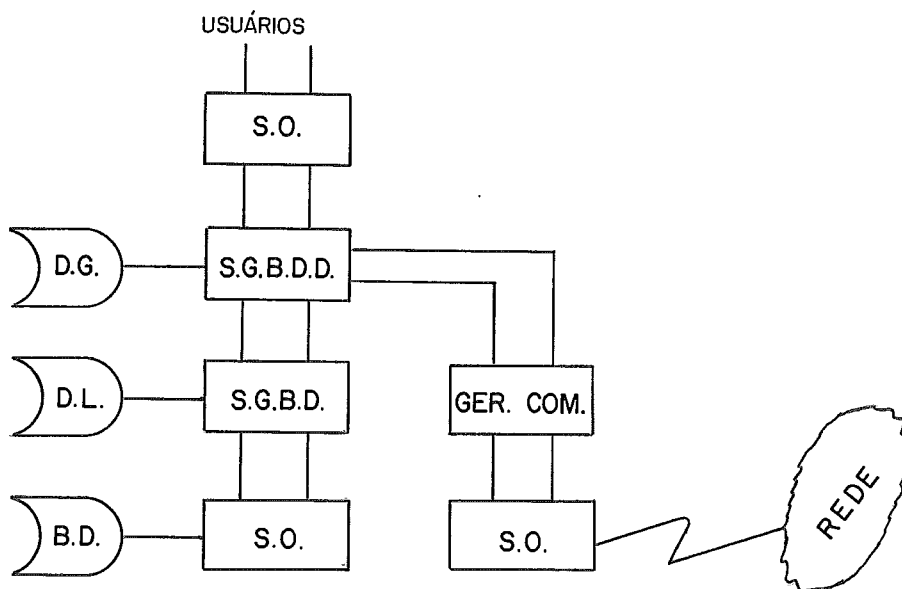


Figura 23

#### 4.3.1.2.2 - Software de Aplicação

Embora uma rede de computadores execute basicamente as funções de transmissão, processamento e armazenamento, a análise das aplicações deve ser executada sob o ponto de vista que combina essas operações com ênfase na informação. Desta forma podemos classificar as aplicações em dois grandes grupos: as aplicações locais e as aplicações distribuídas.

As aplicações locais são caracterizadas por produzirem os seus resultados com dados armazenados no próprio local onde foram submetidas ao processamento. E as aplicações distribuídas são aquelas que utilizam dados armazenados em diferentes locais para produzirem o resultado de uma função global. Neste segundo caso uma boa estratégia de acesso ao dado deve ser escolhida no projeto destas aplicações. As referências HEVNER <sup>48</sup>, BRACHI <sup>9</sup>, WONG <sup>87</sup>, CELLARY <sup>16</sup>, BERSTEIN <sup>7</sup> apresentam diversas alter

nativas para essas estratégias.

Dentro deste tipo de enfoque SCHERR <sup>75</sup> acha necessário considerar as estruturas das aplicações, no que diz respeito à forma como elas podem ser divididas, de maneira a minimizar as comunicações e permitir um balanceamento do processamento entre os diversos locais do Sistema. Mostra ainda que é necessário identificar pontos no fluxo dos programas de aplicação e seus associados dados onde o nível de comunicação seja baixo.

Um outro aspecto a considerar nesta análise é o fato de que as diversas aplicações não requerem as mesmas características de uma rede. Uma aplicação pode requerer um conjunto de características e outra aplicação pode exigir um outro conjunto de características da rede. LICKLIDER e VEZZA <sup>61</sup> apresentam diversos requisitos que uma rede de computadores deve possuir para atender as mais diversas aplicações e os seus respectivos graus de importância numa escala de 0 a 100. Baseado nestes requisitos de LICKLIDER e VEZZA <sup>61</sup> selecionamos os que consideramos mais importantes no contexto desta metodologia e relacionamos na figura 24 a seguir.



A alta taxa de transmissão não é um fator vital a maioria das aplicações em virtude das taxas de transmissão localizadas na faixa de 1.000 a 10.000 BPS satisfazerem a maior parte das aplicações. Entretanto esta característica é muito necessária a procedimentos interativos, pois permitirá um melhor desempenho do Sistema em termos de tempo de resposta.

Seguem-se na escala de importância a conectividade da rede, a privacidade, a prioridade dos serviços, a necessidade de um pequeno atraso e uniforme e finalmente a possibilidade de difusão.

A análise destes itens sugere a seleção de uma mais eficiente rede de comunicação para servir a uma específica aplicação ou um conjunto de aplicações.

#### 4.3.1.3 - Análise das Localidades

Nesta análise as empresas enfrentam um grande número de alternativas de projeto de um sistema computacional, e esta variedade aumenta em número e complexidade quando passamos de um único computador que serve a múltiplos usuários, para diversos computadores que servem a diversos usuários.

A parte mais delicada deste problema é, em geral, a definição do conjunto de locais viáveis, que devem traduzir convenientemente as possibilidades e permitir a enumeração de todas as decisões admissíveis. Isto porque normalmente o número de setores envolvidos em um sistema é muito maior do que o número de nodos necessários à distribuição da rede.

Entretanto algumas regras podem ser estabelecidas no sentido de orientar essa decisão.

- A primeira delas nos diz que a localização de um computador deve, se possível, ficar mais próximo do usuário que possui o maior volume de processamento;
- A segunda regra nos indica que devemos, sempre que possível, agrupar um conjunto de setores candidatos que possuam interesses comuns, de forma a constituir um local que dotado de um processador possa atender os seus interesses e o de todos. DAVENPORT<sup>30</sup> introduz nesse estudo sobre localidades um conceito denominado "Communities of Interest", onde um processador localizado em um local satisfaça todas as necessidades dele mesmo e dos outros vizinhos. Estas comunidades de interesses comuns são determinadas examinando os tipos de transações e seus volumes de processamento em cada local;
- Custos de instalações das linhas de comunicações;
- Frequência de consultas e atualizações.

Esta fase da Distribuição Lógica tem como objetivo final a elaboração de um conjunto de matrizes e vetores que retratarão as necessidades comuns de informações, os recursos disponíveis, os custos estimativos de comunicação e armazenamento e o interrelacionamento existente entre os diversos locais e aplicações com as especificações do conteúdo da Base de Dados. Em síntese constitui-se de uma análise funcional de todos os fatores que con



duzirão ao fluxo geral da rede do Sistema Distribuído, ou seja à definição dos locais, em termos das aplicações e seus dados, que satisfarão as necessidades exigidas pelo fluxo de informações da empresa.

A primeira matriz deste conjunto que podemos individualizar é a Matriz Arquivos-Transações. Esta matriz retrata as associações existentes entre as transações e os respectivos arquivos portadores dos dados necessários a execução destas transações, como mostra a figura 25 abaixo.

ARQUIVOS	TRANSAÇÕES						
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	.....	T <sub>n-2</sub>	T <sub>n-1</sub>	T <sub>n</sub>
ARQUIVO-A	X		X	.....			X
ARQUIVO-C	X	X		.....			X
ARQUIVO-D				.....	X	X	
.				.....			
.				.....			
.				.....			
.				.....			
ARQUIVO-X	X		X	.....	X		
ARQUIVO-Y				.....	X	X	X
ARQUIVO-Z		X		.....	X		X

Figura 25

A segunda matriz do conjunto, denominada de Matriz Arquivos-Locais, indica os diversos locais do Sistema onde serão utilizados os arquivos. Como consequência desta reprodução podemos identificar a existência de dispersão e concentração na geração dos dados como mostra a figura 26 abaixo.

ARQUIVOS	L O C A I S						
	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	.....	N <sub>n-1</sub>	N <sub>n</sub>
ARQUIVO-A	X			X	.....	X	X
ARQUIVO-B	X			X	.....	X	X
ARQUIVO-C	X	X	X		.....	X	X
.					.....		
.					.....		
.					.....		
ARQUIVO-Z	X		X	X	.....		X

Figura 26

A terceira matriz deste conjunto retrata os locais onde são executadas as diversas aplicações, bem como a superposição de execução de atividades, o que poderá conduzir a replicação de arquivos na rede. A esta matriz denominamos de matriz Aplicações-Locais como mostra a figura 27 a seguir.

APLICAÇÕES	L O C A I S					
	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	.....	N <sub>n-1</sub>	N <sub>n</sub>
APLICAÇÃO-A	X			.....	X	X
APLICAÇÃO-B		X		.....	X	
APLICAÇÃO-C		X	X	.....	X	X
.				.....		
.				.....		
.				.....		
.				.....		
APLICAÇÃO-Z	X			.....	X	

Figura 27

E finalmente a quarta matriz denominada Matriz de Transações-Aplicações indica a vinculação existente entre as transações e as aplicações do Sistema. A figura 28 abaixo mostra este tipo de matriz.

APLICAÇÕES	T R A N S A Ç Õ E S					
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	.....	T <sub>n-1</sub>	T <sub>n</sub>
APLICAÇÃO-A	X		X	.....	X	
APLICAÇÃO-B		X		.....		X
APLICAÇÃO-C	X		X	.....	X	
.				.....		
.				.....		
.				.....		
.				.....		
APLICAÇÃO-Z	X			.....	X	X

Figura 28

Com base neste conjunto de matrizes que identificam as áreas de aplicações com necessidades comuns de informações, os locais responsáveis pela entrada de dados e suas respectivas atualizações e os diversos arquivos contendo os dados úteis à execução das diversas aplicações da organização, torna-se possível elaborar os componentes que servirão de entrada para o método de designação de arquivos que será descrito no capítulo seguinte. Estes componentes são:

- Uma matriz que retrate a frequência de acesso para realizar consultas de transações para os arquivos;
- Uma matriz que retrate a frequência de acesso para realizar atualizações de transações para os arquivos;
- Uma matriz de frequência de uso das transações em cada local.

As figuras 29 e 30 abaixo mostram os modelos (exemplos) destas matrizes, que são elementos básicos e fundamentais para o método de alocação de arquivos, proposto nesta tese.

FREQÜÊNCIA DE ACESSO CONSULTA/ATUALIZAÇÕES

ARQUIVOS	TRANSAÇÕES				
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	.....	T <sub>n</sub>
ARQUIVO-A	f <sub>A,1</sub>	f <sub>A,2</sub>	f <sub>A,3</sub>	.....	f <sub>A,N</sub>
ARQUIVO-B	f <sub>B,1</sub>	f <sub>B,2</sub>	f <sub>B,3</sub>	.....	f <sub>B,N</sub>
ARQUIVO-C	f <sub>C,1</sub>	f <sub>C,2</sub>	f <sub>C,3</sub>	.....	f <sub>C,N</sub>
.	.	.	.	.....	.
.	.	.	.	.....	.
.	.	.	.	.....	.
ARQUIVO-Z	f <sub>Z,1</sub>	f <sub>Z,2</sub>	f <sub>Z,3</sub>	.....	f <sub>Z,N</sub>

Figura 29

FREQUÊNCIA DE USO DAS TRANSAÇÕES

LOCAIS	TRANSAÇÕES					
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	.....	T <sub>n</sub>
LOCAL-A	$f_{A1}$	$f_{A2}$	$f_{A3}$	$f_{A4}$	.....	$f_{AN}$
LOCAL-B	$f_{B1}$	$f_{B2}$	$f_{B3}$	$f_{B4}$	.....	$f_{BN}$
LOCAL-C	$f_{C1}$	$f_{C2}$	$f_{C3}$	$f_{C4}$	.....	$f_{CN}$
LOCAL-D	$f_{D1}$	$f_{D2}$	$f_{D3}$	$f_{D4}$	.....	$f_{DN}$
.	.	.	.	.	.....	.
.	.	.	.	.	.....	.
.	.	.	.	.	.....	.
.	.	.	.	.	.....	.
LOCAL-Z	$f_{Z1}$	$f_{Z2}$	$f_{Z3}$	$f_{Z4}$	.....	$f_{ZN}$

Figura 30

## CAPÍTULO V

### DESCRIÇÃO E FORMULAÇÃO DO MÉTODO DE ALOCAÇÃO DE ARQUIVOS

#### 5.1 - DESCRIÇÃO DO MODELO DE ALOCAÇÃO

Um Sistema de Processamento de Dados Distribuídos pode ser visualizado segundo um Grafo Linear dado por  $G=(N,E,Z)$  onde  $N$  é um conjunto não vazio de  $n$  nodos ou vértices,  $E$  é um conjunto de arcos ou arestas e  $Z$  é uma função de otimização.

Os nodos de tal Grafo representam os locais do Sistema onde serão designados equipamentos, dados, software e pessoal. O conjunto de arcos representam os canais de comunicações, entre cada par de nodos  $n_i$  e  $n_j$ , por onde circulam as informações do Sistema. E finalmente a função de otimização que representa um processo, não em busca da solução ótima em termos absolutos para o problema de alocação de arquivos, mais um método sistemático em busca do melhor prático, pois já está comprovado que em muitas situações formuladas, a procura de uma solução ótima torna-se inviável.

Se a comunicação entre os nodos  $n_i$  e  $n_j$  for executada por linhas de transmissão do tipo Half ou Full-Duplex, corresponderá a um arco não orientado  $[n_i, n_j]$ . Por outro lado uma transmissão do tipo Simplex, corresponderá a um arco orientado  $(n_i, n_j)$ . Entretanto nesta formulação serão considerados apenas os arcos do tipo não orientado.

Uma outra importante propriedade que o Sistema deve possuir é permitir que uma informação possa ir de um ponto qualquer a outro ponto qualquer da rede. Isto é, se um nodo qualquer  $x_j$  pertence ao conjunto de vértices adjacentes a um nodo  $x_i$  então  $x_j$  pode se comunicar diretamente com  $x_i$ ; se isso não acontecer, devemos ter, pelo menos um caminho  $(x_i, \dots, x_j)$  entre dois vértices qualquer.

Cada nodo do sistema consiste de uma estação de trabalho cuja principal função é participar direta ou indiretamente no processamento de todas as transações submetidas ao Sistema e que atinjam o nodo, seja através da entrada para comunicação com as transações locais, seja através da entrada para comunicação com os outros nodos. Um nodo é constituído de:

- Computadores, podendo ser de qualquer tipo ou porte;
- Terminais, podendo ser de qualquer tipo;
- Dispositivos para armazenar arquivos de dados e programas;
- Software de apoio;
- Pessoal.

A figura 31 a seguir ilustra a estrutura de funcionamento de um particular nodo.

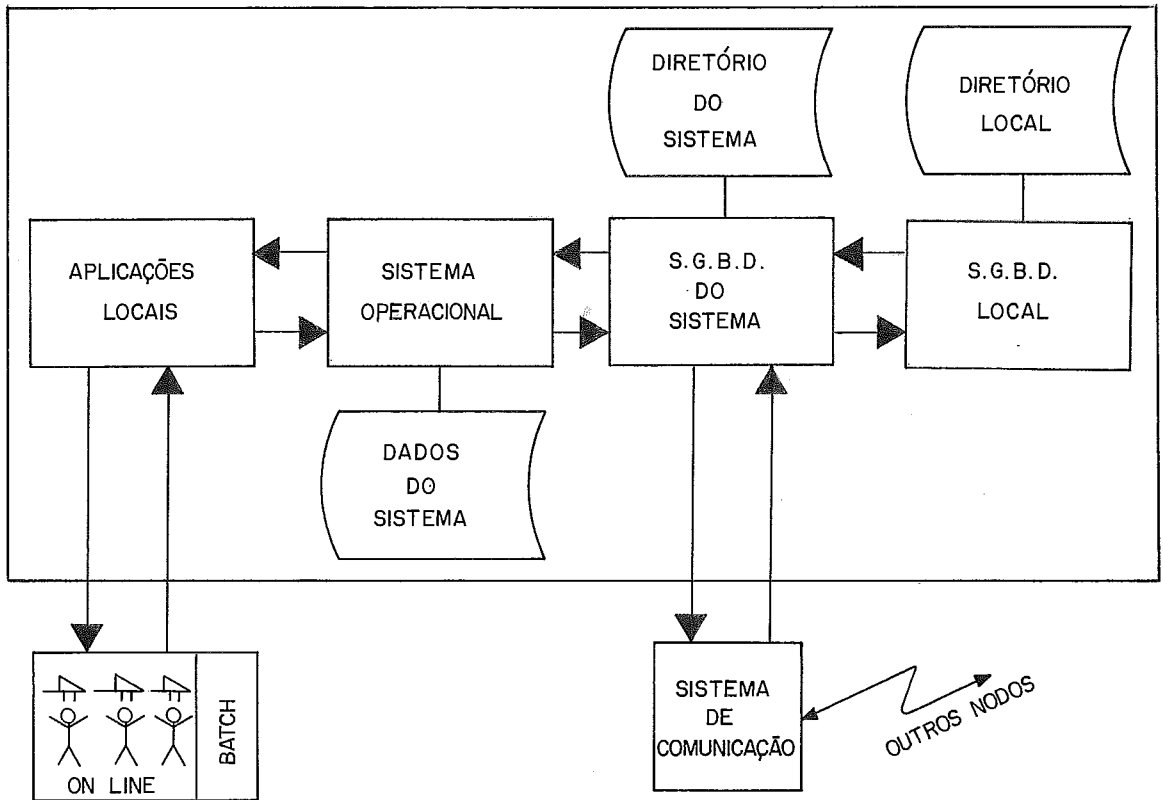


Figura 31

Quando um usuário local deseja consultar ou atualizar uma informação em um dos arquivos do Sistema, inicialmente é verificado se esta informação está definida no "SCHEMA" local, através do SGBD local. Se estiver tudo se passará como se fosse em um ambiente de um Sistema de Banco de Dados convencional, ou seja, o S.G.B.D. é habilitado através do "SCHEMA" definido pelo usuário a localizar os dados e transferí-los para o processo que o solicitou. Em caso contrário, ou seja, se a informação não está definida no "SCHEMA" local, então o controle passará ao S.G.B.D. da rede, que determinará para que nodo da rede deverá ser enviado o pedido. Quando o pedido for atendido a informação será devolvida ao processo que a solicitou.



O problema de Alocação de Arquivos, que nesta metodologia é considerada como uma fase da elaboração do projeto de um Sistema de Processamento de Dados Distribuídos, situa-se no enfoque proposto pelo trabalho do grupo CERI, MARTELLA e PELAGATTI <sup>15</sup>, onde o método de alocação deve ser compatível com o mundo real da organização, integrando parâmetros que podem ser estimados na vida prática. Assim o objetivo geral do método é determinar uma alocação de arquivos em uma rede de computadores que otimize os custos totais de armazenamento, função da replicação dos arquivos, e os custos totais de comunicações, função do volume de dados transmitidos, em razão da necessidade de consultas e atualizações.

## 5.2 - FATORES CONSIDERADOS NO MÉTODO DE ALOCAÇÃO

Para solucionar o problema de alocação de arquivos a necessidade de definir alguns aspectos quantitativos de diversos fatores é mais importante que a definição lógica desses mesmos fatores. Neste método apresentado serão considerados os seguintes fatores:

- Tamanho do arquivo;
- Capacidade de armazenamento de cada nodo;
- Custos de armazenamento;
- Frequências de acessos;
- Distâncias entre os nodos;
- Custos de comunicação.

### 5.2.1 - Tamanho do Arquivo

Este fator representa o tamanho máximo do arquivo expresso em uma unidade de arquivo qualquer tais como: bytes, registros, blocos, etc. Este fator será representado pela variável  $T_k$  onde  $k$  é um índice de referência de um particular arquivo do sistema tal que  $1 \leq k \leq F$ , onde,  $F$  é o conjunto de todos os arquivos do sistema. Neste método adotaremos como unidade padrão de arquivo, para expressar o seu tamanho, o BYTE.

### 5.2.2 - Capacidade de Armazenamento

Este fator representa a capacidade máxima de unidade de armazenamento disponíveis em cada nodo da rede. Este fator será representado pela variável  $V_i$  onde  $i$  é um índice de referência de um determinado nodo da rede tal que  $1 \leq i \leq N$  onde  $N$  é o conjunto de todos os nodos da rede. Utilizaremos neste método o BYTE como unidade padrão de armazenamento.

### 5.2.3 - Custos de Armazenamento

Este fator indica o custo que será dispendido para armazenar uma unidade de arquivo (BYTE) em um nodo da rede. Este fator será representado pela variável  $C_{ki}$ . O custo total necessário para armazenar um arquivo inteiro  $k$  no nodo  $i$  será dado por:

$$\text{CUSTO-ARMAZENAR}_{ki} = T_k \cdot C_{ki}$$

e o custo total de armazenamento do Sistema será obtido fazendo-se:

$$\text{CUSTO-ARMAZENAMENTO} = \sum_{k=1}^F \sum_{i=1}^N T_k \cdot C_{ki} \cdot Y_{ki}$$

onde: 
$$\begin{cases} Y_{ki} = 1 & - \text{ se o arquivo } \underline{k} \text{ está armazenado em } \underline{i} \\ Y_{ki} = 0 & - \text{ se o arquivo } \underline{k} \text{ não está armazenado em } \underline{i} \end{cases}$$

#### 5.2.4 - Freqüências de Acessos

Este fator indica a taxa de acesso de um determinado no do i para um arquivo k. Assim como em CASSEY<sup>12</sup>, AKOKA<sup>1</sup> essa taxa é diferenciada como freqüência de acesso para realizar uma consulta e freqüência de acesso para realizar uma atualização. Esses valores são obtidos a partir dos resultados, catalogados nas matrizes das figuras 29 e 30, e que foram determinados na fase do projeito lógico do sistema onde são identificados:

- Os acessos relativos às consultas e atualizações executados pelas TRANSAÇÕES em arquivos (  $f_{tk}^c$  e  $f_{tk}^a$  ). Uma transação neste método representa um programa compilado cujo objetivo é acessar os arquivos no sentido de processar uma ou mais mensagens de consultas ou atualizações. Essas transações são também identifica-das e definidas na fase do projeto lógico do sistema, bem como a estrutura e formação dos arquivos. As consultas de caráter eventual não são consideradas nes-tas taxas.
- Os acessos das TRANSAÇÕES em cada local do Sistema, ou seja, o número de vezes que cada transação é execu-tada em cada nodo (  $f_{iN}$  ).

A partir destes valores são construídas as freqüências de acessos dos nodos para os arquivos que são representadas pelas seguintes variáveis:

- $f_{ik}^c$  para representar a frequência de acesso de um nodo i para consultar um arquivo k.
- $f_{ik}^a$  para representar a frequência de acesso de um nodo i para atualizar um arquivo k.

Estes valores são dados em número de MENSAGENS/MÊS.

#### 5.2.5 - Distância entre os nodos

Este fator indica a distância existente entre cada par de nodos i e j. A variável representativa deste fator será indicada por  $D_{ij}$  em uma unidade de distância qualquer. Neste método adotaremos o quilômetro (KM) como unidade de distância.

#### 5.2.6 - Custos de Comunicações

Este fator é indicado pela variável  $C_{ij}$  e representa o custo a ser pago pela transmissão de uma unidade de informação entre os pontos i e j. Estes valores são quantificados em função de:

- Capacidade dos canais de comunicações;
- Distância entre os nodos;
- Volume de dados transmitidos.

Antes de apresentarmos a matriz de custos de comunicações utilizada nesta tese, analisaremos primeiramente as três principais formas de conexão entre computadores oferecida pela EMBRA TEL, afim de atender os vários tipos de aplicações de transmissão de dados, com suas respectivas tarifas.

A primeira forma de comunicação denominada "Comunicação de Dados a Distância - CDD", é uma solução adequada para pequenos ou médios tráfego de dados, ou ainda quando se tratar de uso eventual. A figura 32 abaixo ilustra um esquema simplificado deste tipo de comunicação.

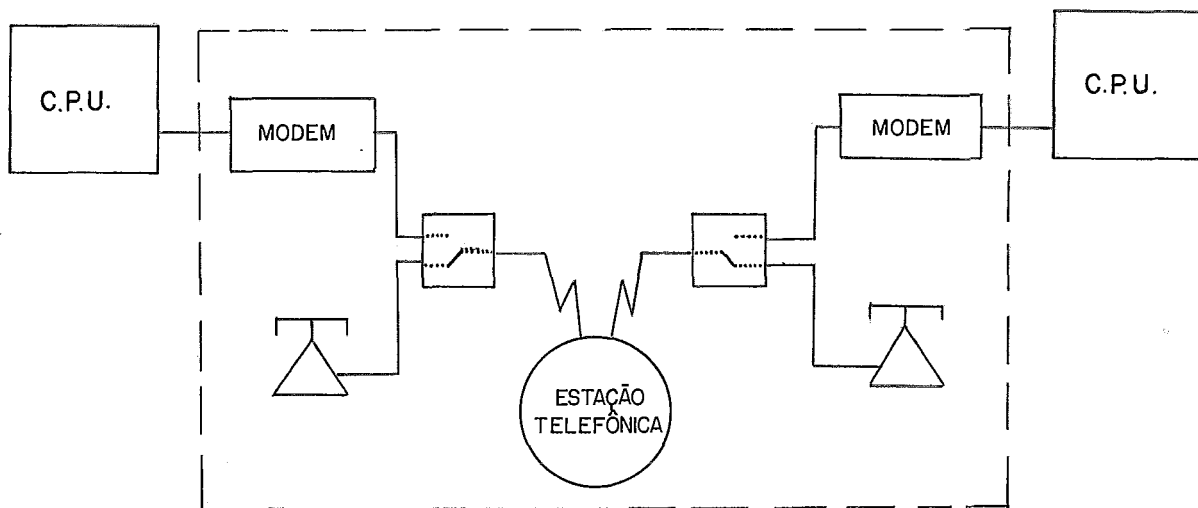


Figura 32

Neste tipo de ligação, a comunicação de dados está, na realidade, sendo executada com os recursos existentes do Sistema de Telefonia normal, portanto sujeito a todo tipo de "ruído" que porventura venha ocorrer na linha.

Um outro aspecto a considerar neste tipo de ligação é que por não ser um serviço específico para comunicação de dados, para que seja possível estabelecer uma ligação entre dois pontos, faz-se necessário inicialmente executar uma ligação telefônica de um ponto para o outro, em seguida acionar uma chave seletora para o MODEM a fim de que então possa ser dado início a transmissão das informações.

O custo deste tipo de ligação é o mesmo de uma chamada telefônica normal, portanto esta tarifa é quantificada em função do tempo que durar a ligação estabelecida. A tabela da figura-33 abaixo mostra as tarifas relativas a este tipo de ligação válidas até outubro de 1983.

# Telefonia

Vigência: 05/07/83

Horários de Tarifação	Feriados Nacionais* Corpus Christi Terça-Feira - Carnaval	Dom.	Seg.	Ter.	Qua.	Qui.	Sex.	Sáb.
		8.00 às 14.00	Tarifa Normal					
14.00 às 20.00								
20.00 às 24.00 e 00.00 às 08.00	Tarifa Reduzida							

\*Feriados Nacionais para efeitos de cobrança de Tarifas Reduzidas em âmbito nacional: 01 de Janeiro, 21 de Abril, 01 de Maio, 07 de Setembro, 12 de Outubro, 02 de Novembro, 15 de Novembro, 25 de Dezembro, Sexta-Feira da Paixão.

## Serviço Automático (DDD) e Serviço Semi-Automático (DDO)

Degrau	Distância Geodésica em Km	Tarifa por Minuto (Cr\$)	
		Telefone a Telefone	
		Normal	Reduzida
02	Até 50	56,98	28,49
03	Acima de 50 Até 100	102,42	51,21
04	Acima de 100 até 200	140,48	70,24
05	Acima de 200 até 300	166,03	83,02
06	Acima de 300 até 500	204,34	102,17
07	Acima de 500 até 700	212,20	106,10
08	Acima de 700 até 1.000	219,81	109,91
09	Acima de 1.000 até 1.500	232,34	116,17
10	Acima de 1.500	245,60	122,80

## Serviço Manual e Semi-Automático (ODD)

Degrau	3 Minutos Iniciais (Cr\$)				Minuto Adicional (Cr\$)	
	Telefone à Telefone		Especial		Telefone/ Telefone Especial	
	Normal	Reduzida	Normal	Reduzida	Normal	Reduzida
02	170,94	85,47	256,41	128,21	56,98	28,49
03	307,26	153,63	460,89	230,45	102,42	51,21
04	421,44	210,72	632,16	316,08	140,48	70,24
05	498,09	249,06	747,14	373,59	166,03	83,02
06	613,02	306,51	919,53	459,77	204,34	102,17
07	636,60	318,30	954,90	477,45	212,20	106,10
08	659,43	329,73	989,15	494,60	219,81	109,91
09	697,02	348,51	1.045,53	522,77	232,34	116,17
10	736,80	368,40	1.105,20	552,60	245,60	122,80

As tarifas reduzidas dos degraus 2 e 3 são válidas somente para o Estado do Paraná. Até 31/03/1985.  
O percentual de incidência da sobre-larita do FNT é de 30%.

