Gerenciamento Pró-ativo: Consideração do Tráfego de Aplicações e Utilização de Séries Temporais

Francisco Watzko Neto

Gerenciamento Pró-ativo: Consideração do Tráfego de Aplicação e Utilização de Séries Temporais

Banca Examinadora:

- Edmundo Roberto Mauro Madeira (Orientador)[1]
- Otto Carlos Muniz Bandeira Duarte[2]
- Célio Cardoso Guimarães[1]
- Paulo Lício de Geus (Suplente)[1]

[1] Professor do Instituto de Computação - IC - UNICAMP
[2] Professor da COPPE/EE - Programa de Engenharia Elétrica da UFRJ - Universidade Federal do Rio de Janeiro

UNICAMP
BIBLIOTECA CENTRAL

UNIDADE BC
N.º CHAMADA: T/UNICAMP
W351g
V. Ex.
TOMBO BC/ 31374
PROC. 281/97
C [ ] D [X]
PREÇO R$ 11,00
DATA 14/08/97
Nº CPD

CM-00099697-1

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA - UNICAMP

Watzko Neto, Francisco
W351g Gerenciamento pró-ativo: consideração do tráfego de aplicação e utilização de séries temporais / Francisco Watzko Neto. --- Campinas, SP : [s.n] , 1997.

Orientador : Edmundo Roberto Mauro Madeira
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Rede de computadores - gerenciamento. 2. Rede de computadores. 3. Análise de séries temporais. 4. Rede local - rede de computadores. I. Madeira, Edmundo Roberto Mauro. I. Universidade Estadual de Campinas - Instituto de Computação. III. Título.

Gerenciamento Pró-ativo: Consideração do Tráfego de Aplicação e Utilização de Séries Temporais

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida pelo Sr. Francisco Watzko Neto e aprovada pela Comissão Julgadora.

Campinas, 26 de junho de 1997.

Prof. Dr. Edmundo R. M. Madeira
Orientador

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Tese de Mestrado defendida e aprovada em 26 de junho de 1997 pela Banca Examinadora composta pelos Professores Doutores

Prof. Dr. Otto Carlos Muniz Bandeira Duarte

Prof. Dr. Célio Cardoso Guimarães

Prof. Dr. Edmundo Roberto Mauro Madeira

Aos meus pais João Américo e Maria José.

AGRADECIMENTOS

Ao Prof. Edmundo Madeira pela ajuda e apoio em todos os momentos.

Aos Prof. Ronaldo Dias e Luiz Koodi Hotta pela ajuda nos estudos de séries temporais.

Ao funcionários do Instituto responsáveis pela administração da rede, em especial ao Niero, pela ajuda sempre prestativa, e à Vera e Roseli pela colaboração.

Aos amigos do Instituto os quais passamos juntos tantos momentos.

A todos os amigos que mesmo distantes sempre estiveram tão próximos compartilhando todos os momentos.

Aos meus pais, João Américo e Maria José, pelo incentivo, compreensão, amor e ensinamentos. Não tenho palavras para agradecer tudo que fizeram para que eu atingisse este momento.

Às minhas irmãs, Fabiana e Caroline, pelo carinho e incentivo.

À Ivana pelo apoio e o amor que sempre me ajudaram a superar as dificuldades.

Aos amigos de república: Ademir, Guto, Faina e Cláudio pelo apoio e companheirismo.

A todas as pessoas que de forma direta ou indireta contribuíram para que eu chegasse até aqui.

À Fapesp e ao CNPq pelo apoio financeiro durante a execução deste trabalho.

RESUMO

A crescente importância das redes de computadores impulsionou os estudos sobre formas de efetuar o gerenciamento de redes de uma forma mais eficiente. Com base neste fato, vem sendo estudado o gerenciamento pró-ativo de redes. Este novo paradigma visa a identificação de situações que possam levar à degradação dos sistemas de computação antes que elas venham a degradar o funcionamento do sistema e tentar evitar que tais situações ocorram através de ações corretivas.

Esta dissertação apresenta duas contribuições no campo do gerenciamento pró-ativo: a consideração do tráfego de aplicação, na proposição de um modelo de gerenciamento e de um conjunto de funções de gerenciamento pró-ativas, e a aplicação de séries temporais na análise e estimativa do comportamento de uma rede.

O conceito de séries temporais foi aplicado inicialmente neste trabalho na tarefa de identificação do funcionamento normal da rede e posteriormente na estimativa do funcionamento futuro da mesma. Segundo [CHA 84], uma série temporal é uma coleção de observações feitas seqüencialmente no tempo. Na teoria de séries temporais procura-se identificar de que forma os valores anteriores de uma variável influenciam o comportamento futuro desta. Através da modelagem deste relacionamento, procura-se estimar o comportamento futuro da variável. Com a utilização desta teoria neste trabalho procurou-se modelar o comportamento de uma das variáveis utilizadas na caracterização do comportamento da rede, visando estimar o funcionamento futuro e desta forma identificar situações que levassem à degradação do sistema.

Recentemente, grande parte dos esforços tem sido voltados para a tarefa de gerenciamento de redes, porém, o gerenciamento dos sistemas conectados a estas redes tornou-se cada vez mais importante. Este trabalho propõem um modelo para o gerenciamento pró-ativo que se utiliza de informações tanto em nível de rede quanto em nível de aplicação. Em acréscimo, é especificado um conjunto de funções de gerenciamento pró-ativo que se utilizam deste nível de informações disponibilizado.

Um protótipo baseado no modelo de gerenciamento pró-ativo proposto foi implementado. Este protótipo utiliza informações tanto em nível de rede quanto em nível aplicação na caracterização do comportamento normal da rede e aplica os conceitos de séries temporais para a estimativa do funcionamento futuro da rede.

ABSTRACT

The increasing importance of the network computer has stimulated research about different kinds of more efficient network management. Based on this, studies have been made about proactive network management. This new paradigm intends to identify situations that may lead to a system performance degradation before it would take place and avoid the occurance of these situations by using corrective actions.

Two new contributions in the field of computer proactive network management are introduced in this dissertation: the consideration of traffic in the application level proposal of a network management model and in a set of proactive network functions, and the use of time series in the analysis and estimate of the network behavior.

The time series concept was used first in the identification of the standard computer network behavior and then in the estimate of its future behavior. Based on [CHA 84], a time series is a collection of observations done sequencely in time. The time series theory tries to identify how the past values of a variable influence its future behavior. The modeling of this relation tries to estimate the future behavior of the variable. Applied to this work, the theory models the behavior of one variable used in the characterization of the computer network behavior, for the purpose of trying to estimate the future behavior and, in this way, identify situations that lead to a system performance degradation.

Many of the recent researches have been focused in the work of computer network management, but the administration of the systems connected to these networks are becoming more important. This work suggests a proactive computer network management model that uses both the network level and the application level information. Another contribution also proposed is a set of proactive network functions that use this level of information.

A prototype based on the proposed proactive network management model was implemented. This prototype uses both level of information: the network and the application level in the characterization of the normal behavior of the computer network and it applies the time series concepts in the estimate of the computer network future behavior.

LISTA DE ABREVIAÇÕES

ACF	*Autocorrelation Function*
ARPA	*Advanced Research Project Agency*
ARPANET	*ARPA experimental packet-switched Network*
ASIC	*Application Specific Integrated Circuits*
BTNG	*Beholder - The Next Generation*
CLTS	*Connectionless Mode Transport Service*
CCR	*Centro de Controle de Rotas*
CRC	*Cyclyc Redundancy Check*
DoD	*Department of Defense*
DNS	*Domain Name Service*
DR	*Dinamic Regression*
FCS	*Frame Check Sequence*
FTP	*File Transfer Protocol*
HTTP	*Hipertext Transfer Protocol*
IAB	*Internet Architecture Board*
ICMP	*Internet Control Message Protocol*
IEC	*International Electrotechnical Committee*
IETF	*Internet Engeneering Task Force*
ISO	*International Organization for Standardization*
LAN	*Local Area Network*
MTA	*Message Transfer Agents*
MIB	*Management Information Base*
MIME	*Multipurpose Internet Mail Extensions*
NFS	*Network File System*
NVT	*Network Virtual Terminal*
OSI	*Open Systems Interconnection*
PACF	*Partial Autocorrelation Function*
PDU	*Protocol Data Unit*
RACF	*Residual Autocorrelation Function*
RPACF	*Residual Partial Autocorrelation Function*
RMON	*Remote Network Monitoring*
RPC	*Remote Procedure Call*
SACF	*Sample Autocorrelation Function*
SIPP	*Simple Internet Protocol Plus*
SPACF	*Sample Partial Autocorrelation Function*
SMI	*Structure of Management Information*

SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
TFTP	*Trivial File Transfer Protocol*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Location*
VFS	*Virtual File System*
WAN	*Wide Area Network*
WWW	*World Wide Web*
XDR	*External Data Representation*

LISTA DE FIGURAS

FIGURA 2.1 - MODELO DE REFERÊNCIA TCP/IP 4
FIGURA 2.2 - CAMINHO PERCORRIDO POR UMA MENSAGEM DO *HOST* A AO *HOST* B 5
FIGURA 2.3 - DATAGRAMA IP 5
FIGURA 2.4 - FORMATO DO SEGMENTO TCP 12
FIGURA 2.5 - FORMATO DO PACOTE UDP 13
FIGURA 2.6 - CONEXÃO TELNET 15
FIGURA 2.7 - CONEXÃO FTP DE CONTROLE E DE TRANSFERÊNCIA DE DADOS 15
FIGURA 2.8 - ARQUITETURA DO CORREIO ELETRÔNICO SMTP 16
FIGURA 2.9 - PROCESSO DE RESOLUÇÃO DE NOMES 18
FIGURA 2.10 - PARTES DO MODELO WWW 20
FIGURA 2.11 - CENÁRIO TÍPICO SNMP 24
FIGURA 3.1 - TOPOLOGIA RMON 29
FIGURA 3.2 - DIAGRAMA DE ESTADOS DE UMA VARIÁVEL DE CONTROLE 30
FIGURA 4.1 - MODELO PARA A GERÊNCIA PRÓ-ATIVA 43
FIGURA 4.2 - MODELO DE GERENCIAMENTO PRÓ-ATIVO PROPOSTO 44
FIGURA 4.3 - ESQUEMA DO FUNCIONAMENTO DA COLETA DE DADOS 49
FIGURA 4.4 - EXEMPLO DE FILTROS, CANAIS E *BUFFERS* CRIADOS 50
FIGURA 4.5 - COLETA DE DADOS SOBRE A APLICAÇÃO E GERAÇÃO DE *BASELINE* 51
FIGURA 4.6 - *FRAME* ETHERNET 52
FIGURA 4.7 - COLISÕES OBSERVADAS 53
FIGURA 4.8 - GRÁFICO DOS PACOTES CURTOS OBSERVADOS 54
FIGURA 4.9 - GRÁFICO DOS PACOTES *MULTICAST* OBSERVADOS 55
FIGURA 4.10 - GRÁFICO DOS PACOTES *BROADCAST* OBSERVADOS 56
FIGURA 4.11 - ESQUEMA DO SISTEMA DE VERIFICAÇÃO 61
FIGURA 4.12 - MÓDULO COLETA 62
FIGURA 4.13 - MÓDULO ESTIMADOR 62
FIGURA 4.14 - MÓDULO ANÁLISE 63
FIGURA 4.15 - ESQUEMA DE FUNCIONAMENTO ENTRE OS NÍVEIS DE APLICAÇÃO E DE TRANSPORTE 64
FIGURA 4.16 - ESQUEMA DE MULTIPLEXAÇÃO DE VÁRIAS CONEXÕES EM NÍVEL DE APLICAÇÃO EM UMA EM NÍVEL DE REDE (A) E VICE-VERSA (B) 64
FIGURA 4.17 - SÍNDROME DA *SILLY WINDOW* 66
FIGURA 4.18 - EXEMPLO DO ALGORITMO DE CONGESTIONAMENTO INTERNET 67
FIGURA 4.19 - ENTIDADE DE TRANSPORTE UTILIZANDO *THREADS* 69
FIGURA 5.1 - ACFS E PACFS TEÓRICOS DE AR(1) 77
FIGURA 5.2 - ACFS E PACFS TEÓRICOS DE AR(2) 77
FIGURA 5.3 - ACFS E PACFS TEÓRICOS DO PROCESSO MA(1) 78
FIGURA 5.4 - ACFS TEÓRICOS DO PROCESSO MA(2) 78
FIGURA 5.5 - ACF E PACF TEÓRICOS DO PROCESSO ARIMA(1,1,1) 79
FIGURA 6.1 - SÉRIE LIDA, SEU ACF E PACF 86
FIGURA 6.2 - PLOTAGEM DA SÉRIE COM D=1, SEU ACF E PACF 87
FIGURA 6.3 - PLOTAGEM DA SÉRIE COM D=D=1, SEU ACF E PACF 88
FIGURA 6.4 - PLOTAGEM DOS VALORES LIDOS E PREVISTOS, RESÍDUOS E ACF DO RESÍDUO 91

LISTA DE TABELAS

TABELA I - ALGUMAS PORTAS *WELL-KNOWN* DO TCP 12
TABELA II - ALGUMAS PORTAS *WELL-KNOWN* DO UDP 13
TABELA III - ALGUNS PROTOCOLOS DE APLICAÇÃO 14
TABELA IV - SÍNTESE DAS APLICAÇÕES MONITORADAS 57
TABELA V - PARÂMETROS DE MONITORAÇÃO E *THRESHOLDS* ASSOCIADOS 58
TABELA VI - *THRESHOLDS* PARA APLICAÇÕES 58
TABELA VII - PREVISÃO 90

SUMÁRIO

1. INTRODUÇÃO 1

1.1. ORGANIZAÇÃO DO TRABALHO 2

2. CONCEITOS BÁSICOS: ARQUITETURA TCP/IP E GERENCIAMENTO DE REDES 3

2.1. MODELO DE REFERÊNCIA INTERNET 3
2.1.1. Arquitetura do Modelo de Referência TCP/IP 3
2.1.2. Protocolos Internet 5
2.1.2.1. Internet Protocol (IP) 5
2.1.2.1.1. Roteamento no Protocolo IP 7
2.1.2.1.2. Roteadores e *Gateways* 9
2.1.2.1.2.1. Configuração de Roteadores 10
2.1.2.1.3. Algoritmos de Roteamento 10
2.1.2.1.3.1. Roteamento Baseado no Fluxo 11
2.1.2.2. Transmission Control Protocol (TCP) 11
2.1.2.3. User Datagram Protocol (UDP) 13
2.1.3. Protocolos de Aplicação 14
2.1.3.1. Telnet 14
2.1.3.2. File Transfer Protocol (FTP) 15
2.1.3.3. Simple Mail Transfer Protocol (SMTP) 16
2.1.3.4. Network File System (NFS) 17
2.1.3.4.1. Virtual File Systems (VFS) e Virtual Nodes 17
2.1.3.4.2. Protocolo NFS 17
2.1.3.5. Simple Network Management Protocol (SNMP) 18
2.1.3.6. Domain Name Service (DNS) 18
2.1.3.7. Hipertext Transfer Protocol (HTTP) 19
2.1.3.7.1. Visão do Cliente 19
2.1.3.7.2. Visão do Servidor 19
2.2. GERÊNCIA DE REDES 20
2.2.1. Modelo de Gerenciamento OSI 21
2.2.1.1. Estrutura do Modelo OSI 21
2.2.1.1.1. Componentes de Gerenciamento 21
2.2.1.1.2. Áreas Funcionais 21
2.2.2. Modelo de Gerenciamento Internet 22
2.2.2.1. Estrutura 22
2.2.2.1.1. Simple Network Management Protocol (SNMP) 23
2.2.2.1.2. Simple Network Management Protocol version 2 (SNMPv2) 24

3. GERENCIAMENTO PRÓ-ATIVO 26

3.1. *BASELINE* OU PERFIL DA REDE 27
3.2. MONITORAÇÃO REMOTA DE REDES 27
3.3. RMON 1 MIB 28
3.3.1. Objetivos da Remote Monitoring 1 MIB - RMON 1 MIB 29
3.3.1.1. Controle dos Dispositivos de Monitoração remota 30
3.3.1.2. Grupos da RMON 1 MIB 30
3.3.2. Implementações Privadas 33
3.3.2.1. Frontier Software RMON Support 33
3.3.2.2. Implementação da RMON feita pela Hewlett-Packard 34
3.3.2.2.1. Power Agents 34
3.3.2.2.2. HP NetMetrix 34
3.3.2.3. SmartAgents RMON desenvolvidos pela 3Com 34
3.3.2.4. Bay Networks' SuperRMON 34
3.3.3. Implementação Acadêmica: Beholder - The Next Generation 35

3.4. RMON 2 MIB 35
3.4.1. Grupos componentes da RMON 2 35
3.5. INFORMAÇÕES UTILIZADAS NO GERENCIAMENTO DE REDES DE COMPUTADORES 37
3.5.1. Gerenciamento de Sistemas 38
3.5.1.1. Informações Relevantes para o gerenciamento de Sistemas 38
3.5.1.1.1. Informações sobre Domain Name Service (DNS) 38
3.5.1.1.2. Informações sobre o Simple Mail Transfer Protocol (SMTP) 39
3.5.1.1.3. Monitoração dos Hosts 40
3.5.1.1.4. Informações sobre o Protocolo File Transfer Protocol (FTP) 41
3.5.1.1.5. Informações relevantes sobre o protocolo Simple Network Management Protocolo (SNMP) 41
3.5.1.1.6. Informações Relevantes sobre o serviço Network Information Service (NIS) 42
3.5.1.1.7. Informações Relevantes sobre o serviço Network FILE SYSTEM (NFS) 42
3.5.1.1.8. Informações Relevantes sobre a Aplicação TELNET 42
3.5.1.1.9. Informações Relevantes sobre o Protocolo Hipertext Transfer Protocol 42

4. GERENCIAMENTO PRÓ-ATIVO PROPOSTO 43

4.1. MODELO DE GERENCIAMENTO PRÓ-ATIVO 43
4.2. MODELO DE GERENCIAMENTO PRÓ-ATIVO DE APLICAÇÕES 44
4.3. AGENTE REMOTO 47
4.3.1. Coleta de Dados 47
4.3.1.1.1. Dados Coletados até o Nível de Rede 47
4.3.1.1.2. Dados Coletados em Nível de Aplicação 47
4.3.1.1.2.1. Formas de Cálculo de Utilização do Meio por uma Aplicação 51
4.4. *BASELINE* 52
4.4.1. *Baselines* até o Nível de Rede 52
4.4.2. *Baselines* do Nível de Aplicação 56
4.5. MÉTRICAS DE DESEMPENHO 57
4.6. FUNÇÕES DE GERENCIAMENTO PRÓ-ATIVO 58
4.6.1. Notificação ao Gerente da Rede 59
4.6.2. Mudança de Rota em um Roteador 59
4.6.3. Controle de Fluxo Baseado em Aplicações 59
4.6.4. Monitoração do Estado de uma Aplicação 60
4.6.5. Controle de Congestionamento Baseado em Aplicações 60
4.6.6. Limitação de Acesso 60
4.6.7. Controle de Prioridade de Processo 60
4.6.8. Execução Condicionada de Processos 60
4.7. SISTEMA DE VERIFICAÇÃO 61
4.8. FUNÇÃO DE CONTROLE DE FLUXO BASEADO EM APLICAÇÃO 63
4.8.1. Comunicação entre a Camada de Transporte e a Camada de Aplicação 63
4.8.2. Política de Transmissão 64
4.8.3. Algoritmo de Nagle 65
4.8.4. Algoritmo de Clarck 66
4.8.5. Controle de Congestionamento 66
4.8.6. Esquema Proposto 68
4.8.7. Aspectos de Implementação 68

5. UTILIZAÇÃO DA TÉCNICA DE SÉRIES TEMPORAIS NO GERENCIAMENTO PRÓ-ATIVO 70

5.1. FATORES A SEREM CONSIDERADOS NA ESCOLHA DE UMA TÉCNICA DE SIMULAÇÃO 71
5.2. ESTRATÉGIA DE MODELAGEM 71
5.3. CONHECIMENTOS BÁSICOS PARA A IDENTIFICAÇÃO DE UM MODELO 72
5.3.1. Modelo ARIMA (Autoregressive Integrated Moving Average) 73
5.3.2. Média de Variância Estacionárias 73
5.3.2.1. Estabilização da Variância 73
5.3.2.2. Estabilização da Média 74
5.3.2.2.1. Diferenciação não Sazonal 74

5.3.2.2.2. Diferenciação Sazonal 74
5.3.3. Auto-correlação 74
5.3.3.1. Função de Auto-correlaçao da Amostra (SACF) 75
5.3.3.2. Função de Auto-correlação Parcial da Amostra (SPACF) 75
5.3.3.3. Processos ARIMA Estacionários 76
5.3.3.3.1. Processo Autoregressivo 76
5.3.3.4. Médias Móveis e Processos Mistos 77
5.3.3.5. Processo Integrados 79
5.3.4. Modelos Sazonais 79
5.3.5. Notação Backshift 80
5.3.6. Processos Sazonais e Não Sazonais Combinados 80
5.4. VERIFICAÇÃO DO MODELO 80
5.4.1. Teste de Portmanteau 81
5.4.2. Identificação de Modelos Aprimorados 81
5.5. CONSIDERAÇÕES SOBRE A APLICAÇÃO DA TÉCNICA DE SÉRIES TEMPORAIS NA ANÁLISE DO COMPORTAMENTO DE UM SEGMENTO 81
5.6. COMO APLICAR A TÉCNICA DE SÉRIES TEMPORAIS NO GERENCIAMENTO DE REDES 82

6. ANÁLISE DA UTILIZAÇÃO DE UM SEGMENTO MONITORADO ATRAVÉS DA TÉCNICA DE SÉRIES TEMPORAIS 84

6.1. IDENTIFICAÇÃO 84
6.2. ESTIMATIVA DOS PARÂMETROS E CHECAGEM DO MODELO 88
6.3. PREVISÃO 89
6.4. CONSIDERAÇÕES SOBRE OS RESULTADOS OBTIDOS 89

7. CONCLUSÃO 92

REFERÊNCIAS BIBLIOGRÁFICAS 94

ANEXOS 99

ANEXO A - PROGRAMA DE CONFIGURAÇÃO DOS FILTROS DO BEHOLDER 99
ANEXO B - *BASELINE* DO NÍVEL DE UTILIZAÇÃO DO SEGMENTO MONITORADO 103
ANEXO C - *BASELINES* COLETADAS EM NÍVEL DE APLICAÇÃO 113

Capítulo 1

INTRODUÇÃO

Nos dias atuais é cada vez maior a importância dos sistemas de computação na vida de todos. Em um mundo onde os continentes estão cada vez mais próximos graças às comunicações, a globalização torna-se cada vez mais real e neste contexto as redes de computadores têm desempenhado um papel fundamental para a concretização desta nova realidade.