A adoção deste tipo de ligação e seus respectivos custos torna-se inadequada aos objetivos propostos neste método, em virtude desta metodologia se propôr a buscar a minimização dos custos em função do volume de dados transmitidos e não em função do tempo de utilização da linha de comunicação.

A segunda opção que analisaremos é a Comunicação de Dados em Circuitos Direto e Exclusivo - TRANSDATA. Este tipo de serviço oferecido pela EMBRATEL é um serviço específico para comunicação de dados para um ou mais pontos e portanto apresenta uma taxa de erro menor que  $5 \times 10^{-5}$ . Este tipo de serviço apresenta uma disponibilidade da ordem de 95% para circuitos urbanos e 99% para circuitos interurbanos, não requerendo nenhum procedimento especial para que seja estabelecida uma comunicação entre dois pontos. Um esquema simplificado deste tipo de ligação está ilustrado na figura 34 abaixo.

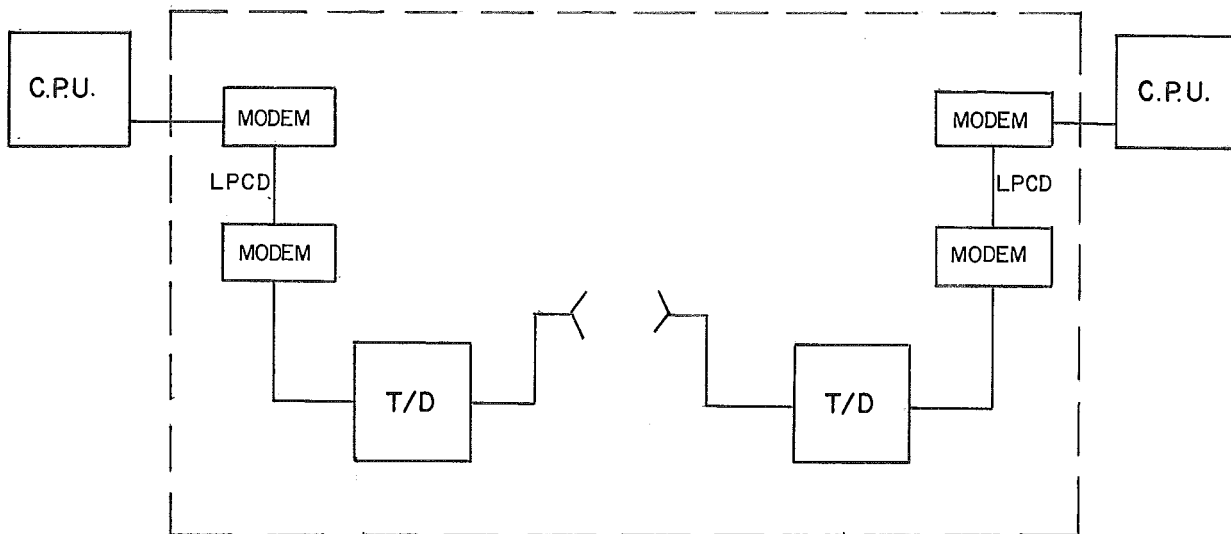


Figura 34



A TRANSDATA tem suas tarifas calculadas em função da distância e da velocidade de transmissão da linha. Os custos independem portanto do volume de informações transmitidas e do tempo de utilização da linha. Isto porque ela permanece disponível para uso durante 24 horas por dia a um custo fixo. A tabela da figura 35 abaixo mostra as tarifas relativas a esta modalidade de comunicação.

## Serviços Especializados de Comunicação de Dados

Vigência: 05/07/83

Comunicação de Dados - Transdata								
	Degrau	Distância Geodésica em Km	Utilização	Tarifa Mensal - Cr\$				
				Até 300 BPS	Até 1.200 BPS	2.400 BPS	4.800 BPS	9.600 BPS
Permanente	D1	Urbano Digital	Mensal	---	* 37.822,00	64.102,00	64.102,00	64.102,00
		Urbano Analógico		41.506,00	63.610,00	131.396,00	357.102,00	608.597,00
	D2	Até 50		97.749,00	182.420,00	293.185,00	429.984,00	651.516,00
	D3	50 a 100		114.171,00	213.066,00	342.440,00	502.222,00	760.970,00
	D4	100 a 300		136.848,00	255.387,00	410.459,00	601.978,00	912.122,00
	D5	300 a 700		273.697,00	510.775,00	820.918,00	1.203.956,00	1.824.243,00
	D6	700 a 1500		342.121,00	638.468,00	1.026.148,00	1.504.945,00	2.280.304,00
	D7	Acima de 1500		390.995,00	729.678,00	1.172.740,00	1.719.937,00	2.606.062,00
	Degrau	Distância Geodésica em Km	Utilização	Valores Tarifários - Cr\$				
				Até 300 BPS	Até 1.200 BPS	2.400 BPS	4.800 BPS	9.600 BPS
Temporário	D1	Urbano Digital	Mínimo 3 dias	---	* 5.673,00	9.615,00	9.615,00	9.615,00
			Por dia	---	* 1.891,00	3.205,00	3.205,00	3.205,00
		Urbano Analógico	Mínimo 3 dias	6.225,00	9.543,00	19.710,00	53.565,00	91.290,00
			Por dia	2.075,00	3.181,00	6.570,00	17.855,00	30.430,00
	D2	Até 50	Mínimo 3 dias	14.661,00	27.363,00	43.977,00	64.497,00	97.728,00
			Por dia	4.887,00	9.121,00	14.659,00	21.499,00	32.576,00
	D3	50 a 100	Mínimo 3 dias	17.127,00	31.959,00	51.366,00	75.333,00	114.147,00
			Por dia	5.709,00	10.653,00	17.122,00	25.111,00	38.049,00
	D4	100 a 300	Mínimo 3 dias	20.526,00	38.307,00	61.569,00	90.297,00	136.818,00
			Por dia	6.842,00	12.769,00	20.523,00	30.099,00	45.606,00
	D5	300 a 700	Mínimo 3 dias	41.055,00	76.617,00	123.138,00	180.594,00	273.636,00
			Por dia	13.685,00	25.539,00	41.046,00	60.198,00	91.212,00
	D6	700 a 1500	Mínimo 3 dias	51.318,00	95.769,00	153.921,00	225.741,00	342.045,00
			Por dia	17.106,00	31.923,00	51.307,00	75.247,00	114.015,00
	D7	Acima de 1500	Mínimo 3 dias	58.650,00	109.452,00	175.911,00	257.991,00	390.909,00
			Por dia	19.550,00	36.484,00	58.637,00	85.997,00	130.303,00

Observação: Para o Serviço Interurbano de comunicação de dados com coordenação ou alternância de voz e dados Aluguel Permanente e Temporário, aplicam-se os valores para a velocidade de 9.600 bps constantes das tabelas anteriores.

\*Síncrono

Aplicar a Sobretaxa de 30% do FNT sobre os Valores Acima

De forma similar a modalidade de comunicação CDD, a TRANSDATA apresenta a mesma inadequação em termos de utilização dos valores de suas tarifas, para compor as variáveis de custos de comunicações para o método de alocação de arquivos apresentado nesta tese.

A terceira e última modalidade de comunicação objeto desta análise é a Rede Nacional de Comutação de Pacotes - RENPAC. Embora este tipo de serviço ainda não esteja disponível comercialmente ao público em geral (previsão para o 2º semestre de 1984) a EMBRATEL já iniciou um programa de divulgação das características e serviços oferecidos pela RENPAC, entre as quais encontramos:

- Uma rede construída a partir de microprocessadores combinados e programados para executar funções específicas de comunicações;
- Serviços de "chamada virtual";
- Serviços de "circuito virtual permanente";
- Circuitos dedicados ou de ligações estabelecidas;
- Protocolo X-25 da CCITT;
- Ligações pela rede pública de telefonia ou pela rede nacional de telex;
- Velocidades síncronas de 2400, 4800 e 9600 BPS;
- Acessos assíncronos em modo character até 1200 BPS;
- Etc.

Com esta modalidade de comunicação a EMBRATEL, através da RENPAC, introduzirá também um novo conceito de custos neste ti

po de tarifa, que estará vinculado ao volume de informações transmitidas. A figura 36 abaixo ilustra este tipo de comunicação.

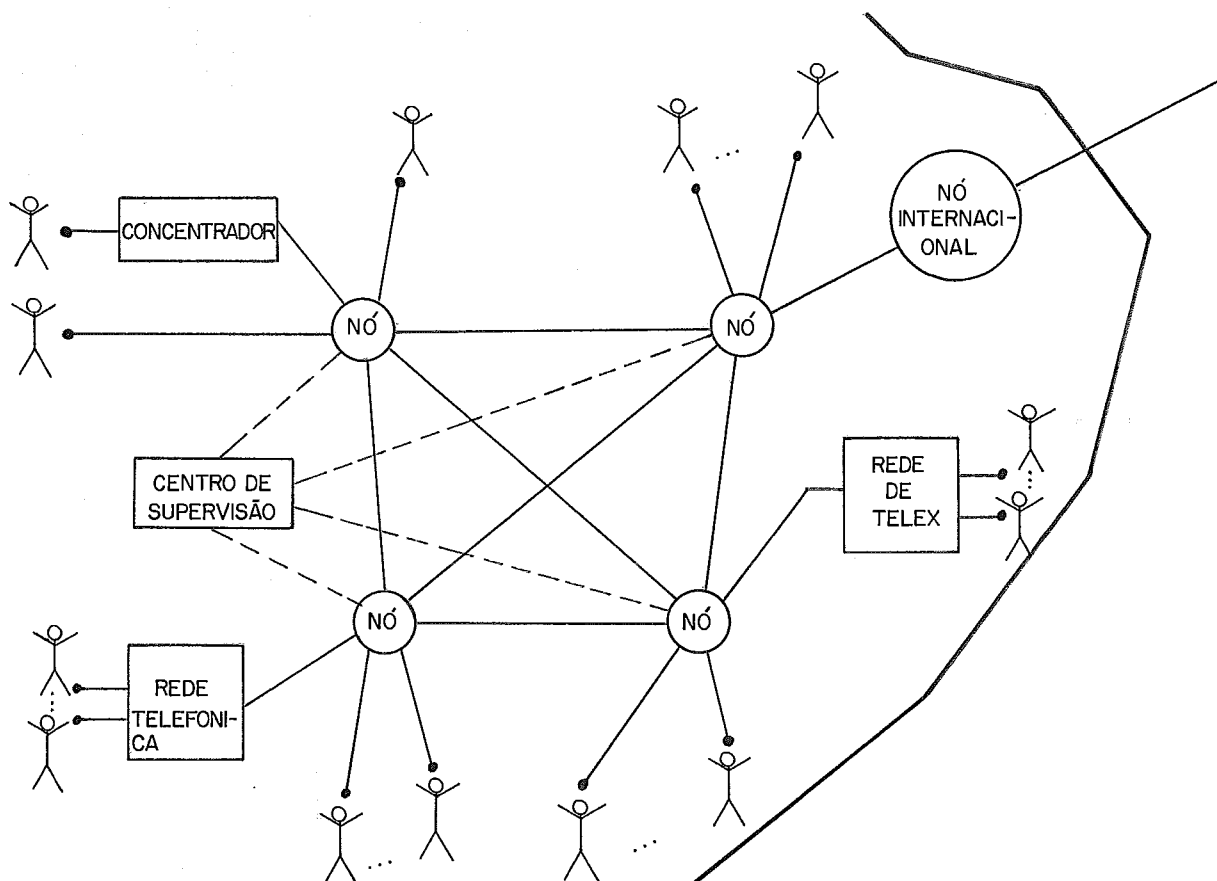


Figura 36

Muito embora este tipo de comunicação e seu conceito de custos sejam adequados aos objetivos estabelecidos nesta tese, a indisponibilidade destas tarifas ao público, impossibilita a adoção dos seus reais valores no método de alocação de arquivos apresentado nesta tese.

Desta forma adotaremos a tabela de custos de comunicação especificada na figura 37 a seguir cujos valores são funções da distância entre os nodos, da capacidade das linhas de comunicações e do volume de dados transmitidos. Esta tabela foi construída a partir dos valores da tabela TRANSDATA sob as seguintes condições:

- 1) Tarifa 300 bps =  $1 / A(B.C.D.E) / F$
- 2) Tarifa 1200 bps = 1,866 da tarifa 300 bps
- 3) Tarifa 2400 bps = 1,607 da tarifa 1200 bps
- 4) Tarifa 4800 bps = 1,466 da tarifa 2400 bps
- 5) Tarifa 9600 bps = 1,515 da tarifa 9600 bps

onde:

A = desempenho relativo a 50%

B = mês de 20 dias

C = dia de 6 horas

D = 3600 segundos

E = 300 bps

F = tarifa TRANSDATA relativa a 300 bps

e:

A introdução das constantes (1,866), (1,607), (1,466) e (1,515) tem o objetivo de guardar as mesmas proporções das tarifas TRANSDATA, relativas respectivamente as velocidades 1200 bps, 2400 bps, 4800 bps e 9600 bps.

TABELA DE CUSTOS DE COMUNICAÇÃO POR BYTE TRANSMITIDO ( $\gamma_{ij}$ )

Velocidades Distâncias	300 bps	1200 bps	2400 bps	4800 bps	9600 bps
até 50	0,012	0,022	0,035	0,051	0,077
50 - 100	0,014	0,026	0,041	0,060	0,090
100 - 300	0,016	0,029	0,046	0,067	0,101
300 - 700	0,033	0,061	0,098	0,143	0,216
700 - 1500	0,042	0,078	0,125	0,183	0,277
acima 1500	0,048	0,089	0,143	0,209	0,316

Figura 37

O custo total de comunicação do Sistema será obtido fazendo-se:

CUSTO-DE-COMUNICAÇÃO =

$$\sum_{k=1}^F \sum_{j=1}^N \left( \sum_{l=1}^N (f_{ki}^c (L_c + L_c^p) + f_{ki}^a L_a) \gamma_{ij} \right) Y_{ki}$$

### 5.3 - CONDIÇÕES ESTABELECIDAS NO MÉTODO

As seguintes condições foram estabelecidas no método apresentado:

5.3.1 - As unidades de custos de acessos remotos das transações são consideradas como pertencentes ao nodo onde será localizado o arquivo.

5.3.2 - Os custos de armazenamento são considerados como pertencentes ao nodo onde está alocado o arquivo.

5.3.3 - A capacidade de armazenamento em cada nodo é considerada limitada. Esta restrição da capacidade de armazenamento em cada nodo é formulada da seguinte forma:

$$\sum_k y_{ki} T_k \leq V_i, \quad \forall i \mid 1 \leq i \leq N$$

5.3.4 - Assumimos que os usuários da rede tem acesso ao Sistema somente através de um dos nodos.

5.3.5 - Nós admitimos que toda transação que vai acessar um arquivo remoto tem uma rota definida.

5.3.6 - Nós admitimos que só pode existir arquivos em nodos que possuam computadores.

5.3.7 - Nós admitimos que a capacidade total de processamento necessária a todas as transações não exceda a capacidade de processamento em cada nodo da rede.

5.4 - FORMULAÇÃO DO MÉTODO DE ALOCAÇÃO

5.4.1 - Dados de Entrada

5.4.1.1 - Matriz Distância

Cada elemento desta matriz, representado pela variável  $D_{ij}$ , indica a distância existente entre cada par de nodos  $i$  e  $j$ . Os elementos simétricos desta matriz possui valores iguais, isto é  $D_{ij} = D_{ji}$ . Uma outra característica desta matriz é que para  $i=j$ , temos que  $D_{ij} = 0$ .

$(D_{ij})$		MATRIZ DISTÂNCIA				
$i \backslash j$	1	2	...	n-1	n	
1	$D_{1,1}$	$D_{1,2}$	...	$D_{1,n-1}$	$D_{1,n}$	
2	$D_{2,1}$	$D_{2,2}$	...	$D_{2,n-1}$	$D_{2,n}$	
.	.	.		.	.	
.	.	.		.	.	
.	.	.		.	.	
n-1	$D_{n-1,1}$	$D_{n-1,2}$	...	$D_{n-1,n-1}$	$D_{n-1,n}$	
n	$D_{n,1}$	$D_{n,2}$	...	$D_{n,n-1}$	$D_{n,n}$	

Figura 38

Cada elemento  $D_{ij}$  desta Matriz Distância é dado em quilômetros (km).



5.4.1.2 - Matriz Frequência de Consultas (Transações/Arquivos)

Cada elemento desta matriz representado pela variável  $f_{tk}^c$ , indica o número de vezes em um determinado limite de tempo, em que uma transação  $t$  acessa o arquivo  $k$ , com o objetivo de realizar uma simples função de consulta. Cada elemento da Matriz de Consultas (Transações/Arquivos) é dado em número de Transações/Mês.

$(f_{kt}^c)$ TRANSAÇÕES/ARQUIVOS					
$t \backslash k$	1	2	...	F-1	F
1	$f_{1,1}^c$	$f_{1,2}^c$	...	$f_{1,F-1}^c$	$f_{1,F}^c$
2	$f_{2,1}^c$	$f_{2,2}^c$	...	$f_{2,F-1}^c$	$f_{2,F}^c$
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
T-1	$f_{T-1,1}^c$	$f_{T-1,2}^c$	...	$f_{T-1,F-1}^c$	$f_{T-1,F}^c$
T	$f_{T,1}^c$	$f_{T,2}^c$	...	$f_{T,F-1}^c$	$f_{T,F}^c$

Figura 39

5.4.1.3 - Matriz Frequência de Atualizações (Transações/Arquivos)

Cada elemento desta matriz representado pela variável  $f_{tk}^a$ , indica o número de vezes em um determinado limite de tempo, em que uma transação  $t$  acessa o arquivo  $k$ , com o objetivo de executar uma simples função de atualização. Cada elemento desta matriz de Atualizações (Transações/Arquivos) é dado em número de Transações/Mês.

$(f_{kt}^a)$		TRANSAÇÕES / ARQUIVOS				
t \ k	1	2	...	F-1	F	
1	$f_{1,1}^a$	$f_{1,2}^a$	...	$f_{1,F-1}^a$	$f_{1,F}^a$	
2	$f_{2,1}^a$	$f_{2,2}^a$	...	$f_{2,F-1}^a$	$f_{2,F}^a$	
.	.	.		.	.	
.	.	.		.	.	
.	.	.		.	.	
T-1	$f_{T-1,1}^a$	$f_{T-1,2}^a$	...	$f_{T-1,F-1}^a$	$f_{T-1,F}^a$	
T	$f_{T,1}^a$	$f_{T,2}^a$	...	$f_{T,F-1}^a$	$f_{T,F}^a$	

Figura 40

5.4.1.4 - Matriz de Frequência Transações/Nodo

Cada elemento desta matriz representado pela variável  $f_{tn}$ , representa o número de vezes em um determinado limite de tempo, em que uma transação  $t$  é utilizada em um determinado nodo  $n$ . Esta frequência de Uso Transações/nodo é dada em um número de uso mensal.

$(f_{ti})$		TRANSAÇÕES / NODOS				
t \ i	1	2	...	N-1	N	
1	$f_{1,1}$	$f_{1,2}$	...	$f_{1,N-1}$	$f_{1,N}$	
2	$f_{2,1}$	$f_{2,2}$	...	$f_{2,N-1}$	$f_{2,N}$	
.	.	.		.	.	
.	.	.		.	.	
.	.	.		.	.	
T-1	$f_{T-1,1}$	$f_{T-1,2}$	...	$f_{T-1,N-1}$	$f_{T-1,N}$	
T	$f_{T,1}$	$f_{T,2}$	...	$f_{T,N-1}$	$f_{T,N}$	

Figura 41

5.4.1.5 - Matriz de Custos de Comunicações

Cada elemento desta matriz representado pela variável  $\delta_{ij}$ , indica o custo a ser dispendido pelo uso de uma linha de comunicação direta entre cada par de nodos  $i$  e  $j$ . Similarmente a Matriz Distância, os elementos simétricos da Matriz de Custos de Comunicação possui valores iguais  $P_{ij} = P_{ji}$  e também para  $i=j$  temos que  $P_{ij} = 0$ .

$(\delta_{ij})$		CUSTOS DE COMUNICAÇÃO				
		$i$	1	2	...	n-1
1	$\delta_{1,1}$	$\delta_{1,2}$	...	$\delta_{1,n-1}$	$\delta_{1,n}$	
2	$\delta_{2,1}$	$\delta_{2,2}$	...	$\delta_{2,n-1}$	$\delta_{2,n}$	
.	.	.		.	.	
.	.	.		.	.	
.	.	.		.	.	
n-1	$\delta_{n-1,1}$	$\delta_{n-1,2}$	...	$\delta_{n-1,n-1}$	$\delta_{n-1,n}$	
n	$\delta_{n,1}$	$\delta_{n,2}$	...	$\delta_{n,n-1}$	$\delta_{n,n}$	

Figura 42

Cada elemento  $P_{ij}$  representa uma taxa a ser paga por BYTE transmitido, de acordo com a Tabela definida no item 5.2.6.

5.4.1.6 - Tamanho dos arquivos (Dado em Megabytes).

5.4.1.7 - Capacidade de armazenamento em cada nodo (dado em Megabytes).

5.4.1.8 - Tamanho médio das mensagens de consultas (dado em Megabytes).

5.4.1.9 - Tamanho médio das mensagens de atualizações (dado em Bytes).

5.4.1.10 - Tamanho médio das mensagens de respostas às consultas - (dado em Bytes).

5.4.1.11 - Número de nodos (N).

5.4.1.12 - Número de Arquivos (K).

5.4.1.13 - Matriz de Direcionamento.

Esta Matriz mostra a rota a ser seguida por todas as mensagens que partem de um nodo  $i$  com destino a um nodo  $j$ .

$(\pi_{ij})$ MATRIZ DE DIRECIONAMENTO					
$i \backslash j$	1	2	...	n-1	n
1	1-1	1-2	...	1-(n-1)	1-n
2	2-1	2-2	...		2-n
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.
n-1	(n-1)-1	(n-1)-2	...	(n-1)-(n-1)	(n-1)-n
n	n-1	n-2	...	n-(n-1)	n-n

Figura 43

5.4.2 - OBJETIVOS A DETERMINAR

5.4.2.1 - Matriz de Alocação

Nesta Matriz de Alocação  $M \begin{bmatrix} y_{ki} \end{bmatrix}$ , cada um dos seus elementos, representado pela variável  $y_{ki}$ , indicará se um determinado arquivo  $k$  está ou não designado para ser armazenado no nodo  $i$ . Esta indicação se refletirá da seguinte forma:

- Se o elemento  $y_{ki} = 1$ , então o arquivo  $k$  está designado para o nodo  $i$ .
- Se o elemento  $y_{ki} = 0$ , então o arquivo  $k$  não está designado para o nodo  $i$ .

Esta designação deverá ser feita de tal forma que minimize os custos de comunicação e armazenamento. Isto é:

$$Z = \sum_{k=1}^F \sum_{j=1}^N \left( \sum_{l=1}^N (f_{ki}^c (L_c + L_c') + f_{ki}^a L_a) \delta_{ij} + T_k C_{kl} \right) y_{ki}$$

seja mínimo

$(y_{ki})$		MATRIZ DE ALOCAÇÃO				
k \ i	1	2	...	n-1	n	
1	$y_{1,1}$	$y_{1,2}$	...	$y_{1,n-1}$	$y_{1,n}$	
2	$y_{2,1}$	$y_{2,2}$	...	$y_{2,n-1}$	$y_{2,n}$	
.	.	.		.	.	
.	.	.		.	.	
.	.	.		.	.	
F-1	$y_{F-1,1}$	$y_{F-1,2}$	...	$y_{F-1,n-1}$	$y_{F-1,n}$	
F	$y_{F,1}$	$y_{F,2}$	...	$y_{F,n-1}$	$y_{F,n}$	

Figura 44

### 5.5 - CONCEITUAÇÃO DO MÉTODO DE SOLUÇÃO

A formulação do problema de alocação de arquivos proposto nesta tese é expressa da seguinte forma: Existem  $\underline{N}$  locais e  $\underline{F}$  arquivos; são considerados também os custos  $(E_{ki})$ , onde estão envolvidos diversos componentes de custos, correspondentes a todas as atribuições possíveis de arquivos e locais. Trata-se portanto de designar os arquivos a determinados locais de modo que os custos de armazenamentos e comunicações sejam mínimos. Esta solução proposta está dividida em duas formas de alocações distintas.

- Solução para o problema sem redundância de arquivo;
- Solução para o problema com redundância de arquivo.

#### 5.5.1 - Solução para Alocação Sem Redundância

Seja um Grafo bipartido  $G(M) = (V,E)$  onde o conjunto de vértices  $\underline{V}$  é representado pela união de dois conjuntos disjuntos, o conjunto  $\underline{F}$  e o conjunto  $\underline{N}$ . O conjunto de arcos  $\underline{E}$  consiste de todos os conjuntos de pares  $\{k,i\}$  tal que  $k \in \underline{F}$  e  $i \in \underline{N}$ . A figura 45 abaixo ilustra este tipo de grafo.

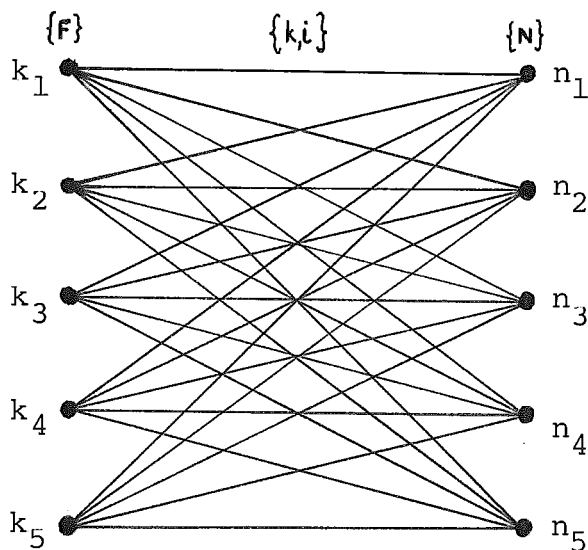


Figura 45

A este Grafo está associado uma Matriz  $M = [E_{ki}]$  tal que:

- As linhas  $\underline{k}$  de  $\underline{M}$  correspondem aos arquivos do Sistema a serem atribuídos aos nodos da rede tal que  $k \in F$ , onde  $F$  é o conjunto de todos os arquivos;
- As colunas  $\underline{i}$  de  $\underline{M}$  correspondem aos nodos da rede que receberão os arquivos tal que  $i \in N$  onde  $N$  é o conjunto de todos os nodos;
- Os elementos  $(E_{ki})$  da matriz  $\underline{M}$  correspondem aos custos do arco  $(k,i)$  tal que  $0 \leq E \leq \infty$ . Estes elementos são obtidos através da expressão dada abaixo.

$$E_{ki} = T_k C_{ki} + \sum_{j=1}^N ((f_{ki}^a L_a) + f_{ki}^c (L_c + L'_c)) \gamma_{ij}$$

onde:

$T_k$  = Tamanho do arquivo  $\underline{k}$ .

$C_{ki}$  = Custos para armazenar uma unidade de arquivo  $\underline{k}$  no nodo  $\underline{i}$ .

$f_{ki}^a$  = Frequência de mensagens de atualização de um nodo  $\underline{i}$  para um arquivo  $\underline{k}$ .

$f_{ki}^c$  = Frequência de mensagens de consulta de um nodo  $\underline{i}$  para um arquivo  $\underline{k}$ .

$L_c$  = Tamanho médio em Bytes das mensagens de consultas de um nodo  $\underline{i}$  para um arquivo  $\underline{k}$ .

$L_a$  = Tamanho médio em Bytes das mensagens de atualização de um nodo  $\underline{i}$  para um arquivo  $\underline{k}$ .

$L_c$  = Tamanho médio em Bytes das mensagens de respostas as consultas efetuados de um nodo i para um arquivo k.

$\delta_{ij}$  = Custo de comunicação por Byte transmitido entre o par de nodos i e j.