A manutenção do perfeito funcionamento dos sistemas de computação é cada vez mais necessária, porém a dificuldade desta tarefa cresce a medida que estes sistemas tornam-se maiores e mais complexos.

Os sistemas de gerenciamento de redes existentes atualmente buscam monitorar as informações sobre a rede de computadores e auxiliar na solução dos problemas encontrados. Estes sistemas são baseados na abordagem reativa, pois os problemas só são resolvidos após terem ocorrido.

Uma alternativa para o gerenciamento reativo é o gerenciamento onde objetiva-se identificar situações que possam levar a um mal funcionamento do sistema e tentar corrigi-las antes que o desempenho do sistema seja prejudicado, chamado de gerenciamento pró-ativo, buscando manter o funcionamento normal da rede baseando-se no perfil de funcionamento da mesma. Pesquisas no campo da inteligência artificial referentes a correção automática de problemas através de sistemas especialistas vem sendo desenvolvidas na Universidade Federal do Rio Grande do Sul [SAE 96a, ROC 96].

É crescente o número de fornecedores de equipamentos e de programas que vem procurando incluir facilidades que permitam o desenvolvimento do gerenciamento pró-ativo, através de facilidades especificadas na RMON MIB.

O conceito de séries temporais foi aplicado neste trabalho na tarefa de identificação do funcionamento normal da rede. Segundo Chatfield [CHA 84], uma série temporal é uma coleção de observações feitas seqüencialmente no tempo. No contexto de gerência pró-ativa, pode-se pensar os diversos parâmetros monitorados de uma rede como sendo séries temporais.

A aplicação deste conceito neste trabalho buscou a identificação de um modelo que representasse o funcionamento de um dos parâmetros monitorados, a saber o nível de utilização da

rede. Com base nesta representação matemática, busca-se estimar o nível de utilização do segmento em um tempo futuro.

Recentemente, grande parte dos esforços eram voltados para a tarefa de gerenciamento de redes, porém, o gerenciamento dos sistemas conectados a estas redes tornou-se cada vez mais importante. A experiência tem mostrado que, sem o perfeito funcionamento dos sistemas-chave , a rede torna-se indisponível para os usuários, ou aplicações. A necessidade de um método consistente que possibilite o gerenciamento dos recursos do sistema e das aplicações críticas torna-se cada vez mais necessário [KRU 95a, KRU 96]. Vários tem sido os esforços neste sentido, destacando-se [KRU 95a, KRU 95b].

Buscando contribuir nestes esforços este trabalho propõe: uma série de parâmetros a serem monitorados sobre as aplicações que estão sendo executadas na rede monitorada, um modelo de gerenciamento pró-ativo que utilize este nível de informações e um conjunto de funções de gerenciamento pró-ativas baseada em aplicações.

Este trabalho tem como objetivos principais o desenvolvimento de estudos sobre a representação do comportamento normal de uma rede monitorada e da aplicação do gerenciamento pró-ativo em nível das aplicações que estão sendo executadas nesta rede.

1.1. Organização do Trabalho

O presente trabalho está organizado em sete capítulos. O capítulo 2 apresenta os conceitos básicos sobre a arquitetura TCP/IP e gerenciamento de redes. O gerenciamento pró-ativo é abordado no capítulo 3. No capítulo 4 são tratados o gerenciamento pró-ativo proposto. A utilização de séries temporais no gerenciamento pró-ativo é apresentada no capítulo 5 e a análise da utilização de segmentos utilizando tais técnicas é abordada no capítulo 6. A conclusão e extensões deste trabalho são apresentadas no capítulo 7.

CAPÍTULO 2

CONCEITOS BÁSICOS: ARQUITETURA TCP/IP E GERENCIAMENTO DE REDES

Neste capítulo serão apresentados os principais conceitos do modelo de referência Internet e de gerenciamento de redes que servirão como conhecimentos básicos para os estudos efetuados neste trabalho.

2.1. MODELO DE REFERÊNCIA INTERNET

O projeto ARPANET, *ARPA experimental packet-switched Network,* constituiu-se de uma pesquisa de redes de computadores apoiada pelo DoD, *Department of Defence.* Ele chegou a conectar centenas de universidades e instalações governamentais através de linhas telefônicas. Posteriormente, quando passou-se a utilizar também a tecnologia de satélites e rádio, os protocolos existentes na época apresentaram problemas em interconectar tais redes, forçando a definição de uma nova arquitetura de referência. Nesta fase, um dos objetivos principais foi a habilidade de conectar várias redes de uma forma semelhante. Algum tempo depois, após os seus dois primeiros protocolos, esta arquitetura tornou-se conhecida como o **modelo de referência TCP/IP**, *Transmission Control Protocol/ Internet Protocol.* Outro objetivo das pesquisas era o de fazer com que a rede fosse capaz de continuar funcionando mesmo com a perda de uma parte dos seus equipamentos tais como *hosts*, roteadores e *gateways.* Além disso, a arquitetura deveria ser flexível para suportar a grande variedade de necessidades das aplicações que já se anteviam [TAN 96].

Com o intuito de aumentar o já grande crescimento apresentado pela ARPANET, nos anos de 1995 e 1996, a *National Science Foundation* (NFS), organização americana de incentivo a ciência, financiou a proliferação da tecnologia para as universidades americanas.

Atualmente, esta rede já conecta universidades, centros de pesquisas , empresas e órgãos governamentais do mundo todo e é conhecida como *Internet.*

2.1.1. ARQUITETURA DO MODELO DE REFERÊNCIA TCP/IP

A arquitetura *Internet* é composta por quatro camadas como mostra a Figura 2.1.

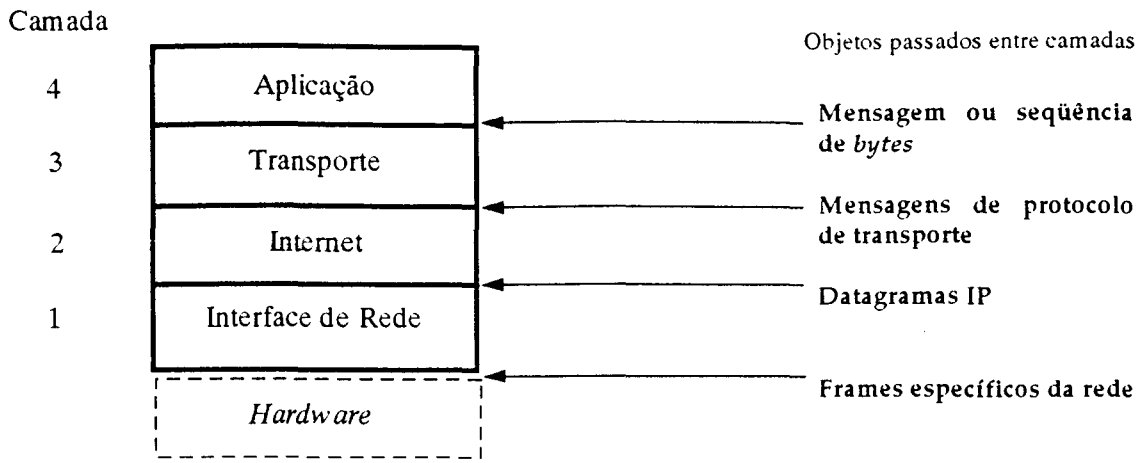


Figura 2.1 - Modelo de Referência TCP/IP

- **Camada de Aplicação**: neste nível são executados os protocolos de mais alto nível e é através deles que os usuários tem acesso à rede. Cabe a cada um dos programas de aplicação que são executados neste nível escolher sua forma de transmissão de dados, por fluxo de *bytes* ou por seqüência de mensagens.

1) **Camada de Transporte**: sua função é prover a comunicação fim-a-fim entre duas entidades de aplicação. Esta camada pode regular o fluxo de informações e fornecer um serviço confiável de transporte assegurando que os *bytes* enviados chegam sem erros e na seqüência correta, cabendo ao destinatário o envio de confirmações de recebimento e ao remetente, a retransmissão dos segmentos perdidos ou danificados.

- **Camada Inter-rede**: responsável pela transferência de dados através da inter-rede de uma máquina origem até a máquina destino. Ela é responsável pelo processamento das solicitações de transmissão da camada superior. Nestas solicitações, é informado o endereço da máquina destino. As informação são encapsuladas em um datagrama Internet e é aplicado o algoritmo de roteamento para se identificar o destino que o datagrama deve seguir. Cabe a ela também o tratamento dos datagramas recebidos, verificando a sua validade e decidindo, através do algoritmo de roteamento, se o datagrama deve ser enviado a outra máquina ou deve ser processado localmente e, neste caso, o programa desta camada é responsável pela passagem das informações recebidas ao nível de transporte. Cabe a esta camada o envio e o tratamento de mensagens ICMP, *Internet Control Message Protocol* [HUN 92, ROS 93].

- **Camada de Interface de Rede**: o protocolo de nível mais baixo é responsável por receber datagramas IP e transmiti-los em um formato específico de rede.

A Figura 2.2 [SOA 95] exemplifica o caminho que uma mensagem passa ao sair de um *host* A até chegar a um *host* B.

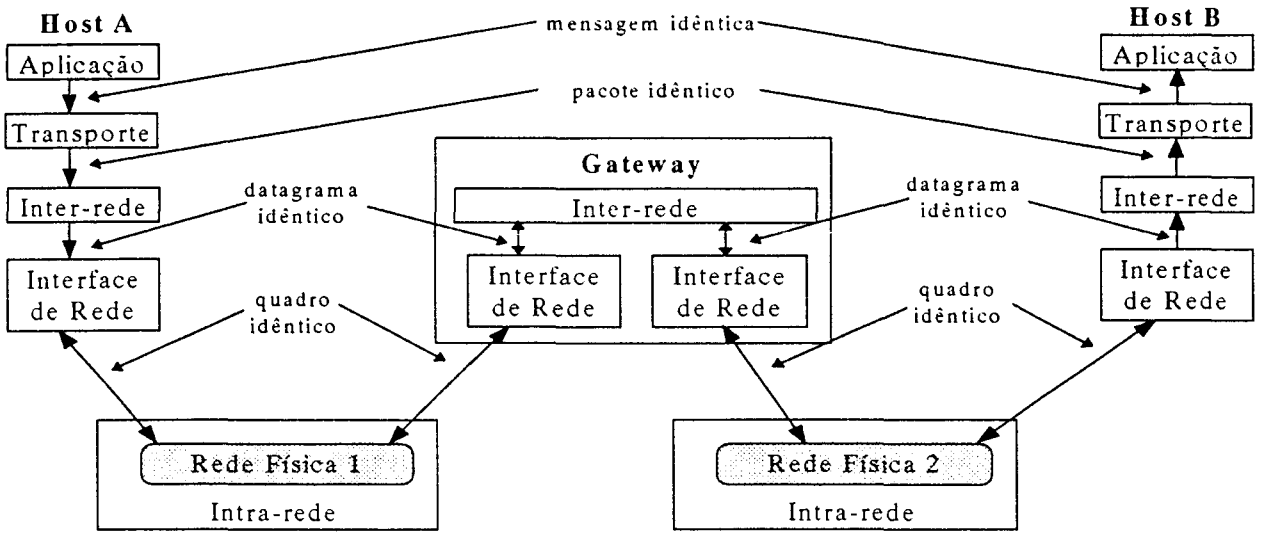


Figura 2.2 - Caminho Percorrido por uma Mensagem do *Host* A ao *Host* B

2.1.2.PROTOCOLOS INTERNET

Serão apresentados a seguir, os protocolos utilizados na Arquitetura Internet

2.1.2.1. INTERNET PROTOCOL (IP)

O serviço primordial da Internet é provido pelo protocolo *Internet Protocol* (IP) que provê o transporte dos datagramas, de um *host* fonte a um *host* destino, passando provavelmente por um ou mais roteadores e redes. Ambos, *hosts* e roteadores, estão envolvidos no processamento dos cabeçalhos IP. Cabe aos *hosts* a sua criação quando do envio do datagrama e o seu processamento quando da sua recepção. Já os roteadores examinam os cabeçalhos para que possam tomar as decisões referentes ao roteamento e modificá-las quando necessário [ROS 93].

Caracteriza-se por um serviço sem conexão, pois cada pacote é tratado de forma independente dos outros, e não confiável pois o pacote pode ser perdido ou atrasado sem que o serviço reporte a ocorrência de tais fatos. Cabe salientar que a ordem com que os pacotes chegarão ao *host* destinatário não é garantida. A Figura 2.3 apresenta o formato do datagrama IP.

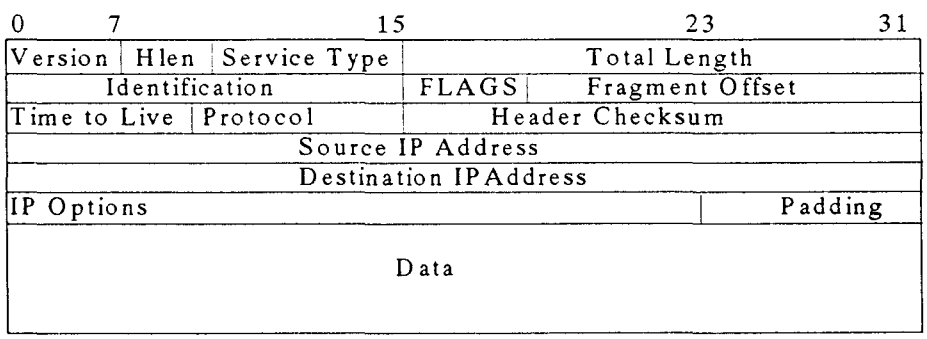


Figura 2.3 - Datagrama IP

1) *Version*: especifica a versão do protocolo IP utilizado na criação do datagrama. É utilizado para verificar que o remetente, destinatário e os roteadores estão em concordância com o formato do datagrama.
2) IHL (*Internet Header Length*): comprimento do cabeçalho IP em palavras de 32 *bits*. Usualmente possui o valor cinco, caso em que o campo de opções encontra-se vazio.
3) *Type of Service*: especifica como o datagrama deve ser processado. É composto pelos seguintes campos:
   - Precedência: indica a prioridade do datagrama, podendo variar de 0, precedência normal, a 7, controle de rede.
   - D: requer pequeno atraso no processamento, quando setado.
   - T: estando setado, significa que o datagrama requer grande vazão de dados.
   - R: quando setado, indica que o datagrama requer alta confiabilidade.
4) *Identification*: número inteiro que identifica unicamente um datagrama ou seus fragmentos.
5) *Flags*: este campo é composto por 3 *bits*:
   - Primeiro *bit*: sempre terá o valor zero.
   - Segundo *bit*: conhecido como *do not fragment*, indica se o datagrama pode ser fragmentado ou não.
   - Terceiro *bit*: chamado de *more fragments*, indica se o fragmento contém dados do meio ou do final do datagrama original.
6) *Fragment Offset*: especifica o deslocamento em relação ao início do datagrama original em valores múltiplos de oito *bytes*, no caso de fragmentos.
7) *Time to Live*: indica o tempo máximo que o datagrama deve permanecer na rede. Cada vez que o datagrama passa por um roteador, este deve decrementar este campo, visando a eliminação dos datagramas que circulam na rede devido a ocorrência de falhas de roteamento. Como as redes atualmente implementadas não fazem o controle de tempo, este campo é atualmente preenchido com o número máximo de roteadores em que o datagrama pode ser processado.
8) *Protocol*: indica que protocolo de nível superior está sendo utilizado. Na maioria das vezes, assume o valor 6, TCP, ou 7, UDP.
9) *Header Checksum*: utilizado na verificação da integridade do cabeçalho IP. É calculado obtendo-se a soma, em complemento de um, do cabeçalho como uma seqüência de inteiros, palavras de 16 *bits*, e então fazendo-se o complemento de um deste resultado.
10) *Source IP Address*: endereço IP do remetente do datagrama.
11) *Destination IP Address*: endereço IP do destinatário do datagrama.
12) *Options*: campo de tamanho variável, sempre ajustável através do *padding* para conter múltiplos de 32 *bits*, utilizados principalmente para testes de rede.

Procurando solucionar alguns problemas existentes na versão em uso do protocolo IP, versão 4, o IETF deu início a novos estudos que solucionassem os seguintes problemas:
1. Suporte a bilhões de máquinas, mesmo com alocação ineficiente de endereços;
2. Simplificação do protocolo;
3. Melhorias na segurança do protocolo;
4. Redução do tamanho das tabelas de roteamento;
5. Levar em consideração o tipo de serviço, em especial para dados em tempo real;
6. Permitir a evolução do protocolo futuramente;
7. Possibilitar a coexistência entre o novo protocolo e as versões anteriores.

Estas pesquisas resultaram na versão 6 do protocolo IP, conhecida como *Simple Internet Protocol Plus* (SIPP). Como melhorias em relação a versão anterior, destacam-se:

1. Aumento do campo de endereço, que passa a ser de 16 *bytes*, suportando bilhões de máquinas;
2. Simplificação do cabeçalho, que passa a conter apenas 7 campos, permitindo um processamento mais rápido dos mesmos;
3. Melhor suporte a opções, transformando campos que antes eram obrigatórios em opcionais além da mudança de como as opções são representadas;
4. Avanços na segurança baseados em autenticação e privacidade.

2.1.2.1.1. ROTEAMENTO NO PROTOCOLO IP

A função principal do protocolo IP é o roteamento inter-redes. Este protocolo parte do pressuposto de que as máquinas são capazes de enviar datagramas para qualquer outra máquina que esteja no mesmo segmento de rede que o seu. Quando se deseja enviar um datagrama IP para uma máquina que esteja fora do segmento da máquina origem, parte da função de roteamento é executada pelos gateways, devendo o módulo IP da máquina origem apenas enviar o datagrama para um dos gateways conectados ao seu segmento.

Por basear-se exclusivamente no identificador de rede do endereço destino, cada computador armazena uma tabela que contêm informações do tipo: endereço de rede/endereço de roteador, conhecida como tabela de roteamento IP. Quando um módulo IP necessita encaminhar um datagrama, ele verifica se a máquina destino está no seu segmento de rede e neste caso, o datagrama é repassado à interface da rede que por sua vez mapeará o endereço IP no endereço físico da máquina destino e transmitirá o datagrama.

Caso a máquina destino encontre-se fora do segmento da máquina origem, o módulo IP procura em sua tabela de roteamento a existência de uma entrada com o endereço destino do datagrama. Se existir, ele descobre qual o endereço do roteador que deve ser utilizado.

O tamanho das tabelas de roteamento é uma desvantagem deste esquema de roteamento. Visando a minimização deste problema, várias são as estratégias utilizadas.

Uma delas é a utilização de roteadores *default*. Por exemplo, pode-se especificar um determinado roteador como caminho *default* para todos os pacotes que são direcionados para fora do segmento, no caso de redes que possuem apenas um roteador para fora do segmento.

Em redes com mais de um roteador, o conceito de roteador *default* é aplicado da seguinte forma: quando não for encontrada uma rota específica para um datagrama, ele é enviado ao roteador *default*. Caberá ao roteador informar ao módulo IP se ele é ou não a melhor escolha para se alcançar uma determinada rede. Esta informação será passada através de uma mensagem chamada *redirect* do protocolo *Internet Control Message Protocol* (ICMP), que contêm o endereço do roteador correto. Quando o módulo IP recebe este tipo de mensagem, ele acrescenta uma entrada em sua tabela de roteamento associando a rede de destino e o endereço do roteador recebido na mensagem. A partir de então, todas as referências à rede em questão serão corretamente encaminhadas ao novo roteador a ela associado.

A implementação do roteamento exige uma estrutura de dados que informe os possíveis caminhos e os seus custos, a fim de que se possa decidir qual o melhor. Dentre os diversos métodos para a manutenção da estrutura de dados, destacam-se:

1) Encaminhamento por rota fixa: a tabela, uma vez criada, não é mais alterada. As rotas são fixas e os caminhos alternativos são tomados só em caso de falhas. Bastante simples porém leva a uma má utilização dos recursos da rede.

2) Encaminhamento adaptativo: a rota é escolhida de acordo com a carga na rede. Nas tabelas de rotas são mantidas informações sobre o tráfego, que são consultadas para a escolha do caminho mais curto. A atualização periódica das tabelas pode ser feita de vários modos:

- Modo isolado: a atualização é realizada com base nas filas de mensagens para os diversos caminhos e outras informações locais.
- Modo distribuído: cada nó envia periodicamente aos outros nós, incluindo os *gateways*, informações locais sobre a carga da rede. Estas informações são utilizadas para o cálculo da nova tabela.
- Modo centralizado: cada nó envia a um ponto central da rede as informações locais sobre a carga. Essas informações são utilizadas pelo ponto central para o cálculo das novas tabelas, que são então enviadas aos *gateways* e demais nós.

Dentre as várias formas de se efetuar o roteamento de pacotes, destacam-se as seguintes:

- **Roteamento centralizado**

Parte do pressuposto de que existe, em algum lugar da rede, um Centro de Controle de Rotas (CCR) responsável pelo cálculo das tabelas de rotas. Este modelo apresenta como vantagem o fato do CCR sempre poder tomar decisões precisas sobre o caminho ótimo, uma vez que possui todas as informações da rede.

Uma desvantagem do mesmo é o fato de que, visando adaptar-se a um tráfego mutável, o cálculo das tabelas deve ser realizado com muita freqüência. Para uma rede com muitos nós esse cálculo pode exigir um processamento muito grande. Outras desvantagens deste modelo seria o alto tráfego que as linhas que levam ao CCR deverão suportar e a confiabilidade do mesmo. Além disso, devido aos nós receberem suas tabelas em tempos diferentes, devido a atrasos diferentes sofridos pelas mensagens que as transportam, pode-se gerar inconsistências de roteamento causando maiores atrasos, inclusive nas mensagens que transportam as tabelas, realimentando o processo.

- **Roteamento isolado**

A atualização é realizada com base nas filas de mensagens para os diversos caminhos e outras informações locais.

Um algoritmo de roteamento isolado simples é baseado na idéia de que ao receber um pacote, um nó tenta livrar-se dele imediatamente pelo enlace que possui a fila mais curta no momento. Uma melhora neste algoritmo pode ser conseguida combinando-o com o encaminhamento por rota fixa. Desta forma, ao encaminhar-se um pacote, deve-se levar em consideração não só a rota estática especificada, mas também o tamanho das filas.

- **Roteamento distribuído**

No modo distribuído, cada nó envia periodicamente aos outros nós, incluindo os *gateways*, informações locais sobre a carga na rede. Essas informações são utilizadas para o cálculo da nova tabela. Apresenta-se a seguir, um algoritmo que utiliza este tipo de roteamento.

Quando um nó quer descobrir o caminho mais curto até um nó de destino, ele envia um pacote de requisição por difusão na rede, com um campo contendo o endereço do nó de destino e um campo contendo um caminho, inicialmente com seu endereço. Cada nó intermediário ao receber o pacote de difusão verifica se o campo caminho contém seu endereço. Se não contiver, o nó acrescenta ao caminho seu endereço e o difunde por todos os seus enlaces. Caso contrário, simplesmente descarta o pacote. O nó de destino vai receber vários pacotes, pelos diversos caminhos que possui desde o nó de origem. O primeiro deles contém no campo caminho a rota de menor retardo. O nó de destino envia então, como resposta ao pacote recebido, um outro pacote contendo essa rota.

Visando a manutenção da tabela de rotas, um nó deve periodicamente enviar pacotes de difusão para descobrimento das rotas. Quanto menor o período, mais adaptável às flutuações de tráfego é o algoritmo. Quanto maior é o período, menor é o tráfego que o algoritmo gera na rede.

- **Roteamento hierárquico**

Aconselhável para grandes redes. Neste roteamento, os nós são divididos em regiões, com cada nó capaz de manter as informações de rotas das regiões a que pertence. Para muitas redes, no entanto, essa hierarquia em dois níveis é insuficiente, podendo ser necessário agrupar regiões em super-regiões e assim sucessivamente.

2.1.2.1.2. ROTEADORES E *GATEWAYS*

Os *gateways* são classificados usualmente em dois tipos:

- ***gateways* conversores de meio (*media-conversion gateway*)**: os mais simples. Bastante utilizados em inter-redes que oferecem o serviço de datagrama, suas funções resumem-se em receber um pacote do nível inferior, tratar o cabeçalho inter-redes do pacote, descobrindo o roteamento necessário, construir novo pacote com novo cabeçalho inter-redes, se necessário, e enviar esse novo pacote ao próximo destino, segundo o protocolo da rede local em que este se encontra. Este tipo de *gateway* da camada de rede é também chamado de roteador.

- ***gateways* tradutores de protocolo (*protocol-translation gateway*)**: são mais utilizados em inter-redes que utilizam circuitos virtuais passo a passo. Eles atuam traduzindo mensagens de uma rede, em mensagens de outra rede, com a mesma semântica de protocolo. Entretanto nem todos os protocolos podem ser mapeados entre si, e o subconjunto formado pela interseção dos serviços comuns é o serviço que deverá ser oferecido como base para a interligação. As dificuldades na tradução dos protocolos tornam tais *gateways* bastante complexos e de difícil realização, o que pode aumentar em muito o custo da interligação. Este tipo de *gateway* pode atuar em qualquer nível acima do nível de enlace RM-OSI.

Pode-se observar também que a utilização de *gateways* para a conexão de redes locais idênticas não sofre nenhuma das restrições existentes para pontes e repetidores como, por exemplo, tamanhos dos quadros, esquemas de prioridade, entre outros. A desvantagem está na sua maior complexidade, na exigência de um protocolo inter-redes, enfim, no custo da interligação.

2.1.2.1.2.1.CONFIGURAÇÃO DE ROTEADORES

Roteamento é a "cola" que mantém a Internet unida. Sem isso, o tráfego TCP/IP seria limitado a uma única rede física. O roteamento permite que o tráfego de uma rede local atinja seu destino em qualquer outro local do mundo. Destacam-se dois conceitos básicos:

- Roteamento: ato de se passar datagramas baseado nas informações contidas na tabela de roteamento.
- Protocolo de Roteamento: programas que exportam a informação utilizada para a construção da tabela de roteamento.

A configuração de roteamento de uma estação específica nem sempre requer um protocolo de roteamento. Em situações onde a informação de roteamento não precisa mudar, por exemplo, quando há apenas uma rota possível, o administrador normalmente constrói a tabela de roteamento manualmente. As três configurações mais comuns de roteamento são:

- Roteamento Mínimo: uma rede completamente isolada requer apenas um roteamento mínimo. Uma tabela de roteamento mínima é construída pelo comando ifconfig quando a interface de rede é configurada.
- Roteamento Estático: uma rede com um número limitado de gateways para outras redes TCP/IP podem ser configuradas com roteamento estático. Uma tabela de roteamento estático é construída manualmente pelo administrador através do comando route. Tabelas de roteamento estáticas não se ajustam às mudanças da rede, devem ser utilizadas apenas onde as rotas não mudam. Porém, quando destinos remotos podem ser alcançados apenas por uma rota, uma rota estática é a melhor escolha.
- Roteamento Dinâmico: uma rede com mais de uma rota possível para o mesmo destino deve utilizar este tipo de roteamento. Uma tabela de roteamento dinâmico é construída a partir de informações exportadas pelos protocolos de roteamento. Os protocolos são planejados para distribuir informações que dinamicamente ajustam as rotas para refletir as mudanças nas condições da rede. Os protocolos de roteamento tratam com situações complexas de roteamento mais rápido e apurado que o administrador é capaz. Os protocolos de roteamento são planejados não apenas para desviar para uma rota reserva quando a rota primária torna-se inoperante, são planejados para decidir qual é a "melhor" rota para um destino. Em qualquer rede onde há múltiplos caminhos para o mesmo destino, um protocolo de roteamento deve ser utilizado.

A combinação destes tipos de roteamento é utilizada por vários sistemas.

2.1.2.1.3. ALGORITMOS DE ROTEAMENTO

O algoritmo de roteamento é responsável por decidir qual linha um determinado pacote deverá tomar. Se a subrede em questão utiliza datagramas, esta decisão deve ser tomada para cada novo pacote a ser transmitido desde a última alteração na especificação da melhor rota. Se a

subrede em questão utiliza-se de circuitos virtuais, as decisões de roteamento são feitas apenas quando um novo circuito virtual está sendo estabelecido. Após esta fase, os pacotes seguem a rota previamente estabelecida.

2.1.2.1.3.1. ROTEAMENTO BASEADO NO FLUXO

Uma outra categoria de algoritmo de roteamento é a que leva em consideração o fluxo de informações entre os **hosts**.

Em algumas redes, a média do fluxo de dados entre os nós é relativamente estável e previsível. Sob condições nas quais a média do tráfego é conhecida *a priori*, é possível analisar os fluxos matematicamente visando a otimização do tráfego [TAN 96].

A idéia básica por trás desta análise é a de que, para uma determinada linha, se a capacidade e a média do fluxo são conhecidas, é possível calcular o *delay* médio dos pacotes naquela linha através da teoria das filas. Da média dos *delays* de todas as linhas, pode-se calcular um "peso" de fluxo médio para se conseguir o delay médio dos pacotes de toda a subrede. O problema de roteamento reduz-se a encontrar um algoritmo de roteamento que produz o *delay* médio mínimo para a subrede.

2.1.2.2. TRANSMISSION CONTROL PROTOCOL (TCP)

O protocolo TCP foi planejado para operar sobre o protocolo IP, provendo uma comunicação segura, seqüenciada e com controle de fluxo fim-a-fim.

Em um nível mais alto, as aplicações enviam grandes volumes de dados de uma máquina para outra. Se todo protocolo de aplicação tivesse de se ocupar no gerenciamento do transporte dos pacotes enviados e recebidos, a carga de trabalho necessária para o desenvolvimento e implementação de aplicações seria maior do que a existente hoje.

Visando solucionar este problema, projetou-se um protocolo de transporte confiável chamado *Transmission Control Protocol* (TCP).

Neste protocolo, as aplicações que estão se comunicando são identificadas através do uso de portas. Entretanto, as portas são mais complexas pois representam, na verdade, conexões identificáveis por um par de pontos através de tuplas (máquina, porta).

Baseado nesta abstração e em um conjunto de campos, o protocolo TCP implementa as facilidades de controle de fluxo, retransmissão e reconhecimento de pacotes, tornando-o um protocolo confiável.

A unidade de transferência utilizada neste protocolo chama-se segmento. A estrutura de um segmento TCP é apresentada na Figura 2.4.

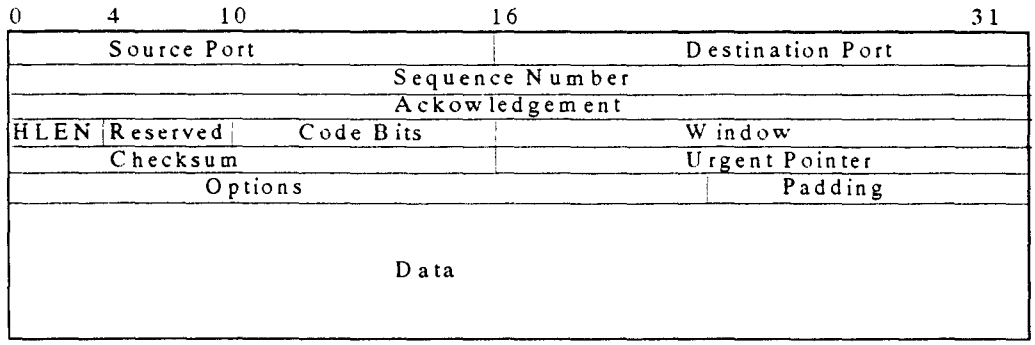


Figura 2.4 - Formato do Segmento TCP

- *Source Port*: número que identifica a porta TCP remetente;
- *Destination Port*: número que identifica a porta TCP destino;
- *Sequence Number*: campo utilizado para o controle de seqüência dos dados;
- *Acknowledgment*: utilizado para a confirmação de segmentos recebidos;
- HLEN: comprimento do cabeçalho do segmento, em palavras de 32 *bits*;
- *Code Bits*: utilizados para a definição de regras de prioridade no tratamento do segmento;
- *Window*: campo utilizado para o controle do tamanho da janela utilizada pela conexão;
- *Checksum*: utilizado para checar a integridade do segmento;
- *Urgent Pointer*: aponta para a posição em que se localizam os dados urgentes. Válido apenas quando o *Code Bit* mais significativo, URG, estiver setado.

Este protocolo possui algumas portas reservadas para aplicações mais utilizadas, chamadas de *well-known ports*. A Tabela I apresenta algumas destas portas e as respectivas aplicações a elas associadas [POS 94].

Número da Porta (Servidor)	Palavra Chave	Descrição
0	-	Reservado
11	USERS	Usuários ativos
13	DAYTIME	Hora corrente
20	FTP-DATA	Dados de controle do File Transfer Protocol
21	FTP	Dados de arquivo do File Transfer Protocol
23	TELNET	Conexão remota de terminal
25	SMTP	Correio Eletrônico
42	NAMESERVER	Servidor de nomes
70	GOPHER	Servidor Gopher
79	FINGER	Finger
80	WWW	Servidor WWW

Tabela I - Algumas portas *well-known* do TCP

2.1.2.3. USER DATAGRAM PROTOCOL (UDP)

O protocolo UDP proporciona o serviço de transporte sem conexão com o mecanismo mais simples que um protocolo de aplicação pode dispor. A distinção entre as diferentes aplicações em uma mesma máquina é feita através de campos chamados *ports*. Cada pacote UDP possui uma tupla *source/destination port* que possibilita a comunicação entre duas aplicações. A Figura 2.5 apresenta o formato do pacote UDP.

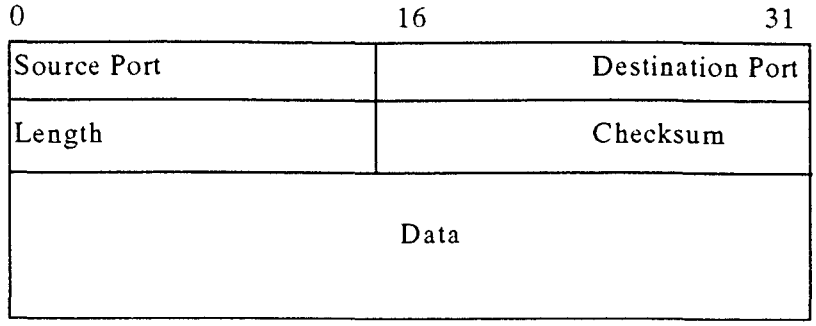


Figura 2.5 - Formato do Pacote UDP

- *Source Port*: número de 16 *bits* identificando a aplicação na máquina origem;
- *Destination Port*: número de 16 *bits* identificando a aplicação na máquina destino;
- *Message Length*: comprimento do pacote, incluindo o cabeçalho e os dados, em unidades de oito *bits*;
- *Checksum*: o cálculo deste campo é opcional visando a minimização na carga de processamento caso seja necessário. Quando apresenta o valor zero, significa que não foi calculado.

Algumas portas, conhecidas como *well-known ports*, apresentam o seu valor padronizado para que os serviços mais conhecidos sejam acessados com mais rapidez, evitando a negociação de uma determinada porta entre as aplicações, antes do início da transmissão propriamente dita. A Tabela II apresenta algumas destas portas e as aplicações a elas associadas.

Número da Porta (Servidor)	Palavra Chave	Descrição
0	-	Reservado
11	USERS	Usuários Ativos
13	DAYTIME	Hora Corrente
42	NAMESERVER	Servidor de Nomes
69	TFTP	Trivial File Transfer
111	SUNRPC	Sun Microsystems RPC
161	SNMP	Gerente SNMP
162	SNMP	Traps SNMP
513	WHO	Unix Rwho Daemon

Tabela II - Algumas portas *well-known* do UDP

2.1.3. PROTOCOLOS DE APLICAÇÃO

Os protocolos inferiores à camada de aplicação apenas proporcionam o transporte de mensagens confiável ou não. Cabe aos protocolos de aplicação utilizar as funcionalidades oferecidas por tais níveis a fim de prover os recursos para que as verdadeiras aplicações, tais como FTP e TELNET, possam ser executadas. A Tabela III apresenta alguns dos protocolos de aplicação mais conhecidos.

Palavra Chave	Significado	Descrição
FTP	File Transfer Protocol	Transferência de Arquivos
SMTP	Simple Mail Transfer Protocol	Correio Eletrônico
SNMP	Simple Network Management Protocol	Protocolo de Gerenciamento de Redes
DNS	Domain Name Server	Serviço de Nomes
HTTP	Hipertext Transfer Protocol	Protocolo do utilitário WWW

Tabela III - Alguns Protocolos de Aplicação

A grande maioria das aplicações baseadas no protocolo TCP/IP foi criada inspirada no modelo cliente-servidor. Este modelo executa a comunicação de rede baseada na troca de mensagens entre dois programas ou processos, sendo que um é o processo do usuário e o outro, o processo servidor.

O processo de usuário pode ser de duas formas: controlado diretamente pelo usuário, através de uma interface, ou um conjunto de instruções pré-definidas. Este processo normalmente é a parte ativa na comunicação, cabendo-lhe o envio de pedidos de informações ou serviços para o processo servidor e a interpretação das respostas.

O processo servidor normalmente está conectado a um conjunto de recursos do sistema. Normalmente é a parte passiva que aguarda por uma solicitação de serviço, executando-a e enviando a resposta.

2.1.3.1. TELNET

O protocolo TELNET [POS 83] permite a um usuário de uma máquina A estabeleça uma sessão interativa com uma máquina B na rede, fazendo com que a máquina A passe a ser um terminal da máquina B.

A identificação da máquina à qual o módulo cliente deseja se conectar pode ser feita tanto pelo nome quanto pelo endereço IP, forma esta que possibilita o estabelecimento de conexões remotas em ambientes onde a associação entre nomes e endereços IP não seja possível.

Três são as principais idéias deste protocolo:

1. Conceito de Terminal Virtual: dispositivo lógico que fornece uma representação padronizada de um terminal. Possibilita a abstração das características das máquinas que estão comunicando-se.
2. Princípio de Negociação de Opções: permite a negociação de opções sobre o comportamento do terminal virtual.

3. Tratamento Equivalente de Terminais e Processos: possibilita que o cliente seja um terminal virtual ou um processo de aplicação e permite que a ambas as extremidades a iniciativa da negociação das opções.

A Figura 2.6 [SOA 95] ilustra o funcionamento de uma conexão TELNET.

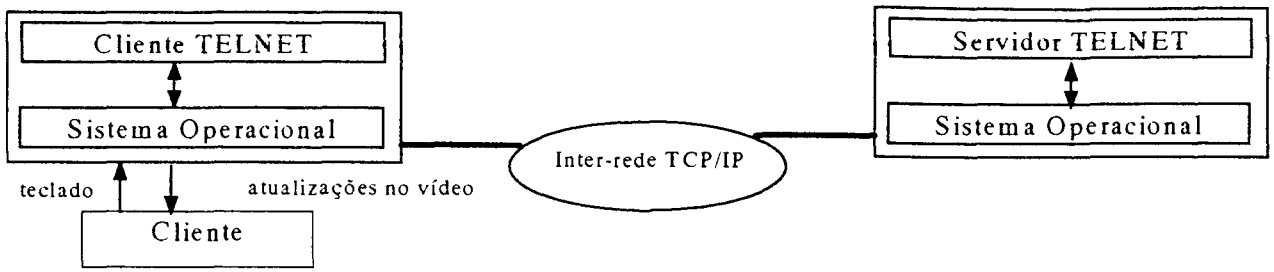


Figura 2.6 - Conexão TELNET

2.1.3.2. FILE TRANSFER PROTOCOL (FTP)

O *File Transfer Protocol* (FTP) [POS 85], possibilita a transferência e manipulação de arquivos e diretórios remotos. A Figura 2.7 apresenta um esquema da arquitetura FTP.