Esta expressão indica o custo  $E_{ki}$  do arco (k,i) necessário para se manter armazenado um arquivo k no nodo i. A primeira parcela de  $E_{ki}$  indica os custos de armazenamento propriamente dito ( $T_k \cdot C_{ki}$ ). A segunda parcela indica o somatório dos custos pagos pelo acesso remoto de todos os outros nodos ao arquivo k. Seja para realizar uma operação de atualização, seja para realizar uma operação de consulta, através do caminho que ofereça o menor custo de comunicação entre o nodo de origem i e o nodo de destino j onde está armazenado o arquivo k.

De forma semelhante, o problema de Designação da Programação Linear guarda as mesmas características da conceituação apresentada acima e constitui uma eficiente e simples ferramenta para solução de um considerável número de questões relativas à alocação de recursos. A formulação matemática do problema de Designação é apresentada por diversos autores.

Assim sendo dado  $M[E_{ki}]$  a solução ótima do problema de alocação de arquivos sem redundância consiste em achar S, o conjunto de n pares (k,i) onde todos os elementos são distintos tal que:

$$Z = \sum_{i=1}^N \sum_{k=1}^F E_{ki} \cdot Y_{ki}$$

seja mínimo



sujeito a

$$\sum_{k=1}^F Y_{ki} = 1 \quad (i = 1, 2, \dots, N)$$

$$\sum_{i=1}^N Y_{ki} = 1 \quad (k = 1, 2, \dots, F)$$

e  $Y_{ki} \geq 0 \quad (k = 1, 2, \dots, F) \quad (i = 1, 2, \dots, N)$

onde

$$\begin{cases} Y_{ki} = 1 & \text{se o arquivo } \underline{k} \text{ for atribuído ao local} \\ & \underline{i}. \\ Y_{ki} = 0 & \text{se o arquivo } \underline{k} \text{ não for atribuído ao} \\ & \text{local } \underline{i}. \end{cases}$$

O processo interativo que conduzirá à Designação Ótima está baseado no seguinte princípio e que está demonstrado em NOVAES <sup>68</sup>.

"Ao se adicionar uma constante a cada elemento de uma linha (coluna) qualquer da matriz de custos, também chamada de matriz de Eficiência de um problema de Designação, a solução ótima da matriz alterada será também a solução ótima da matriz original".

O processo consistirá em se obter uma Matriz de Eficiência alterada que seja equivalente a original, não possuindo nenhum elemento negativo e apresentando o maior número possível de elementos nulos, de maneira que a solução ótima seja aparente e igual a soma das constantes que foram utilizadas na transformação da Matriz de Eficiência.

Está claro que esta formulação apresenta três restrições para o problema de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos.

A primeira restrição apresenta uma limitação indicando que a quantidade de recursos a serem distribuídos deve ser igual à quantidade de destinos que irão receber os recursos,  $F = N$ , ou seja, a Matriz de Eficiência deve ser do tipo quadrática.

A segunda restrição nos aponta que cada recurso será atribuído apenas para um único destino, isto é, cada local receberá um e somente um recurso.

E finalmente a terceira restrição indica que nenhum local deixará de receber um recurso.

Estas duas últimas restrições estão explícitas nas seguintes formulações:

$$\sum_{k=1}^F Y_{ki} = 1 \quad (i = 1, 2, \dots, N)$$

$$\sum_{i=1}^N Y_{ki} = 1 \quad (k = 1, 2, \dots, F)$$

As três restrições mencionadas acima poderiam a princípio inviabilizar a aplicação do método de designação para solucionar o problema de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos, pelo fato de que o método proposto nesta tese admite que:

- A quantidade de nodos existentes no Sistema seja diferente da quantidade de arquivos a serem distribuídos;

- Qualquer nodo do Sistema pode armazenar nenhum, um ou mais de um arquivo, desde que satisfaça às condições de limitação da capacidade de armazenamento.

Entretanto estas restrições podem ser facilmente solucionada pelo fato de que o algoritmo de Designação da Programação Linear permite que a Matriz de Eficiência originariamente da forma  $(n \times m)$  possa ser transformada em uma matriz do tipo  $(n \times n)$  através da introdução de linhas ou colunas fictícias, cujos elementos receberão valores nulos, sem que interfiram no processo de designação. Com esta providência duas condições especiais ocorrerão.

A primeira condição é aquela em que a quantidade de arquivos a serem distribuídos é menor do que a quantidade de locais que irão recebê-los, isto é, para

$$|F| < |N| \text{ sujeito a}$$

$$\sum_{k=1}^F Y_{ki} = 1 \quad (i = 1, 2, \dots, N)$$
$$\sum_{i=1}^N Y_{ki} = 1 \quad (k = 1, 2, \dots, F)$$

teremos como consequência do processo de designação, a atribuição de um arquivo "DUMMY" para um local qualquer da rede, como mostra os grafos da figura 46 a seguir em um exemplo onde  $|F| = 3$  e  $|N| = 4$ .

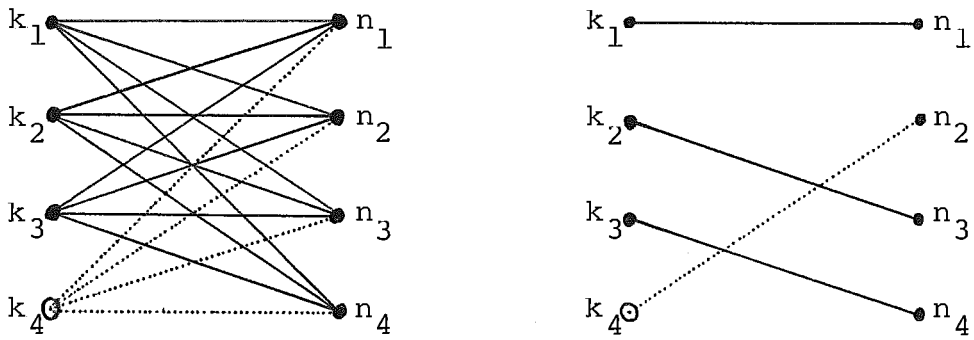


Figura 46

Este fato contudo não contraria as premissas estabelecidas no método de alocação proposto nesta tese, bastando apenas que se faça:

$$\sum_{i=1}^N Y_{ki} \geq 0 \quad (k = 1, 2, \dots, F)$$

A segunda condição é aquela em que a quantidade de arquivos a serem distribuídos é maior do que a quantidade de locais que irão recebê-los, isto é, para

$$|F| > |N| \quad \text{sujeito a}$$

$$\sum_{k=1}^F Y_{ki} = 1 \quad (i = 1, 2, \dots, N)$$

$$\sum_{i=1}^N Y_{ki} = 1 \quad (k = 1, 2, \dots, F)$$

teremos como consequência do processo de designação, a atribuição de arquivos para locais "DUMMY", como mostra os Grafos da figura 47 a seguir em um exemplo onde  $|F| = 5$  e  $|N| = 3$ .

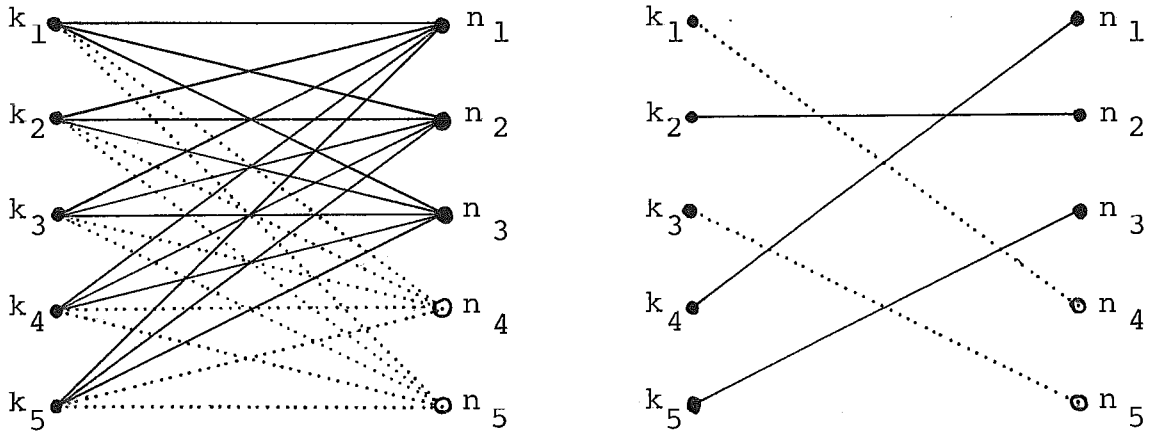


Figura 47

Este fato entretanto não pode ocorrer, porque na realidade ficaremos com arquivos sem locais a eles destinados. Desta forma temos que fazer:

$$\sum_{i=1}^N Y_{ki} \geq 1 \quad (k = 1, 2, \dots, F)$$

o que satisfará as premissas estabelecidas no método de alocação proposto.

Um fato ainda a considerar é que a condição para  $|F| > |N|$  sujeito a

$$\sum_{i=1}^N Y_{ki} = 1 \quad (k = 1, 2, \dots, F)$$

poderá conduzir a situações em que determinados nodos da rede ficarão sujeitos a um excessivo tráfego de consultas e atualizações devido às atribuições de diversos arquivos. Este problema entretanto será objeto de análise no Capítulo VI onde estudaremos o desempenho do sistema em função da alocação obtida através deste método.

5.5.2 - Solução para o Problema de Alocação Com Redundância

Se cada arquivo k, do conjunto de arquivos F que constitui a Base de Dados do Sistema, for designado para ser armazenado em um ou mais locais de um Sistema de Processamento de Dados Distribuídos caracteriza-se um problema de alocação de arquivos com redundância. Neste tipo de alocação fica evidente a necessidade de existir um balanceamento entre, os custos de comunicação, a integridade dos dados e o tempo de resposta de forma que o desempenho do Sistema seja maximizado por um custo mínimo.

Supondo que se deseje minimizar os custos de comunicações relativos ao tráfego de consulta, em um determinado Sistema, admitindo-se para isso uma configuração de replicação extrema, na qual todo arquivo seja armazenado em cada local da rede, como mostra a figura 48 abaixo.

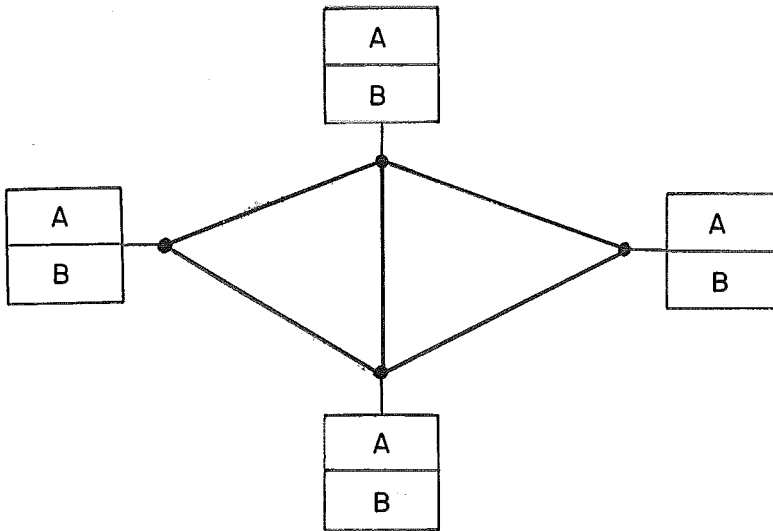


Figura 48

Está claro, com este exemplo, que o tráfego de consulta ficará reduzido a um custo zero, se considerarmos que as consultas locais não geram custos de comunicação. Entretanto o tráfego gerado pela necessidade de se manter atualizada as diversas cópias dos arquivos, em função da configuração assumida, provavelmente crescerá de tal forma que os custos de comunicações tornem-se significativamente elevados. Além do que os custos adicionais pagos pelo armazenamento das cópias dos arquivos poderão atingir o seu limite máximo.

Por outro lado uma arquitetura que retrate uma situação diretamente contrária, como no caso em que muitos arquivos são armazenados em somente um nodo da rede, como no exemplo da figura 49 abaixo, resultará em uma grande quantidade do tráfego de consultas dirigido a este nodo, e ele se transformará em um "gargalo" para o Sistema.

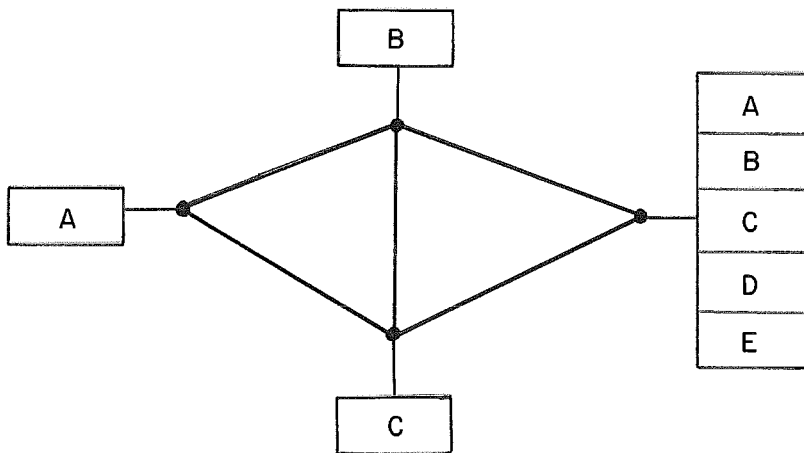


Figura 49

Esta concentração demasiada de serviços em um único nó compromete o tempo de resposta, em razão deste tempo ser função do tamanho da mensagem, da capacidade do canal de comunicação e do fluxo do tráfego, caracterizando, em alguns casos, uma situação de "node-bound".

Como podemos observar, por meio destes dois exemplos ilustrados acima, a existência de um equilíbrio entre a replicação dos arquivos, os custos de comunicações, armazenamento e desempenho, é essencial em um Sistema de Processamento de Dados Distribuídos. Além deste equilíbrio, outro importante aspecto a ser considerado para a solução do problema de uma ótima alocação com replicação, é o que se refere ao crescimento exponencial do tempo de processamento requerido pelo método de programação essencialmente matemáticos, em busca da solução ótima, mencionadas no item 3.2 desta tese. No sentido de tentar reduzir este tipo de problema GRAPA<sup>46</sup> introduz em seu trabalho alguns teoremas que podem ser aplicados a priori no sentido de indicar que certos locais de uma distribuição poderiam ou não serem incluídos em um problema de alocação. Apresentamos em seguida os três teoremas desenvolvidos por GRAPA, cujas demonstrações encontram-se em

- "O primeiro teorema afirma que se o custo de ter uma cópia de um arquivo, em um local, é menor do que o mínimo custo possível de enviar as consultas geradas neste local, para qualquer outra parte, então este local inquestionavelmente poderá ser incluído em uma alocação ótima".
- "O segundo teorema afirma exatamente o outro lado da moeda, ou seja, se custa mais manter uma cópia de um



arquivo em um local, do que seja possível economizar com mais esta cópia, então não podemos eleger este local para receber uma cópia do arquivo".

- "E finalmente o terceiro teorema permite que se elimine inicialmente certos locais de qualquer ulterior consideração em uma pesquisa do ótimo. Este teorema afirma que um local i não deve ser incluído em alguma ótima alocação, se existe um outro local j, na rede, tal que seja mais econômico manter a cópia do arquivo em j".

Estes teoremas foram aplicados sobre os dados do modelo de CASSEY<sup>12</sup>, e determinaram os mesmos resultados alcançados por CASSEY. Similarmente, KOLLIAS<sup>58</sup>, baseado nestes teoremas estabeleceu uma série de critérios que além de determinar a priori os mesmos resultados de GRAPA<sup>46</sup>, eles poderiam fazer parte integrante de um procedimento para determinar a solução ótima do problema de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos.

O critério adotado na metodologia apresentada nesta tese, para selecionar os locais de um Sistema de Processamento de Dados Distribuídos, que poderão receber cópias de arquivos, está fundamentado nos conceitos estabelecidos nos trabalhos de GRAPA<sup>46</sup> e KOLLIAS<sup>58</sup>.

### 5.5.2.1 - Condição Geral para Duplicação de Arquivos

O conjunto de nodos  $D$ , que deverá conter cópias do arquivo  $k$ , tal que  $D \subset N$ , será constituído dos nodos  $j \in N$  que satisfaçam a seguinte condição geral:

"Um determinado arquivo  $k$  armazenado em um nodo  $i$ , poderá ser duplicado em um outro nodo  $j$ , se e somente se os custos adicionais, provocados pelo armazenamento do arquivo  $k$  no nodo  $j$  ( $T_k C_{kj}$ ) e pelo tráfego gerado pela necessidade de atualização das cópias do arquivo  $k$  ( $\sum_{c=1}^N f_{ki}^a \gamma_{ij}$ ) for menor do que os custos de comunicações relativo ao tráfego de consulta dirigido ao arquivo  $k$  ( $\sum_{j=1}^N f_{ki}^c \gamma_{ji}$ ) armazenado em  $i$ , que poderem ser economizados com esta duplicação".

Desta forma, para que se possa determinar esse conjunto de nodos que irão receber as cópias do arquivo  $k$ , alguns procedimentos devem ser executados, na forma estabelecida no item que se segue.

### 5.5.2.2 - Procedimentos para Duplicação

Sejam:

- $j'$  .... um nodo candidato a armazenar uma cópia do arquivo  $k$ .
- $j$  .... um nodo eleito para receber uma cópia do arquivo  $k$ .
- $D'$  .... um conjunto de nodos candidatos a armazenar cópias do arquivo  $k$ .
- $D$  .... um conjunto de nodos eleitos para armazenar cópias do arquivo  $k$ .

Determinar:

- 1) Para todo  $j' \in N$ , o custo  $CR_{kj'}^i$ , que representa o valor a ser reduzido, em termos do tráfego de consulta, admitindo-se a duplicação do arquivo  $k$  no nodo  $j'$ , estando  $k$  alocado no nodo  $i$ .

onde:

$$CR_{kj'}^i = \sum_{j=1}^N f_{kj}^c \gamma_{ji}$$

sujeito a

$$f_{kj}^c (\gamma_{ji} - \gamma_{jj'}) > 0$$

Esta variável  $CR_{kj'}^i$ , guarda a soma de todos os custos de comunicação relativos ao tráfego de consulta ao arquivo  $k$ , localizado em  $i$ , de todos os nodos da rede, desde que o custo para executar essa consulta seja mais custosa, através do caminho onde se encontra originariamente o arquivo  $k(j,i)$ , do que pelo caminho onde se encontra a cópias do arquivo  $k(j,j')$ .

- 2) Para todo  $j' \in N$ , determinar o custo  $AD_{kj'}^i$ , que representa o valor a ser adicionado, em termos do tráfego de atualização requerido com a duplicação do arquivo  $k$ , no nodo  $j$ .

onde:

$$AD_{kj'}^i = \sum_{j=1}^N f_{kj}^a \gamma_{ji} + \infty$$

tal que:

$$\alpha = \begin{cases} \sum_{j=1}^N f_{kj}^c \gamma_{ji} & \text{se } f_{kj}^c (\gamma_{ji} - \gamma_{jj'}) < 0 \\ \text{ZERO} & \text{se } f_{kj}^c (\gamma_{ji} - \gamma_{jj'}) \geq 0 \end{cases}$$

Esta variável  $AD_{kj}^i$ , guarda a soma de duas parcelas. A primeira parcela indica a soma de todos os custos de atualização da cópia do arquivo  $k$  duplicado em  $j'$ , por todos os nodos da rede. E a segunda parcela indica os custos de comunicação relativo ao tráfego de consulta ao arquivo  $k$  armazenado em  $j$ , desde que os custos desta consulta seja menor que o custo da mesma consulta direcionada ao arquivo  $k$  armazenado em  $i$ .

3) Se para cada  $j' \in N$

$$CR_{kj'}^i > AR_{kj'}^i,$$

então o arquivo  $k$  localizado no nodo  $i$  satisfaz a condição geral de duplicação. Desta forma deve ser eleito o nodo  $j' \in N$  que oferecer o máximo  $(CR_{kj'}^i - AR_{kj'}^i)$ , para fazer parte do conjunto  $D'$ .

4) Verificar se não existe um outro local  $j''$  para o qual  $k$  satisfaça a condição geral, considerando agora que  $k$  já está duplicado em  $j'$ . Caso a condição geral de duplicação não possa mais ser satisfeita, o conjunto

$D'$ , contendo todos os nodos eleitos, é o conjunto candidato a armazenar cópia do arquivo  $k$ .

- 5) O conjunto  $D'$  eleito como conjunto  $D$  será aquele que, para todo  $i \in N$ , a duplicação do arquivo  $k$  ofereça um valor máximo para:

$$\sum_{j=1}^N (CR_{kj}^i - AR_{kj}^i)$$

- 6) Repetir todos estes procedimentos para todo  $k \in F$ .

## 5.6 - CONSIDERAÇÕES FINAIS DA ALOCAÇÃO DE ARQUIVOS

Neste Capítulo, propomos uma metodologia para solução do problema de alocação de arquivos de dados em uma Rede de Computadores, onde os usuários em vários nodos, tem acessos aos arquivos de dados, através de programas que processam mensagens de consultas e atualizações. E para isto, a metodologia prever inicialmente, o desenvolvimento de uma série de procedimentos que constituem o Projeto Lógico do Sistema de Informações Distribuídas. Esta distribuição lógica está fundamentada nos princípios básicos da Análise de Sistemas de Informações e cujos resultados permitem a construção dos "inputs" necessários ao método de Alocação de Arquivos. Desta maneira, dois tipos de alocação de arquivos em um Sistema de Processamento de Dados Distribuídos foram propostos, tal que, uma alocação selecionada deva ser também um resultado de uma bem balanceada Rede de Informações. Estas formas de Alocações denominam-se de Alocação sem Redundância e Alocação com Redundância.

Para solucionar a primeira forma de alocação proposta, a metodologia buscou, no isomorfismo existente entre o problema de DESIGNAÇÃO de recursos da Pesquisa Operacional e a Alocação sem Redundância de arquivos, uma forma simples e eficiente de determinar a distribuição dos arquivos de dados para os diversos nodos de uma rede de computadores, a um custo mínimo. Muito embora algumas modificações tivessem que ser introduzidas nas restrições da formulação original do problema de designação, com o objetivo de tornar possível a utilização deste algoritmo para solução do problema de alocação sem redundância de arquivos, o quadro demonstrativo a seguir mostra um mapeamento deste isomorfismo existente.

Alocação de Arquivos		Designação de Recursos	
- Conjunto de Nodos	N	- Conjunto de Locais	L
- Conjunto de Arquivos	F	- Conjunto de Recursos	R
- Um particular arquivo	K	- Um particular Recurso	i
- Um particular nodo	i	- Um particular local	j
- Quantidade de Acessos de <u>i</u> para <u>k</u>	$f_{ki}$	- Demanda do Recurso <u>j</u>	D <sub>j</sub>
- Custo de Comunicação entre <u>i</u> e <u>j</u>	$\delta_{ij}$	- Custo de enviar um recurso de <u>j</u> para <u>k</u>	T <sub>jk</sub>
- Custo de alocar o arquivo <u>k</u> em <u>i</u>	E <sub>ki</sub>	- Custo de designar o recurso <u>i</u> em <u>j</u>	C <sub>ij</sub>

A solução da segunda forma de alocação de arquivos proposta na metodologia e denominada de Alocação com Redundância, am parou-se nos teoremas apresentados por GRAPA<sup>46</sup> e KOLLIAS<sup>58</sup>, que resultaram nos procedimentos básicos descritos no item 5.5 deste capítulo.

Em ambos os casos, o método buscou a minimização dos custos de armazenamentos e comunicações, considerando as restrições impostas pela capacidade de armazenamento disponível e em cada nodo do sistema.

O processamento dos procedimentos propostos para as duas formas de alocação foram implementados em um programa codificado em FORTRAN e submetidos em um ambiente DOS/VSE para Sistema IBM Modelo 4341 cujas listagens fonte do programa e os resultados relativos a exemplificação do método encontra-se no ANEXO A desta Tese.

No ANEXO B apresentamos também o comportamento do método exposto, para os mesmos dados do modelo de CASSEY<sup>12</sup> cujo resultado obtido são os mesmos alcançados por CASSEY. E para exemplificar a flexibilidade do método para as diversas variações de topologias de redes o ANEXO C mostra os resultados conseguidos para as diversas formas de arquiteturas de redes.

Desejamos também ressaltar um dos problemas, que deixou de ser considerado nesta tese, e que é consequência da localização de arquivos em uma rede de computadores, ou seja, os Diretórios desses arquivos. O diretório é um conjunto de informações pelas quais os arquivos são identificados e localizados na rede. Em um Sistema de Processamento de Dados Distribuídos, três modos de criação de diretórios foram considerados por CHU<sup>3</sup>. O diretório Centralizado, o diretório Distribuído e o diretório Local. Existem diversas vantagens e desvantagens na utilização destes tipos de diretórios, e algumas combinações desses diretórios com arquivos Distribuídos são utilizadas. O trabalho de OHANA<sup>69</sup> mostra um estudo detalhado sobre organização de diretórios para Sistemas de Processamento de Dados Distribuídos.

Finalmente queremos salientar, que os dados utilizados nos exemplos do ANEXO A e ANEXO C, não representam informações relativas a qualquer sistema, entretanto servem de base para exemplificação da utilização do método, enquanto o exemplo apresentado no ANEXO 2 serve de base para avaliar o seu comportamento.



CAPÍTULO VI

MÉTODO PARA AVALIAÇÃO DO DESEMPENHO DO SISTEMA

6.1 - INTRODUÇÃO

No Capítulo anterior descrevemos um método de alocação de arquivos, para os diversos locais de um Sistema de Processamento de Dados Distribuídos, cuja principal ênfase do problema está voltada para a minimização dos custos relativos ao armazenamento das cópias dos arquivos nos diversos locais e para os custos de comunicações necessários ao envio das mensagens de consultas e atualização para os arquivos situados em locais remotos do Sistema. Assim sendo, os arquivos disjuntos  $\{f_1\}$ ,  $\{f_2\}$ , ...,  $\{f_k\}$  que unidos constituem virtualmente o Banco de Dados do Sistema, podem ser acessados de qualquer local e nestes arquivos serem realizadas operações de consultas e atualizações a um custo mínimo.

$$Z = \sum_{i=1}^N \sum_{k=1}^F E_{ki} \cdot Y_{ki}$$

Por outro lado os fatores responsáveis pelas medidas que avaliam o desempenho do Sistema tais como: Tempo de Resposta, Capacidade dos Canais de Comunicação, Atrasos, Fluxos, Throughput, etc., não foram considerados na formulação do modelo de alocação. Desta maneira ficaram introduzidas no método apresentado, questões da seguinte ordem: Com esta designação estabelecida pelo método de alocação, qual o tempo médio de resposta oferecido pelo Sistema, aos seus usuários, por suas consultas efetuadas? Será este tempo oferecido, suficientemente satisfatório para atender os requisitos dos usuários?

Este Capítulo tem como objetivo, oferecer respostas para as questões levantadas acima, fazendo primeiramente um estudo simplificado das ferramentas analíticas utilizadas para avaliação destas questões e em seguida apresentar um enfoque para solução do problema de forma que possibilite aos usuários do Sistema decidir por outras arquiteturas de distribuição de arquivos, pelos diversos nodos de uma rede de computadores, que ofereçam índices de desempenhos mais compatíveis com as necessidades reais de suas aplicações.

Neste contexto algumas importantes medidas de desempenho de redes de computadores são estudadas. Entre elas destacamos aquelas que medem o tempo que decorre entre a chegada de uma mensagem em seu nodo de origem até a bem sucedida entrega desta mensagem em seu nodo de destino (ATRASO). Em um projeto de um Sistema de Processamento de Dados Distribuídos, que é fundamentalmente apoiado em uma rede de comunicação de computadores, é importante que este tempo seja razoavelmente pequeno, em razão dele ser um influente componente no cálculo do Tempo de Resposta para o usuário.

Outra importante medida de desempenho em uma rede de comunicação de computadores é o número de mensagens entregues pela rede em uma unidade de tempo (VAZÃO).