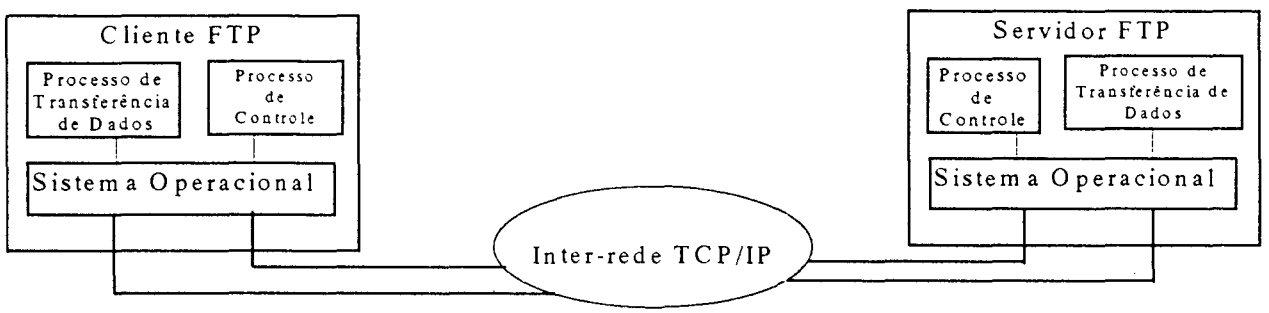


Figura 2.7 - Conexão FTP de Controle e de Transferência de Dados

O início desta aplicação é caracterizado pela identificação do usuário que deseja manipular arquivos/diretórios de forma remota, com a identificação do mesmo através de seu nome (*login name*) e de sua senha de acesso. Contas especiais criadas pelos administradores do sistema, conhecidas através do nome do usuário anônimo, permitem a usuários não cadastrados no sistema possam manipular os arquivos da máquina, normalmente com restrições de acesso.

Após a permissão de acesso tenha sido dada ao usuário, duas conexões são estabelecidas entre o módulo que solicitou a conexão (cliente) e o módulo que fornece o acesso aos seus arquivos (servidor):

- Conexão de controle: utilizada na transferência de comandos, permanece aberta enquanto durar a sessão de manipulação remota de arquivos;
- Conexão de transferência de dados: utilizada na transferência de arquivos propriamente dita. Esta conexão existe apenas quando está sendo feita uma transferência de arquivo.

Além do protocolo FTP para a transferência de arquivos, a arquitetura UDP/IP define o protocolo *Trivial File Transfer Protocol* (TFTP) [SOL 81] que fornece um serviço simplificado de transferência de dados, não implementando mecanismos de autenticação e operando em uma única

conexão. Este protocolo é utilizado principalmente por máquinas que precisam transferir arquivos de algum servidor quando da sua inicialização.

2.1.3.3. SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

O protocolo *Simple Mail Transfer Protocol*, SMTP, é utilizado no sistema de correio eletrônico. A Figura 2.8 [SOA 95] apresenta os componentes conceituais utilizados no sistema de correio eletrônico.

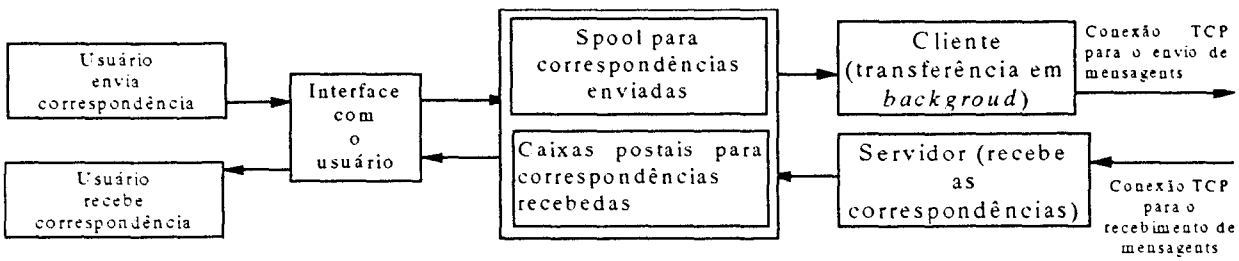


Figura 2.8 - Arquitetura do Correio Eletrônico SMTP

A operação do sistema de correio eletrônico segue os seguintes passos:

1. O usuário compõe a mensagem e solicita ao sistema de correio eletrônico, através da interface com o usuário, a sua entrega;
2. O sistema de correio eletrônico recebe a mensagem, armazena uma cópia da mesma juntamente com o horário que o armazenamento ocorreu e a identificação do remetente e do destinatário. A partir deste momento, o processo do usuário fica livre para executar outras tarefas;
3. Cabe ao processo de transferência verificar a existência de mensagens a serem transmitidas e a transmissão das mesmas. Quando este processo tenta enviar uma mensagem, ele passa a ser cliente do processo servidor responsável pela recepção de mensagens da máquina destino. Neste contexto, o processo cliente tenta estabelecer uma conexão TCP com o processo servidor da máquina destino. Se a conexão for estabelecida, o processo cliente envia uma cópia da mensagem para o servidor que se encarregará de armazená-la e de enviar uma mensagem de confirmação do recebimento da mesma. Ao receber esta confirmação, o processo cliente retira a cópia armazenada localmente. Caso a confirmação não seja recebida, o processo cliente grava o horário da tentativa e suspende a sua execução. Se o cliente recebe a confirmação esperada, este retira a cópia armazenada localmente. Caso contrário, o processo cliente grava o horário da tentativa e suspende a sua execução. Uma mensagem que não foi transmitida em um determinado período, é devolvida pelo serviço de correio eletrônico ao remetente, informando que não conseguiu transmiti-la.

Uma mensagem SMTP é composta de cabeçalho e corpo, separados por uma linha em branco. O cabeçalho contém as informações necessárias para o envio da mensagem na forma de palavra-chave e valor como, por exemplo, em: **to: endereço do destinatário**. O corpo contém as informações da mensagem propriamente dita, seu formato é livre e sua transferência dá-se no formato texto.

Os usuários do sistema de correio eletrônico são identificados através de um par de identificadores da forma: **nome local@nome do domínio**, onde **nome local** identifica a conta local do usuário e **nome do domínio** identifica o domínio ao qual a máquina pertence.

2.1.3.4. NETWORK FILE SYSTEM (NFS)

O protocolo NFS, *Network File System*, desenvolvido pela Sun Microsystems, tem como objetivo prover um mecanismo consistente de acesso a arquivos.

Este protocolo provê transparência tanto ao sistema de arquivos quanto a rede, pois ele é responsável pelo gerenciamento físico dos arquivos, permitindo que todas as máquinas de um determinado domínio compartilhem um único sistema de arquivos, responsabilizando-se pelo tratamento de problemas como cópias duplicadas de arquivos em máquinas diferentes.

Sua implementação é baseada em um conjunto de procedimentos *Remote Procedure Call* (RPC) que utiliza-se da codificação *External Data Representation* (XDR) para a passagem de argumentos entre cliente e servidor. Um sistema de arquivos baseado neste protocolo oferece dois níveis de transparência [STE 91]:

1. O sistema de arquivos parece estar no disco local da máquina, e todos os arquivos e diretórios são vistos da mesma forma, sejam eles locais ou não pois o protocolo trata de ocultar a localização do arquivo na rede. Este nível de transparência é conseguido através da utilização de um conjunto de operações genéricas de sistema de arquivos, que são processadas em um VFS, *Virtual File System.*
2. O sistema de arquivos não mantêm nenhuma informação sobre o servidor de arquivos que ele está utilizando, possibilitando desta forma, a comunicação transparente entre máquinas de diferentes arquiteturas ou sistemas operacionais. Disponibilizado através da definição de nós virtuais, *virtual nodes*, semelhantes à estrutura *inode* do sistema de arquivos Unix.

2.1.3.4.1. VIRTUAL FILE SYSTEMS (VFS) E VIRTUAL NODES

O VFS permite a um sistema cliente acessar diversos tipos diferentes de sistemas de arquivos de forma transparente, ocultando as diferenças entre as diversas implementações sobre uma interface consistente.

As ações que operam sobre todo o sistema de arquivos, tal como descobrir a quantidade de espaço livre no sistema, são chamadas de operações VFS, já as chamadas que operam sobre arquivos ou diretórios são conhecidas como operações *vnode*. Na visão do servidor, a implementação de uma VFS requer a conversão das operações genéricas sobre VFS e *vnode* e a conversão delas em ações do sistema operacional local, conversão esta que deve ser feita de forma transparente para o processo NFS cliente.

A "nomeação" de objetos no sistema é feita através da utilização de *file handles*, que são "apontadores" opacos para os arquivos no sistema remoto. Seu valor tem significado apenas no contexto do sistema de arquivos remoto. Uma chamada NFS deve conter um *file handle* para o *vnode*. É de responsabilidade da camada de sistema de arquivos virtual traduzir o descritor de arquivos do nível de usuário em *file handle* a nível de *kernel*.

2.1.3.4.2. PROTOCOLO NFS

Este protocolo é baseado em RPC e no paradigma de cliente-servidor.

Os *daemons* de servidores NFS , conhecidos como *daemons* **nfsd**, executam na máquina servidora e processam as chamadas RPC dos clientes. O *daemon* **rpc.mountd** também é executado no servidor para tratar de solicitações de "*mount*" de sistemas de arquivos e algumas traduções de *pathnames*.

No lado do cliente, cada processo que utiliza arquivos NFS é um cliente do servidor. As chamadas de sistema do cliente que utilizam arquivos NFS executam chamadas RPC para o servidor NFS onde estes arquivos estão "montados".

2.1.3.5. SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Este protocolo é utilizado no gerenciamento de redes de computadores e está comentado na Seção 2.2.2.1.1 .

2.1.3.6. DOMAIN NAME SERVICE (DNS)

Este serviço define um esquema de gerenciamento de nomes hierárquico e distribuído. O esquema de nomes é baseado em domínios e implementado através de um banco de dados distribuído que associa endereços IP de 32 *bits*, normalmente utilizados pelos programas, a nomes que identificam máquinas, geralmente utilizados pelas pessoas.

Visando evitar os problemas associados à existência de uma única fonte de informação, o espaço de nomes do DNS é dividido em zonas não sobrepostas. Cada zona contém uma parte do conjunto global de nomes e também o nome dos servidores que tem a autoridade de fornecer informações sobre aquela zona. Usualmente, uma zona possui um servidor de nomes primário, o qual contém as informações em sua unidade de armazenamento, e um ou mais servidores de nomes secundário os quais pegam suas informações do servidor primário [ALB 92].

Quando um processo precisa resolver uma pergunta sobre um nome, ele repassa-a ao servidor de nomes local. Se o domínio que está sendo procurado pertence a jurisdição do servidor local, ele retorna uma informação proveniente da autoridade local que sempre está correta conhecida como ***authoritative record***. Caso se trate de um domínio remoto e não exista nenhuma informação disponível localmente sobre aquele domínio, o servidor de nomes envia uma pergunta ao servidor de nomes de um nível superior ao seu. Caso este também não consiga resolver a pergunta, repassa-a ao servidor superior a ele e assim sucessivamente até chegar ao nível mais elevado, caso necessário, e a resposta retornará ao servidor que iniciou o processo. A Figura 2.9 ilustra este processo, caso a máquina itajaí.dcc.unicamp.br queira se comunicar coma máquina venus.inf.usfc.br.

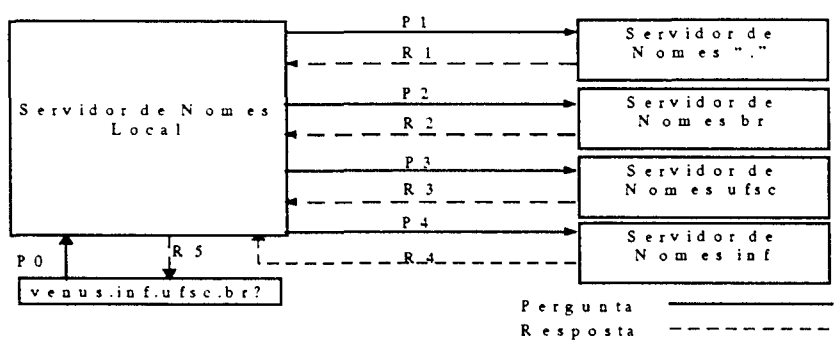


Figura 2.9 - Processo de Resolução de Nomes

2.1.3.7. HIPERTEXT TRANSFER PROTOCOL (HTTP)

O HTTP é o padrão de transferência da aplicação WWW, *World Wide Web*. Cada interação entre o processo cliente e o processo servidor consiste de um pedido em formato ASCII seguido por uma resposta no formato MIME, *Multipurpose Internet Mail Extensions*, especificado em [BOR 93]. Este protocolo encontra-se ainda em evolução, várias versões encontram-se atualmente em uso e outras estão sendo desenvolvidas.

Este protocolo consiste de dois itens distintos: o conjunto de pedidos do *browser* para o servidor e o conjunto de resposta no caminho contrário.

Todas as novas versões deste protocolo suportam dois tipos de solicitação:

- Solicitação simples: formada por uma única sentença do tipo GET *nome da página desejada* e tem como resposta apenas a página solicitada.
- Solicitação completa: pode ser formada por várias linhas, seguido por uma linha em branco que indica o final da solicitação. A primeira linha contém um comando que pode ser o comando GET ou não e o número do protocolo/versão. As linhas subseqüentes possuem cabeçalhos no formato especificado na RFC 1521.

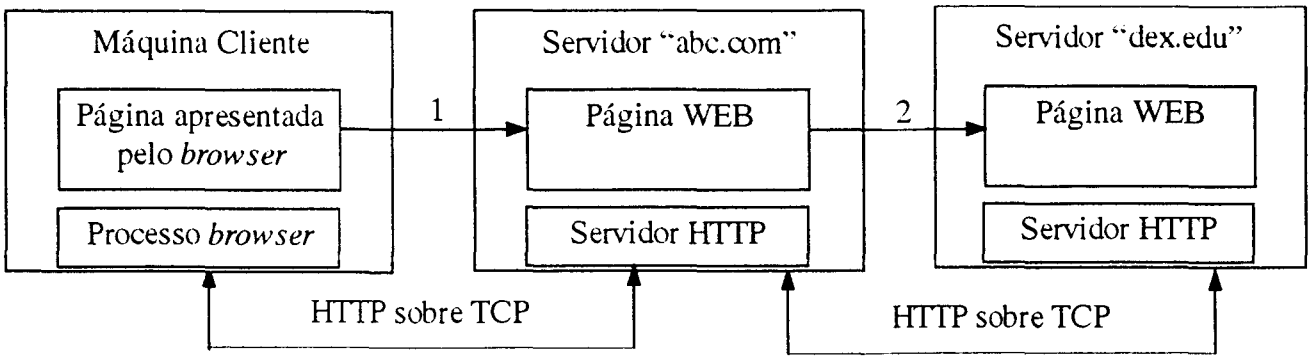
2.1.3.7.1. VISÃO DO CLIENTE

Na visão do usuário, o banco de dados, que pode ser acessado através da aplicação WWW, é uma vasta coleção mundial de documentos, normalmente conhecidos como "páginas". Cada página pode conter referências para outras páginas relacionadas, localizadas em qualquer lugar do mundo. Os usuários podem ir "navegando" através das páginas, passando por centenas de referências. As páginas que referenciam outras páginas são chamadas de hipertextos [PIC 95, LIU 94].

As páginas são visualizadas com um programa chamado *browser*. Cabe a este buscar a página solicitada, interpretar o texto e os comando formatados que ele possa conter, e apresentar a página, em um formato apropriado, na tela.

2.1.3.7.2. VISÃO DO SERVIDOR

O processo servidor localizado em um *site* WWW monitora a porta 80 do protocolo TCP em busca de solicitações de conexão dos clientes [HAN 95]. Após o estabelecimento da conexão, o cliente envia um pedido e após o servidor tê-la respondido, a conexão é desfeita. A necessidade de se estabelecer uma conexão TCP a cada elemento gráfico que deve ser trazido não é eficiente, mas torna a implementação simples, fato que tem impulsionado novos estudos visando a melhora da eficiência. A Figura 2.10 [TAN 96] apresenta como as diversas partes do modelo WWW se comportam.



1 - Hyperlink para abc.com
2 - Hyperlink para dex.edu

Figura 2.10 - Partes do Modelo WWW

Quando o usuário solicita uma página através de seu *browser*, os seguintes passos acontecem:

1. *Browser* determina o *Uniform Resource Location*, URL, a localização daquela página. Uma URL apresenta o seguinte formato: http://endereço da maquina/localização do arquivo;
2. *Browser* pergunta ao DNS o endereço IP do endereço especificado no URL;
3. DNS retorna o endereço IP;
4. *Browser* faz uma conexão TCP na porta 80 no endereço IP que ele recebeu;
5. *Browser* envia um comando do tipo GET arquivo;
6. Servidor responde ao *browser* enviando o arquivo solicitado;
7. A conexão TCP é desfeita;
8. *Browser* apresenta todo o texto contido na página solicitada;
9. *Browser* busca e apresenta todas as imagens da página solicita.

2.2. GERÊNCIA DE REDES

Dentre várias definições para gerenciamento de redes, destaca-se a da ISO, *International Organization for Standardization* [ISO 10040]:

"O gerenciamento de redes provê mecanismos para a monitoração, controle e coordenação de recursos em um ambiente OSI e define padrões de protocolo OSI para troca de informações entre estes recursos".

Uma outra definição é citada em [BRISA 93]:

"O gerenciamento de redes é o conjunto de funções que visa promover a produtividade da planta, dos recursos disponíveis e integrar, de forma organizada, as funções de geração, administração e manutenção de todos os elementos da rede e dos serviços de telecomunicações".

Com base neste contexto, dois são os principais padrões que procuram solucionar o problema de interoperabilidade entre sistemas heterogêneos: os padrões propostos pelo modelo de gerenciamento OSI e o proposto pela *Internet*.

Enquanto o padrão OSI é um padrão de direito que é um grande marco na terminologia de redes [ROS 91], o padrão *Internet* tornou-se um padrão *de facto* devido a simplicidade por ele apresentada.

2.2.1. Modelo de Gerenciamento OSI

O modelo de referência OSI foi proposto pela ISO/IEC, *International Organization for Standardization/International Electrotechnical Committee*, em 1979.

2.2.1.1. Estrutura do Modelo OSI

A estrutura do modelo de gerenciamento OSI subdivide-se em:

- Gerenciamento de Sistemas: exige funções de apoio em todas os sete níveis especificados no modelo.
- Gerenciamento de Camada: restringe-se à gerência de recursos relacionados com as atividades de comunicação de um único nível.
- Operação de Camada: forma de gerenciamento restrita a uma única instância de comunicação em um nível.

2.2.1.1.1. Componentes de Gerenciamento

O ambiente de gerenciamento definido pelo modelo OSI inclui os conceitos de gerente, agente e objeto gerenciado.

Cabe ao gerente transmitir operações de gerenciamento aos agentes, a fim de obter informações atualizadas sobre os objetos gerenciados e controlá-los.

O agente tem como função receber as operações de gerenciamento emitidas pelo gerente e executar as ações necessárias sobre os objetos gerenciados, podendo ainda transmitir ao gerente notificações geradas pelos objetos gerenciados ou sobre a ocorrência de eventos.

Um objeto gerenciado representa um recurso sujeito ao gerenciamento. É definido em termos de seus atributos, operações a que pode ser submetido, notificações que pode emitir e de seus relacionamentos com outros objetos gerenciados.

O conjunto de objetos gerenciados, juntamente com seus atributos, operações e notificações, constitui a MIB, *Management Information Base.*

2.2.1.1.2. Áreas Funcionais

Os requisitos que devem ser satisfeitos pelas atividades de gerenciamento de redes são classificados em cinco áreas funcionais:

- Gerenciamento de desempenho possibilita a avaliação do comportamento de recursos no ambiente OSI, bem como o cálculo da eficiência das atividades de comunicação.
- Gerenciamento de falhas engloba a detecção, isolamento e correção de operações anormais do ambiente OSI.

- Gerenciamento de segurança dá apoio à aplicação de políticas de segurança através de um conjunto de funções para criar, controlar e eliminar mecanismos de segurança, entre outras.
- Gerenciamento de contabilização inclui funções para informar aos usuários os custos ou recursos consumidos e permitir a associação do uso de recursos com escalas de tarifação.
- Gerenciamento de configuração é responsável pelo controle das condições do ambiente de comunicação, identificando mudanças significativas e modelando a configuração dos recursos físicos e lógicos da rede.

Este modelo é utilizado atualmente e serve para a classificação da gerência de redes em geral.

2.2.2. MODELO DE GERENCIAMENTO INTERNET

Será visto a seguir, a estrutura do modelo de gerenciamento *Internet*.

2.2.2.1. ESTRUTURA

A estrutura definida pelo modelo de gerenciamento *Internet* é baseada em quatro componentes:

- agentes de gerenciamento em elementos de rede: são encontrados em elementos de rede, ou seja, nos dispositivos que formam a rede, como, por exemplo, estações de trabalho e servidores de arquivos, entre outros.

  Elementos de rede gerenciáveis têm um agente de gerenciamento responsável pelo recebimento das mensagens de gerenciamento de rede e de seus respectivos tratamentos.

  Seu funcionamento pode ser passivo, apenas respondendo quando perguntado ou ativo, quando do envio de uma mensagem, *trap*, que tem como objetivo informar a ocorrência de um evento. A comunicação entre o agente e a estação de gerenciamento é realizada através dos protocolos UDP/IP.

  Quando o sistema de gerenciamento não provê os protocolos necessários para que tal comunicação ocorra de forma direta, utiliza-se um agente *proxy*. Tal agente possui os recursos necessários e a autorização para responder a comandos e perguntas indiretamente em nome de um objeto gerenciado. Para que isto seja possível, o agente *proxy* comunica-se com o sistema gerenciado através de um protocolo privado estabelecido de comum acordo entre ambos;
- gerente de rede: no ambiente *Internet*, uma ou mais estações de gerenciamento executam aplicações que monitoram e controlam elementos de rede. Estas aplicações são responsáveis pela formulação e transmissão das requisições apropriadas, ficando então, aguardando as respostas enviadas pelos agentes para os seus devidos tratamentos. Em geral, uma estação de gerenciamento é responsável por vários elementos de rede;
- protocolo de comunicação: implementado tanto em agentes quanto nos gerentes de rede. Maiores detalhes na Seção 3.2.1.1.
- informação de gerenciamento: compartilhada entre gerentes e agentes através do protocolo de gerenciamento sendo que os gerentes podem ler ou escrever nas variáveis de gerenciamento e

os agentes, encarregam-se de atualizá-las. Várias formas são admitidas por estas informações, tais como inteiras, seqüências de octetos, entre outras.

Sua estrutura é definida na SMI, *Structure of Management Information*, que define um modelo de objeto de informação e especifica regras para descrever os objetos existentes na MIB, bem como regras para descrever novos objetos para a MIB. A MIB guarda as informações transferidas ou modificadas pelo uso dos protocolos de gerenciamento.

2.2.2.1.1. SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Segundo [STA 93], o *Simple Network Management Protocol*, SNMP, foi desenvolvido visando prover uma ferramenta de gerenciamento de redes fundamental e de simples implementação para ambientes TCP/IP. Por ser um protocolo independente da camada de transporte, pode ser implementado tanto sobre o protocolo UDP em redes do tipo Ethernet quanto sobre outros protocolos de transporte, tais como o *Open System Interconnection - Connectionless Mode Transport Service*, OSI CLTS.

No SNMP são suportados dois tipos de transações:

- *polling*: mecanismo síncrono baseado no envio de requisições pelos gerentes que permanecem aguardando por respostas dos agentes;

- *trap*: mecanismo assíncrono baseado no envio de mensagens pelos agentes ao gerentes de forma unilateral. Normalmente utilizadas para notificar os gerentes da ocorrência de algum evento previamente determinado.

Quatro são as primitivas de serviço deste protocolo: *get-request*, *get-next-request*, *set-request* e *trap*. A Figura 2.11 [STA 93] ilustra o ambiente típico em que estas primitivas são utilizadas.

Em [STA 93] é apresentada uma lista de limitações para o protocolo SNMP:

- Não é apropriado para a recuperação de um grande volume de dados,
- O serviço de *trap* não é confirmado. Em um caso típico em que a arquitetura UDP/IP é utilizada para a entrega de mensagens deste tipo, o agente não tem como confirmar que a sua mensagem *trap* chegou a estação gerente.
- O protocolo básico padrão não apresenta nenhuma forma de autenticação, sendo mais indicado para a monitoração do que o controle de um sistema.
- não oferece suporte direto para comandos imperativos. A única forma de disparar um evento é através da setagem do valor de um objeto.
- O modelo da MIB é limitado e não suporta aplicações que executam consultas sofisticadas de gerenciamento baseadas em valores ou tipos de objetos.
- Não suporta a comunicação gerente-gerente.

Muitas destas limitações já foram tratadas por outras especificações, tais como a comunicação gerente-gerente e a ausência de mecanismos de segurança, ambos presentes na especificação do protocolo *Simple Network Management Protocol version* 2 (SNMPv2) abordado a seguir.

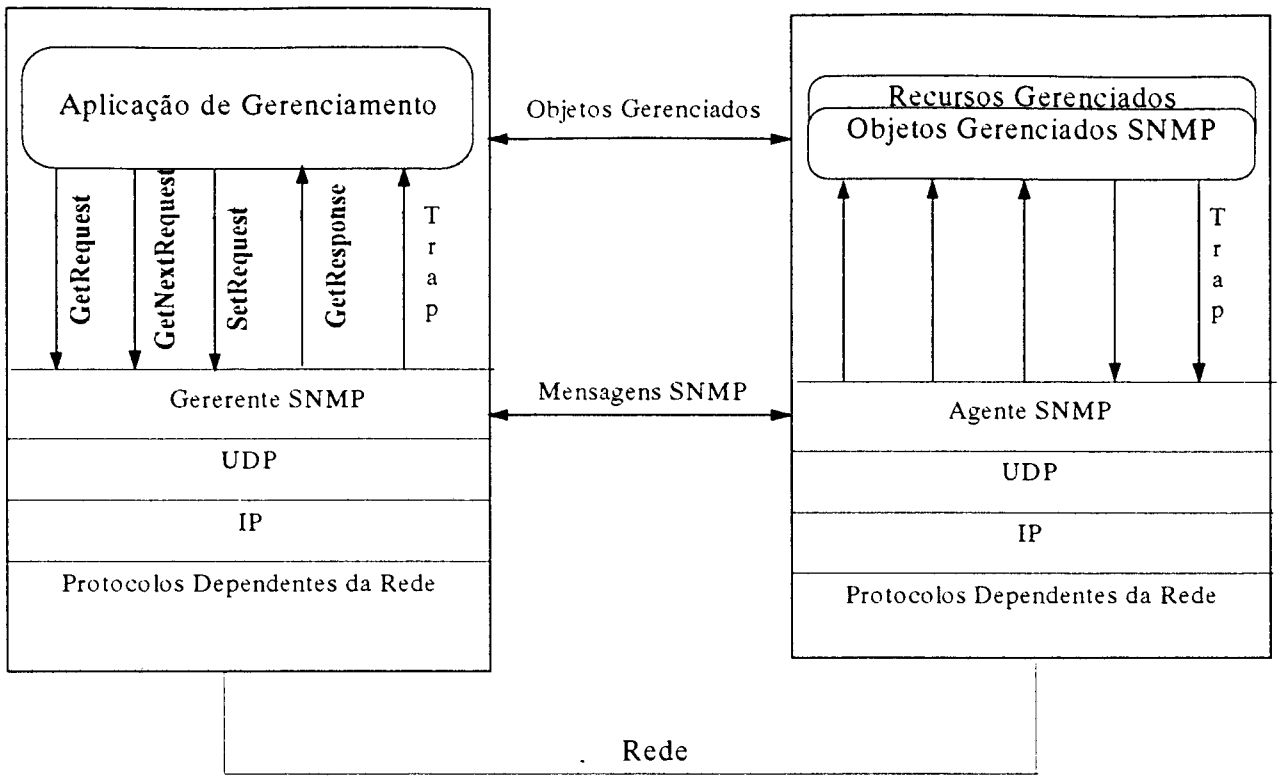


Figura 2.11 - Cenário Típico SNMP

2.2.2.1.2. SIMPLE NETWORK MANAGEMENT PROTOCOL VERSION 2 (SNMPV2)

Esta versão do protocolo SNMP pode suportar uma estratégia de gerenciamento de redes altamente centralizada ou de forma distribuída e, neste caso, alguns sistemas irão operar como uma entidade gerente ou agente. No papel de agente, este sistema aceitará comandos de um sistema de gerenciamento superior, podendo estes comandos tanto acessar informações armazenadas localmente em um gerente intermediário ou podem solicitar ao gerente intermediário que providencie informações resumidas sobre os agentes a ele subordinados [STA 93]. Além disso, um agente intermediário pode enviar mensagens do tipo *trap* ao gerente superior.

Os principais melhoramentos oferecidos por esta versão são [CAR 94, STA 93]:

1) Estrutura da informação de gerenciamento, SMI: entre outros melhoramentos, destaca-se que a macro utilizada para definir os tipos de objeto foi expandida para incluir vários novos tipos de dados e aumentar a documentação associada a um objeto.
2) Operações de protocolo: foram incluídas duas novas operações:
   - **GetBulkRequest**: habilita o gerente a recuperar eficientemente grandes blocos de dados.
   - **InformRequest**: habilita a um gerente enviar mensagens do tipo *trap* para outro gerente.
3) As MIBs especificadas nesta versão contêm informações básicas do tráfego sobre as operações de protocolo além de outras informações relacionadas a configuração de um gerente ou agente deste protocolo.

4) Comunicação gerente-gerente: foi especificada uma MIB especialmente voltada para suportar as arquiteturas de gerenciamento distribuídas.
5) Segurança: esta versão apresenta capacidades de segurança baseadas no protocolo SNMP seguro.

CAPÍTULO 3

GERENCIAMENTO PRÓ-ATIVO

Atualmente, na grande maioria dos casos, o gerenciamento de redes tem sido feito de uma forma **reativa**, ou seja, é necessário que ocorra algum problema que leve a degradação dos serviços para que não só o gerente da rede passe a se preocupar com a performance da rede, mas também a grande maioria dos sistemas de gerenciamento comece a emitir alarmes e eventos [WAT 96].

Logo, a gerência pró-ativa é aquela que tem por objetivo a identificação de situações que caracterizam estados que possam vir a degradar os serviços oferecidos [JAN 93, ROC 96, STE 95a, WAT 96].

Quanto mais as empresas se conscientizam da importância do perfeito funcionamento de sua rede de computadores, mais elas pressionam os fabricantes a oferecer produtos que ofereçam facilidades pró-ativas [DAU 91], resultando em uma crescente oferta de produtos que oferecem facilidades que viabilizem o gerenciamento pró-ativo. Grande parte destes produtos oferecem implementações da RMON MIB, utilizada para a monitoração remota de redes.

A identificação dos estados críticos é obtida através da comparação das informações sobre a atual situação do funcionamento corrente do sistema com o seu perfil de funcionamento, também conhecido como *baseline*. Após esta etapa, deve-se procurar identificar a razão de tal estado e as respectivas soluções para tal.

Este passo, o da identificação da razão do estado crítico e as respectivas soluções para tal, é bastante prejudicado pelo fato da RMON MIB coletar informações apenas até o nível de rede impossibilitando assim que se considerem soluções de níveis mais elevados.

Duas outras áreas podem ser aplicadas juntamente com a gerência pró-ativa: inteligência artificial e simulação. A inteligência artificial é aplicada através de sistemas especialistas que tem por objetivo prever situações que podem levar à degradação do sistema [JAN 93]. A simulação matemática é utilizada para efetuar a análise do comportamento da rede submetendo-a a variadas situações. A análise do comportamento da rede utilizando a teoria de séries temporais enquadra-se no campo da simulação matemática.

3.1. *BASELINE* OU PERFIL DA REDE

A *baseline*, ou perfil da rede, é conseguida através da realização de uma atividade de amostragem durante intervalos de tempo, identificando o desempenho normal através de médias e cálculos estatísticos.

Procura-se produzir uma caracterização estatística válida do comportamento normal da rede sobre um período extenso ao invés de um intervalo específico, levando em conta a variação de níveis de tráfego em diferentes períodos do dia e em diferentes dias da semana [JAN 93] [DAU 91].

Alguns especialistas sugerem que as *baselines* sejam atualizadas periodicamente para que estas sempre reflitam a situação atual da rede.

Uma das formas possíveis de se conseguir algumas das características de uma *baseline* é através da utilização de um agente remoto que seria responsável pela manutenção de uma RMON MIB.

3.2. MONITORAÇÃO REMOTA DE REDES

A descentralização das organizações, o crescimento das redes locais e o aumento da complexidade resultante da heterogeneidade dos ambientes têm impostos um desafio aos administradores de redes: a responsabilidade de manter várias redes locais em vários locais remotos funcionando [BUR 93]. Uma grande parte deste problema é originado pelo fato de que a maior parte destes pequenos lugares não comportam uma equipe de suporte técnico para solucionar os problemas.

Segundo [PHA 94], a tarefa principal da monitoração é prover informações detalhadas sobre a rede de forma contínua, pois é através destas informações que se pode exercer o gerenciamento da rede através da tomada de decisões e a execução de ações de controle sobre a rede.

Ainda segundo [PHA 94], dois são os estilos de monitoração:

- **monitoração apenas quando necessário**: colocada apenas em locais-chave onde sabe-se previamente que existem problemas. Serão colocados dispositivos apropriadamente configurados para auxiliar no diagnóstico do problema, caracterizando um gerenciamento reativo.

- **monitoração contínua**: uso típico da tecnologia RMON, procura-se monitorar o maior número possível de segmentos, através do baixo custo por segmento. Cada um desses investigadores monitoraria a transmissão de dados no seu segmento e organiza as informações coletadas em um formato acessível a um administrador em uma central para analisar os padrões de tráfego e diagnosticar problemas inclusive em locais remotos [SAL 95]. A grande cobertura de informações proporcionada, possibilita a construção de aplicações de gerenciamento baseadas neste tipo de monitoração, que permite a determinação do comportamento da rede antes, durante e após a ocorrência de um problema [HUG 94]. Podem detectar vários tipos de

problemas antes que eles venham a prejudicar o funcionamento do sistema, caracterizando o gerenciamento pró-ativo.

Neste trabalho utilizou-se a monitoração contínua, visando a criação de uma *baseline* e o aprendizado sobre o comportamento dos objetos gerenciados.

3.3.RMON 1 MIB

A RMON 1 MIB [WAL 91, WAL 95a], *Remote Network Monitoring Management Information Base*, define a próxima geração da monitoração de redes através de um conjunto de informações detalhadas sobre o tráfego de um segmento de rede que está sendo monitorado.

Entre as novas características acrescentadas pela RMON aos monitores remotos, destacam-se [DNP 94]:

- Contadores adicionais de erros de pacotes;
- Análises estatísticas;
- Matriz de tráfego em nível Ethernet;
- Alarmes mais compreensivos;
- Maior capacidade na filtragem e captura de pacotes

Destaca-se também o fato de que esta MIB foi definida para ser executada em um agente, normalmente chamado de monitor ou investigador (*probe*), que não necessita estar em contato ininterrupto com a aplicação de gerenciamento, diminuindo assim não apenas a dependência mas também o tráfego de informações entre ambos [STE 95a]. Freqüentemente, estes monitores são dispositivos do tipo *stand-alone* e destinam uma grande parte dos seus recursos computacionais para a tarefa de gerenciamento de rede [WAL 91, WAL 95a]. Vários são os dispositivos onde estes agentes podem ser instalados:

- interconectores de redes tais como pontes, roteadores ou *hubs*;
- *hosts* dedicados ou não dedicados;
- plataformas especialmente desenvolvidas para o gerenciamento de redes.

A Figura 3.1 [STE 95b] ilustra um ambiente com vários monitores RMON.

Para um efetivo gerenciamento de redes, estes monitores devem se comunicar com uma estação de gerenciamento central e, neste contexto, são referenciados como monitores remotos [STA 94].

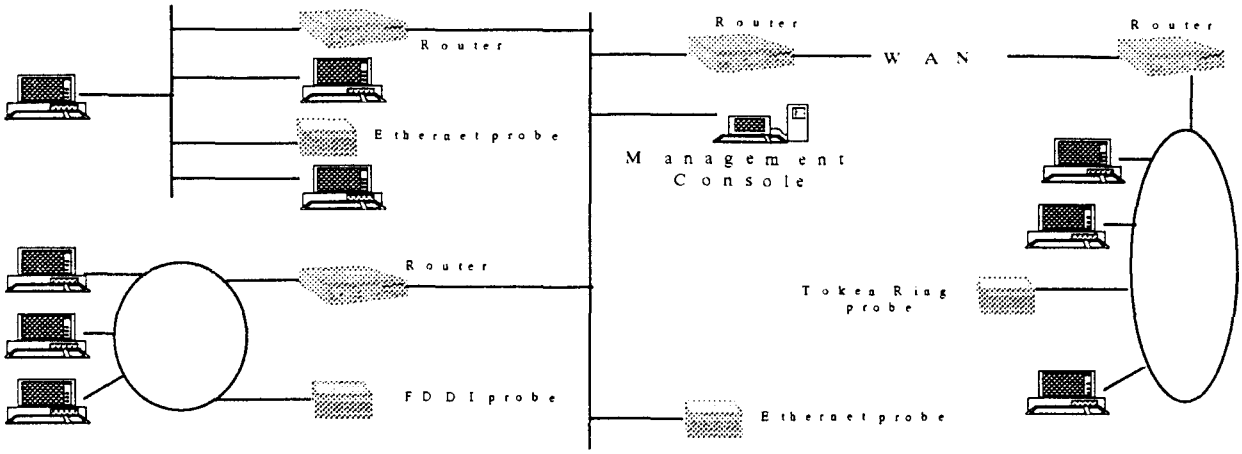


Figura 3.1 - Topologia RMON

3.3.1. OBJETIVOS DA REMOTE MONITORING 1 MIB - RMON 1 MIB

O IETF, *Internet Enginneering Task Force*, lançou em novembro de 1991 a RFC 1271, intitulada *Remote Network Monitoring Management Information Base*. Em setembro de 1993, foi lançada a RFC 1513, uma extensão a RFC 1271 para permitir a monitoração remota de redes do tipo *Token Ring*. A RFC 1757, de fevereiro de 1995 veio a substituir a RFC 1271.

A gerência de redes remotas através de agentes remotos possui cinco funções [WAL 91, WAL 95a], a saber:

- Operações *Off-line*: quando deseja-se ou necessita-se limitar ou interromper a rotina de *polling* entre a estação gerente e o monitor, diminuindo os custos de comunicação ou em caso de falha na linha de comunicação ou na estação de gerenciamento. Mesmo que tais situações aconteçam, o monitor deve continuar a coletar informações sobre falhas, desempenho e configuração continuamente. O monitor também pode tentar notificar a estação de gerenciamento caso um evento pré-definido ocorra.

- Monitoração Preemptiva: se o monitor tiver recursos suficientes e se não degradar o funcionamento da rede, a monitoração da rede pode ser feita continuamente. Quando da ocorrência de falha na rede, o monitor pode reportar à estação de gerenciamento e prover informações significativas para o diagnóstico da falha.

- Detecção e registro de problemas: o monitor pode ser configurado para reconhecer passivamente determinadas situações como, por exemplo, erros, congestionamentos. Quando uma destas condições for atingida, ele pode registrá-la e tentar reportar à estação de gerenciamento.

- Valorização dos dados coletados: o monitor pode executar análises específicas sobre as informações coletadas em sua subrede, poupando a estação de gerenciamento de mais esta responsabilidade. Tal nível de informação sobre uma determinada subrede não estaria disponível de outra forma a uma estação de gerenciamento que não estivesse conectada a este segmento.

- Múltiplos Gerentes: o monitor pode ser configurado para trabalhar com mais de uma estação de gerenciamento concorrentemente

3.3.1.1. CONTROLE DOS DISPOSITIVOS DE MONITORAÇÃO REMOTA

Devido à complexa natureza das funções disponíveis nos dispositivos de monitoração remota, estas normalmente precisam ser configuradas. Em muitos casos, as funções requerem a configuração de parâmetros para a operação de coleta de dados pois esta operação só pode acontecer depois que esta configuração tenha sido feita [WAL 91, WAL 95a].