Estas duas medidas de desempenho estão intimamente relacionadas, mas são usualmente aplicadas como critérios de desempenho para diferentes tipos de tráfegos. As medidas que avaliam o Tempo de Resposta são particularmente importantes no caso de tráfego do tipo interativo, que se caracteriza por ter que ser libe-

rado rapidamente e é usualmente constituído de mensagens curtas. Por outro lado quando estamos tratando de uma avaliação sobre um tráfego gerado por uma transferência demorada de um arquivo, a medida do tempo de resposta deixa de ser o centro da questão, o interesse maior está ligado à medida da capacidade de vazão da rede, ou seja, quantos bits por segundo pode ser "bombeado" através da rede. De qualquer forma não só estas duas medidas de desempenhos exemplificadas acima como também outras medidas tais como: o controle do fluxo a fim de evitar o congestionamento da rede, os custos de comunicação, a integridade da rede, etc, tem sido usados intensivamente por diversos autores, utilizando para isto modelos baseados na Teoria das Filas para rede de computadores, entre os quais destacamos os trabalhos de KLEINROCK<sup>56</sup>, CHANDY<sup>18</sup>, GERLA<sup>43</sup>, LANING<sup>59</sup>, etc. Nos itens seguintes faremos uma análise sintetizada de algumas destas medidas de desempenho.

## 6.2 - MEDIDAS DE DESEMPENHO

A utilização de modelos baseados na teoria das Filas oferecem um efetivo conjunto de ferramentas analíticas para o estudo do desempenho em uma rede de comutação de pacotes. Este esforço, no sentido de determinar modelos de desempenhos de Sistemas, tem sido executado procurando-se primeiramente prognosticar o tempo médio de atraso encontrado por uma mensagem em seu caminho através da rede, para em seguida determinar a capacidade ótima de atribuição dos canais de comunicação de forma a se conseguir um tempo de resposta mínimo.

### 6.2.1 - Análise do Atraso

O atraso é uma das medidas de maior importância para o estudo do desempenho de uma rede de comunicação de computadores. Esta medida representa um tempo médio (T), consumindo por uma mensagem para percorrer a sua rota, entre o seu nodo fonte (i) e o seu nodo de destino (j). Podemos ainda caracterizar o atraso como sendo função do tamanho da mensagem, da capacidade dos canais de comunicações e do fluxo de mensagens existentes entre cada par de nodos da rede, ou seja o tráfego de mensagens.

KLEINROCK <sup>55</sup> deduziu uma expressão representativa deste tempo para uma rede de comunicação de computadores, baseada em um tráfego de "pacotes", considerando os seguintes aspectos.

- As chegadas externas em um nodo de comutação obedecem à distribuição de POISSON, isto é, o tráfego que entra na rede através de uma fonte externa, como por exemplo um computador hospedeiro, obedece a uma distribuição cuja probabilidade  $P_k(t)$  de acontecer  $k$  chegadas em um intervalo de tempo (t) segundos e dado por

$$P_{k(t)} = \left[ \frac{(\lambda t)^k}{k!} \right] e^{-\lambda t}$$

onde  $\lambda$  indica a taxa de chegada;

- A capacidade de armazenamento das mensagens em cada nodo é considerada ilimitada;
- Todos os pacotes são considerados como tendo um tamanho que é obtido independentemente de uma distribuição exponencial, com um tamanho médio de  $1/u$  bits por pacote;

- O direcionamento das mensagens é feito sobre uma rota fixa. O que implica na existência de um único caminho através da rede para um dado par origem-destino;
- Os canais são livres de erros;
- Existe uma independência entre os intervalos das chegadas e o tempo de transmissão em cada canal.

Levando em conta todas as premissas relacionadas acima e utilizando o suporte oferecido pela teoria das filas na análise do desempenho de uma rede de computadores o valor médio do atraso (T) é obtido de acordo com a expressão abaixo, cuja evolução pode ser encontrada em KLEINROCK<sup>55</sup>.

$$T = \frac{1}{\gamma} \sum_{i=1}^b \frac{\lambda_i}{\mu C_i - \lambda_i}$$

onde:

- $1/\mu$  - tamanho médio dos pacotes (bits/pacotes);
- $C_i$  - capacidade do canal (i) (bits/segundos);
- $\lambda_i$  - frequência média do tráfego no canal (i), dados em (pacotes/segundos);
- b - número de arcos da rede
- $\gamma = \sum_{j,k} \gamma_{kj}$  onde  $\gamma_{jk}$  é a taxa média de pacotes que circulam do nodo (j) ao nodo (k).

Embora esta expressão do atraso seja suficientemente apurada para soluções de muitos propósitos na análise de desempe-

nho de uma rede de computadores, uma expressão mais refinada pode ser obtida com a inclusão dos efeitos provocados pelos fatores:

- $P_i$  Tempo de Propagação no Canal ( $i$ ) em segundos;
- $K_i$  Tempo de Processamento Nodal (pacotes/segundos).

Com estas novas condições obtém-se então a seguinte expressão representativa do atraso médio em uma rede.

$$T = K + \sum_{i=1}^b \frac{\lambda_i}{\gamma} \left[ \frac{1}{\mu C_i} + \frac{\lambda_i / \mu C_i}{\mu C_i - \lambda_i} + P_i + K \right]$$

onde:

- $1/\mu'$  representa o número médio de bits em uma mensagem de informação;
- $1/\mu$  representa o tamanho médio de um pacote incluindo o ACK, paridade, header, etc.;
- $K$  tempo de processamento nodal (constante para todos os nodos);
- $P_i$  tempo de propagação no canal  $i$ .

O detalhamento e evolução destas expressões do atraso podem ser encontradas em KLEINROCK<sup>5 6</sup>, GERLA<sup>4 3</sup>.

### 6.2.2 - Vazão

A vazão é provavelmente a mais comum das medidas de desempenho de controle de fluxo em uma rede de computadores. Esta medida nos oferece uma taxa representativa da capacidade de processamento de uma Unidade de Informação durante um intervalo de tempo. Podendo ser, de acordo com os critérios de análise desen-

volvidos, representada por: bits/seg, pacotes/seg, transações/seg, etc.

### 6.2.3 - Capacidade dos Canais de Comunicações

Um outro fator a ser considerado na análise do desempenho é a otimização da distribuição da capacidade dos canais de comunicações em função dos seus custos.

Sendo  $Q_i$  a capacidade do canal de comunicação  $i$ , e sendo  $d_i(Q_i)$  o custo a ser dispendido pelo aluguel da linha  $i$  de capacidade  $Q_i$  o custo total de comunicação é definido como:

$$D = \sum_{i=1}^b d_i(Q_i)$$

Em KLEINROCK <sup>56</sup> encontramos uma síntese sobre a otimização de várias funções de custos de linhas de comunicações.

### 6.2.4 - Tráfego

O tráfego médio entre os nodos, representado pela Matriz  $R = [R_{jk}]$  onde  $\{R_{jk}\}$  é a taxa média de transmissão do nodo fonte  $j$  ao nodo de destino  $k$  é um outro fator importante no estudo das medidas de desempenho. A análise dos diversos tipos de tráfego tem mostrado que um projeto ótimo é melhor formulado para tráfegos que possuem variações padrões. Esta propriedade tem um significado especial para redes distribuídas o que em geral justifica a utilização de tráfego médio.

O tráfego de mensagens em qualquer nodo  $i$  associado com um arquivo  $j$  localizado em algum outro nodo e representado em mensagens/segundos, é obtido, segundo MAHMOUD<sup>62</sup>, através de quatro componentes como mostra a figura 50 abaixo.

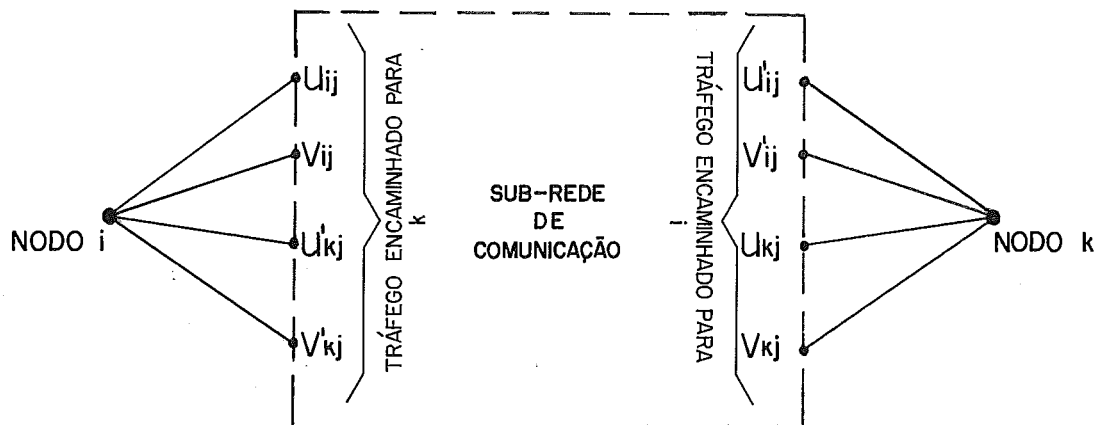


Figura 50

- $U_{ij}$  tráfego de consultas originado no nodo  $i$  para o arquivo  $j$ ;
- $V_{ij}$  tráfego de atualizações originado no nodo  $i$  para o arquivo  $j$ ;
- $U'_{ij}$  tráfego de retorno para o nodo  $i$  resultante do tráfego de consultas  $u_{ij}$ ;
- $V'_{ij}$  tráfego de retorno ao nodo  $i$  resultante do tráfego de atualização  $v_{ij}$ .



Baseado nestes componentes e considerando que se um arquivo pode ser alocado em  $q_j$  diferentes nodos em uma rede onde:

$$q_j = \sum_{i=1}^N x_{ij} \quad \text{tal que} \quad \begin{cases} x_{ij} = 1 & \text{se o nodo } \underline{i} \text{ contém o arquivo} \\ & \underline{j}. \\ x_{ij} = 0 & \text{em caso contrário.} \end{cases}$$

então é assumido que as consultas e seus respectivos tráfegos de retorno do nodo  $\underline{i}$  são divididas igualmente entre aqueles  $q_j$  nodos providos do arquivo  $\underline{j}$  no qual o arquivo  $\underline{j}$  não está armazenado no nodo  $\underline{i}$ . Nestas condições MAHMOUD<sup>6 2</sup> desenvolveu uma expressão para determinar o tráfego existente entre o nodo  $\underline{i}$  e o nodo  $\underline{k}$  da seguinte forma:

$$\gamma_{ik} = \sum_{j=1}^N \left\{ \left[ \left( \frac{\mu_{ij}}{q_j} \right) (1 - X_{ij}) + V_{ij} \right] X_{ki} + \left[ \left( \frac{\mu_{kj}}{q_j} \right) (1 - X_{kj}) + V_{kj} \right] X_{ij} \right\}$$

o tráfego total da rede será portanto expresso por:

$$\gamma = \sum_{i=1}^N \sum_{k=1}^N \gamma_{ik}$$

Outros fatores tais como uma política de direcionamento, integridade da rede, fluxos nas linhas de comunicações, etc., são todas medidas propostas e estudadas por alguns autores que podem ser facilmente incluídas como variáveis ou restrições em um modelo de alocação que considere a análise de desempenho de redes de comunicação de computadores.

### 6.3 - FORMULAÇÕES

Todos os aspectos vistos acima são importantes na opção de formulação a ser feita. Veremos em seguida alguns tipos de formulação para o problema de alocação de arquivos que considera alguns destes fatores tais como capacidade das linhas de comunicações, direcionamento, fluxos, etc. Os algoritmos para solução destas formulações podem ser encontrado no trabalho de GERLA<sup>43</sup>.

#### 6.3.1 - Formulação Considerando a Capacidade da Linha

- Dados:
- Topologia da Rede
  - Matriz de Tráfego  $\underline{R}$
  - Política de Direcionamento (e portanto um vetor de fluxo de linha  $f = (f_1, f_2, \dots, f_b)$ )

Minimizar: 
$$D = \sum_{i=1}^b d_i (C_i)$$

Sobre as variáveis:  $C = (C_1, C_2, \dots, C_b)$

Sujeito a: -  $F \leq C$

- 
$$T = \frac{1}{\delta} \sum_{i=1}^b \frac{f_i}{C_i - h_i} \leq T_{MAX}$$

6.3.2 - Formulação Considerando o Direcionamento

- Dados:
- Topologia da Rede
  - Capacidade dos Canais de Comunicações ( $C_i$ )
  - Matriz de Tráfego  $\underline{R}$

Minimizar: 
$$T = \frac{1}{\gamma} \sum_{i=1}^b f_i \left[ \frac{1}{C_i - f_i} + \mu (P_i + K_i) \right]$$

Sobre as variáveis:  $f = (f_1, f_2, \dots, f_b)$

- Sujeito a:
- $f$  é um fluxo variável que satisfaz  $\underline{R}$
  - $f \leq C$

6.3.3 - Formulação Considerando a Capacidade e Fluxos

- Dados:
- Topologia da Rede
  - Matriz de Tráfego  $\underline{R}$
  - $d_i(C_i)$  funções custo-capacidade

Minimizar: 
$$D(C) = \sum_{i=1}^b d_i(C_i)$$

Sobre as variáveis:  $f, C$

- Tal que:
- $f$  é um fluxo variável que satisfaz  $\underline{R}$
  - $f \leq C$

- 
$$T(f, C) = \frac{1}{\gamma} \sum_{i=1}^b \frac{f_i}{C_i - f_i} \leq T_{\text{MAX}}$$

Diversos pesquisadores formularam os seus modelos de análise de desempenho dentro deste contexto apresentado. Assim por exemplo RAMAMOORTHY <sup>72</sup> investigou um modelo que procurava minimizar o tempo de acesso ao arquivo, sujeito a restrições de custos de armazenamento, em uma memória hierárquica. ARORA <sup>2</sup> formulou um modelo que minimizava o tempo de execução dos programas e o tempo de transferência de arquivos que não podem ser processados localmente. BUZEN <sup>11</sup> considerou um modelo, baseado na teoria das filas, sobre uma topologia em estrela para maximizar o Throughput e finalmente o trabalho de CHEN <sup>21</sup> procurava minimizar o tempo de resposta. Desta forma o problema da análise de desempenho de uma rede de computadores tem sido exaustivamente estudado e das mais variadas formas de enfoque do problema. E para concluir este estudo sobre a análise de desempenho de redes de computadores, mostraremos o trabalho de DENNING e BUZEN <sup>32</sup> que utilizaram um enfoque operacional para estudar o comportamento da vazão e do Tempo de Resposta. Na análise operacional existem dois componentes básicos para todo problema: Um Sistema, o qual pode ser Real ou Hipotético e um período de tempo, podendo ser presente, passado e futuro. O objetivo de uma análise operacional é equacionar as quantidades medidas de um sistema durante um dado período de tempo.

Este trabalho baseia-se na análise sobre um modelo de Sistema de fila com apenas um servidor i com quatro variáveis operacionais, as quais, como quantidades básicas podem ser diretamente medidas durante um intervalo de tempo, ou ainda, como quantidades derivadas podem ser calculadas através da manipulação das quan

tidades básicas. A figura 51 abaixo mostra este modelo e suas respectivas quantidades básicas e derivadas.

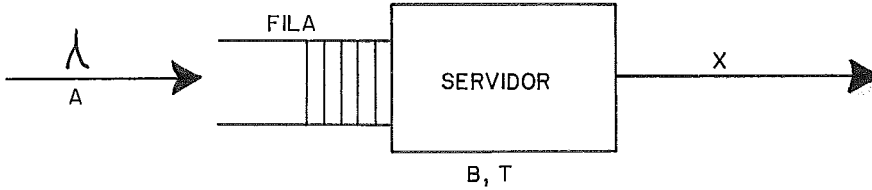


Figura 51

- Quantidade Básicas:

T ... o tamanho do período de observação.

A ... número de chegadas ocorridas em T.

B ... a quantidade total de tempo durante o qual o sistema estava ocupado durante o período de observação ( $B > T$ ).

C ... o número de serviços completos ocorridos durante o período de observação T.

- Quantidades Derivadas:

$\lambda$  ...  $A/T$ , taxa de chegada em Jobs/seg.

X ...  $C/T$ , taxa de saída em Jobs/seg.

U ...  $B/T$ , fração de tempo em que o sistema está ocupado.

S ...  $B/C$ , o tempo de serviço por Job.

A partir destas quantidades DENNING<sup>32</sup> define uma sê-rie de equações operacionais necessárias ao cálculo e prognóstico das medidas de desempenho. Estas equações operacionais estão relacionadas na tabela da figura 52 abaixo, e o completo desenvolvimento de cada uma delas podemos encontrar no trabalho de DENNING<sup>32</sup>.

Equações Operacionais de Buzen

Lei da Utilização .....	$U_i = X_i S_i$
Lei da Little's .....	$\bar{n} = X_i R_i$
Lei do Fluxo Forçado .....	$X_i = V_i X_i$
Lei do Fluxo de Saída .....	$X_o = X_i q_{io}$
Lei Geral do Tempo de Resposta .	$R = V_i R_i$
Lei do Tempo de Resposta Intera tivo.	$R = M/X_o - Z$

Figura 52

6.4 - AVALIAÇÃO DE DESEMPENHO DE UM SISTEMA DE PROCESSAMENTO DE DADOS DISTRIBUÍDOS

6.4.1 - Objetivo

O objetivo fundamental do processo de avaliação do desempenho de um Sistema de Processamento de Dados Distribuídos, desenvolvido segundo o enfoque proposto nesta metodologia, tem em vista a determinação do tempo de acesso  $T$ , consumido por uma mensagem que parte de um nodo qualquer  $i$  para alcançar um nodo de

destino qualquer  $j$ , em razão de ter que realizar uma atividade de consulta ou atualização sobre um arquivo  $k$ , situado no nodo  $j$ . Esta avaliação é executada em função do tráfego gerado na rede como consequência da distribuição dos arquivos estabelecidas no método de alocação proposto no Capítulo V desta tese.

Com os resultados obtidos nesta avaliação, torna-se possível a realização de uma análise dos tempos alcançados, de forma que se esses tempos não satisfizerem as expectativas desejadas pelos usuários do Sistema, podem ser introduzidas modificações nos parâmetros que induziram à distribuição dos arquivos, de maneira que, novos tempos possam ser conseguidos que ofereçam uma relação custo/benefício mais satisfatória aos interesses dos usuários do Sistema.

#### 6.4.2 - Descrição do Modelo

O modelo básico para avaliação do desempenho de uma rede de comunicação de computadores, que dá suporte ao Sistema de Processamento de Dados Distribuídos, proposto nesta tese, está fundamentado no estudo sobre teoria das filas e é representado por três componentes principais. As Interfaces de Comunicações (IC), as linhas de Transmissões de Dados (LT) e a Capacidade de processamento local representada pelos computadores (C). A figura 53 a seguir mostra o modelo proposto.

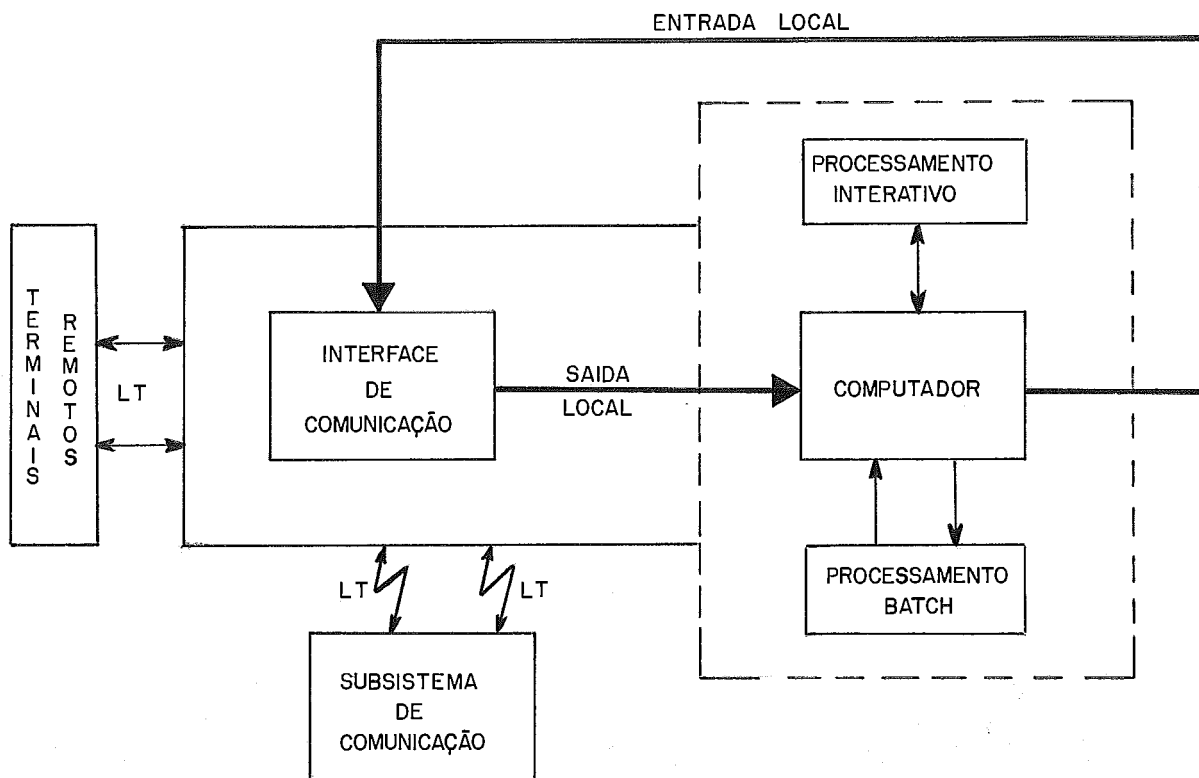


Figura 53

Uma Interface de Comunicação (IC) que na realidade é um processador FRONT-END, representando um nodo de comutação é constituído por uma fila de chegada (FC), um ou mais processadores (P), diversas filas de saídas (FS) e equipamentos responsáveis pelas interfaces de linhas (IL). Cada linha de transmissão tem associada a si uma fila para reter a mensagem o tempo necessário para que a linha esteja capacitada e liberada para transmitir a mensagem para o outro nodo. A figura 54 a seguir mostra o modelo de uma Interface de Comunicação apresentada por SAMARI<sup>74</sup>.



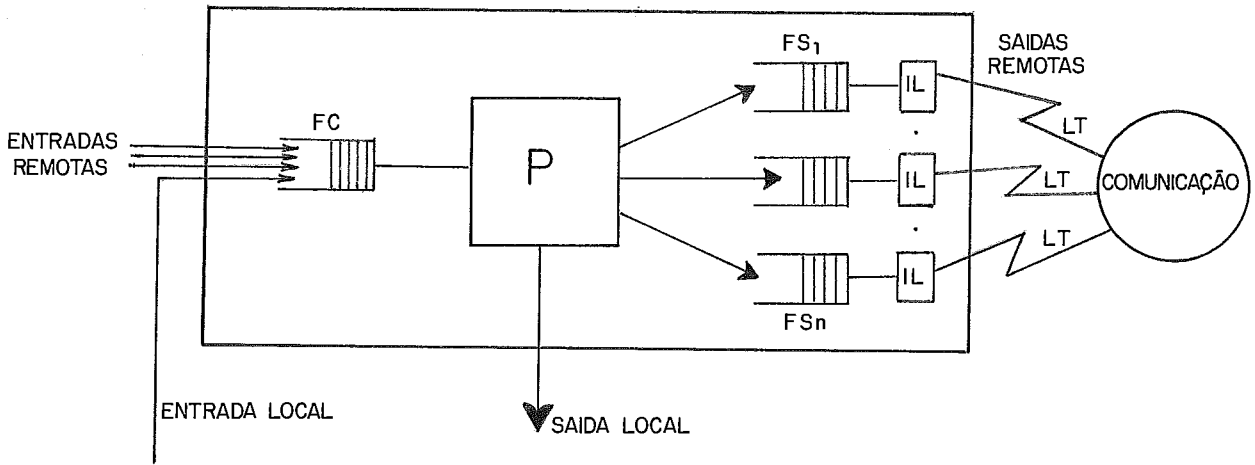


Figura 54

O tráfego de mensagens que circularão entre uma Interface de Comunicação  $i$  e uma outra Interface de Comunicação  $j$  será dado por:

$$\phi_{ij} = \sum_{k=1}^F \left\{ \left[ f_{kij}^c (1 - y_{ki}) + f_{kij}^a \right] y_{jk} + \left[ f_{kji}^c (1 - y_{kj}) \right] y_{ki} \right\}$$

onde:

$f_{kij}^c$  = freqüência de consultas de um nodo  $i$  para um arquivo  $k$  localizado no nodo  $j$ .

$f_{kij}^a$  = freqüência de atualizações de um nodo  $i$  para um arquivo  $k$  localizado no nodo  $j$ .

$y_{ki}$  = variável de decisão que indica se o arquivo  $k$  está ou não alocado no nodo  $i$ , da seguinte forma:

$$\begin{cases} y_{ki} = 1 & \text{se está alocado} \\ y_{ki} = 0 & \text{se não está} \end{cases}$$

As mensagens originárias de usuários de terminais e de computadores hospedeiros são tratadas pela Interface de Comunicação como chegadas externas e a sua entrada na Interface de Comunicação se dá através de sua fila de entrada. Estas mensagens são posteriormente processadas e encaminhadas para as filas de saídas ou para uma saída local. Toda vez que uma mensagem entra em uma interface de comunicação diversas instruções são executadas necessárias as tarefas de armazenamento da mensagem, determinação de sua rota, manuseio de erros, controle de fluxo, etc. O tempo consumido para executar essas instruções é chamado de Tempo de Processamento Nodal (P). Sendo  $P_i$  o Tempo de Processamento nodal de um particular nodo.

As linhas de transmissão têm cada uma um canal com capacidade  $Q_l$  bits/seg, ligando a Interface de Comunicação  $i$  com a Interface de Comunicação  $j$ . O tráfego médio de mensagens que fluem através de um canal qualquer  $l$  é dado pela soma de todos os tráfegos cujo caminho passa pelo canal  $l$ . Considerando que  $\pi_{ij}$  é o caminho entre o nodo  $i$  e o nodo  $j$  podemos afirmar que o canal  $l$  está incluído no caminho  $\pi_{ij}$ , se este canal  $l$  é atravessado por mensagens que fluem neste caminho. E este fato é denotado por  $l \in \pi_{ij}$ . Assim sendo o tráfego de mensagens que fluem através de um canal  $l$  é expresso por:

$$\beta_l = \sum_{\substack{ij: l \in \pi_{ij} \\ i \neq j}}^N \sum_{i \neq j}^N \phi_{ij}$$

O componente computador do modelo representa os computadores "hospedeiros", responsáveis pela execução propriamente dita das transações solicitadas, permitindo dois tipos de processamento interativo e o processamento em batch.

## 6.5 - FORMULAÇÃO DO MÉTODO DE AVALIAÇÃO

### 6.5.1 - Dados de Entrada

a) Matriz de alocação  $M \left[ Y_{ki} \right]$ , onde cada elemento da matriz representa uma variável de decisão que indica se existe ou não um arquivo  $k$  alocado no nodo  $i$ , da seguinte forma:

$$\text{quando: } \begin{cases} Y_{ki} = 1 & \rightarrow \text{ existe uma alocação} \\ Y_{ki} = 0 & \rightarrow \text{ não existe} \end{cases}$$

b) Matriz de Frequência de Consultas  $M \left[ f_{ki}^c \right]$ , onde cada elemento da matriz, indica a quantidade de vezes em que o nodo  $i$  consulta o arquivo  $k$ .

c) Matriz de Frequência de Atualização  $M \left[ f_{ki}^a \right]$ , onde cada elemento da matriz, indica a quantidade de vezes em que o nodo  $i$  atualiza o arquivo  $k$ .

d) Matriz de Direcionamento  $M \left[ \pi_{ij} \right]$ , onde cada elemento da matriz indica o caminho percorrido por uma mensagem do nodo  $i$  ao nodo  $j$ .

e) Matriz de Capacidade das Linhas de Comunicações  $M \left[ Q_{ij} \right]$  onde cada elemento da matriz indica a capacidade em Bps da linha de comunicação direta entre  $i$  e  $j$ .

- f) Tamanho médio de uma mensagem de consulta  $L_c$ , dado em bits.
- g) Tamanho médio de uma mensagem de atualização,  $L_a$  dado em bits.
- h) Tamanho médio de uma mensagem em resposta a uma consulta  $L'_c$ , dado em bits.
- i) Tamanho do Pacote  $l/u$ . O tamanho de um pacote é normalmente expresso em termos da quantidade de conjunto de 8 bits que ele contém. Este conjunto de 8 bits é denominado de "octeto" e, assim o tamanho máximo do pacote é limitado a  $x$  octetos, onde  $x$  varia de rede para rede. Na França, por exemplo, a rede TRANSPAC opera com pacotes de 32 a 128 octetos, enquanto que na rede DATAPAC (Canada) os pacotes têm tamanho de 128 a 256 octetos.

#### 6.5.2 - Objetivos a Determinar

A matriz de Tempo de Acesso  $M \begin{bmatrix} T_{ki}^j \end{bmatrix}$ , onde cada elemento  $T_{ki}^j$ , da matriz, indica o tempo médio que uma mensagem consome para a partir de um nodo  $i$  de origem atingir um nodo  $j$  de destino, através de uma rota fixa  $\pi_{ij}$ , no intuito de realizar uma consulta ou atualização em um arquivo  $k$  situado no nodo  $j$ .

### 6.5.3 - Condições do Método

- Consideramos que existensomente linhas de comunicações do tipo Full-duplex, interligando os nodos da rede.
- O tráfego na rede não deve exceder a capacidade dos canais de comunicações.

$$\mu_{li} - \beta_{li} \geq 0$$

- São consideradas também as condições estabelecidas por KLEINROCK, quando deduziu uma expressão representativa do atraso, apresentada no item 6.2.1 deste capítulo.
- É considerado um Tempo de Processamento Nodal fixo para todo nodo  $n$  dado por  $P_j$  onde  $j$  é o nodo que recebe a mensagem.

### 6.5.4 - Procedimentos Básicos do Processo de Avaliação

O processo de avaliação é composto de quatro procedimentos fundamentais e que estão descritos a seguir:

6.5.4.1 - Inicialmente é determinado o tráfego  $\phi_{ij}$  existente entre cada par de nodos  $(i, j)$  em termos de PACOTES/SEG. Cada elemento desta Matriz de Tráfego é obtido segundo a seguinte expressão:

$$\phi_{ij} = \sum_{k=1}^F \left\{ \left[ f_{kij}^c L_c (1 - y_{ki}) + f_{kij}^a L_a \right] y_{jk} + \left[ f_{kji}^c L_c (1 - y_{kj}) \right] y_{ki} \right\}$$

6.5.4.2 - O segundo procedimento calcula o valor do tráfego existente em cada linha  $l$ , em termos de PACOTES/SEG. Estes valores são obtidos fazendo:

$$\theta_l = \sum_{\substack{ij: l \in \pi_{ij} \\ i \neq j}}^N \sum_{i \neq j}^N \phi_{ij}$$

6.5.4.3 - O terceiro procedimento determina para cada linha  $l$ , o tempo de espera na fila mais o tempo de transmissão na linha  $l$ . Estes valores são obtidos através da expressão do Atraso definida por KLEINROCK <sup>55</sup> e descrita no item 6.2.1 deste Capítulo.

$$T_l = \frac{1}{\mu Q_l - \theta_l}$$

6.5.4.4 - E finalmente o quarto procedimento é responsável pela construção da matriz de tempos  $M \left[ T_{kl}^j \right]$ , onde cada elemento desta matriz representa o tempo consumido por uma mensagem que parte de um nodo de origem  $i$  para atingir o nodo de destino  $j$  onde se encontra armazenado o arquivo  $k$ . Estes elementos são obtidos fazendo-se:

$$T_{kl}^j = \sum_{l \in \pi_{ij}}^E (T_l + P_j) - P_j$$

Adicionalmente o processo de avaliação determina ainda, como complementos para o processo de análise e avaliação do Sistema as seguintes informações:

- A relação entre o Tempo de Acesso e a Carga na rede;
- Uma matriz que mostra os pontos de saturação das linhas da rede. Estes valores são obtidos fazendo-se:

$$S_l = \left( \frac{Q_l}{\mu} \right) / Q_l$$

- Ajustamento da capacidade das linhas de comunicações. Este ajustamento é realizado em função do tráfego existente em cada linha.

## 6.6 - CONSIDERAÇÕES FINAIS DA AVALIAÇÃO

Neste Capítulo, foi desenvolvido um método para determinar o tempo consumido por uma mensagem de consulta ou atualização que parte de um nodo qualquer i para atingir um outro nodo qualquer j, onde se encontra armazenado o arquivo k, para o qual é destinada a mensagem.

Este método foi exposto segundo o enfoque da Teoria das Filas, para rede de computadores, analisados no item 6.2 deste capítulo, e se propõe a medir o tempo de acesso entre cada nodo fonte e destino, a partir do tráfego de mensagens existentes em cada arco da rede. Este tráfego é calculado através das frequências de mensagens de consultas e atualizações de cada nodo para os arquivos estando os arquivos armazenados segundo a distribuição obtida pelo método de alocação.

O método desenvolvido também é bastante flexível para permitir a existência de um processo de realimentação, afim de que, melhores resultados possam ser conseguidos tornando mais eficiente o desempenho do Sistema. Esta realimentação pode ser realizada através de um reajustamento da capacidade das linhas de comunicações e através de alterações nos tamanhos das mensagens ou dos pacotes. Com isto uma nova matriz de alocação com novos custos é obtida, um novo tráfego é gerado e como consequência final novos tempos de acessos são determinados.

Os procedimentos descritos, para determinar os objetivos propostos pelo método de avaliação, foram implementados em um programa codificado em FORTRAN e submetidos em um ambiente DOS/VSE para um Sistema IBM Modelo 4341. A listagem do programa fon-



te e os resultados relativos a um exemplo de aplicação do método fazem parte do ANEXO D desta Tese.

CAPÍTULO VII

C O N C L U S ã O

7.1 - Atualmente a Indústria de Computadores e seus usuários estão se aproximando de um estágio de desenvolvimento que tende ao uso de um Sistema de Banco de Dados Distribuídos. Nem todas as informações do Sistema seguirão esta tendência dominante. Muitas poderão e deverão continuar a servir satisfatoriamente através da estrutura convencional do Sistema.

Entretanto o avanço da tecnologia de Hardware e Software torna atrativa, às arquiteturas Distribuídas, um crescente número de uso. O desenvolvimento de Rede de Informações torna-se-ão mais e mais comuns nos próximos 5 a 10 anos. Atualmente a integração dos recursos de computação e das comunicações na teleinformática viabiliza a chamada "sociedade da informação", ampliando enormemente a utilização dos computadores nas mais diversas atividades e com isto alterando profundamente os hábitos dos homens e da sociedade como um todo.

Acompanhando portanto essas redes de informações e as arquiteturas de Sistemas Distribuídos, será também crescente o uso de Banco de Dados Distribuídos. Esses Sistemas de Banco de Dados Distribuídos, embora complexos, prover um grau de eficiência e custos/benefícios frequentemente impossível de acontecer em um Sistema Centralizado. Dentro deste contexto o problema de alocação de arquivos, para os diversos locais de um Sistema de Processamento de Dados Distribuídos, como o modelo apresentado nesta Tese, assu

me um papel de grande importância e tem estimulado diversos estudos e pesquisas.

7.2 - Os trabalhos que foram realizados nesta Tese procuram de um forma geral estabelecer uma metodologia para alocação de arquivos em um Sistema de Processamento de Dados Distribuídos e também um método de avaliação do desempenho do Sistema em termos do tempo de acesso ao arquivo, segundo a alocação estabelecida.

Inicialmente foram estudados os importantes fatores que determinam as formas de processamento nas empresas, as vantagens e desvantagens existentes nas arquiteturas centralizadas e distribuídas, e em seguida foram estudados os problemas que estão envolvidos com a alocação de arquivos propriamente dita, as técnicas para solução e alguns modelos propostos por diversos autores.

A metodologia apresentada em seguida, como produto final de um esforço de estudos e pesquisas as referências bibliográficas, resultou em um prático e viável método para solução e análise de um projeto de Alocação de Arquivos em um Sistema de Processamento de Dados Distribuídos. Além disso a metodologia, que foi desenvolvida dentro de um enfoque Sistêmico, torna-se bastante flexível para considerar qualquer variação de um problema de projeto de Sistemas.

Sabemos que os estudos sobre Sistemas de Processamento de Dados Distribuídos, são recentes e por isto, talvez, muita coisa ainda deve ser entendida e desenvolvida. Assim alguns fatores deixaram de ser abordados neste trabalho, como por exemplo o pro-

blema de otimização de consultas e projetos de diretórios. Nós es  
peramos contudo que este trabalho possa estimular mais pesquisas  
nesta área, tal que um mais completo modelo que combine a alocação  
de arquivos com a otimização de consultas e o projeto de diretó-  
rios, possa ser desenvolvido em um futuro próximo.

REFERÊNCIAS

1. AKOKA J. - Design of Optimal Distributed Database Systems, in Distributed Databases, North-Holland Publishing Company, INRIA, 1980.
2. ARORA S.R. and GALLO A. - Optimization of Static Loading and Sizing of Multilevel Memory Systems, J. ACM 20,2 (Apr.1973), pp. 307-319.
3. BAKER C.T. - Logical Distribution of Applications and Data, IBM Systems Journal, Vol. 19, Nº 2, 1980.
4. BERNSTEIN A.P. and GOODMAN N. - Concurrency Control in Distributed Database Systems, ACM Computing Surveys, Vol. 13(2) , 1981, pp. 185-221.
5. \_\_\_\_\_ - Timeslamp - Base Algorithms for Concurrency Control in Distributed Database Systems, Sixth International Conference on Very Large Data Bases, Montreal, October 1980.
6. BERNSTEIN A.P. et al. - Concurrency Control in Distributed Database Systems, ACM - TODBS, Vol. 5 - Nº 1, March 1980, pp. 18-51.
7. \_\_\_\_\_ - Query Processing in a System for Distributed Databases (SDD-1), ACM Transactions on Database Systems, Vol. 6(4), Dec. 1981, pp.602-625.
8. BOOTH G.M. - Distributed Information Systems, National Computer Conference, 1976, AFIPS PRESS, Vol. 45, pp. 789-794.

9. BRACHI C. et al. - Distributed Query Processing in Distributed Data Bases, ed. Draffan I. W. e Poole F., Cambridge Press, 1980.
10. BUCCI G. and STREETER D. N. - A Methodology for the Design of Distributed Information Systems, Communications of ACM, Vol. 22, N° 4, 1979.
11. BUZEN J. P. - Queueing Network Models of Multiprogramming, Ph. D. Dissertation, Div. Engineering and Applied Physics, Harvard Univ. Cambridge, Mass, 1971.
12. CASEY R. G. - Allocation of Copies of a File in an Information Network, in AFIPS Conference Proceedings Atlantic N. J., May 16-18, 1972, Spring Joint Computer Conference.
13. \_\_\_\_\_ - Design of Tree Network for Distributed Data, Proc. AFIPS 1973, NCC 42 (AFIPS PRESS, Montvale N. J. 1973, pp. 341-348).
14. CARRARESI P. and GALLO G. - Optimal Location of Files and Programs in Computer Networks, Mathematical Programming Study 20(1982) pp. 39-53.
15. CERI S., MARTELLA G. and PELAGATTI G. - Optimal File Allocation in a Computer Network: a Solution Method Based on the Knapsak Problem, Computer Network 6, (1982), pp. 345-357.
16. CELLARY W. et al. - A Multiquery Approach to Distributed Processing in a Relational Distributed Data Base Management System. In Distributed Data Bases, ed. Delobel e Lelwin, North Holland Publishing Company, 1980, pp. 99-119.

17. CHACHRA V., GHARE P.M. and MOORE J.M. - Applications of Graph Theory Algorithms, New York, North-Holland, 1979.
18. CHANDY K.M. and HEWES J.E. - File Allocation in Distributed Systems, in Proc. Int. Symp. Computer Performance Modeling, Measurement, an Evaluation (Cambridge, Mass, March 1976) pp. 10-13.
19. CHANDY K.M. and SAUER C.H. - Approximate Methods for Analyzing Queueing Network Models of Computing Systems, Computing Surveys, Vol. 10, No 3, September 1978.
20. CHEN P.P.S. - The entity - relationship model - Toward a unified view of data, ACM transactions on Database Systems, Vol. 1, No 1, pp. 9-36.
21. \_\_\_\_\_ - Optimal File Allocation in Multilevel Storage Systems, Proc. AFIPS 1973, NCC, Vol. 42, pp. 277-282.
22. CHEN P.P.S. and AKOKA J. - Optimal Design of Distributed Information Systems, IEEE Transactions on Computers, Vol. C-29, No 12, December 1980.
23. CHU W.W. - Performance of File Directory Systems for Database in Star and Distributed Network, National Computer Conference, 1976, AFIPS PRESS, Vol. 45, pp. 577-587.
24. \_\_\_\_\_ - Optimal File Allocation in a Multiple Compute Systems, IEEE Transaction on Computer, Vol. C-18, pp. 885-889.
25. CHU W.W. and HURLEY P. - Optimal Query Processing for Distributed Database Systems, IEEE Transactions on Computers, Vol. C-31, No 9, September 1982.

26. CODD E.F. - Normalized Database Structure: A Brief Tutorial, Proc. 1971, ACM Sigfidet Workshop on Data Description, Access and Control, Available from ACM.
27. COURLOIS P.J. et al. - Concurrente Control with Readers and Writers, Com. of ACM, Vol. 14, N<sup>o</sup> 10, Oct. 1971, pp. 667-668.
28. CYPSEK R.J. - Communications Architecture for Distributed Systems, Reading, Addison-Wesley Publishing Company, Cop. 1978.
29. DAVENPORT R.A. - Design of Distributed Database Systems The Computer Journal, London, Heyden & Son, Vol. 24, 1-Feb 1981, pp. 31-41.
30. \_\_\_\_\_ - Distributed or Centralized Database, The Computer Journal, Vol. 21, 1, 1978, pp. 7-14.
31. DAVIES D.W., BARDER D.L.A., PRICE W.L. and SOLOMONIDES C.M. - Computer Network and Their Protocols, Wiley, New York, 1979.
32. DENNING P.J. and BUZEN J.P. - The Operational Analysis of Queuing Network Models, Computing Surveys, Vol. 10, N<sup>o</sup> 3, September 1978.
33. DRAFFAN I.W. and POOLE F. - The Classification of Distributed Database Management Systems, In Distributed Databases, Cambridge, Cambridge University Press, 1980, Chap. 3, pp.57-81.
34. DOWDY L.W. - Optimal Branching Probabilities and Their Relationship to Computer Network File Distribution, Ph.D. Dissertation, Dep. Computer Science, Duke Univ., Durhan, N. C. 1977.



35. DOWDY L.W. and FOSTER D.V. - Comparative Models of the File Assignment Problem, Computing Surveys, Vol. 14, No 2, June 1982.
36. ENSLOW P.H. Jr. - What is a Distributed Data Processing Systems?, Computer, January 1978, pp. 13-21.
37. FISHER M.L. and HOCHBAUM D.S. - Database Location in Computer Network, Journal of Association for Computing Machinery, Vol. 27, No 4, October 1980, pp. 718-735.
38. FOSTER D.V., DOWDY L.W. and AMES J.E. - File Assignment in a Computer Network, Computer Networks 5, 1981, pp. 341-349.
39. FOSTER D.V. and BROWNE J.C. - File Assignment in Memory Hierarchies, In Proc. Modeling and Performance Evaluation of Computer Systems, Oct 1976, Beilner and Gelender, Eds. North-Holland Publ. Amsterdam, 1976, pp. 119-127.
40. FRANK H. and CHOU W. - Topological Optimization of Computer Network, Proceedings of the IEEE, November 1972.
41. FRANK H., FAHN R.E. and KLEINROCK L. - Computer Communication Network Design: Experience with Theory and Practice, Network, Vol. 2, pp. 135-166, 1972.
42. GAVISH B. - Topological Design of Centralized Computer Network Formulations and Algorithms, Network, Vol. 12, 1982, pp.355-377.
43. GERLA M. and KLEINROCK L. - On the Topological Design of Distributed Computer Network, IEEE Transactions on Communications, Vol. Com-25, No 1, January 1977.

44. GERLA M. and KLEINROCK L. - Flow Control: A Comparative Survey, IEEE Transactions on Communications, Vol. Com-28, No 4, April 1980.
45. GHOSH S.P. - Distributing a Data Base with Logical Associations on a Computer Network for Parallel Searching, IEEE Transactions on Software Engineering Piscataway, N.J. 2(2), pp.106-113.
46. GRAPA E. and BELFORD G.G. - Some Theorems to Aid in Solving the File Allocation Problem, ACM Communications, New York, NY 20(11), November 1977, pp. 878-882.
47. HATZOPOULOS M. and KOLLIAS J.G. - The File Allocation Problem Under Dynamic Usage, Information Systems Vol. 5, pp. 197-201, 1980.
48. HEVNER A., et al. - Query Processing in Distributed Database System, IEEE Transaction Soft Engin, Vol. SE-5, No 3, May 1979.
49. HUGHES P.H. and MOE G. - A Structural Approach to Computer Performance Analysis, In Proc. AFIPS 1973 Spring Jt. Computer Conf., AFIPS PRESS. Arlington, Va. pp. 109-120.
50. IGNIZIO J.P., PALMER D.F. and MURPHY C.M. - A Multicriteria Approach to Supersystem Architecture Definition, IEEE Transactions on Computers, Vol. C-31, No 5, May 1982.
51. IRANI K.B. and KHABBAZ N.G. - A Methodology for the Design of Communication Network and the Distribution of Data in Distributed Supercomputer Systems, IEEE Transactions on Computers, Vol. C-31, No 5, May 1982.

52. JONES L.G., FOSTER D.V. and KLOLAK P.D. - A Goal Programming Formulation Used in the File Assignment Problem, In Proc. R. J., Duffin Conf. (Pittsburgh, Pa. July 1978).
53. KATZAN H. - An Introduction to Distributed Data Processing, New York, Petrocelli, 1978.
54. KING J. - Evaluating Data Base Management Systems, New York, Van Nostrand Reinhold, 1981.
55. KLEINROCK L. - Queueing Systems, Vol. 2: Computer Applications, John Wiley and Sons, New York, 1976.
56. \_\_\_\_\_ - Analitic and Simulation Methods in Computer Network Design, In. Conf. Rec. Spring Joint Comput. Conf., AFIPS Conf. Proc. Vol. 36, Montvale, N.J., AFIPS PRESS, 1970, pp. 568 - 579.
57. KOLLIAS J.G. and HATZOPOULOS M. - Allocation of Copies of Distinct Files in a Information Network, Information Systems Vol. 6, N<sup>o</sup> 3, pp. 201-204, 1981.
58. \_\_\_\_\_ - Criteria to Aid in Solving the Problem of Allocating Copies of a File in a Computer Network, The Computer Journal, Vol. 24, N<sup>o</sup> 1, 1981.
59. LANING L.J. and LEONARD M.S. - File Allocation in a Distributed Computer Communication Network, IEEE Transaction on Computer, Vol. C-32, N<sup>o</sup> 3, March 1983.
60. LANN LE. G. - Consistency, Synchronization and Concurrency Control, Edit I.W. Draffan e Poole, Cambridge University Press, 1980, pp. 195-220
61. LICKLIDER J.C.R. and VEZZA A. - Application of Information Network, Proceedings of the IEEE, Vol. 66, N<sup>o</sup> 11, November 1978.

62. MAHMOUD S.A. and RIORDON J.S. - Optimal Allocation of Resources in Distributed Information Network, ACM Transaction on Database Systems, N.Y., 1(1) pp. 66-78, Mar 1976.
63. MARTELLA G., RONCHETTI B. and SCHREIDER F.A. - Availability Evaluation 1, pp. 201-211, 1981.
64. MARTIN J. - Computer Network and Distributed Processing, Software, Techniques and Architecture, Englewood Cliffs N.J., Prentice-Hall, 1981.
65. MATTHEWS D.Q. - The Design of the Management Information Systems, Princeton, Auerbach Pub. 1971.
66. MENASCE D. et al. - A Locking Protocol for Resource Coordination in Distributed Databases, ACM Transactions on Database Systems, Vol. 5, Nº 2, June 1980.
67. MORGAN H.L. and LEVIN K.D. - Optimal Program and Data Locations in Computer Network, ACM Communications, New York, Vol. 20 , Nº 5, pp. 315-322.
68. NOVAES, A.G. - Métodos de Otimização: Aplicações aos Transportes, São Paulo: Edgard Blücher, Transesp, 1978.
69. OHANA I. - Arquivos e Diretórios em Banco de Dados Distribuídos, Congresso Nacional de Processamento de Dados, 13, Rio de Janeiro, Out. 1980, Anais do XIII CNPD, RJ SUCESU, pp.57-65.
70. PRICE R.T. and OHANA I. - Alocação de Arquivos em Banco de Dados Distribuídos, Congresso Nacional de Processamento de Dados, 12 São Paulo, 8.12, Out. 1979, Anais do XII CNPD São Paulo, SUCESU, pp. 273-281, 1979.

71. RAMAMOORTHY C.V. and WAH B.W. - The Isomorphism of Simple File Allocation, IEEE Transactions on Computers, Vol. C-32, No 3, March 1983.
72. \_\_\_\_\_ - Data Management in Distributed Databases, In AFIPS Conference Proceedings, New York, June 4-7. 1979, National Computer Conference, Montvale, N.J., AFIPS PRESS, 1979, Vol. 48, pp. 667-680.
73. ROSENKVANTZ D.J. et al., - A System level Concurrency for Distributed Database Systems, Proceeding of 2<sup>o</sup> Berkeley Workshop on Distributed Data Management and Computer Network, May 1977, pp. 132-145.
74. SAMARI N.K. & SCHNEIDER G.M. - The Analysis of Distributed Computer Network Using M/D/R and M/M/1 Queues. In: International Conference on Distributed Computing Systems Huntsville, Alabama, Oct. 1-5, 1979. Proceedings. Piscataway, IEEE-PRESS, 1979, pp. 143-155.
75. SCHERR A.L. - Distributed Data Processing, IBM Systems Journal, Vol. 17, No 4, 1978.
76. SCHREIBER F.A. et al. - Distributed Database Applications: On Overview, In Draffan, J.W. & Poole, F. Distributed Databases, Cambridge, Cambridge University Press, 1980, Chap. 12, pp. 323-336.
77. SPACCAPIETRA S. - Heterogeneous Database Distribution, In Draffan I. W. & Poole F., Distributed Databases, Cambridge, Cambridge University Press., 1980, Chap.6, pp. 155-193.

78. SWARTZLANDER E.E.Jr. and GILBERT B.K. - Supersystems: Technology and Architecture, IEEE Transaction on Computers, Vol. C-31, Nº 5, May 1982.
79. TANENBAUM A.S. - Computer Network, Prentice-Hall, Englewood Cliffs, N.J., 1981.
80. \_\_\_\_\_ - Network Protocols, Computer Surveys, Vol. 13, Nº 4 December 1981.
81. TAROUÇO L.M.R. - Rede de Comunicação de Dados, Rio de Janeiro, Livros Técnicos e Científicos, 1977.
82. THIÉRAUF R.J. - Distributed Processing Systems, Englewood Cliffs, Prentice-Hall, 1978.
83. WAINWRIGHT, ROBERT - Practical Engineering of a Data Communication Network, El Toro, Institute for Advanced Technology , 1974.
84. WHITNEY, V.K.M. - A Study of Optimal File Assignment and Communication in Remote-Access Computer Message Processing and Communications Systems. Sel Tech. Rep. Nº 48, U. of Michigan, Ann Arbor, Mich. Sept. 1970.
85. WILKOV R.S. - Analysis and Design of Reliable Computer Networks, IEEE Transactions on Communications, June 1972, pp. 158-176.
86. WONG J.W. - Queueing Network Modeling of Computer Communication Network, Computing Surveys, Vol. 10, Nº 3, September 1978.

87. WONG E. and YOUSSEFI K. - Decomposition - A Strategy for Query Processing, ACM Transaction Database Systems, Vol. 1, N<sup>o</sup> 3, September 1976.
88. YEH R.T. and CHANDY M.K. - On the Design of Elementary Distributed Systems, Computer Network 3, (1979) pp. 24-35.
89. ZIEGLER K. Jr. - A Distributed Information System Study, IBM Systems Journal, Vol. 18, N<sup>o</sup> 3, 1979.

A N E X O S

. ANEXO A	.....	A1
. ANEXO B	.....	B1
. ANEXO C	.....	C1
. ANEXO D	.....	D1



## ANEXO - A

### I - PARTE

Na primeira parte deste anexo apresentaremos a listagem fonte do programa, codificado em FORTRAN, e que determina a alocação dos arquivos em um Sistema de Processamento de Dados Distribuído. Embora a linguagem de programação FORTRAN possua reduzidos recursos de estruturação, o que dificulta o entendimento para que deseje ler o programa, a escolha desta linguagem, para execução dos trabalhos, deveu-se ao fato do FORTRAN possuir um relativo grau de portabilidade, o que facilita a sua utilização na maioria dos computadores.

### II - PARTE

Na segunda parte deste anexo apresentaremos os resultados de um exemplo da aplicação do método de Alocação de Arquivos, segundo o enfoque descrito nesta tese. Embora esses dados não representem a realidade de qualquer sistema existente, eles servem contudo para exemplificar a aplicação do método. Salientamos também que para impressão dos resultados, foram utilizados os recursos de um ambiente CMS do Sistema 4341, afim de oferecer uma melhor apresentação.