Muitos grupos que compõem a RMON MIB possuem uma ou mais tabelas nas quais estão os parâmetros de controle e uma ou mais tabelas de dados onde são colocados os resultados das operações. As tabelas de controle normalmente são do tipo que permitem leitura e escrita, enquanto as de dados, normalmente são do tipo de leitura. Em virtude dos parâmetros na tabela de controle algumas vezes descreverem informações resultantes da tabela de dados, como o número de octetos capturados em um determinado *buffer*, muitos parâmetros das tabelas de controle podem ser modificados apenas quando a entrada de controle estiver inválida. Desta forma, o método para modificar estes parâmetros é o de invalidar a entrada de controle, forçando a sua destruição e a de qualquer entrada de dados a ela associada, e só então criar uma nova entrada de controle com os parâmetros apropriados. Quatro são os valores possíveis a estas variáveis: **valid(1), createrequest(2), undercreation(3), invalid(4)**. A Figura 3.2 [KOO 94] apresenta um diagrama de estados deste tipo de variável.

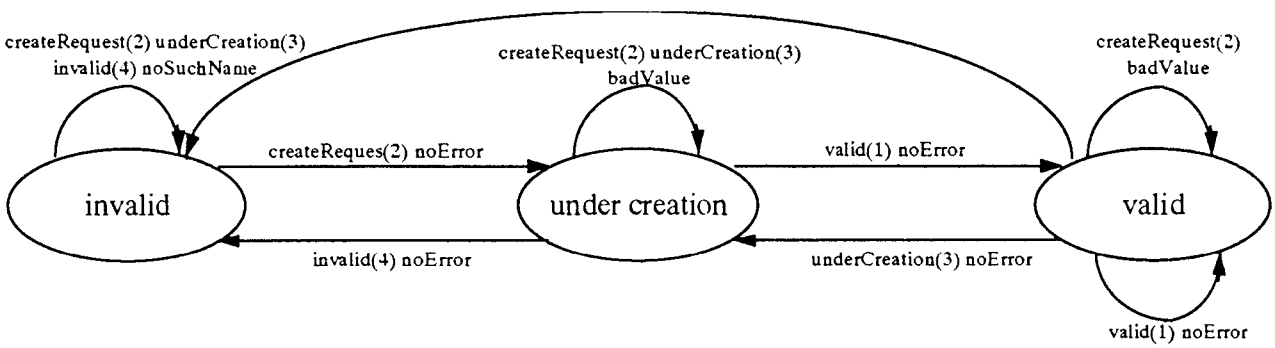


Figura 3.2 - Diagrama de Estados de uma Variável de Controle

Alguns objetos especificados provêem um mecanismo para executar uma ação no dispositivo de monitoração remota como resultado na mudança no estado de um objeto.

3.3.1.2. GRUPOS DA RMON 1 MIB

A especificação RMON consiste de duas RFCs: RFC 1757 [WAL 95a] que contém nove grupos Ethernet e Ethernet/*Token Ring*; e a RFC 1513 [WAL 95b] que define dez extensões específicas para a RMON de *Token Ring*.

Statistics Group

Este grupo contém as estatísticas básicas para cada subrede monitorada. Consiste de uma única tabela com uma entrada (linha) para cada interface (subrede) monitorada. Estas estatísticas

apresentam o número de pacotes, octetos, *broadcasts*, *multicasts*, colisões e o número de vezes que o monitor descartou pacotes devido a falta de recursos. Além destas estatísticas, são monitorados também os erros para cinco diferentes tipos de pacotes. Eles são:

- *Undersizes*: pacotes Ethernet de tamanho menor que 64 *bytes*, os quais são usualmente causados por colisão no segmento. Um pacote Ethernet normal é de 1518 *bytes*. Um número excessivo deste tipo de erro pode indicar que alguma estação transmissora do segmento pode não estar configurada apropriadamente.
- *Fragments*: um pacote cujo tamanho total seja menor que 64 *bytes*, excluindo os *bits* de *frame*, e não é um número integral de octetos de comprimento ou que possui um *frame check sequence*, FCS, errado.
- *CRC/Alignment Errors*: *Cyclic Redundancy Check* (CRC) é composto pelos últimos 32 *bits* de informação contidos em um pacote/*frame* e é utilizado para a detecção de erros de transmissão.
- *Collisions*: refere-se a dados enviados por um dispositivo em uma rede sem que este preocupe-se com qualquer outro dispositivo que poderia também estar tentando transmitir. Uma colisão ocorre quando dois ou mais dispositivos tentam transmitir ao mesmo tempo, situação esta que causa a colisão de sinais e os dados tornam-se inutilizáveis.
- *Oversizes*: pacotes que excedem os 1518 *bytes* de comprimento máximo, excluindo os *bits* de *frame* mas incluindo os de FCS e que possuem um FCS correto. Este tipo de pacote pode ser causado por um nó que não está configurado corretamente.

History Group

Este grupo provê o histórico das estatísticas providas pelo grupo Statistics. Este grupo permite ao usuário definir os intervalos de amostragem e o número de amostras que ele deseja armazenar. É composto de duas tabelas:

- historyControlTable: especifica a interface e os detalhes sobre a função de amostragem;
- etherHistoryTable: tabela onde os dados são armazenados.

Alarm Group

Disponibiliza um mecanismo para a definição de *thresholds*, tanto de subida quanto o de descida, e intervalos de amostragem para a geração de eventos sobre qualquer contador ou inteiro mantido pelo monitor. Os *thresholds* podem ser estabelecidos baseados em dois tipos de valores: o valor absoluto que é o valor de um objeto no momento da amostragem ou o valor delta, que é a diferença entre dois valores observados em duas amostragens sucessivas.

É utilizado um mecanismo chamado *hysteresis mechanism* que evita a geração desnecessária de alarmes causada por pequenas flutuações. Pode-se pensar no mecanismo de geração de alarmes como tendo dois estados. Quando está no estado de alarme de subida, o mecanismo irá gerar um alarme de subida quando o valor da variável observada atingir ou ultrapassar o *threshold* de subida; enquanto estiver neste estado, o mecanismo está desabilitado para a geração de um alarme de descida. Uma vez que o alarme de subida foi gerado, o mecanismo passa para o estado de alarme de descida e permanecerá desta forma até que a variável monitorada atinja ou ultrapasse o *threshold* de descida. De forma similar, enquanto estiver em estado de alarme de descida, o mecanismo gerará um alarme de descida quando o valor da variável observada atingir ou ultrapassar o *threshold* de alarme de descida; enquanto permanecer neste estado, o mecanismo fica desabilitado de gerar um alarme de subida.

Host Group

Este grupo é composto por três tabelas:

- Host Control Table: tabela de controle do grupo;
- Host Table: contém estatísticas sobre o tráfego de cada *host* do segmento monitorado tais como: pacotes enviados e recebidos, octetos enviados e recebidos, *broadcasts*, *multicasts* e o número de pacotes com erro enviados, que compreendem os pacotes com erros do tipo *undersize*, *fragment*, erros de CRC/*Alignment*, *collisions* e *oversizes*.
- Host Time Table: apresenta exatamente as mesmas informações da tabela anterior indexadas pela ordem de criação. A descoberta de novos *hosts* em um segmento é feita através da observação dos endereços de origem e de destino da camada *Medium-Access Control*, MAC, de pacotes corretos.

Host Top N Group

Extensão da *host* table, oferece as estatísticas dos *hosts* ordenadas por um parâmetro definido pelo usuário. O conjunto de estatísticas para um objeto grupo-*host* de uma interface coletado durante um intervalo de amostragem, é referido como sendo um relatório. Cada relatório contém os resultados para apenas uma variável e esta representa a quantidade de alteração de um objeto grupo-*host* no intervalo de amostragem. Assim, o relatório lista os *hosts* em um determinado segmento com a maior média de alteração de uma variável em particular. É composto por duas tabelas:

- Host Top N Control Table: tabela de controle onde define-se a variável a ser monitorada, o intervalo de amostragem e o número máximo de *hosts* que podem constar em um relatório.
- Host Top N Table: tabela onde os relatórios gerados ficam armazenados e disponíveis à estação gerente.

Matrix Group

Este grupo é utilizado para armazenar informações sobre o tráfego entre pares de *hosts* de um segmento. É composto por três tabelas:

- Matrix Control Table: tabela de controle onde é definida a interface a ser monitorada e o tamanho das duas tabelas apresentadas a seguir.
- Matrix SD Table: utilizada para armazenar estatísticas sobre o tráfego de um *host* fonte em particular para um número de destinos.
- Matrix DS Table: utilizada para armazenar estatística sobre o tráfego de um certo número de destinos para uma determinada origem.

Filter Group

Este grupo provê um mecanismo de filtragem genérico que implementa todas as funções de captura de pacotes e a geração de eventos. O seu *buffer* de captura de pacotes recebe apenas os pacotes que combinarem com o critério de filtragem especificado pelo usuário.

Os usuários podem capturar pacotes que são válidos ou não. São mantidos também contadores para cada filtro proporcionando análises estatísticas dos mesmos. Este grupo é composto por duas tabelas de controle:

- Filter Table: tabela onde são definidos os filtros a serem aplicados nos pacotes;
- Channel Table: o grupo de pacotes que passaram pelo teste do filtro é referenciado como sendo um canal. Este canal pode ser configurado para gerar um evento, definido no **Event Group**, quando um pacote passa pelo canal e este encontra-se em um estado habilitado. Além disso, os pacotes que passam pelo canal podem ser capturados se tal mecanismo for ativado em **Capture Group**.

Packet-Capture Group

Pode ser utilizado para o armazenamento dos pacotes de um dos canais definidos no **Filter Group**. Este grupo é composto por duas tabelas:

- Buffer Control Table: permite a associação de um determinado canal a um *buffer*, define qual a ação deve ser tomada quando o *buffer* estiver cheio e informa o número de pacotes que foram capturados no *buffer*, entre outras informações.
- Capture Buffer Table: tabela onde são armazenados os pacotes coletados.

Event Group

Este grupo permite a definição de eventos. Um evento é disparado por uma condição em qualquer lugar da MIB, e um evento pode disparar uma ação definida em qualquer lugar da MIB, efetuar um log da informação ou gerar uma mensagem SNMP de *trap*, que pode ser um destes:

1. *link up*;
2. *link down*;
3. *warm start*;
4. *cold start*;
5. *authentication failure*;
6. EGP *neighboor loss*;
7. *Enterprise Specific*;
8. *Rising Threshold*;
9. *Falling Threshold*;
10. *Packet Match*;

Sendo que os três últimos são específicos da RMON MIB.

Este grupo é composto das seguintes tabelas:

- Event Table: tabela de controle onde é definido o evento;
- Log Table: tabela que armazenará as informações de log geradas.

3.3.2. Implementações Privadas

Vários fabricantes privados oferecem produtos com as facilidades da RMON 1. Dentre estes, destacam-se Frontier Software, Hewlett-Packard, 3Com e Bay Networks [MUL 95].

3.3.2.1. Frontier Software RMON Support

Esta empresa desenvolveu o programa Netscout Manager 3.3 [FRONTIER] que oferece suporte RMON para a monitoração do tráfego de aplicações. Estas aplicações incluem Note,

Mosaic, ccMail, NFS entre outras. Este programa também permite a monitoração da utilização de LAN/WAN por aplicação e por departamento ou usuário.

Utilizando-se do método *point and click*, qualquer grupo RMON pode ser habilitado pelo administrador para coletar informações estatísticas sobre os aplicativos anteriormente mencionados.

Este produto também monitora recursos críticos de rede e envia alertas quando *thresholds* definidos pelo usuário são alcançados. Ele é capaz também de tomar ações corretivas baseados nos *thresholds* através da definição de comandos ou *scripts* específicos para alguns dispositivos tais como roteadores, *bridges*, servidores. As ações corretivas incluem o envio de mensagens eletrônicas para o administrador e a geração de relatórios, entre outras.

3.3.2.2. Implementação da RMON feita pela Hewlett-Packard

A implementação feita pela HP é composta de dois elementos: *Power Agents* e um conjunto de aplicações chamadas *NetMetrix*.

3.3.2.2.1. Power Agents

Estes programas são os monitores que rodam em estações de trabalho Unix. São responsáveis pela coleta de dados, processamento dos pacotes e disponibilização de informações sobre o tráfego da rede. Um rico conjunto de extensões privadas são implementadas por estes agentes, das quais, destacam-se:

- Informações sobre os sete níveis da pilha de protocolo de rede.
- Correlação entre tempo, protocolo, fonte e destino.

Tais programas podem ser executados tanto em redes do tipo LAN quanto em WAN.

3.3.2.2.2. HP NetMetrix

Este produto é responsável pelo controle, visualização e fácil compreensão dos dados coletados através de um conjunto de aplicações e permite visualização dos diversos segmentos monitorados, análise do tráfego, gerenciamento de NFS, análise de protocolos, gerador de tráfego, setagem de alarme e criação de *baselines*.

3.3.2.3. SmartAgents RMON desenvolvidos pela 3Com

Sua implementação em nível de *hardware* chama-se *Application Specific Integrated Circuits*, ASICs e não sobrecarrega o processador, diminuindo os custos de gerenciamento. As funcionalidades básicas da RMON são implementadas pelos SmartAgents além de possibilitar a associação de execução de eventos à geração de alarmes.

3.3.2.4. Bay Networks' SuperRMON

O agente *Advanced Analyzer* integra o gerenciamento de *hub*, monitoração baseada em RMON e a análise de funcionalidades no seu ambiente de gerenciamento. Sua implementação inclui como extensões à RMON padrão variáveis específicas sobre *hubs*.

3.3.3.IMPLEMENTAÇÃO ACADÊMICA: BEHOLDER - THE NEXT GENERATION

O programa Beholder - *The Next Generation*, BTNG, é um monitor de redes do tipo *Ethernet*, desenvolvido na *Technical University of Delft, The Netherlands*, Holanda. De domínio público, pode ser conseguido através de FTP para o endereço dnpap.et.tudelft.nl .

Comunica-se através do protocolo SNMP e pode ser utilizado no sistema operacional SunOs, Solaris, OS/2 e Linux.

Este monitor implementa os nove grupos de objetos definidos na RMON 1 MIB, e é utilizado na coleta de dados deste trabalho, vide a Seção 4.3.1.

3.4.RMON 2 MIB

Esta nova MIB vem como uma complementação à RMON 1, permitindo efetuar análises acima da camada de rede, através da leitura do *header* do protocolo do nível de rede, que normalmente é o IP [STA 96], visualizando-se os protocolos de alto nível que são executados acima do protocolo desta camada, através da leitura do *header* do protocolo TCP ou UDP e até mesmo o *header* do protocolo do nível de aplicação [BIE 97]. Esta nova gama de informações permite ao gerente da rede monitorar o tráfego com um grande nível de detalhes, possibilitando a criação de um gerenciamento das aplicações que caracterizará o tráfego através de porcentagem de utilização de cada aplicação ou protocolo, sendo de grande relevância para contenção da carga e o gerenciamento de desempenho.

No contexto da RMON 2, qualquer protocolo acima da camada de rede é considerado "nível de aplicação".

3.4.1.GRUPOS COMPONENTES DA RMON 2

A seguir, serão descritos os grupos definidos na RMON 2 [STA 96][WAL 97].

Protocol Directory Group

Este grupo provê as facilidades para que um gerente RMON 2 defina quais protocolos em particular o monitor RMON 2 irá interpretar. Isto é importante especialmente quando gerente e monitor são de fabricantes diferentes. Em uma determinada rede, pode haver vários protocolos diferentes sendo utilizados. Alguns são padronizados ou, pelo menos, bem conhecidos, enquanto outros podem ser protocolos específicos para um determinado produto. Este grupo é composto por uma tabela, *Protocol Dir Table*, na qual o monitor lista uma entrada para cada protocolo que ele pode decodificar e calcular o número de *protocol data units* (PDUs). Esta tabela cobre o protocolo MAC, rede e de níveis mais altos.

Protocol Distribution Group

Este grupo contabiliza quantos pacotes e octetos foram enviados por cada protocolo suportado.

Address Map Group

Este grupo compara cada endereço de rede com um endereço da camada MAC específico e, desta forma, com um número *port* no dispositivo da rede, sendo útil na descoberta de nós e na topologia das aplicações da rede.

Network-Layer Host Group

Seu objetivo é coletar estatísticas básicas do tráfego destinado e originado de cada *host* descoberto, divididos por endereços da camada de rede.

Network-Layer Matrix

Provê estatísticas baseadas no tráfego em nível de rede entre pares de *hosts*.

Application-Layer Host Group

Monitora a quantidade de tráfego, por protocolo, destinado e originado de cada endereço de rede descoberto pelo monitor. Tais estatísticas incluem o número de pacotes e octetos destinados e originados por um endereço.

Application-Layer Matrix Group

Monitora a quantidade de tráfego, por protocolo, entre cada par de endereços de rede descobertos na rede. Este grupo é composto pelas seguintes tabelas:

- Al Matrix SD Table: lista das entradas da matriz do tráfego de aplicações entre dois endereços do nível de rede, ordenada primeiro pelo endereço fonte e, em seguida, pelo endereço destino.
- Al Matrix DS Table: lista das entradas da matriz do tráfego de aplicações entre dois endereços do nível de rede, ordenada primeiro pelo endereço destino e, em seguida, pelo endereço fonte.
- Al Matrix Top N Control Table: tabela que contém o conjunto de parâmetros utilizados na criação do relatório sobre as maiores entradas da matriz de acordo com uma métrica selecionada.
- Al Matrix Top N Table: conjunto de estatísticas para as entradas da matriz do nível de aplicação que apresentam os mais altos números de octetos ou pacotes.

User History Group

Este grupo é utilizado para definir funções de amostragem para um monitor. O monitor periodicamente coleta amostras das variáveis definidas pelo gerente e realiza um log destas. Com este grupo, o gerente tem a possibilidade de configurar estudos históricos de qualquer contador no sistema. Tais facilidades propiciam a aplicação de gerenciamento de redes analisar tendências de um determinado parâmetro. Este grupo é composto das seguintes tabelas:

- User History Control Table: tabela onde são configurados o grupo de objetos da MIB a serem amostrados periodicamente, número de coletas a serem armazenadas, intervalo de amostragem, etc.

- User History Object Table: contém uma lista das instâncias da MIB a serem amostradas periodicamente.
- User History Table: tabela onde são armazenadas as informações coletadas.

Probe Configuration Group

Este grupo permite uma maior interoperabilidade entre os monitores RMON e os gerentes através da definição de um conjunto padrão de parâmetros de configuração para os monitores, facilitando a configuração de um monitor RMON de um vendedor por outro monitor RMON de um fabricante diferente.

Timing Filtering

Uma função comum da aplicação de gerenciamento de redes é a de questionar periodicamente todos os monitores subordinados para a coleta de informações. Visando o aumento da eficiência, seria desejável que os monitores reportassem apenas os objetos que sofreram alteração desde a última consulta. A viabilização desta funcionalidade na RMON MIB 2 dá-se da seguinte forma: cada linha de uma tabela da RMON 2 pode ser equipada com um objeto *timestamp* que registra a última vez que tal objeto sofreu alteração. Quando a estação de gerenciamento solicita informações sobre uma tabela, ele inclui no pedido um filtro de tempo para cada linha solicitada. Os únicos valores retornados pelo monitor serão sobre aquelas linhas que sofreram atualização após o valor especificado no filtro de tempo da solicitação.

3.5. Informações Utilizadas no Gerenciamento de Redes de Computadores

Por ser uma área onde as pesquisas vem ocorrendo a mais tempo, várias são as informações utilizadas pelos gerentes de redes e pelos programas de gerenciamento para se monitorar o comportamento de uma rede de computadores. Dentre elas, destacamos:

- *Throughput*;
- Número de pacotes;
- Número de octetos;
- Utilização do meio;
- Quantidade de pacotes de diversos tamanhos.

Na monitoração dos *hosts* que compõem as redes, as seguintes métricas são as mais utilizadas [FER 83] :

- *Thoughput* do *host*;
- Tempo de resposta;
- Capacidade;
- Disponibilidade;
- Utilização da CPU;
- Média de paginação;
- Tempo de reação.

Tais informações, apesar de úteis são de um nível bastante básico. Se os administradores de redes tivessem a sua disposição sobre nível mais altos de protocolos, como por exemplo, a nível de aplicação, suas decisões poderiam dar-se a níveis bem mais altos.

3.5.1. GERENCIAMENTO DE SISTEMAS

Recentemente, os esforços eram voltados para a tarefa de gerenciamento de redes, porém, o gerenciamento dos sistemas conectados a estas redes tornou-se cada vez mais importante. A experiência tem mostrado que, sem o perfeito funcionamento dos sistemas-chave , a rede torna-se indisponível para os usuários, ou aplicações. A necessidade de um método consistente que possibilite o gerenciamento dos recursos do sistema e das aplicações críticas torna-se cada vez mais necessário [KRU 95a, KRU 96]. Vários tem sido os esforços neste sentido, destacando-se [KRU 95a, KRU 95b].