I - PARTE

LISTAGEM DO PROGRAMA DE ALOCAÇÃO



```

C.....PROCEDIMENTO PARA LER, ARMAZENAR E IMPRIMIR AS MATRIZES DE
C-----CUSTO DE COMUNICACAO
C-----CUSTO DE ARMAZENAMENTO
C-----FREQUENCIA DE ACESSO P/ CONSULTAS DE TRANSACOES PARA ARQUIVOS
C-----FREQUENCIA DE ACESSO P/ ATUALIZACOES DE TRANSACOES PARA ARQUIVOS
C-----FREQUENCIA DAS TRANSACOES EM CADA NODO
C-----MATRIZ DE DISTANCIAS ENTRE OS NODOS
C-----MATRIZ DE DIRECIONAMENTOS
C-----MATRIZ DAS CAPACIDADES DAS LINHAS DE COMUNICACOES
C.....LEITURA E IMPRESSAO DA MATRIZ DE COMUNICACAO
C
DO 32 I=1,6
  READ(9,5003)(MATCOM(I,J),J=1,5)
  WRITE(W,6002)
DO 33 I=1,6
  WRITE(W,6009)(MATCOM(I,J),J=1,5)
C.....LEITURA E IMPRESSAO DA MATRIZ DE CUSTOS ARMAZENAMENTO
C
DO 35 K=1,F
  READ(10,5003)(MATARM(K,I),I=1,N)
  WRITE(W,6003)
DO 36 K=1,F
  WRITE(W,6009)(MATARM(K,I),I=1,N)
C.....LEITURA DO TAMANHO DOS ARQUIVOS
C
READ(18,5002)(TAMARQ(K),K=1,F)
WRITE(W,6018)
WRITE(W,6010)(TAMARQ(K),K=1,F)
C.....LEITURA E IMPRESSAO DAS CAPACIDADES DE ARMAZENAMENTO
C
DO 41 READ(19,5002)(CAPARM(I),I=1,N)
  WRITE(W,6019)
  WRITE(W,6010)(CAPARM(I),I=1,N)
DO 43 I=1,N
  VEICAP(I)=CAPARM(I)
 43 CONTINUE
C.....LEITURA E IMPRESSAO DA MATRIZ DE FREQUENCIA DE CONSULTA
C-----DE TRANSACOES PARA ARQUIVOS
C
DO 45 K=1,I
  READ(11,5002)(MATFFC(K,I),I=1,F)
  WRITE(W,6004)
DO 46 K=1,I
  WRITE(W,6010)(MATFFC(K,I),I=1,F)
C.....LEITURA E IMPRESSAO DA MATRIZ DE FREQUENCIA DE ATUALIZACAO
C-----DE TRANSACOES PARA ARQUIVOS
C
DO 50 K=1,T

```

SPD000550  
 SPD000560  
 SPD000570  
 SPD000580  
 SPD000590  
 SPD000600  
 SPD000610  
 SPD000620  
 SPD000630  
 SPD000640  
 SPD000650  
 SPD000660  
 SPD000670  
 SPD000680  
 SPD000690  
 SPD000700  
 SPD000710  
 SPD000720  
 SPD000730  
 SPD000740  
 SPD000750  
 SPD000760  
 SPD000770  
 SPD000780  
 SPD000790  
 SPD000800  
 SPD000810  
 SPD000820  
 SPD000830  
 SPD000840  
 SPD000850  
 SPD000860  
 SPD000870  
 SPD000880  
 SPD000890  
 SPD000900  
 SPD000910  
 SPD000920  
 SPD000930  
 SPD000940  
 SPD000950  
 SPD000960  
 SPD000970  
 SPD000980  
 SPD000990  
 SPD010000  
 SPD010100  
 SPD010200  
 SPD010300  
 SPD010400  
 SPD010500  
 SPD010600  
 SPD010700  
 SPD010800  
 SPD010900

```

50 READ(12,5002)(MATFIA(K,I),I=1,F)
   WRITE(W,6005)
DO 52 K=1,I
52 WRITE(W,6010)(MATFIA(K,I),I=1,F)
C... LEITURA E IMPRESSAO DA MATRIZ DE FREQUENCIAS DAS TRANSACOES EM
C... CADA NODO
DO 55 K=1,I
55 READ(13,5002)(MATFIN(K,I),I=1,N)
   WRITE(W,6006)
DO 56 K=1,I
56 WRITE(W,6010)(MATFIN(K,I),I=1,N)
C... LEITURA E IMPRESSAO DA MATRIZ DISTANCIAS
C...
DO 57 I=1,N
57 READ(20,5000)(MATDIS(I,J),J=1,N)
   WRITE(W,6020)
DO 58 I=1,N
58 WRITE(W,6010)(MATDIS(I,J),J=1,N)
C... LEITURA E IMPRESSAO DA MATRIZ DE DIRECIONAMENTO
C...
DO 59 I=1,N
59 READ(21,5005)(MATDIR(I,J),J=1,N)
   WRITE(W,6021)
DO 60 I=1,N
60 WRITE(W,6010)(MATDIR(I,J),J=1,N)
C... LEITURA E IMPRESSAO DAS CAPACIDADES DAS LINHAS DE COMUNICACOES
C...
DO 61 I=1,N
61 READ(22,5000)(MATLIN(I,J),J=1,N)
   WRITE(W,6022)
DO 62 I=1,N
62 WRITE(W,6010)(MATLIN(I,J),J=1,N)
C... PROCEDIMENTO PARA CALCULAR AS MATRIZES DEFREQUENCIAS DE
C... CONSULTAS E ATUALIZACOES DOS NODOS PARA OS ARQUIVOS
DO 67 K=1,F
DO 66 I=1,N
   MATFOC(K,I)=0
   MATFOA(K,I)=0
DO 65 J=1,I
   MATFOC(K,I)=MATFOC(K,I)+(MATFIN(J,I)*MATFIC(J,K))
   MATFOA(K,I)=MATFOA(K,I)+(MATFIN(J,I)*MATFIA(J,K))
65 CONTINUE
66 CONTINUE
67 CONTINUE
C... IMPRIMIR AS MATRIZES DE FREQUENCIAS DE CONSULTAS E ATUALIZACOES
C...
WRITE(W,6007)

```

SPD01650  
 SPD01660  
 SPD01670  
 SPD01680  
 SPD01690  
 SPD01700  
 SPD01710  
 SPD01720  
 SPD01730  
 SPD01740  
 SPD01750  
 SPD01760  
 SPD01770  
 SPD01780  
 SPD01790  
 SPD01800  
 SPD01810  
 SPD01820  
 SPD01830  
 SPD01840  
 SPD01850  
 SPD01860  
 SPD01870  
 SPD01880  
 SPD01890  
 SPD01900  
 SPD01910  
 SPD01920  
 SPD01930  
 SPD01940  
 SPD01950  
 SPD01960  
 SPD01970  
 SPD01980  
 SPD01990  
 SPD02000  
 SPD02010  
 SPD02020  
 SPD02030  
 SPD02040  
 SPD02050  
 SPD02060  
 SPD02070  
 SPD02080  
 SPD02090  
 SPD02100  
 SPD02110  
 SPD02120  
 SPD02130  
 SPD02140  
 SPD02150  
 SPD02160  
 SPD02170  
 SPD02180  
 SPD02190

```

68      DC 68, K=1, F
        WRITE(14,6016)(MATFQC(K, I), I=1, N)
        WRITE(W,6010)(MATFQC(K, I), I=1, N)
        DC 69, K=1, F
        WRITE(15,6016)(MATFQA(K, I), I=1, N)
        WRITE(W,6010)(MATFQA(K, I), I=1, N)
C.....PROCEDIMENTO PARA GERAR A MATRIZ DE CUSTOS DE TRANSMISSAO
C
        WRITE(W,6002)
        DO 110 I=1, N
          DO 109 J=1, N
            JJ=I
            IF(I-J) 75, 109, 75
            IF(MATLIN(II, JJ)) 82, 80, 82
            JJ=MATDIR(II, JJ)
            IF(MATLIN(II, JJ)) -1200, 83, 84, 85
            JJJ=1
            GO TO 89
          JJJ=2
            GO TO 89
          JJJ=3
            GO TO 89
          JJJ=4
            GO TO 89
          JJJ=5
            IF(MATDIS(II, JJ)-50) 90, 90, 92
            II=1
            GO TO 102
          IF(MATDIS(II, JJ)-100) 93, 93, 94
            III=2
            GO TO 102
          IF(MATDIS(II, JJ)-300) 95, 95, 96
            III=3
            GO TO 102
          IF(MATDIS(II, JJ)-700) 97, 97, 98
            III=4
            GO TO 102
          IF(MATDIS(II, JJ)-1500) 99, 99, 100
            III=5
            GO TO 102
            III=6
            MATCST(I, J)=MATCST(II, J)+MATCOM(III, JJJ)
            IF(MATDIR(II, JJ)-J) 105, 109, 105
            II=JJ
            JJJ=J
            GO TO 75
          CONTINUE
        WRITE(W,6009)(MATCST(I, J), J=1, N)
        110 CONTINUE
C.....PROCEDIMENTO PARA GERAR E IMPRIMIR A MATRIZ DE EFICIENCIA

```

```

C..  TORRAR A MATRIZ DE EFICIENCIA QUADRATICA
C
112  SF=F
     SN=N
     IF(F-N)114,118,116
114  F=N
     GO TO 118
116  N=F

C..  GERAR OS ELEMENTOS DA MATRIZ DE CUSTO
C
118  DO 140 K=1,F
     DO 130 I=1,N
     CUSTKI=0
     CUSTAR=0
     DO 120 J=1,N
     CUSTKI=ANINT(CUSTKI+((MATFOA(K,J)*TAMATUJ)+
+      (MATFOC(K,J)*{(TAMCON+TAMRET)}))*MATCSI(J,I))
120  CONTINUE
122  IF(CAPARM(I)-TAMARQ(K))122,125,125
     MATEFI(K,I)=999999
     GO TO 129
125  CUSTAR=ANINT(TAMARQ(K)*1000000*MATARM(K,I))
     MATEFI(K,I)=CUSTKI+CUSTAR
129  MATCKI(K,I)=MATEFI(K,I)
130  CONTINUE
140  CONTINUE

C..  IMPRIMIR A MATRIZ DE CUSTOS
C
     WRITE(M,6011)
     DO 145 K=1,SF
     WRITE(M,6010)(MATEFI(K,I),I=1,SN)

C..  INICIO DOS PROCEDIMENTOS PARA ALOCCAD SEM REDUNDANCIA
C
C..  PESQUISAR O ELEMENTO MINIMO DE CADA LINHA E SUBTRAIR ESTE VALOR
C     DE TODOS OS ELEMENTOS DA LINHA
149  DO 190 K=1,F
     MIN=MATEFI(K,1)
     DO 160 I=1,N
     IF(MATEFI(K,I)-MIN)150,150,160
     MIN=MATEFI(K,I)
150  CONTINUE
160  IF(MIN)170,190,170
     DO 180 I=1,N
     MATEFI(K,I)=MATEFI(K,I)-MIN
170  CONTINUE
180  CONTINUE
190  CONTINUE

C..  PESQUISAR O ELEMENTO MINIMO DE CADA COLUNA E SUBTRAIR ESTE VALOR
C     DE TODOS OS ELEMENTOS DA COLUNA

```

```

SPD02200
SPD02210
SPD02220
SPD02230
SPD02240
SPD02250
SPD02260
SPD02270
SPD02280
SPD02290
SPD02300
SPD02310
SPD02320
SPD02330
SPD02340
SPD02350
SPD02360
SPD02370
SPD02380
SPD02390
SPD02400
SPD02410
SPD02420
SPD02430
SPD02440
SPD02450
SPD02460
SPD02470
SPD02480
SPD02490
SPD02500
SPD02510
SPD02520
SPD02530
SPD02540
SPD02550
SPD02560
SPD02570
SPD02580
SPD02590
SPD02600
SPD02610
SPD02620
SPD02630
SPD02640
SPD02650
SPD02660
SPD02670
SPD02680
SPD02690
SPD02700
SPD02710
SPD02720
SPD02730
SPD02740

```

```

DO 240 I=1,N
  MIN=MATEFI(1,I)
DO 210 K=1,F
  IF(MATEFI(K,I)-MIN)200,200,210
  MIN=MATEFI(K,I)
CONTINUE
200 IF(MIN)220,240,220
210 DO 230 K=1,F
220 MATEFI(K,I)=MATEFI(K,I)-MIN
230 CONTINUE
240 C..  OBTER O NUMERO DE ZEROS DE CADA LINHA E CADA COLUNA
C..
C.. 999 NUMDSG=0
DO 370 K=1,F
DO 360 I=1,N
  IF(MATEFI(K,I))360,300,360
  NUZLIN(K)=NUZLIN(K)+1
  NUZCOL(I)=NUZCOL(I)+1
  IF(NUZLIN(K)-1)310,310,320
  INDCOL(K)=I
  ATRLIN(K)=1
  GO TO 330
  ATRLIN(K)=0
  INDCOL(K)=0
300 IF(NUZCOL(I)-1)340,340,350
310 INDLIN(I)=K
  ATRCOL(I)=1
  GO TO 360
  ATRCOL(I)=0
  INDLIN(I)=0
320 CONTINUE
370 C..  DESIGNACAO FINAL EM CADA LINHA SEMPRE QUE POSSIVEL
C..
C.. 400 DO 430 K=1,F
  IF(ATRLIN(K)-1)430,410,430
  IF(NUZLIN(K))420,430,420
  I=INDCOL(K)
  CALL DESIGN(K,I)
  CALL ELIMIN(K,I)
  CONTINUE
430 C..  DESIGNACAO FINAL EM CADA COLUNA SEMPRE QUE POSSIVEL
C..
C.. DO 450 I=1,N
  IF(ATRCOL(I)-1)450,440,450
  IF(NUZCOL(I))445,450,445
  K=INDLIN(I)
  CALL DESIGN(K,I)
  CALL ELIMIN(K,I)
  CONTINUE
450 C..  VERIFICAR SE AINDA EXISTE LINHAS E COLUNAS COM APENAS UM ELEMENTO
C..

```



```

C      NULO QUE ESTEJA DISPONIVEL PARA DESIGNACAO. CASO ISID OCORRA DEVE-
C      RETORNAR AO COMANDO 400
C
C      DO 460 K=1,F
C      IF(NUZCOL(K)-1)460,400,460
C      CONTINUE
C      DO 470 I=1,N
C      IF(NUZLIN(I)-1)470,400,470
C      CONTINUE
C
C      VERIFICAR SE A DESIGNACAO ESTA COMPLETA
C
C      IF(NUMDSG-N)480,900,900
C
C      VERIFICAR SE NAO EXITE MAIS ELEMENTOS NULOS DISPONIVEIS PARA
C      DESIGNACAO
C
C      DO 480 DO 540 K=1,F
C      IF(NUZLIN(K))490,540,490
C      DO 530 I=1,N
C      IF(ATRCOL(I))530,500,530
C      IF(MATEFI(K,I))530,520,530
C      CALL DESIGN(K,I)
C      CALL ELIMIN(K,I)
C      GO TO 400
C      CONTINUE
C      540 CONTINUE
C
C      MARCAR TODAS AS LINHAS QUE NAO TENHAM DESIGNACAO
C
C      DO 610 K=1,F
C      IF(ATRLIN(K))610,600,610
C      NUZLIN(K)=1
C      CONTINUE
C      600
C      610 CONTINUE
C
C      MARCAR TODAS AS COLUNAS QUE TENHAM ZEROS EM LINHAS MARCADAS
C
C      ALTER=0
C      DO 670 K=1,F
C      IF(NUZLIN(K)-1)670,630,670
C      DO 660 I=1,N
C      IF(MATEFI(K,I))660,640,660
C      IF(NUZCOL(I)-1)650,660,650
C      NUZCOL(I)=1
C      ALTER=1
C      CONTINUE
C      670 CONTINUE
C
C      MARCAR TODAS AS LINHAS QUE TENHAM DESIGNACOES EM COLUNAS MARCADAS
C
C      DO 730 I=1,N
C      IF(NUZCOL(I)-1)730,700,730
C      IF(ATRCOL(I))710,730,710
C      K=INDLIN(I)
C      IF(NUZLIN(K)-1)720,730,720
C      700
C      710

```

```

SPD003300
SPD003310
SPD003320
SPD003330
SPD003340
SPD003350
SPD003360
SPD003370
SPD003380
SPD003390
SPD003400
SPD003410
SPD003420
SPD003430
SPD003440
SPD003450
SPD003460
SPD003470
SPD003480
SPD003490
SPD003500
SPD003510
SPD003520
SPD003530
SPD003540
SPD003550
SPD003560
SPD003570
SPD003580
SPD003590
SPD003600
SPD003610
SPD003620
SPD003630
SPD003640
SPD003650
SPD003660
SPD003670
SPD003680
SPD003690
SPD003700
SPD003710
SPD003720
SPD003730
SPD003740
SPD003750
SPD003760
SPD003770
SPD003780
SPD003790
SPD003800
SPD003810
SPD003820
SPD003830
SPD003840

```

```

720 ALIER=1
    NUZLIN(K)=1
730 CONTINUE
    IF(ALTER)620,750,620
C.....
C.....
C.....
C.....
750 MIN=999999
    DO 800 K=1,F
        IF(NUZLIN(K))760,800,760
    DO 790 I=1,N
        IF(NUZCOL(I)-1)770,790,770
        IF(MATEFI(K,I)-MIN)780,780,790
770     MIN=MATEFI(K,I)
780     CONTINUE
790     CONTINUE
800 CONTINUE
C.....
C.....
C.....
C.....
810 DO 810 K=1,F
        DO 810 I=1,N
            MATEFI(K,I)=MATEFI(K,I)-MIN
820 DO 850 K=1,F
        IF(NUZLIN(K)-1)830,850,830
830 DO 840 I=1,N
        MATEFI(K,I)=MATEFI(K,I)+MIN
840 CONTINUE
850 DO 880 I=1,N
        IF(NUZCOL(I))860,880,860
860 DO 870 K=1,F
        MATEFI(K,I)=MATEFI(K,I)+MIN
870 CONTINUE
880 CONTINUE
C.....
C.....
C.....
C.....
890 DO 890 I=1,N
        NUZLIN(I)=0
        NUZCOL(I)=0
890 CONTINUE
    GO TO 999
C.....
C.....
C.....
C.....
900 DO 960 K=1,F
        DO 950 I=1,N
            IF(I-INDCOL(K))950,915,950
            IF(I-SN)916,916,920
            IF(VETCAP(I)-TAMARQ(K))920,917,917
            MATALC(K,I)=1
            VETCAP(I)=VETCAP(I)-TAMARQ(K)
            GO TO 960
        MIN=999999999
915
916
917
920

```

```

PD03850
SPD03860
SPD03870
SPD03880
SPD03890
SPD03900
SPD03910
SPD03920
SPD03930
SPD03940
SPD03950
SPD03960
SPD03970
SPD03980
SPD03990
SPD04000
SPD04010
SPD04020
SPD04030
SPD04040
SPD04050
SPD04060
SPD04070
SPD04080
SPD04090
SPD04100
SPD04110
SPD04120
SPD04130
SPD04140
SPD04150
SPD04160
SPD04170
SPD04180
SPD04190
SPD04200
SPD04210
SPD04220
SPD04230
SPD04240
SPD04250
SPD04260
SPD04270
SPD04280
SPD04290
SPD04300
SPD04310
SPD04320
SPD04330
SPD04340
SPD04350
SPD04360
SPD04370
SPD04380
SPD04390

```

```

DO 930 J=1,SN
  IF(MATCKI(K,J))930,924,924
  IF(MATCKI(K,J)-MIN)925,925,930
  MIN=MATCKI(K,J)
  JJ=J
CONTINUE
IF(MIN-9999999)934,945,934
IF(VETCAP(JJ)-TAMARQ(K))940,935,935
MATALC(K,JJ)=1
VETCAP(JJ)=VETCAP(JJ)-TAMARQ(K)
GO TO 960
MATCKI(K,JJ)=MATCKI(K,JJ)*(-1)
GO TO 920
WRITE(W,*)'NAO FOI POSSIVEL DETERMINAR A ALOCACAO'
GO TO 961
CONTINUE
CONTINUE
C.. GRAVAR E IMPRIMIR A MATRIZ DE ALOCACAO SEM REDUNDANCIA
C
C
961 WRITE(W,6012)
DO 962 K=1,SF
  WRITE(W,6010)(MATALC(K,I),I=1,SN)
  WRITE(I16,6017)(MATALC(K,I),I=1,SN)
962 CONTINUE
C.....IMPRIMIR O CUSTO DA ALOCACAO SEM REDUNDANCIA
C
C
Z=0
DO 965 K=1,SF
  DO 965 I=1,SN
    IF(MATALC(K,I))963,965,963
    Z=Z+MATCKI(K,I)
963 CONTINUE
965 WRITE(W,6014)Z
C.....SALVAR A CAPACIDADE DE ARMAZENAMENTO
C
C
DO 970 I=1,SN
  SALVAR(I)=VETCAP(I)
  SALARM(I)=VETCAP(I)
970
C.....INICIO DOS PROCEDIMENTOS PARA DETERMINAR A MATRIZ DE ALOCACAO
C.....ALOCACAO COM REPLICACAO E O SEU CUSTO DE ALOCACAO
C
C
1000 CONTRO=0
N=SN
F=SF
DO 1001 K=1,F
  VETARQ(K)=K
1001 CONTINUE
VETDSGIA)=9999999
DO 1300 KKK=1,F
1002 Z=0

```

SPD04400  
SPD04410  
SPD04420  
SPD04430  
SPD04440  
SPD04450  
SPD04460  
SPD04470  
SPD04480  
SPD04490  
SPD04500  
SPD04510  
SPD04520  
SPD04530  
SPD04540  
SPD04550  
SPD04560  
SPD04570  
SPD04580  
SPD04590  
SPD04600  
SPD04610  
SPD04620  
SPD04630  
SPD04640  
SPD04650  
SPD04660  
SPD04670  
SPD04680  
SPD04690  
SPD04700  
SPD04710  
SPD04720  
SPD04730  
SPD04740  
SPD04750  
SPD04760  
SPD04770  
SPD04780  
SPD04790  
SPD04800  
SPD04810  
SPD04820  
SPD04830  
SPD04840  
SPD04850  
SPD04860  
SPD04870  
SPD04880  
SPD04890  
SPD04900  
SPD04910  
SPD04920  
SPD04930  
SPD04940

```

1005 DO 1220 KK=1,F
      K=VETARQ(K,K)
      IF(K)1005,1220,1005
      VETALC(A)=999999
      DO 1170 I=1,N
        IF(MATALC(K,I))1008,1170,1008
        I2=I
        I1=I
        ZR=MATCKI(K,I)
1009 DO 1010 J=1,N
      VETALC(J)=0
      VETIND(J)=J
      MATIND(I,J)=I
      CONTINUE
1010 VETALC(I)=1
1015 DO 1120 L=1,N
      DO 1070 V=1,N
        CUSTRZ=0
        CUSTAR=0
        X=VETIND(V)
        IF(1-VETALC(X))1060,1060,1020
        IF(1-VETCAP(X)-TAMARQ(K))1060,1025,1025
        CUSTAR=ANINT(TAMARQ(K)*1000000/MATARM(K,X))
1020 DO 1040 J=1,N
      CUSTAZ=CUSTAZ+(MATFQA(K,J)*TAMATU)*MATCST(J,X)
      MATIND(X,J)=MATIND(I2,J)
      Y=MATIND(I2,J)
      IF(MATCST(J,X)-MATCST(J,Y))1030,1040,1040
1030 * CUSTAZ=ANINT(CUSTAZ+(MATFQC(K,J)*TAMCON+TAMRET))*
      * CUSTRZ=ANINT(CUSTRZ+(MATFQC(K,J)*TAMCON+TAMRET))*
      MATIND(X,J)=X
      CONTINUE
1040 VETDIF(V)=CUSTRZ-CUSTAZ-CUSTAR
      IF(VETDIF(V))1060,1070,1070
      VETDIF(V)=0
      CONTINUE
      S=0
      DO 1100 P=1,M
        IF(VETDIF(P)-VETDIF(P+1))1090,1100,1100
        S=1
        VETEMP(1)=VETDIF(P+1)
        VETEMP(2)=VETIND(P+1)
        VETDIF(P+1)=VETDIF(P)
        VETIND(P+1)=VETIND(P)
        VETDIF(P)=VETEMP(1)
        VETIND(P)=VETEMP(2)
        CONTINUE
1100 IF(S)1080,1105,1080
1105 IF(VETDIF(1))1130,1130,1115
1115 I2=VETIND(1)
      VETALC(I2)=1
      ZR=ZR-VETDIF(1)

```

SPD04950  
SPD04960  
SPD04970  
SPD04980  
SPD04990  
SPD05000  
SPD05010  
SPD05020  
SPD05030  
SPD05040  
SPD05050  
SPD05060  
SPD05070  
SPD05080  
SPD05090  
SPD05100  
SPD05110  
SPD05120  
SPD05130  
SPD05140  
SPD05150  
SPD05160  
SPD05170  
SPD05180  
SPD05190  
SPD05200  
SPD05210  
SPD05220  
SPD05230  
SPD05240  
SPD05250  
SPD05260  
SPD05270  
SPD05280  
SPD05290  
SPD05300  
SPD05310  
SPD05320  
SPD05330  
SPD05340  
SPD05350  
SPD05360  
SPD05370  
SPD05380  
SPD05390  
SPD05400  
SPD05410  
SPD05420  
SPD05430  
SPD05440  
SPD05450  
SPD05460  
SPD05470  
SPD05480  
SPD05490

```

1116 VETCAP(I12)=VETCAP(I2)-TAMARQ(K)
1117 IF(VETCAP(I2))I116,I120,I120
1118 VETINUE
1119 VETALC(A)=ZR
1120 GO TO I1175
1121 CONTINUE
1122 VETALC(A)=ZR
1123 GO TO I1175
1124 CONTINUE
1125 IF(CONTRO)I176,I181,I176
1126 IF(I177 J=1,N
1127 DO MATALC(K,J)=VETALC(J)
1128 Z=Z+VETALC(A)
1129 GO TO I220
1130 IF(VETALC(A))-VETDSC(A))I182,I182,I184
1131 DO I183 I=1,A
1132 VETDSC(I)=VETALC(I)
1133 SK=K
1134 SKK=KK
1135 DO I185 I=1,N
1136 VETCAP(I)=SALARM(I)
1137 CONTINUE
1138 GO TO I220
1139 CONTINUE
1140 IF(CONTRO)1500,1230,1500
1141 VETESP(KKK)=SK
1142 DO I1250 I=1,N
1143 IF(I-I1)I235,1250,1235
1144 IF(VETDSC(I))I240,I250,1240
1145 SALARM(I)=SALARM(I)-TAMARQ(SK)
1146 VETCAP(I)=SALARM(I)-TAMARQ(SK)
1147 CONTINUE
1148 VETARQ(SKK)=0
1149 VETDSC(A)=99999999
1150 CONTINUE
1151 DO I1350 K=1,F
1152 VETARQ(K)=VETESP(K)
1153 CONTINUE
1154 CONTRO=1
1155 DO I1360 I=1,N
1156 SALARM(I)=SALVAR(I)
1157 VETCAP(I)=SALVAR(I)
1158 CONTINUE
1159 GO TO I002
1160 CONTINUE
C.....IMPRIMIR A MATRIZ DE ALOCACAO COM REPLICAO E O CUSTO Z
C
1500 WRITE(W,6013)
DO I1600 K=1,F
WRITE(W,6010)(MATALC(K,J),J=1,N)
WRITE(I17,6017)(MATALC(K,J),J=1,N)
1600 CONTINUE
C
WRITE(W,6015)Z
C
STOP
C

```

SPD05500  
SPD05510  
SPD05520  
SPD05530  
SPD05540  
SPD05550  
SPD05560  
SPD05570  
SPD05580  
SPD05590  
SPD05600  
SPD05610  
SPD05620  
SPD05630  
SPD05640  
SPD05650  
SPD05660  
SPD05670  
SPD05680  
SPD05690  
SPD05700  
SPD05710  
SPD05720  
SPD05730  
SPD05740  
SPD05750  
SPD05760  
SPD05770  
SPD05780  
SPD05790  
SPD05800  
SPD05810  
SPD05820  
SPD05830  
SPD05840  
SPD05850  
SPD05860  
SPD05870  
SPD05880  
SPD05890  
SPD05900  
SPD05910  
SPD05920  
SPD05930  
SPD05940  
SPD05950  
SPD05960  
SPD05970  
SPD05980  
SPD05990  
SPD06000  
SPD06010  
SPD06020  
SPD06030  
SPD06040

```

C.....DESCRICAO DOS FORMATOS.
C
5000 FORMAT(20I4)
5001 FORMAT(I2)
5002 FORMAT(20I3)
5003 FORMAT(20F4.3)
5004 FORMAT(3I3)
5005 FORMAT(20I2)
6002 FORMAT(IH1,5X,'TABELA CUSIO DE TRANSMISSAO')
6003 FORMAT(IH1,5X,'TABELA CUSIO DE ARMAZENAMENTO')
6004 +FORMAT(IH1,5X,'TABELA DE FREQUENCIA CONSULTA TRANSACOES/ARQUIVOS(I
+RANSACOES/HORA)')
6005 +FORMAT(IH1,5X,'TABELA DE FREQUENCIA ATUALIZACOES TRANSACOES/ARQUIV
+OS(TRANSACOES/HORA)')
6006 FORMAT(IH1,5X,'TABELA DE FREQUENCIA DAS TRANSACOES NOS NODOS')
6008 FORMAT(IH1,5X,'MATRIZ DE CONSULTAS(NODO/ARQUIVO)')
6009 FORMAT(IH1,5X,'MATRIZ DE FREQUENCIA DE ATUALIZACOES(NODO/ARQ.)')
6010 FORMAT(I0,5X,20F7.3/)
6011 FORMAT(IH1,5X,'MATRIZ DE C U S T O S')
6012 FORMAT(IH1,5X,'MATRIZ DE ALOCACAO SEM REDUNDANCIA')
6013 FORMAT(IH1,5X,'MATRIZ DE ALOCACAO COM REDUNDANCIA')
6014 FORMAT(IH1,5X,'CUSTO DE ALOCACAO SEM REDUNDANCIA Z=,18)
6015 FORMAT(IH1,5X,'CUSTO DE ALOCACAO COM REDUNDANCIA Z=,18)
6016 FORMAT(20I4)
6017 FORMAT(20I1)
6018 FORMAT(IH1,5X,'TAMANHO DOS ARQUIVOS')
6019 FORMAT(IH1,5X,'CAPACIDADES DE ARMAZENAMENTOS')
6020 FORMAT(IH1,5X,'MATRIZ DE DISTANCIAS (KM)')
6021 FORMAT(IH1,5X,'MATRIZ DE DIRECIONAMENTO')
6022 FORMAT(IH1,5X,'CAPACIDADE DAS LINHAS DE COMUNICACOES(BP S)')
6000 FORMAT(I18)
C.....FIM DO PROGRAMA PRINCIPAL
C
C
C
C
C.....SUBROTINA PARA FAZER A DESIGNACAO DE UM ARQUIVO PARA UM NODO
C
SUBROUTINE DESIGN(KL,IC)
DIMENSION NUZLIN(20),NUZCOL(20),INDLIN(20),INDCOL(20)
INTEGER ATRLIN(20),ATRCOL(20),NUMDSG
COMMON NUZLIN,NUZCOL,INDCOL,INDLIN,ATRLIN,ATRCOL,NUMDSG
NUZLIN(KL)=0
ATRLIN(KL)=1
INDCOL(KL)=IC
NUZCOL(IC)=0
ATRCOL(IC)=1
INDLIN(IC)=KL
NUMDSG=NUMDSG+1
RETURN
C

```

```

SPD060650
SSPD060660
SSPD060670
SSPD060680
SSPD060690
SSPD06100
SSPD06110
SSPD06120
SSPD06130
SSPD06140
SSPD06150
SSPD06160
SSPD06170
SSPD06180
SSPD06190
SSPD06200
SSPD06210
SSPD06220
SSPD06230
SSPD06240
SSPD06250
SSPD06260
SSPD06270
SSPD06280
SSPD06290
SSPD06300
SSPD06310
SSPD06320
SSPD06330
SSPD06340
SSPD06350
SSPD06360
SSPD06370
SSPD06380
SSPD06390
SSPD06400
SSPD06410
SSPD06420
SSPD06430
SSPD06440
SSPD06450
SSPD06460
SSPD06470
SSPD06480
SSPD06490
SSPD06500
SSPD06510
SSPD06520
SSPD06530
SSPD06540
SSPD06550
SSPD06560
SSPD06570
SSPD06580
SSPD06590

```



```

C.....FIM DA SUBROTINA DESIGN
C
C      END
C
C.....SUBROTINA PARA ELIMINAR OS DEMAIS ELEMENTOS NULOS DA LINHA E
C      COLUNA APOS UMA DESIGNACAO
C
SUBROUTINE ELIMIN(KK, II)
DIMENSION NUZLIN(20), NUZCOL(20), INDLIN(20), INDCOL(20),
1  INTEGER ATRLIN(20), ATRCOL(20), NUMDSG, N, F, A, B
COMMON NUZLIN, NUZCOL, INDCOL, INDLIN, ATRLIN, ATRCOL, NUMDSG, N, F,
1  MATEFI
C
DO 91 A=1, F
IF(NUZLIN(A))11, 91, 11
IF(MATEFI(A, II))91, 21, 91
11 NUZLIN(A)=NUZLIN(A)-1
21 IF(NUZLIN(A))31, 31, 41
31 ATRLIN(A)=0
INDCOL(A)=0
GO TO 91
41 IF(NUZLIN(A)-1)91, 51, 91
51 ATRLIN(A)=1
DO 81 B=1, N
IF(MATEFI(A, B))81, 61, 81
61 IF(NUZCOL(B))81, 81, 71
71 INDCOL(A)=B
81 CONTINUE
91 CONTINUE
C..
DO 191 B=1, N
IF(NUZCOL(B))101, 191, 101
101 IF(MATEFI(KK, B))191, 111, 191
111 NUZCOL(B)=NUZCOL(B)-1
121 IF(NUZCOL(B))121, 121, 131
121 ATRCOL(B)=0
INDLIN(B)=0
GO TO 191
131 IF(NUZCOL(B)-1)191, 141, 191
141 ATRCOL(B)=1
DO 181 A=1, N
IF(MATEFI(A, B))181, 151, 181
151 IF(NUZLIN(A))181, 181, 161
161 INDLIN(B)=A
181 CONTINUE
191 CONTINUE
RETURN
END

```

SPD06600  
SPD06610  
SPD06620  
SPD06630  
SPD06640  
SPD06650  
SPD06660  
SPD06670  
SPD06680  
SPD06690  
SPD06700  
SPD06710  
SPD06720  
SPD06730  
SPD06740  
SPD06750  
SPD06760  
SPD06770  
SPD06780  
SPD06790  
SPD06800  
SPD06810  
SPD06820  
SPD06830  
SPD06840  
SPD06850  
SPD06860  
SPD06870  
SPD06880  
SPD06890  
SPD06900  
SPD06910  
SPD06920  
SPD06930  
SPD06940  
SPD06950  
SPD06960  
SPD06970  
SPD06980  
SPD06990  
SPD07000  
SPD07010  
SPD07020  
SPD07030  
SPD07040  
SPD07050  
SPD07060  
SPD07070  
SPD07080  
SPD07090

II - PARTE

EXEMPLIFICAÇÃO DO METODO DE ALOCAÇÃO



DADOS DE ENTRADA

- 1) Matriz Distância;
- 2) Capacidade das Linhas de Comunicações;
- 3) Matriz de Direcionamento;
- 4) Tabela de Custos de Comunicações (figura 37, p.113);
- 5) Tabela de Freqüência de Consultas (TRANSAÇÕES/ARQUIVOS);
- 6) Tabela de Freqüência de Atualizações (TRANSAÇÕES/ARQUIVOS);
- 7) Tabela de Freqüência (TRANSAÇÕES/NODOS);
- 8) Tamanho dos Arquivos;
- 9) Capacidade de Armazenamento;
- 10) Custo de Armazenamento (no exemplo apresentado foi utilizado um custo constante de armazenamento para todos os nodos no valor de 0,005 por byte armazenado);
- 11) Tamanho médio das mensagens de consultas = 128 bytes;
- 12) Tamanho médio das mensagens de atualizações = 128 bytes;
- 13) Tamanho médio das mensagens de respostas as consultas = 80 bytes.

DADOS INTERMEDIÁRIOS DE SAÍDA

- 1) Matriz de Custos de Transmissão;
- 2) Matriz de Freqüência de Consultas (NODO/ARQUIVO);
- 3) Matriz de Freqüência de Atualização (NODO/ARQUIVO);
- 4) Matriz de custos (EFICIÊNCIA).

DADOS FINAIS DE SAÍDA

- 1) Matriz de Alocação sem Redundância;
- 2) Custo da Alocação sem Redundância;
- 3) Matriz de Alocação com Redundância;
- 4) Custo da Alocação com Redundância.

MATRIZ DISTANCIAS (KM)

I/J	N1	N2	N3	N4	N5
N1	0	300	650	1100	900
N2	300	0	600	1200	1120
N3	650	600	0	750	850
N4	1100	1200	750	0	450
N5	900	1120	850	450	0

CAPACIDADE DAS LINHAS DE COMUNICACOES (BPS)

I/J	N1	N2	N3	N4	N5
N1	0	9600	0	9600	9600
N2	9600	0	9600	9600	0
N3	0	9600	0	9600	9600
N4	9600	9600	9600	0	9600
N5	9600	0	9600	9600	0

MATRIZ DE DIRECIONAMENTO

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	1-1	I	1-2	I	1-2-3	I	1-4	I	1-5
N2	I	2-1	I	2-2	I	2-3	I	2-4	I	2-1-5
N3	I	3-2-1	I	3-2	I	3-3	I	3-4	I	3-5
N4	I	4-1	I	4-2	I	4-3	I	4-4	I	4-5
N5	I	5-1	I	5-1-2	I	5-3	I	5-4	I	5-5

TABELA CUSTO DE TRANSMISSAO

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	0.000	I	0.101	I	0.317	I	0.277	I	0.277
N2	I	0.101	I	0.000	I	0.216	I	0.277	I	0.378
N3	I	0.317	I	0.216	I	0.000	I	0.277	I	0.277
N4	I	0.277	I	0.277	I	0.277	I	0.000	I	0.216
N5	I	0.277	I	0.378	I	0.277	I	0.216	I	0.000

TABELA DE FREQUENCIA DE CONSULTAS TRANSACOES/ARQUIVOS

T/K	I	K1	I	K2	I	K3	I	K4	I	K5	I	K6	I	K7	I	K8
T1	I	50	I	0	I	30	I	0	I	20	I	10	I	0	I	0
T2	I	50	I	10	I	40	I	0	I	0	I	0	I	0	I	0
T3	I	10	I	0	I	0	I	20	I	0	I	80	I	0	I	30
T4	I	0	I	20	I	0	I	0	I	0	I	0	I	90	I	0
T5	I	0	I	40	I	0	I	20	I	10	I	40	I	0	I	0
T6	I	0	I	0	I	80	I	0	I	10	I	0	I	80	I	40
T7	I	10	I	30	I	30	I	15	I	0	I	0	I	0	I	50
T8	I	20	I	0	I	20	I	0	I	0	I	0	I	0	I	30
T9	I	0	I	10	I	0	I	0	I	0	I	100	I	0	I	0
T10	I	0	I	0	I	0	I	0	I	50	I	0	I	0	I	0

TABELA DE FREQUENCIA DE ATUALIZACOES TRANSACOES ARQUIVOS

T/K	I	K1	I	K2	I	K3	I	K4	I	K5	I	K6	I	K7	I	K8
T1	I	10	I	0	I	6	I	0	I	4	I	2	I	0	I	0
T2	I	10	I	2	I	8	I	0	I	0	I	0	I	0	I	0
T3	I	2	I	0	I	0	I	4	I	0	I	16	I	0	I	6
T4	I	0	I	4	I	0	I	0	I	0	I	0	I	18	I	0
T5	I	0	I	8	I	0	I	4	I	2	I	8	I	0	I	0
T6	I	0	I	0	I	16	I	0	I	2	I	0	I	16	I	8
T7	I	2	I	6	I	6	I	3	I	0	I	0	I	0	I	10
T8	I	4	I	0	I	4	I	0	I	0	I	0	I	0	I	6
T9	I	0	I	2	I	0	I	0	I	0	I	20	I	0	I	0
T10	I	0	I	0	I	0	I	0	I	10	I	0	I	0	I	0

TABELA DE FREQUENCIA DAS TRANSAÇÕES NOS NÓDOS

T/N	I	N1	I	N2	I	N3	I	N4	I	N5
T1	I	50	I	0	I	10	I	0	I	0
T2	I	0	I	20	I	50	I	0	I	0
T3	I	0	I	10	I	10	I	50	I	0
T4	I	10	I	0	I	20	I	0	I	20
T5	I	0	I	0	I	0	I	10	I	0
T6	I	20	I	10	I	0	I	0	I	20
T7	I	0	I	0	I	10	I	30	I	0
T8	I	30	I	50	I	0	I	0	I	30
T9	I	0	I	0	I	0	I	20	I	50
T10	I	0	I	0	I	0	I	50	I	10

---

---

TAMANHO DOS ARQUIVOS (MBYTES)

---

---

K1 I K2 I K3 I K4 I K5 I K6 I K7 I K8

---

---

2 I 4 I 10 I 8 I 5 I 2 I 4 I 4

---

---

---

---

CAPACIDADES DE ARMAZENAMENTOS (MBYTES)

---

---

N1 I N2 I N3 I N4 I N5

---

---

25 I 25 I 25 I 25 I 25

---

---

MATRIZ DE FRECUENCIA DE CONSULTAS(NODO/ARQUIVO)

K/N	I	N1	I	N2	I	N3	I	N4	I	N5
K1	I	3100	I	2100	I	3200	I	800	I	600
K2	I	200	I	200	I	1200	I	1500	I	900
K3	I	3700	I	2600	I	2600	I	900	I	2200
K4	I	0	I	200	I	350	I	1650	I	0
K5	I	1200	I	100	I	200	I	2600	I	700
K6	I	500	I	800	I	900	I	6400	I	5000
K7	I	2500	I	800	I	1800	I	0	I	3400
K8	I	1700	I	2200	I	800	I	3000	I	1700

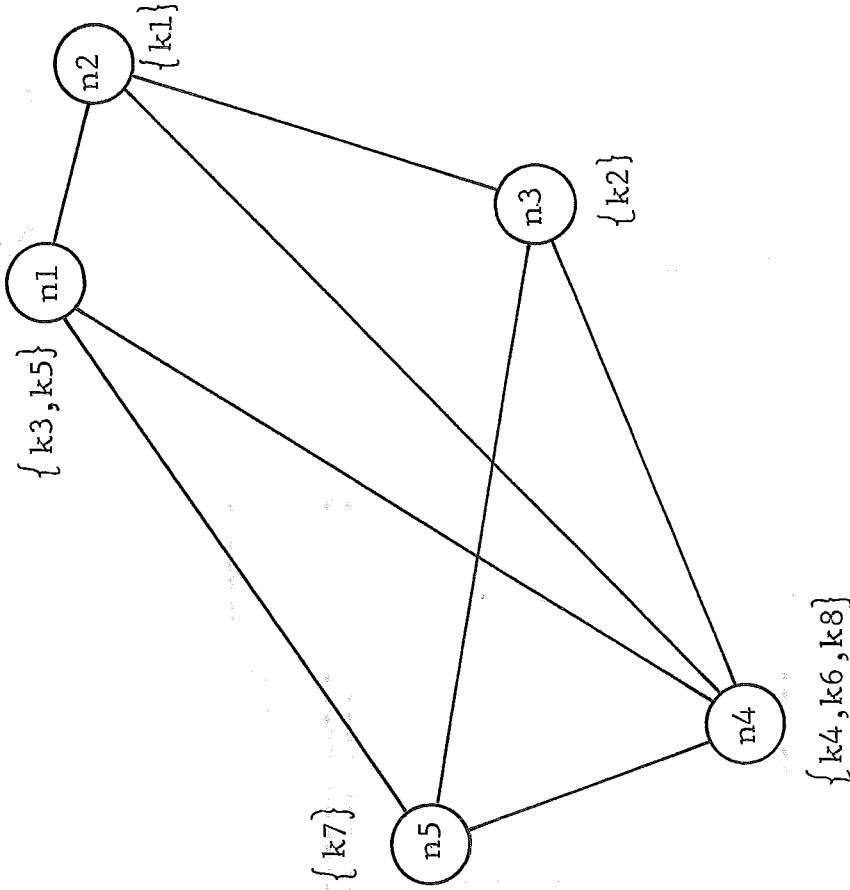


MATRIZ DE FREQUENCIA DE ATUALIZACOES(INQDC/ARQ.)

K/N	I	N1	I	N2	I	N3	I	N4	I	N5
K1	I	620	I	420	I	640	I	160	I	120
K2	I	40	I	40	I	240	I	300	I	180
K3	I	740	I	520	I	520	I	180	I	440
K4	I	0	I	40	I	70	I	330	I	0
K5	I	240	I	20	I	40	I	520	I	140
K6	I	100	I	160	I	180	I	1280	I	1000
K7	I	500	I	160	I	360	I	0	I	680
K8	I	340	I	440	I	160	I	600	I	340

MATRIZ DE C U S T O S

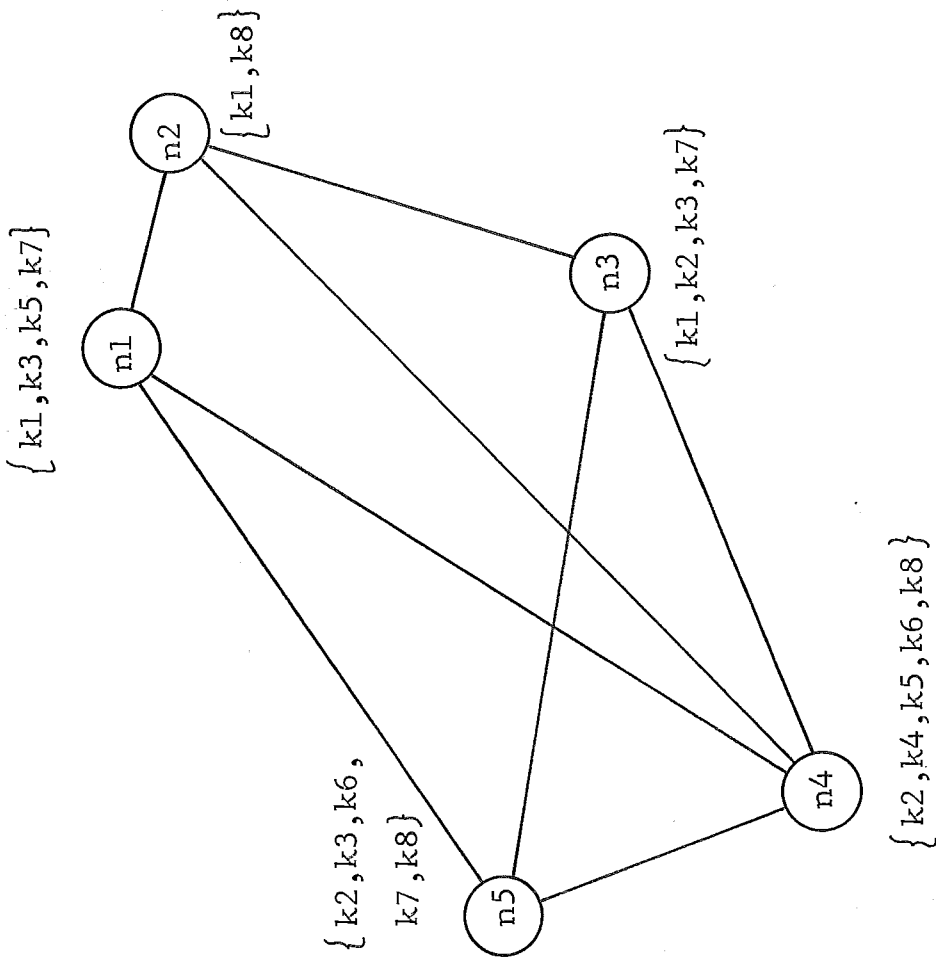
K/N	I	N1	I	N2	I	N3	I	N4	I	N5
K1	I	387101	I	349350	I	436110	I	583815	I	643453
K2	I	268877	I	261800	I	200199	I	166943	I	203936
K3	I	504468	I	520984	I	655771	I	736902	I	732650
K4	I	177404	I	164427	I	156859	I	75589	I	163563
K5	I	255703	I	293454	I	332441	I	157381	I	255610
K6	I	833183	I	922839	I	825054	I	404644	I	494160
K7	I	392171	I	470031	I	465498	I	521563	I	368882
K8	I	435271	I	444708	I	561018	I	409902	I	527403



MATRIZ DE ALOCAÇÃO SEM REPLICACAO

K/N	I	N1	I	N2	I	N3	I	N4	I	N5
K1	I	0	I	1	I	0	I	0	I	0
K2	I	0	I	0	I	1	I	0	I	0
K3	I	1	I	0	I	0	I	0	I	0
K4	I	0	I	0	I	0	I	1	I	0
K5	I	1	I	0	I	0	I	0	I	0
K6	I	0	I	0	I	0	I	1	I	0
K7	I	0	I	0	I	0	I	0	I	1
K8	I	0	I	0	I	0	I	1	I	0

I CUSTO DE ALOCAÇÃO SEM REPLICACAO Z= 2568737 I



MATRIZ DE ALOCAÇÃO COM REPLICACAO

K/N	I	N1	I	N2	I	N3	I	N4	I	N5
K1	I	1	I	1	I	1	I	0	I	0
K2	I	0	I	0	I	1	I	1	I	1
K3	I	1	I	0	I	1	I	0	I	1
K4	I	0	I	0	I	0	I	1	I	0
K5	I	1	I	0	I	0	I	1	I	0
K6	I	0	I	0	I	0	I	1	I	1
K7	I	1	I	0	I	1	I	0	I	1
K8	I	0	I	1	I	0	I	1	I	1

I CUSTO DE ALOCAÇÃO COM REPLICACAO Z= 1743343 I

Foram ainda executados exemplos de Alocações com o objetivo de caracterizar a influência da restrição da capacidade de armazenamento no custo final da Alocação. Para estes tipos de exemplos apresentaremos apenas os custos alcançados, para as diversas capacidades de armazenamentos relacionadas abaixo.

CAPACIDADE DE ARMAZENAMENTO	ALOCAÇÃO SEM REDUNDÂNCIA	ALOCAÇÃO COM REDUNDÂNCIA
30 MBS	2.568.737	1.743.343
25 MBS	2.568.737	1.743.343
20 MBS	2.568.737	1.801.581
15 MBS	2.568.737	1.997.780
10 MBS	2.603.450	2.383.782

O tempo de processamento requerido para o exemplo apresentado neste anexo, onde a quantidade de nodos era igual a 5 e a quantidade de arquivos a ser distribuído era igual a 8, foi de , estando incluído neste tempo a compilação e execução do programa.

ANEXO - B

RESULTADOS COM OS DADOS DO MODELO DE CASSEY<sup>12</sup>

=====

FRECUENCIA DE CONSULTAS(NODG/ARQUIVO)

=====

K/N I N1 I N2 I N3 I N4 I N5

=====

K1 I 10 I 15 I 20 I 30 I 40

=====

=====

FRECUENCIA DE ATUALIZACOES(NODG/ARQUIVO)

=====

K/N I N1 I N2 I N3 I N4 I N5

=====

K1 I 2 I 3 I 4 I 6 I 8

=====

---



---

TABELA CUSTO DE COMUNICAO

---



---

I/J I N1 I N2 I N3 I N4 IN5

---



---

N1 I 0 I 6 I 12 I 9 I 6  
 N2 I 6 I 0 I 6 I 12 I 9  
 N3 I 12 I 6 I 0 I 6 I 12  
 N4 I 9 I 12 I 6 I 0 I 6  
 N5 I 6 I 9 I 12 I 6 I 0

---



---



---



---

C U S T O S

---



---

K/N I N1 I N2 I N3 I N4 I N5

---



---

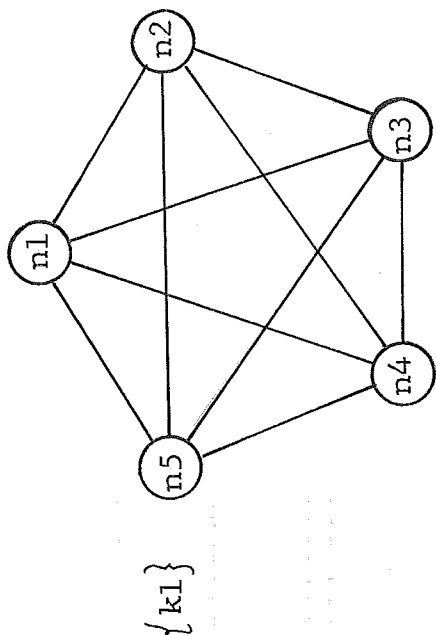
K1 I 1008 I 1080 I 1044 I 756 I 738

---



---



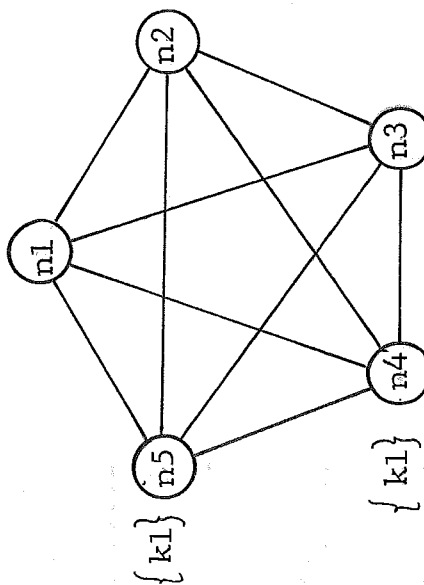


ALOCACAO SEM REDUNDANCIA

K/N I N1 I N2 I N3 I N4 I N5

K1 I 0 I 0 I 0 I 0 I 1

I CUSTO DE ALOCACAO SEM REDUNDANCIA Z = 738 I



ALOCACAO COM REDUNDANCIA

K/N I N1 I N2 I N3 I N4 I N5

K1 I 0 I 0 I 0 I 1 I 1

I CUSTO DE ALOCACAO COM REPLICACAO Z = 564 I

ANEXO - C

RESULTADOS PARA OS DIVERSOS TIPOS DE TOPOLOGIAS

Este anexo contém exemplos que mostram a flexibilidade do método de Alocação, para os mais variados tipos de estruturas de redes de computadores. Os dados utilizados foram os mesmos constantes do exemplo apresentado no Anexo-A, exceto para as matrizes de Distâncias e Direcionamento. Por esta razão os resultados deste anexo refere-se apenas as matrizes de Alocação e seus respectivos custos.

ESTRUTURA EM ANEL

MATRIZ DE ALOCAÇÃO SEM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I 0	I 1	I 0	I 0	I 0
K2	I 0	I 0	I 1	I 0	I 0
K3	I 1	I 0	I 0	I 0	I 0
K4	I 0	I 0	I 0	I 1	I 0
K5	I 1	I 0	I 0	I 0	I 0
K6	I 0	I 0	I 0	I 1	I 0
K7	I 0	I 0	I 0	I 0	I 1
K8	I 0	I 0	I 0	I 1	I 0

I CUSTO DE ALOCAÇÃO SEM REDUNDANCIA Z= 1058070 I

MATRIZ DE ALOCAÇÃO COM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I	I	I	I	0
K2	0	I	I	I	0
K3	I	I	0	0	0
K4	0	I	0	I	0
K5	I	I	0	I	0
K6	0	I	0	I	I
K7	I	I	0	0	I
K8	0	I	0	I	0

I CUSTO DE ALOCAÇÃO COM REDUNDANCIA Z = 892447 I

ESTRUTURA EM BARRA

MATRIZ DE ALOCAÇÃO SEM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I 0	I 1	I 0	I 0	I 0
K2	I 0	I 0	I 1	I 0	I 0
K3	I 1	I 0	I 0	I 0	I 0
K4	I 0	I 0	I 0	I 1	I 0
K5	I 1	I 0	I 0	I 0	I 0
K6	I 0	I 0	I 0	I 1	I 0
K7	I 0	I 0	I 0	I 0	I 1
K8	I 0	I 0	I 0	I 1	I 0

I CUSTO DE ALOCAÇÃO SEM REDUNDANCIA Z= 1077131 I

MATRIZ DE ALOCAÇÃO COM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I	I	I	I	0
K2	I	0	I	I	0
K3	I	I	I	0	0
K4	I	0	I	I	0
K5	I	I	0	I	0
K6	I	0	I	I	I
K7	I	I	0	I	I
K8	I	0	I	I	0

I CUSTO DE ALOCAÇÃO COM REDUNDANCIA Z = 905599 I

ESTRUTURA EM ESTRELA

MATRIZ DE ALOCAÇAO SEM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I 0	I 0	I 0	I 0	I 1
K2	I 0	I 0	I 1	I 0	I 0
K3	I 0	I 0	I 0	I 0	I 1
K4	I 0	I 0	I 0	I 1	I 0
K5	I 1	I 0	I 0	I 0	I 0
K6	I 0	I 0	I 0	I 1	I 0
K7	I 0	I 0	I 0	I 0	I 1
K8	I 0	I 1	I 0	I 0	I 0

I CUSTO DE ALOCAÇAO SEM REDUNDANCIA Z = 945848 I

MATRIZ DE ALOCAÇÃO COM REDUNDANCIA

K/N	I N1	I N2	I N3	I N4	I N5
K1	I	I	0	I	I
K2	I	0	I	I	I
K3	I	0	I	0	I
K4	I	0	I	0	I
K5	I	I	0	I	I
K6	I	0	I	0	I
K7	I	I	0	I	I
K8	I	0	I	I	I

I CUSTO DE ALOCAÇÃO COM REDUNDANCIA Z= 782598 I



ANEXO - D

I - PARTE

Na primeira parte deste anexo apresentaremos a listagem fonte do programa de avaliação, codificado em FORTRAN.

II - PARTE

Na segunda parte apresentaremos um exemplo de aplicação do método de avaliação.