A definição de gerenciamento de sistemas aqui adotada, enfoca o gerenciamento e a operação dos sistemas nos quais os seus componentes estejam interligados por redes. Incluem-se nesta categoria os sistemas operacionais inferiores, os equipamentos e dispositivos nos quais o sistema opera, suas aplicações, sistemas de arquivo e seus processos [KRU 95a].

Através da integração do gerenciamento de redes e de sistemas pode-se disponibilizar uma interface padrão para o gerenciamento de sistemas, acima de detalhes de sistemas operacionais, além de permitir a utilização de programas de gerenciamento de redes na execução de funções de gerenciamento de sistemas.

O gerenciamento das camadas mais elevadas é importante pois o desempenho dos sistemas pode apresentar degradação mesmo que sejam corrigidos os problemas detectados nas camadas inferiores, pois o desempenho destes sistemas pode ser influenciado também por problemas a nível de sistema operacional ou por erros na própria aplicação.

3.5.1.1. INFORMAÇÕES RELEVANTES PARA O GERENCIAMENTO DE SISTEMAS

A comunidade Internet tem feito grandes progressos na especificação de uma interface comum para a realização do gerenciamento de sistemas através da definição de várias MIB, dentre eles destacam-se [CAR 94, GRI 93, KIL 94a, KIL 94b, SAP 96], em especial [WAL 97], que busca abranger todos os níveis de informações acima da camada de redes.

Com a disponibilização de tal nível de informação, o administrador poderá efetuar as análise das situações através de refinamentos sucessivos no nível de detalhamento em que sua abordagem ocorre, por exemplo, ele pode primeiramente obter informações sobre os níveis mais básicos sobre o segmento monitorado e só então, partir para uma análise das informações sobre as aplicações que estão sendo executadas no segmento. Caso ele deseje um grau de aprofundamento maior ainda, ele pode consultar informações específicas sobre uma aplicação em particular.

3.5.1.1.1. INFORMAÇÕES SOBRE DOMAIN NAME SERVICE (DNS)

Visando mensurar o impacto desta aplicação sobre o tráfego na rede, propõem-se as seguintes métricas:

- número de *bytes* que trafegam em um determinado segmento;
- número de pacotes que trafegam em um determinado segmento;
- número de *bytes* com erro que trafegam no segmento;
- número de pacotes com erro que trafegam em um determinado segmento;
- lista com os nomes mais acessados e quais os *hosts* que os acessaram.

3.5.1.1.2. INFORMAÇÕES SOBRE O SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

O gerenciamento do serviço de correio eletrônico foi abordado em [KIL 94b], onde são definidos os requisitos visando a monitoração dos MTAs, *Message Transfer Agents*. Para a monitoração do protocolo SNMP na arquitetura Internet, são apresentados os seguintes parâmetros:

1) Com relação ao *spool* que armazena as mensagens a serem transmitidas:
   - volume total de mensagens atualmente armazenado no *spool*, em kilo-octetos;
   - tempo decorrido desde que a mensagem mais antiga no *spool* foi posta lá;

2) Com relação as mensagens transmitidas:
   - volume total de mensagens transmitidos pelo processo de transferência de mensagens desde a sua inicialização, em kilo-octetos;

3) Com relação as mensagens recebidas armazenadas:
   - número total de receptores especificados em todas as mensagens atualmente armazenadas nas caixas postais para correspondências recebidas;
   - número total de receptores especificados em todas as mensagens transmitidas desde a inicialização do sistema;

4) Com relação as conexões estabelecidas pelo processo transmissor:
   - número total de conexões estabelecidas desde a inicialização do sistema, onde o processo transmissor é o servidor;
   - número de conexões estabelecidas desde a inicialização do sistema, onde o processo transmissor é o cliente;
   - tempo decorrido desde a última vez que o processo transmissor foi ativado pelo estabelecimento de uma conexão para o recebimento de mensagens;
   - tempo decorrido desde a última vez que o processo transmissor foi ativado pelo estabelecimento de uma conexão para a transmissão de mensagens;

5) Com relação a erros no estabelecimento de conexão:
   - número total de conexões não estabelecidas, desde a inicialização do sistema;
   - número total de conexões onde o processo transmissor era o cliente e ocorreu uma falha no estabelecimento da conexão, desde a inicialização do sistema;
   - a razão, caso exista, para que o processo transmissor tenha recusado o estabelecimento da última conexão. Um *string* vazio representaria que a última conexão solicitada foi bem sucedida. Se nenhum pedido de conexão foi realizado desde a inicialização do sistema, seu valor deverá ser *never*;
   - a razão da falha, caso exista, para que a última solicitação de estabelecimento de conexão que o processo transmissor iniciou. Um *string* vazio representaria que a última solicitação

foi bem sucedida. Se nenhum pedido de conexão foi realizado desde a inicialização do sistema, seu valor deverá ser *never*;

6) Com relação a carga de tráfego gerado por esta aplicação na rede, visando o gerenciamento de desempenho:
   - número de *bytes* que trafegam em um segmento;
   - número de pacotes com erro em um segmento.

3.5.1.1.3. MONITORAÇÃO DOS HOSTS

Em [GRI 93] temos a definição de uma MIB para o gerenciamento dos sistemas instalados em um *host*.

Entende-se por *host* qualquer computador que se comunica com outros computadores através de uma rede Internet.

O aspecto principal abordado nesta MIB é o de configuração tanto de *software* quanto de *hardware*, limitando suas variáveis ao escopo do *host*. Dentre elas, destacam-se:
- aplicações instaladas em um *host*;
- nome do fabricante da aplicação;
- nome dado pelo fabricante da aplicação a mesma;
- versão da aplicação instalada;
- número de série da aplicação;
- data e hora da instalação da aplicação;
- identificação de quem instalou a aplicação;
- nome do *path* onde os executáveis são armazenados para a aplicação;
- nome completo do *path* de localização dos executáveis primários associados à aplicação;
- nome e tamanho dos arquivos que compõem a aplicação.

Procurando um detalhamento maior sobre o estado da aplicação, as seguintes métricas são sugeridas:
- data e hora da última vez que a aplicação foi iniciada;
- data e hora em que a aplicação finalizou;
- estado atual da aplicação;
- estado dos *hosts* onde a aplicação está sendo executada;
- tempo de CPU gasto por cada aplicação;
- utilização média de memória utilizada pela aplicação, em *kilobytes*.

Com relação aos recursos disponíveis no host, sugere-se as seguintes informações:
- quantidade de tempo em que a CPU encontrou-se ociosa;
- quantidade física de memória principal contida pelo *host*;
- quantidade física total do *disco rígido* do *host*;
- quantidade livre do *disco rígido* do *host*;
- quantidade de tempo em que o *host* onde a aplicação está instalada encontra-se ativo;
- número de sessões de usuário para as quais o *host* está armazenando informações de estado;
- número de contextos de processos atualmente carregados ou sendo executados no sistema;

- número máximo de contextos de processos que o *host* pode suportar;
- número de conexões de transporte utilizadas pelo processo;
- número de arquivos abertos pela aplicação.

3.5.1.1.4. INFORMAÇÕES SOBRE O PROTOCOLO FILE TRANSFER PROTOCOL (FTP)

Dentre os esforços realizados na especificação de métricas para esta aplicação, destaca-se [CAR 94]. A seguir serão apresentadas algumas destas métricas:
- número de mensagens enviadas e recebidas pelo sistema;
- endereço do *host* que mais aparece como *destination* em pacotes deste utilitário;
- *hosts* da rede que estão executando esta aplicação no momento;
- número de conexões de controle e de transferência de dados estabelecidas;
- tempo médio das conexões;
- número de arquivos transmitidos;
- arquivos mais acessados nos servidores;
- lista dos usuários que acessaram os n arquivos mais procurados;

Além dessas, sugere-se as métricas apresentadas a seguir, que visam cobrir escopos ainda não haviam sido endereçados:
- número de pacotes normais que trafegam em um determinado segmento;
- número de *bytes* normais que trafegam em um determinado segmento;
- número de pacotes com erro que trafegam em um segmento;
- número de *bytes* com erro que trafegam em um segmento;
- número de *bytes* enviados e recebidos.

3.5.1.1.5. INFORMAÇÕES RELEVANTES SOBRE O PROTOCOLO SIMPLE NETWORK MANAGEMENT PROTOCOLO (SNMP)

A seguir, serão apresentadas as informações propostas para a monitoração do protocolo SNMP.
- número de *bytes* trafegando em um segmento;
- número de mensagens com erro trafegando no segmento;
- número de *bytes* com erro trafegando no segmento;
- número total de mensagens recebidas pelo agente;
- número de mensagens, separadas por tipo, recebidas pelo agente;
- número de *traps* enviados pelo agente.

Em um segundo nível de detalhamento, poderíamos ter informações sobre o estado atual dos *daemons* responsáveis por esta aplicação:
- estado atual do agente;
- estado atual do gerente;
- data da última inicialização do agente;
- data da última inicialização do gerente;

3.5.1.1.6. INFORMAÇÕES RELEVANTES SOBRE O SERVIÇO NETWORK INFORMATION SERVICE (NIS)

Dentre as informações relevantes sobre este serviço, destacam-se:

- número de *bytes* em um segmento;
- número de octetos em um segmento;
- data da última atualização do mapa de NIS do servidor;
- data da última atualização do mapa de NIS do cliente.

3.5.1.1.7. INFORMAÇÕES RELEVANTES SOBRE O SERVIÇO NETWORK FILE SYSTEM (NFS)

Visando avaliar o impacto desta aplicação sobre o tráfego de um segmento de rede, sugere-se as seguintes métricas:

- quantidade de *bytes* que trafegam em um determinado segmento;
- número de pacotes que trafegam no segmento;
- número de pacotes com erro que trafegam no segmento;
- número de pacotes com erro trafegando em um segmento;
- arquivos acessados via NFS por um determinado *host*;
- arquivos mais acessados.

3.5.1.1.8. INFORMAÇÕES RELEVANTES SOBRE A APLICAÇÃO TELNET

A aplicação Telnet pode ser monitorada segundo os seguintes parâmetros:

- número de *bytes* trafegando em um segmento;
- número de pacotes com erro circulando em um segmento;
- número de conexões em um *host*;
- controle dos endereços envolvidos na comunicação;
- lista com os usuários que utilizaram a aplicação.

3.5.1.1.9. INFORMAÇÕES RELEVANTES SOBRE O PROTOCOLO HIPERTEXT TRANSFER PROTOCOL

Entre os trabalhos que buscam possibilitar o gerenciamento deste protocolo, destacamos [PIC 95]. A seguir, serão apresentadas as métricas propostas para a monitoração deste protocolo.

- número de *bytes* em um segmento;
- número de pacotes com erro circulando em um segmento;
- número de pacotes do protocolo HTTP que passam por um segmento;

Em um segundo nível de detalhamento, são apresentadas as seguintes informações:

- *hosts* onde os processos deste aplicativo são executados;
- lista dos usuários que mais utilizaram a aplicação recentemente;
- lista dos *sites* acessados recentemente;
- lista dos documentos mais acessados.

CAPÍTULO 4

GERENCIAMENTO PRÓ-ATIVO PROPOSTO

Este capítulo propõe um modelo para o gerenciamento pró-ativo que leva em consideração as informações em nível de rede e de aplicação, apresenta a sua implementação parcial, introduz um conjunto de funções de gerenciamento pró-ativas que se utilizam do nível de informações disponibilizado pelo modelo e a especificação da função de controle de fluxo baseado em aplicação.

4.1. MODELO DE GERENCIAMENTO PRÓ-ATIVO

Em [DEF 96a, DEF 96b], é apresentada a definição de um modelo para o gerenciamento pró-ativo, composto dos seguintes componentes:

1) **Módulo de Gerência Pró-ativa**: responsável pela monitoração e análise da rede, visando verificar as tendências de degradação da rede. Caso alguma tendência seja verificada, uma notificação deve ser gerada.
   - ***Baseline***: registro do comportamento normal da rede, utilizado pelo Serviço de Verificação na análise do estado atual da rede.
   - **Monitor de Rede ou Agente Remoto**: coleta as informações sobre os objetos gerenciados.
   - **Serviço de Verificação**: responsável pela verificação das tendências de degradação da rede.
2) **Plataforma de Gerência de Redes**: responsável por receber as notificações e executar as ações corretivas.
3) **Serviço de Comunicação**: responsável pela interconexão do Módulo de Gerência Pró-ativa e a Plataforma de Gerência: pode utilizar-se do protocolo SNMP ou de um protocolo de transporte.

A Figura 4.1 [DEF 96a, DEF 96b] ilustra estes componentes.

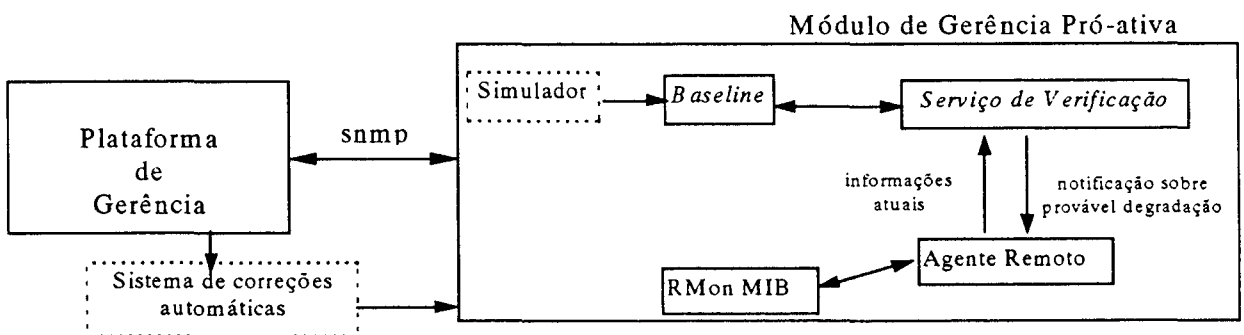


Figura 4.1 - Modelo para a Gerência Pró-ativa

4.2. MODELO DE GERENCIAMENTO PRÓ-ATIVO DE APLICAÇÕES

Visando complementar o modelo anteriormente apresentado para lidar com informações sobre as aplicações foram incluídos no modelo proposto neste trabalho as facilidades de disponibilização e tratamento das informações em nível de aplicação, a técnica de séries temporais na análise da *baseline* e novas funções para o módulo Sistema de Correção Automática. Este módulo passou a fazer parte do módulo de Gerenciamento Pró-ativo com o objetivo de distribuir o gerenciamento da rede. A Figura 4.2 ilustra o modelo de gerenciamento pró-ativo proposto.

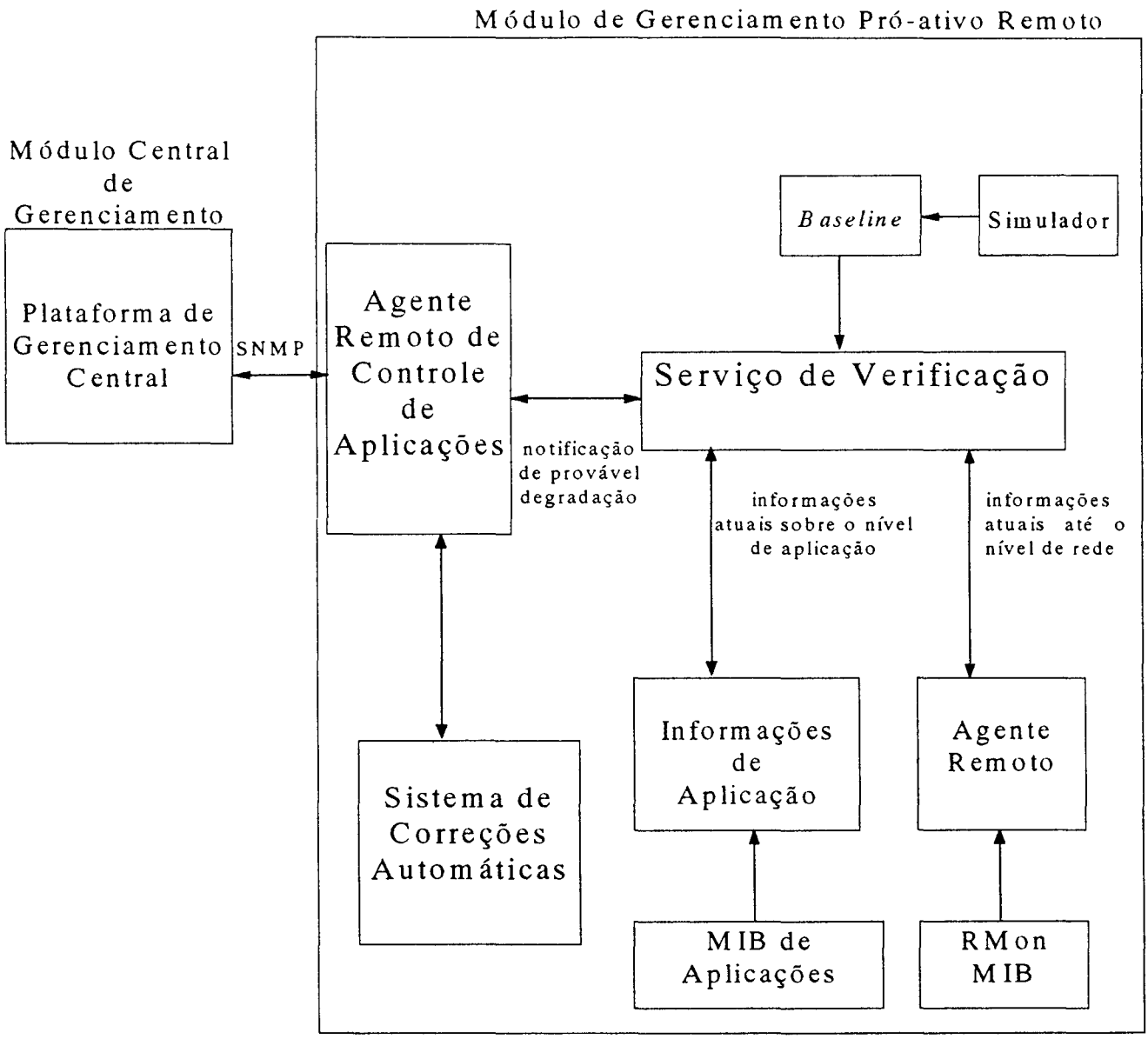


Figura 4.2 - Modelo de Gerenciamento Pró-ativo Proposto

O modelo proposto tem como componentes:

1) **Módulo Central de Gerenciamento**

Este módulo é composto pela Plataforma Geral de Gerenciamento, e tem como tarefa o gerenciamento centralizado da rede.

2) **Módulo de Gerenciamento Pró-ativo Remoto**

Este módulo tem como objetivo a implementação do gerenciamento pró-ativo propriamente dito. Seus componentes são:

- ***Baseline***

Este componente conterá informações sobre o funcionamento normal da rede. Atualmente a *baseline* é gerada através da monitoração da rede, mas também existe a possibilidade de definição de um modelo da rede, para que se possa estimar, através do emprego de simulações, o funcionamento da rede sob várias cargas de trabalho, buscando-se assim uma *baseline* do comportamento mais próximo do ideal possível. As informações nela contidas não são apenas sobre as camadas até o nível de rede mas também sobre as aplicações que estão sendo utilizadas no segmento monitorado, cabendo ao administrador da rede informar o nível de detalhamento que será utilizado.

- **Agente Remoto**

Este componente tem como tarefa a coleta de informações sobre o segmento monitorado até o nível de rede.

- **Informações de Aplicação**

Este componente tem como função a disponibilização de informações sobre as aplicações que estão sendo monitoradas ao módulo **Serviço de Verificação**. O nível de profundidade das informações a serem monitoradas também deve ser definido pelo administrador da rede.

- **Serviço de Verificação**

Este componente tem como funções:

1) a identificação de situações que possam levar a rede a um estado degradado. Sua análise pode ser baseada tanto em informações de mais baixo nível quanto sobre as aplicações que estão sendo monitoradas.

   Esta análise pode ser efetuada de duas formas: através da comparação das informações coletadas com as contidas na *baseline*, ou através do emprego de análises estatísticas sobre os dados coletados (vide seção 5.6).

2) reportar as situações identificadas ao módulo Agente Remoto de Controle de Aplicações.

- **Agente Remoto de Controle de Aplicações**

Este módulo é responsável pelas tarefas de controle local do segmento, tanto a nível de aplicações quanto a níveis mais baixos de informação.

Cabe a este módulo a tarefa de interagir com a **Plataforma de Gerenciamento Central**, possibilitando que esta possa exercer o gerenciamento sobre o segmento remoto. É também do seu escopo informar a Plataforma de Gerenciamento Central da ocorrência de situações que possam levar a degradação do sistema, sua possíveis causas e as soluções adotadas. Além disso, ele é responsável pela interação com o módulo **Sistema de Correções Automáticas**, para que as ações preventivas sejam tomadas.