I - PARTE

LISTAGEM DO PROGRAMA DE AVALIAÇÃO

```

C.....PROGRAMA PRINCIPAL
C.....AVALIACAO DO DESEMPENHO
C...DEFINICAO DAS AREAS DE TRABALHO
C
1 DIMENSION MATALC(20,20),MATFOC(20,20),MATFOA(20,20),MATRES(20,20),
MATDIR(20,20),MATRAZ(20,20),MATLIN(20,20)
C
C INTEGER N,F,I,J,K,VEIRAZ(50),R,W,II,JJ,SATURA,MAX,ACABOU
C
1 REAL MATRAF(20,20),MATARC(20,20),FATOR,SOMTRF,I,TAMENS,TAMPAC,
MAISAT(20,20),SOMA
C
C.....PROCEDIMENTO PARA LER A QUANTIDADE DE NODOS E ARQUIVOS DA REDE
C
R=1
W=3
MAX=20
ACABOU=1
WRITE(6,*)' INFORME TAMANHO DAS MENSAGENS(BITS) - 4 POS.'
READ(5,*)TAMENS
WRITE(6,*)' INFORME TAMANHO DOS PACOTES(BITS) - 4 POS.'
READ(5,*)TAMPAC
WRITE(6,*)' INFORME QUANTIDADE DE NODOS DA REDE - 2 POS.'
READ(5,*)N
IF(IN-MAX)10,10,20
10 WRITE(6,*)' INFORME QUANTIDADE DE ARQUIVOS DO SISTEMA - 2 POS.'
READ(5,*)F
IF(F-MAX)30,30,20
20 WRITE(6,*)
STOP
C.....PROCEDIMENTO PARA LER E ARMAZENAR
C - FREQUENCIA DE ACESSO PARA CONSULTAS(MENSAGENS/HORA)
C - FREQUENCIA DE ACESSO PARA ATUALIZACAO(MENSAGENS/HORA)
C - MATRIZ DE DIRECIONAMENTO(ROTA)
C - MATRIZ DE ALOCACAO DOS ARQUIVOS NOS NODOS
C - CAPACIDADE DAS LINHAS DE COMUNICACOES(KBPS)
C
C... LEITURA E IMPRESSAO DA MATRIZ DE FREQUENCIAS DE CONSULTAS
30 DO 40 K=1,F
40 READ(9,5002)(MATFOC(K,I),I=1,N)
WRITE(10,6002)
DO 50 K=1,F
50 WRITE(10,6012)(MATFOC(K,I),I=1,N)
C... LEITURA E IMPRESSAO DA MATRIZ DE FREQUENCIAS DE ATUALIZACAO
DO 60 K=1,F
60 READ(10,5002)(MATFOA(K,I),I=1,N)
WRITE(10,6003)

```

AVA00010

AVA00020

AVA00030

AVA00040

AVA00050

AVA00060

AVA00070

AVA00080

AVA00090

AVA00100

AVA00110

AVA00120

AVA00130

AVA00140

AVA00150

AVA00160

AVA00170

AVA00180

AVA00190

AVA00200

AVA00210

AVA00220

AVA00230

AVA00240

AVA00250

AVA00260

AVA00270

AVA00280

AVA00290

AVA00300

AVA00310

AVA00320

AVA00330

AVA00340

AVA00350

AVA00360

AVA00370

AVA00380

AVA00390

AVA00400

AVA00410

AVA00420

AVA00430

AVA00440

AVA00450

AVA00460

AVA00470

AVA00480

AVA00490

AVA00500

AVA00510

AVA00520

AVA00530

```

70 DO 70 K=1,F
  WRITE(W,6012)(MATFQA(K,I),I=1,N)
C
C... LEITURA E IMPRESSAO DA MATRIZ DE DIRECIONAMENTO
C
80 DO 80 I=1,N
  READ(I,5003)(MATDIR(I,J),J=1,N)
  WRITE(W,6004)
  DO 90 J=1,N
    WRITE(W,6012)(MATDIR(I,J),J=1,N)
C
C... LEITURA E IMPRESSAO DA MATRIZ DE ALOCAAO
C
100 DO 100 K=1,F
  READ(I2,5004)(MATALC(K,I),I=1,N)
  WRITE(W,6005)
  DO 110 K=1,F
    WRITE(W,6012)(MATALC(K,I),I=1,N)
C
C... LEITURA E IMPRESSAO DAS CAPACIDADES DAS LINHAS DE COMUNICACOES
C
120 DO 120 I=1,N
  READ(I3,5002)(MATLIN(I,J),J=1,N)
  WRITE(W,6007)
  DO 130 I=1,N
    WRITE(W,6012)(MATLIN(I,J),J=1,N)
C
C.....PROCEDIMENTO PARA GERAR O TRAFEGO FONTE-DESTINO(PACOTES-SEGUNDOS)
C
WRITE(W,6008)
SOMTRF=0.
DO 190 I=1,N
  DO 180 J=1,N
    MATRES(I,J)=0
    MATARC(I,J)=0
    MATSAT(I,J)=0
    MATRAZ(I,J)=0
    IF(I-J)150,170,150
    I=0
  DO 160 K=1,F
    T=I+1(MATFQC(K,I)*(1-MATALC(K,I))+MATFQA(K,I))*MATALC(K,J))
    CONTINUE
    T=T*(IAMENS/IAMPAC)
    MATRAF(I,J)=T/3600.
    SOMTRF=SOMTRF+MATRAF(I,J)
    GO TO 180
  MATRAF(I,J)=0.
CONTINUE
WRITE(W,6013)(MATRAF(I,J),J=1,N)
170 CONTINUE
C
C.....PROCEDIMENTO PARA GERAR O TRAFEGO NAS LINHAS(PACOTES/SEGUNDOS)
C
WRITE(W,6009)

```

AVA00540  
 AVA00550  
 AVA00560  
 AVA00570  
 AVA00580  
 AVA00590  
 AVA00600  
 AVA00610  
 AVA00620  
 AVA00630  
 AVA00640  
 AVA00650  
 AVA00660  
 AVA00670  
 AVA00680  
 AVA00690  
 AVA00700  
 AVA00710  
 AVA00720  
 AVA00730  
 AVA00740  
 AVA00750  
 AVA00760  
 AVA00770  
 AVA00780  
 AVA00790  
 AVA00800  
 AVA00810  
 AVA00820  
 AVA00830  
 AVA00840  
 AVA00850  
 AVA00860  
 AVA00870  
 AVA00880  
 AVA00890  
 AVA00900  
 AVA00910  
 AVA00920  
 AVA00930  
 AVA00940  
 AVA00950  
 AVA00960  
 AVA00970  
 AVA00980  
 AVA00990  
 AVA01000  
 AVA01010  
 AVA01020  
 AVA01030  
 AVA01040  
 AVA01050  
 AVA01060  
 AVA01070  
 AVA01080

```

DO 230 I=1,N
DO 220 J=1,N
  II=I
  JJ=J
  IF(I-J)200,220,200
  IF(MAIDIR(II,J)-J)210,215,210
  JJ=MAIDIR(II,J)
  MATARC(II,JJ)=MATARC(II,JJ)+MATRAF(I,J)
  II=JJ
  GO TO 200
200 MATARC(II,JJ)=MATARC(II,JJ)+MATRAF(I,J)
210 CONTINUE
220 CONTINUE
230 CONTINUE
C.....IMPRIMIR A MATRIZ DE TRAFEGO NAS LINHAS (PACOTES/SEGUNDOS)
C
DO 240 I=1,N
  WRITE(W,6013)(MATARC(I,J),J=1,N)
240 CONTINUE
C.....PROCEDIMENTO PARA CALCULAR O ATRAZO POR LINHA(MSEG)
C
DO 250 I=1,N
DO 280 J=1,N
  IF(MATARC(I,J))270,280,270
  MATRAZ(I,J)=ANINT(1/(MATLIN(I,J)/TAMPAC)-MATARC(I,J))*
  1000)
  IF(MATRAZ(I,J))271,271,280
  IF(MATLIN(I,J)-1200)272,274,275
  MATLIN(I,J)=1200
  MATLIN(J,I)=1200
  WRITE(W,6019)I,J,MATLIN(I,J)
  WRITE(W,6019)J,I,MATLIN(J,I)
  GO TO 270
270 MATLIN(I,J)=2400
  MATLIN(J,I)=2400
  GO TO 273
271 IF(MATLIN(I,J)-4800)276,277,278
272 MATLIN(I,J)=4800
  MATLIN(J,I)=4800
  GO TO 273
273 MATLIN(I,J)=9600
  MATLIN(J,I)=9600
  GO TO 273
274 WRITE(W,6020)I,J
  WRITE(W,6020)J,I
  CONTINUE
275 CONTINUE
276 CONTINUE
277 CONTINUE
278 CONTINUE
280 CONTINUE
290 CONTINUE
C.....IMPRIMIR A MATRIZ DE ATRAZO EM CADA LINHA
C
WRITE(W,6010)
DO 295 I=1,N
  WRITE(W,6012)(MATRAZ(I,J),J=1,N)
295 CONTINUE

```

AVA01090  
 AVA01100  
 AVA01110  
 AVA01120  
 AVA01130  
 AVA01140  
 AVA01150  
 AVA01160  
 AVA01170  
 AVA01180  
 AVA01190  
 AVA01200  
 AVA01210  
 AVA01220  
 AVA01230  
 AVA01240  
 AVA01250  
 AVA01260  
 AVA01270  
 AVA01280  
 AVA01290  
 AVA01300  
 AVA01310  
 AVA01320  
 AVA01330  
 AVA01340  
 AVA01350  
 AVA01360  
 AVA01370  
 AVA01380  
 AVA01390  
 AVA01400  
 AVA01410  
 AVA01420  
 AVA01430  
 AVA01440  
 AVA01450  
 AVA01460  
 AVA01470  
 AVA01480  
 AVA01490  
 AVA01500  
 AVA01510  
 AVA01520  
 AVA01530  
 AVA01540  
 AVA01550  
 AVA01560  
 AVA01570  
 AVA01580  
 AVA01590  
 AVA01600  
 AVA01610  
 AVA01620  
 AVA01630

```

C.....PROCEDIMENTO PARA GERAR A MATRIZ DE TEMPO DE RESPOSTA(MSEG)
C
WRITE(W,6011)
DO 340 I=1,N
  DO 330 J=1,N
    II=I
    JJ=J
    IF(I-J)300,330,300
    IF(MAIDIR(II,J)-J)310,320,310
    JJ=MAIDIR(II,J)
    MATRES(I,J)=MATRES(I,J)+MATRAZ(II,JJ)
    II=MAIDIR(II,J)
    GO TO 300
  300 MATRES(I,J)=MATRES(I,J)+MATRAZ(II,J)
  310 CONTINUE
  320 WRITE(W,6012)(MATRES(I,J),J=1,N)
  340 CONTINUE
C.....PROCEDIMENTO ESPECIAL
C
WRITE(W,6015)
FATOR=0.
SATURA=1
DO 450 K=1,50
  SOMA=0.
  DO 440 I=1,N
    DO 430 J=1,N
      IF(MATARC(I,J))410,430,410
      I=MATARC(I,J)/(MATLIN(I,J)/TAMPAC)-(MATARC(I,J)*FATOR)
      SOMA=SOMA+I
      IF(I)420,420,430
      SATURA=-I
    CONTINUE
  CONTINUE
  VETRAZ(K)=ANINT(SOMA/(SOMTRF*.001))
  WRITE(W,6014)VETRAZ(K),FATOR
  FATOR=FATOR+.1
  IF(SATURA)500,450,450
  450 CONTINUE
C.....PROCEDIMENTO PARA CALCULAR O FATOR DE SATURACAO DAS LINHAS
C
WRITE(W,6016)
SOMA=99999.
DO 540 I=1,N
  DO 530 J=1,N
    IF(MATARC(I,J))510,530,510
    MATSAT(I,J)=(MATLIN(I,J)/TAMPAC)/MATARC(I,J)
    IF(SOMA-MATSAT(I,J))520,530,520
    SOMA=MATSAT(I,J)
  510 CONTINUE
  520 JJ=J
  530 CONTINUE
  540 WRITE(W,6013)(MATSAT(I,J),J=1,N)
  CONTINUE

```

AVA01640  
 AVA01650  
 AVA01660  
 AVA01670  
 AVA01680  
 AVA01690  
 AVA01700  
 AVA01710  
 AVA01720  
 AVA01730  
 AVA01740  
 AVA01750  
 AVA01760  
 AVA01770  
 AVA01780  
 AVA01790  
 AVA01800  
 AVA01810  
 AVA01820  
 AVA01830  
 AVA01840  
 AVA01850  
 AVA01860  
 AVA01870  
 AVA01880  
 AVA01890  
 AVA01900  
 AVA01910  
 AVA01920  
 AVA01930  
 AVA01940  
 AVA01950  
 AVA01960  
 AVA01970  
 AVA01980  
 AVA01990  
 AVA02000  
 AVA02010  
 AVA02020  
 AVA02030  
 AVA02040  
 AVA02050  
 AVA02060  
 AVA02070  
 AVA02080  
 AVA02090  
 AVA02100  
 AVA02110  
 AVA02120  
 AVA02130  
 AVA02140  
 AVA02150  
 AVA02160  
 AVA02170  
 AVA02180

```

WRITE(N,6017) I,I,J,J,SOMA
IF(ACABOU)600,999,600
C.....PROCEDIMENTO PARA AJUSTAR AS CAPACIDADES DAS LINHAS
C
600 DO 710 I=1,N
      DO 700 J=1,N
        IF(MATARC(I,J))610,605,610
          MATLIN(I,J)=0
          GO TO 700
        MATLIN(I,J)=ANINT((SOMA*MATARC(I,J)*TAMPAC))
        IF(MATLIN(I,J)-300)620,620,630
          MATLIN(I,J)=300
          GO TO 700
        IF(MATLIN(I,J)-1200)640,640,650
          MATLIN(I,J)=1200
          GO TO 700
        IF(MATLIN(I,J)-2400)660,660,670
          MATLIN(I,J)=2400
          GO TO 700
        IF(MATLIN(I,J)-4800)680,680,690
          MATLIN(I,J)=4800
          GO TO 700
          MATLIN(I,J)=9600
          CONTINUE
700 CONTINUE
710 CONTINUE
C
C.....GRAVACAO E IMPRESSAO DAS CAPACIDADES DAS LINHAS AJUSTADAS
C
WRITE(N,6018)
DO 760 I=1,N
  WRITE(14,5002)(MATLIN(I,J),J=1,N)
  WRITE(N,6012)(MATLIN(I,J),J=1,N)
  CONTINUE
760 ACABOU=0
      GO TO 250
999 STOP
C.....DEFINICAO DOS FORMATOS
C
5001 FORMAT(I2)
5002 FORMAT(20I4)
5003 FORMAT(20I2)
5004 FORMAT(20I1)
6001 FORMAT(IH1,5X)
6002 FORMAT(IH1,5X)
6003 FORMAT(IH1,5X)
6004 FORMAT(IH1,5X)
6005 FORMAT(IH1,5X)
6006 FORMAT(IH1,5X)
6007 FORMAT(IH1,5X)
6008 FORMAT(IH1,5X)
6009 FORMAT(IH1,5X)
6010 FORMAT(IH1,5X)
6011 FORMAT(IH1,5X)
        FIM DE EXECUCAO POR ERRO)
        FREQUENCIAS DE CONSULTAS(TRANSACOES/HORA)*)
        FREQUENCIAS DE ATUALIZACOES(TRANSACOES/HORA)*)
        MATRIZ DE DIRECIONAMENTO*)
        MATRIZ DE ALDACAOK(I,I)*)
        MATRIZ DE ADJACENCIA(I,J)*)
        MATRIZ DE CAPACIDADE DAS LINHAS(KBPS)*)
        TRAFEGO DE FONTE -- DESTINO(PACOTES/SEGUNDOS)*)
        TRAFEGO NAS LINHAS-COMUNICACOES(PACOTES/SEGUNDOS)*)
        TEMPO DE ATRAZO NAS LINHAS DE COMUNICACOES(MSEG)*)
        TEMPO DE RESPOSTA(MSEG)*)

```

```

6012 FORMAT(0,5X,10I8/)
6013 FORMAT(0,7X,10F8.2/)
6014 FORMAT(0,5X,I8,5X,F3.1)
6015 FORMAT(1H1,5X,RELACAO DE RESPOSTA-CARGA NA REDE)
6016 FORMAT(1H1,5X,FATORES DE SATURACAO DAS LINHAS DA REDE)
6017 FORMAT(1H1,5X,LINHA SATURADA,I2,--,F5.3/)
6018 FORMAT(1H1,5X,CAPACIDADES DAS LINHAS DE COMUNICACOES AJUSTADA)
6019 FORMAT(1H1,5X,REAJUSTADA A CAPACIDADE DA LINHA,I2,--,I2, PARA
C,I4, ,KBS)
6020 FORMAT(1H1,5X,CAPACIDADE DE 9600 BPS INSUFICIENTE PARA TRAFEGO NA
C LINHA,I2,--,I2)
C.....FIM DO PROGRAMA
C
END

```

```

AVA02740
AVA02750
AVA02760
AVA02770
AVA02780
AVA02790
AVA02800
AVA02810
AVA02820
AVA02830
AVA02840
AVA02850
AVA02860
AVA02870
AVA02880

```



II - PARTE

EXEMPLIFICAÇÃO DO METODO DE AVALIAÇÃO

DADOS DE ENTRADA

- 1) Matriz de Frequências de Consultas (NODOS/ARQUIVOS);
- 2) Matriz de Frequências de Atualizações (NODOS/ARQUIVOS);
- 3) Matriz de direcionamento;
- 4) Matriz de Alocação (no exemplo foi utilizada a matriz de alocação com redundância);
- 5) Capacidade das linhas de Comunicações;
- 6) Tamanho médio das mensagens (= 1024 bits);
- 7) Tamanho médio dos "pacotes" (= 168 bits);
- 8) Número de nodos (= 5);
- 9) Número de arquivos (= 8);
- 10) Tempo de Processamento nodal constante para todos os nodos ( $= 10^{-3}$  mmseg).

DADOS INTERMEDIÁRIOS DE SAÍDA

- 1) Matriz de Tráfego de Pacotes (FONTE/DESTINO);
- 2) Matriz de Tráfego de Pacotes (nas linhas);
- 3) Matriz de Atraso (nas linhas).

DADOS FINAIS DE SAÍDA

- 1) Matriz de Tempo de Acesso (FONTE/DESTINO);
- 2) Fatores de Saturações nas Linhas;
- 3) Capacidade das Linhas de Comunicações Ajustadas;
- 4) Relação entre o Tempo de acesso e a Carga na Rede.

TRAFEGO FONTE-DESTINO(PACOTES/SEGUNDOS)

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	0.00	I	5.91	I	4.67	I	6.93	I	9.16
N2	I	10.27	I	0.00	I	10.53	I	4.44	I	12.71
N3	I	3.91	I	3.56	I	0.00	I	6.53	I	7.02
N4	I	5.69	I	3.47	I	5.20	I	0.00	I	7.24
N5	I	5.96	I	2.36	I	4.49	I	5.24	I	0.00

TRAFEGO NAS LINHAS-COMUNICACOES(PACOTES/SEGUNDOS)

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	0.00	I	12.93	I	0.00	I	6.93	I	21.87
N2	I	26.89	I	0.00	I	15.20	I	4.44	I	0.00
N3	I	0.00	I	7.47	I	0.00	I	6.53	I	7.02
N4	I	5.69	I	3.47	I	5.20	I	0.00	I	7.24
N5	I	8.31	I	0.00	I	4.49	I	5.24	I	0.00

TEMPO DE ATRAZO NAS LINHAS DE COMUNICACOES(MSEG)

I/J I N1 I N2 I N3 I N4 I N5

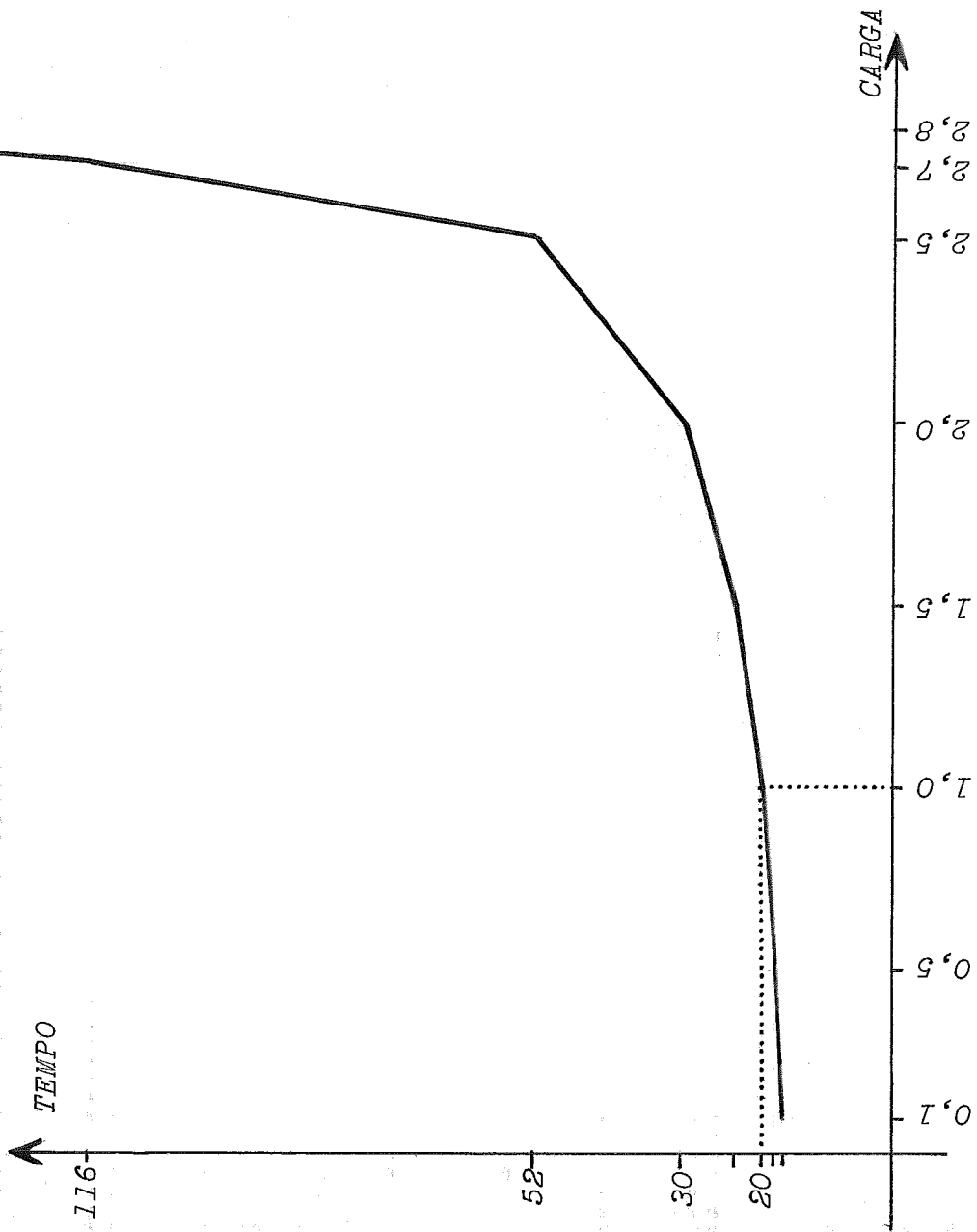
N1 I 0 I 16 I 0 I 15 I 19  
N2 I 21 I 0 I 17 I 14 I 0  
N3 I 0 I 15 I 0 I 15 I 15  
N4 I 14 I 14 I 14 I 0 I 15  
N5 I 15 I 0 I 14 I 14 I 0

TEMPO DE ACESSO FONTE-DESTINO (MSEG)

I/J I N1 I N2 I N3 I N4 I N5

N1 I 0 I 16 I 33 I 15 I 19  
N2 I 21 I 0 I 17 I 14 I 40  
N3 I 36 I 15 I 0 I 15 I 15  
N4 I 14 I 14 I 14 I 0 I 15  
N5 I 15 I 31 I 14 I 14 I 0

RELACAO (TEMPO DE RESPOSTA-CARGA NA REDE)



0.0  
0.1  
0.2  
0.3  
0.4  
0.5  
0.6  
0.7  
0.8  
0.9  
1.0  
1.1  
1.2  
1.3  
1.4  
1.5  
1.6  
1.7  
1.8  
1.9  
2.0  
2.1  
2.2  
2.3  
2.4  
2.5  
2.6  
2.7  
2.8

16  
16  
17  
18  
18  
19  
20  
20  
21  
22  
23  
24  
25  
27  
28  
30  
33  
35  
38  
43  
52  
68  
116  
-715

FATORES DE SATURACAO DAS LINHAS DA REDE

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	0.00	I	5.80	I	0.00	I	10.82	I	3.43
N2	I	2.79	I	0.00	I	4.93	I	16.88	I	0.00
N3	I	0.00	I	10.04	I	0.00	I	11.48	I	10.68
N4	I	13.18	I	21.63	I	14.42	I	0.00	I	10.35
N5	I	9.02	I	0.00	I	16.71	I	14.30	I	0.00

LINHA SATURADA 2- I CARGA=2.789

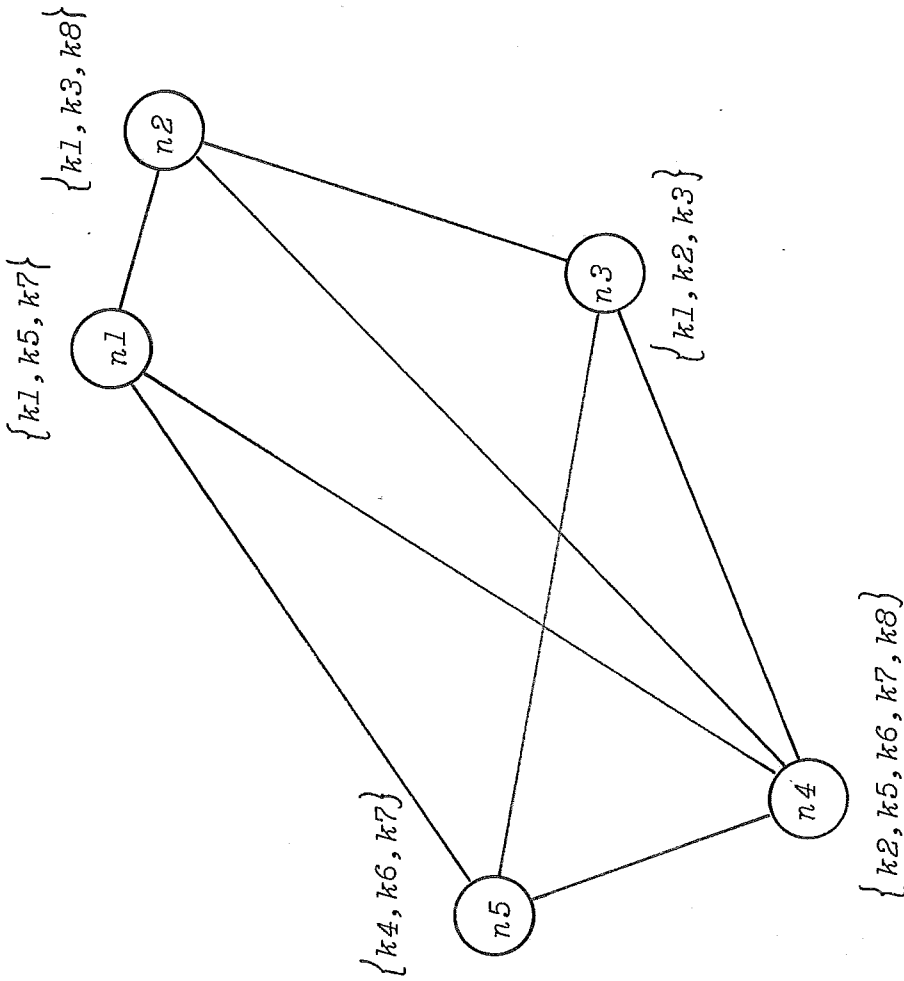
CAPACIDADE DAS LINHAS DE COMUNICACOES AJUSTADA

I/J	I	N1	I	N2	I	N3	I	N4	I	N5
N1	I	0	I	4800	I	0	I	4800	I	9600
N2	I	9600	I	0	I	9600	I	2400	I	0
N3	I	0	I	4800	I	0	I	2400	I	4800
N4	I	2400	I	2400	I	2400	I	0	I	4800
N5	I	4800	I	0	I	2400	I	2400	I	0

Com esta nova Matriz de Capacidade das Linhas de Comunicações Ajustadas, o processo de Alocação pode ser realimentado, passando a oferecer novos resultados que caracterizam a influência dos custos de comunicações nos custos finais de Alocação.

- 1) Uma nova Matriz de Custos de Transmissão;
- 2) Uma nova Matriz de Custos (Matriz Eficiência);
- 3) Novas Matrizes de Alocações;
- 4) Novos custos de Alocações.

Apresentaremos apenas como resultado do processo de realimentação do método de Alocação, a Matriz de Alocação com Redundância e seu respectivo custo.



MATRIZ DE ALOCACAO COM REDUNDANCIA

K/I	I N1	I N2	I N3	I N4	I N5
K1	I	I	I	I	I
K2	I	I	I	I	I
K3	I	I	I	I	I
K4	I	I	I	I	I
K5	I	I	I	I	I
K6	I	I	I	I	I
K7	I	I	I	I	I
K8	I	I	I	I	I

I CUSTO DE ALOCACAO COM REDUNDANCIA Z = 1310170



A partir destas novas Alocações determinadas pela realimentação do processo de Alocação, o processo de Avaliação poderá ser realimentado passando a oferecer novos tempos de acessos. Outros parâmetros poderão também ser incluídos no processo de realimentação do método de Avaliação, assim como por exemplo o tamanho das mensagens ou o tamanho dos pacotes. Na página (D18), apresentamos um gráfico que mostra o comportamento do tempo de acesso em relação a variação do tamanho do pacote.

A quantidade de ciclos a ser executado dependerá do equilíbrio que se deseja alcançar entre os tempos de acessos e os custos de Alocações, que satisfaça aos interesses do Sistema em questão.

O tempo consumido para execução do programa de Avaliação, para o exemplo apresentado, foi de \_\_\_\_\_, estando incluído neste tempo a compilação e execução do programa.