- **Sistema de Correções Automáticas**

Este componente tem como tarefas a identificação da causa da provável degradação do sistema, a tomada de ações que evitem esta degradação e a notificação do **Agente Remoto de Controle de Aplicações** das causas identificadas e das ações que foram tomadas ou as que são aconselhadas para que a degradação do sistema possa ser evitada ou o problema seja resolvido.

A tomada de decisão sobre as ações que evitem a degradação do sistema, baseiam-se não apenas nas informações em nível de rede, mas também em nível de aplicação, através de refinamentos sucessivos das informações. Para auxiliar na tomada de decisão sobre as ações corretivas que podem ser tomadas, pode-se aplicar um sistema especialista, como proposto em [SAE 96a, SAE 96b]. Devido a complexidade envolvida no desenvolvimento deste componente, o mesmo não foi aprimorado neste trabalho.

Considerando a proposição dos componentes anteriormente descritos, aliado às facilidades disponibilizadas para que o gerenciamento do nível de aplicação fosse possível, a seguinte situação é possível: quando a plataforma de gerenciamento central desejar efetuar ela própria o gerenciamento do módulo remoto, esta poderá fazê-lo utilizando-se das facilidades que disponibilizam através de refinamentos sucessivos das informações. Inicialmente, por exemplo, o administrador poderá informar-se sobre o funcionamento da rede com base em informações como o número de *broadcasts* e o nível de utilização do meio. Se for necessário, o administrador terá a sua disposição a facilidade de aumentar o nível de detalhamento das informações sobre o segmento, obtendo informações sobre o nível de utilização do meio por cada aplicação previamente definida para ser monitorada. Com tal nível de detalhamento de informações, o gerente poderá efetuar sua tarefa em um nível bastante elevado de abstração se comparado com o atualmente utilizado.

Com base no modelo proposto, implementou-se um protótipo do Módulo de Gerenciamento Pró-ativo Remoto, com os seguintes componentes: Agente Remoto, RMON MIB, MIB de Aplicações Informações de Aplicações, Serviço de Verificação e *Baseline*. Para o Sistema de Correções Automáticas foi definido um conjunto de funções de gerenciamento pró-ativas que utilizam informações em nível de aplicação.

4.3.AGENTE REMOTO

Esta seção apresenta a utilização do componente Agente Remoto e a implementação do componente Informação de Aplicação.

O agente remoto utilizado neste trabalho foi o agente *Beholder*. Ele foi instalado em um segmento da rede do Instituto de Computação da Unicamp, formado por diversas estações de trabalho Sun e microcomputadores instalados nas salas dos professores.

4.3.1.COLETA DE DADOS

Visando atender às necessidades de monitoração de dados apresentadas anteriormente, utilizou-se alguns grupos da RMON MIB, a saber:

- History Table: utilizado para a coleta de informações em nível de rede;
- Filter Table: utilizado na filtragem de pacotes das aplicações monitoradas;
- Buffer Table: utilizado para o armazenamento dos pacotes das aplicações monitoradas para posterior filtragem.

4.3.1.1.1. DADOS COLETADOS ATÉ O NÍVEL DE REDE

A coleta de informações até o nível de rede foi feita utilizando-se o agente Beholder. Utilizando-se do grupo HISTORY, vide seção 3.6.1.2, foram criadas duas instâncias, ambas coletando as mesmas informações, divergindo entre elas apenas no período de amostragem:

1. a cada 60 segundos, visando capturar oscilações rápidas;
2. a cada 30 minutos, buscando uma caracterização mais geral do ambiente.

4.3.1.1.2. DADOS COLETADOS EM NÍVEL DE APLICAÇÃO

O agente remoto captura todo tráfego possível. Visando a seleção do tráfego das aplicações a serem monitoradas, serviu-se das facilidades de filtragem de pacotes, uma das mais poderosas disponíveis na RMON MIB [STA 94]. Foram instanciados oito filtros no grupo FILTER, e cada dois destes selecionam um determinado *well-known port*:

- WWW através do *port* de número 80;
- SMTP através do *port* de número 25;
- DNS através do *port* de número 53;
- TELNET através do *port* 23;
- FTP DATA através do *port* 20;
- FTP CONTROL através do *port* 21;

Um canal, instanciado através da tabela CHANNEL, foi associado a cada par de filtros que monitora uma determinada *well-known port*. Por sua vez, estes canais foram associados a um *buffer* do grupo PACKET-CAPTURE, (vide Figura 4.3). Todo pacote que chega à interface de rede do computador é analisado pelo Agente Beholder, que verifica se o pacote atende a especificação dos filtros associados a um determinado canal. Caso positivo, este pacote é armazenado em um *buffer* para posterior processamento. A Figura 4.4 ilustra um relatório gerado pelo agente Beholder sobre a configuração dos grupos *Filter*, *Channel* e *Buffer*. No grupo *Filter*

são especificados as posições do pacote que serão analisados, os valores buscados nestas posições. O grupo *Channel* é configurado com os filtros que compõe cada canal. No grupo *Buffer* são configurados o canal associado e a ação que deve ser tomada quando o *buffer* estiver cheio: se o armazenamento deve ser interrompido, aqui escolhido, ou se o seu funcionamento é o de um *buffer* circular. O programa que efetua a configuração encontra-se no Anexo A .

Quando os pacotes encontram-se disponíveis nos *buffers*, um programa desenvolvido neste trabalho recupera tais informações de um *buffer* em particular executando uma chamada **snmp-get** e **snmp-getnext** ao agente remoto. Este mesmo programa interpreta tais dados, calculando o número de octetos e o número de pacotes que trafegaram na rede de uma determinada aplicação. Estas informações são repassadas a um outro programa que é encarregado do cálculo da utilização do segmento por cada aplicação.

Para a análise e a geração das *baseline*s de aplicação foram desenvolvidos dois *daemons* utilizando-se a linguagem C. O primeiro, que aqui será chamado de aplicd é responsável pelo somatório do número de pacotes e de octetos de uma determinada aplicação que trafegam pelo segmento monitorado e possui um processamento contínuo. O segundo, baplic, é responsável pela geração da *baseline* propriamente dita, arquivando periodicamente as informações geradas por aplicd. Seu funcionamento é periódico e é ativado pela facilidade do sistema operacional chamada Crontab, responsável pela ativação de programas pré-determinados em horários preestabelecidos. O sincronismo no acesso às informações foi implementado através de acesso exclusivo a arquivo, como mostrado na Figura 4.5.

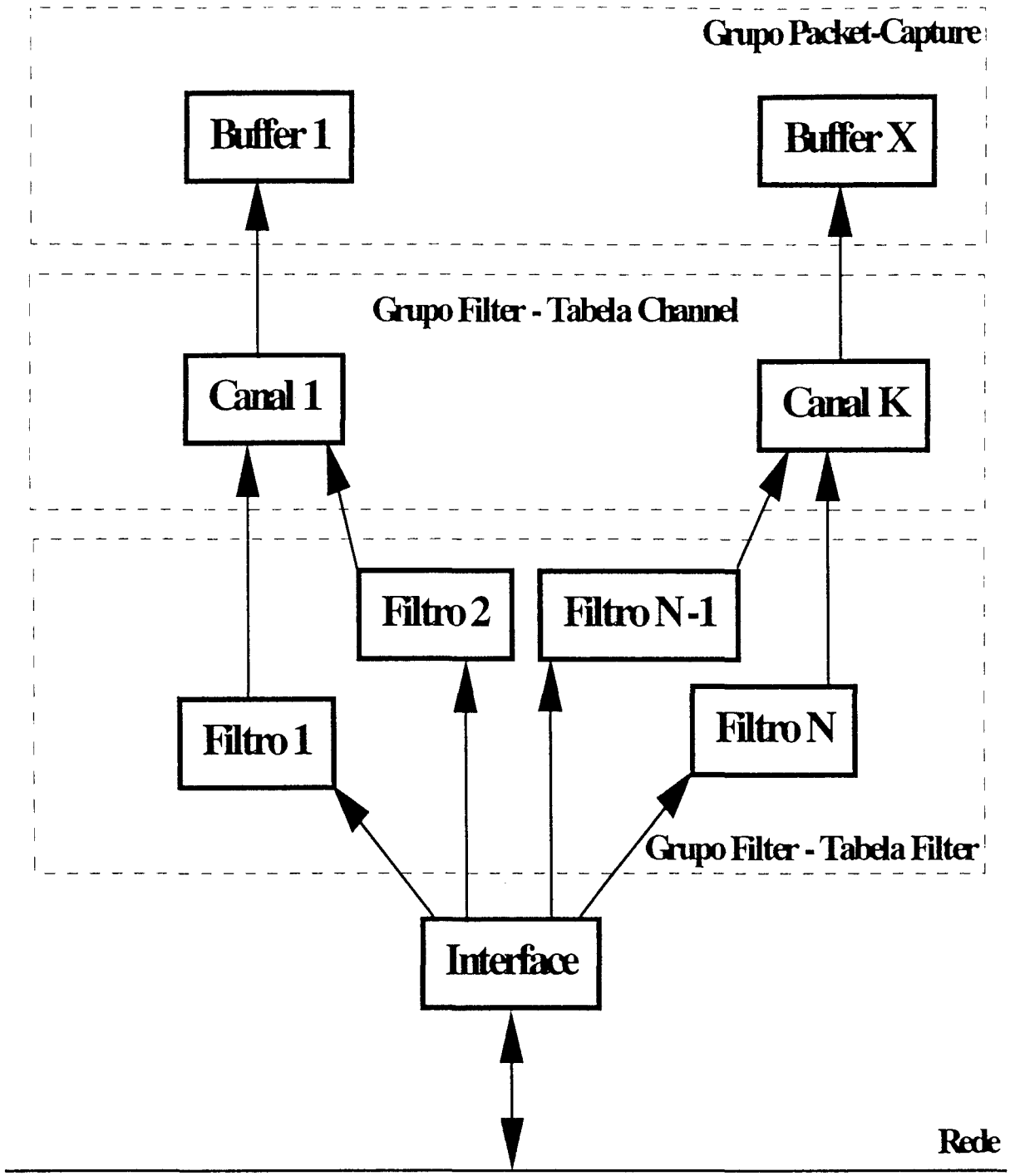


Figura 4.3 - Esquema do funcionamento da coleta de dados

```
watzko@itajai[78]% echo 'filterTable[1]'|snmp-tbl itajai
filterIndex[1]=1
filterIndex[2]=2
filterChannelIndex[1]=1
filterChannelIndex[2]=1
filterPktDataOffset[1]=14
filterPktDataOffset[2]=14
filterPktData[1]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:
00:50
filterPktData[2]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:
00:00:00:50
filterPktDataMask[1]=0f:00:00:00:00:00:00:00:00:ff:00:00:00:00:00:00:00:00:00
:00:ff:ff
filterPktDataMask[2]=0f:00:00:00:00:00:00:00:00:ff:00:00:00:00:00:00:00:00:00
:00:00:00:ff:ff
filterPktDataNotMask[1]=""
filterPktDataNotMask[2]=""
filterPktStatus[1]=0
filterPktStatus[2]=0
filterPktStatusMask[1]=0
filterPktStatusMask[2]=0
filterPktStatusNotMask[1]=0
filterPktStatusNotMask[2]=0
filterOwner[1]=""
filterOwner[2]=""
filterStatus[1]=1
filterStatus[2]=1
watzko@ttl[79]% echo 'channelTable[1]'|snmp-tbl ttl
channelIndex[1]=1
channelIfIndex[1]=1
channelAcceptType[1]=1
channelDataControl[1]=1
channelTurnOnEventIndex[1]=0
channelTurnOffEventIndex[1]=0
channelEventIndex[1]=0
channelEventStatus[1]=1
channelMatches[1]=0
channelDescription[1]=""
channelOwner[1]=""
channelStatus[1]=1
watzko@ttl[80]% echo 'bufferControlTable[1]'|snmp-tbl ttl
bufferControlIndex[1]=1
bufferControlChannelIndex[1]=1
bufferControlFullStatus[1]=1
bufferControlFullAction[1]=1
bufferControlCaptureSliceSize[1]=100
bufferControlDownloadSliceSize[1]=45
bufferControlDownloadOffset[1]=14
bufferControlMaxOctetsRequested[1]=-1
bufferControlMaxOctetsGranted[1]=-1
bufferControlCapturedPackets[1]=0
bufferControlTurnOnTime[1]=33322651
bufferControlOwner[1]=""
bufferControlStatus[1]=1
```

Figura 4.4 - Exemplo de Filtros, Canais e *Buffers* Criados

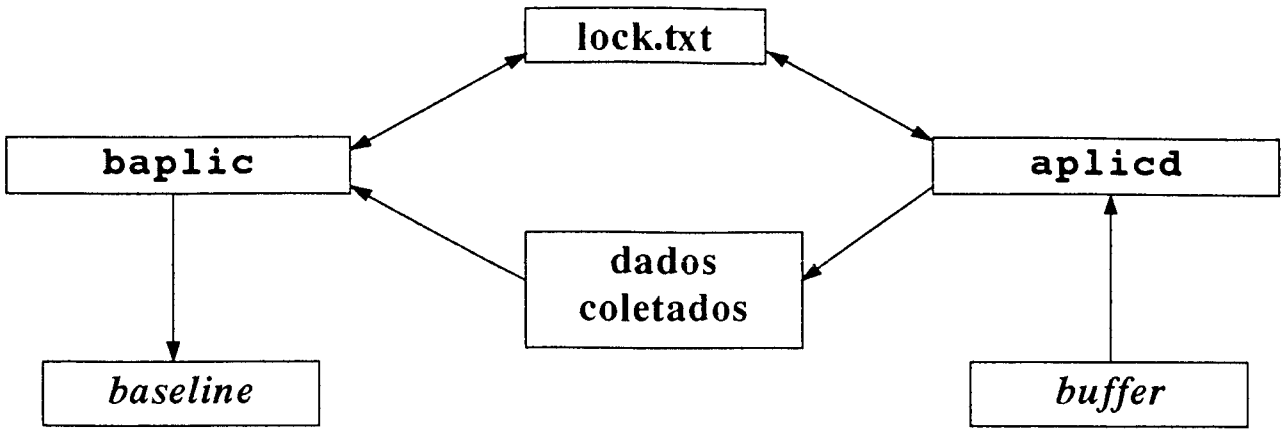


Figura 4.5 - Coleta de Dados Sobre a Aplicação e Geração de *Baseline*

Quando o programa aplicd verifica que o *buffer* está cheio ou que já foi ultrapassado um determinado número de verificações, ele tenta conseguir o acesso exclusivo ao arquivo **lock.txt**. Após conseguir, ele verifica se o conteúdo do arquivo possui uma indicação de que o programa baplic efetuou a leitura dos dados gerados por ele, caso isto seja verdade, ele escreve no arquivo que existem novas informações, inicializa suas variáveis internas, processa o *buffer*, grava as novas informações coletadas e libera o acesso ao arquivo de sincronização.

Por sua vez, o programa baplic também tenta periodicamente conseguir o acesso exclusivo ao arquivo **lock.txt** e, quando consegue, verifica se o seu conteúdo possui alguma informação de que já estão disponíveis novas informações geradas pelo outro programa. Caso isso seja verdade, ele lê as novas informações, libera o acesso ao arquivo de sincronização e só então processa as informações anteriormente lidas, gerando a *baseline*.

4.3.1.1.2.1.FORMAS DE CÁLCULO DE UTILIZAÇÃO DO MEIO POR UMA APLICAÇÃO

Baseando-se nas informações relevantes para o gerenciamento de aplicações, como foi explanado anteriormente, foram implementadas as seguintes métricas: número de pacotes, número de octetos e porcentagem de utilização do meio.

A informação sobre o número de pacotes é obtida através da variável **bufferControlCapturedPackets**, que informa quantos pacotes foram capturados no *buffer*.

O número de octetos é calculado através do campo *total length* da *frame* IP dos pacotes capturados.

O cálculo da utilização do meio por uma aplicação, considerando uma rede Ethernet (Figura 4.6) é feita utilizando-se a equação 4.1 [STA 96].

$$\text{utilização} = \frac{(\text{Pacotes} * [96 + 64]) + (\text{Octetos} * 8)}{\text{Intervalo} * 10^7} \qquad (4.1)$$

onde:

- 64 *bits* = preâmbulo

- 96 *bits* = espaço vazio entre os pacotes
- 10^7 *bits* por segundo = Capacidade do meio físico (rede Ethernet)
- Intervalo = Intervalo de amostragem em segundos
- Pacotes = Número de pacotes que passaram pelo canal
- Octetos = Número de octetos calculados pelo programa

7	1	6	6	2	>=0	> =0	>=0	4
preamble	SFD	DA	SA	*length*/type	LLC	data	pad	FCS
Preâmbulo				**Octetos**				

Figura 4.6 - *Frame* Ethernet

Com base nesta informação é possível calcular a porcentagem de utilização gerada por cada aplicação no segmento.

Estas informações serão utilizadas pelo módulo de verificação, que reportará estes dados ao gerente remoto para que este possa tomar decisões baseadas nestas informações.

4.4.*BASELINE*

Visando caracterizar o comportamento do segmento monitorado, para que se possa determinar se a rede está com problemas que podem ocasionar a degradação no seu desempenho, foram coletadas *baseline*s tanto em nível de rede quanto em nível de aplicação.

4.4.1.*BASELINES* ATÉ O NÍVEL DE REDE

Vários são os problemas que podem ser identificados com a análise das *baselines* coletadas até o nível de rede. Entre eles destacam-se:

- Problemas de degradação no nível físico: como principais problemas que afetam o desempenho da rede no nível físico destacam-se conectores e placas de rede em mal estado, gerando erros de transmissão classificados como [NEM 92]: erros de CRC/alinhamento e pacotes curtos/longos. Taxas significativas destes tipos de erros indicam mal funcionamento de equipamento;
- Problema de Configuração: uma rede com uma topologia mal configurada pode causar problemas como altas taxas de colisões e de utilização [SAE 96a];
- Problemas com Tormenta de Pacotes *Broadcast*: caracteriza o envio de pacotes com o endereço *Broadcast Ethernet* em um volume grande o suficiente para causar problemas em máquinas da rede. As principais causas deste tipo de erro são: problemas nos protocolos, programas mal configurados e implementações com defeitos;
- Sobrecarga da Rede: este problema pode ser detectado através da monitoração do nível de utilização da rede e da taxa de colisões. Valores acima de 35% para o nível de utilização e de 2% para a taxa de colisões são um forte indicativo de que a rede está sobrecarregada;

Com relação a pacotes do tipo *multicast*, um número excessivo destes pacotes pode ser um indicativo de problemas nos equipamentos da rede, excesso de máquinas em um segmento ou apenas refletir a configuração de algum equipamento.

É apresentado a seguir, um exemplo de análise de uma *baseline* que foi gerada baseada na monitoração de um segmento do Instituto de Computação do qual fazem parte microcomputadores e estações de trabalho. O equipamento utilizado para a coleta dos dados foi um computador Pentium de 133 Mhz de relógio e 48 Mbytes de memória RAM, utilizando o sistema operacional Linux Slackware 2.2 e o agente remoto *Beholder*. A monitoração foi feita durante 40 dias entre os meses de fevereiro e março. Durante este período não foi registrado nenhum evento de perda de pacotes, também não foram constatados erros dos tipos: pacotes grandes, pacotes com erro de CRC ou de alinhamento, fragmentos e *jabbers*. Será apresentada a seguir uma análise dos dados coletados.

Colisões

Durante o período monitorado, o nível de colisões apresentou-se bastante baixo com média de 0,0002% e máximo de 0,035% do número de pacotes trafegando no momento. Este evento ocorreu em uma quarta-feira às 10 horas da manhã. Neste mesmo instante a utilização do segmento era de 6%, o número de pacotes trafegando era de $2{,}53*10^5$, o número de pacotes *multicast* era inferior a 0,04%, o de pacotes *broadcast* era inferior a 0,4%, os pacotes curtos correspondiam a 0,3%, valor máximo observado naquele dia. Neste instante nenhuma das cinco aplicações monitoradas destacou-se e a utilização de cada uma foi abaixo de 0,5% do nível de utilização do segmento.

Nas segundas-feiras observadas, o nível de colisões foi maior no período das 12 às 18 horas. Terças-feiras tiveram o período das 12 às 18 como o de maior número de colisões seguido pelo horário das 6 às 12 horas. O período das 6 às 12 horas foi o de maior níveis de colisões nas quartas. Nas quintas, observadas, o nível de colisões foi maior das 12 às 18 horas, comportamento este que se repetiu nas sextas. O comportamento observado durante os finais de semana apresentaram valores bastante baixos em todos os horários. De uma forma geral, os dias com maiores taxas de colisões foram terça e quarta. A Figura 4.7 apresenta a plotagem das colisões observadas.

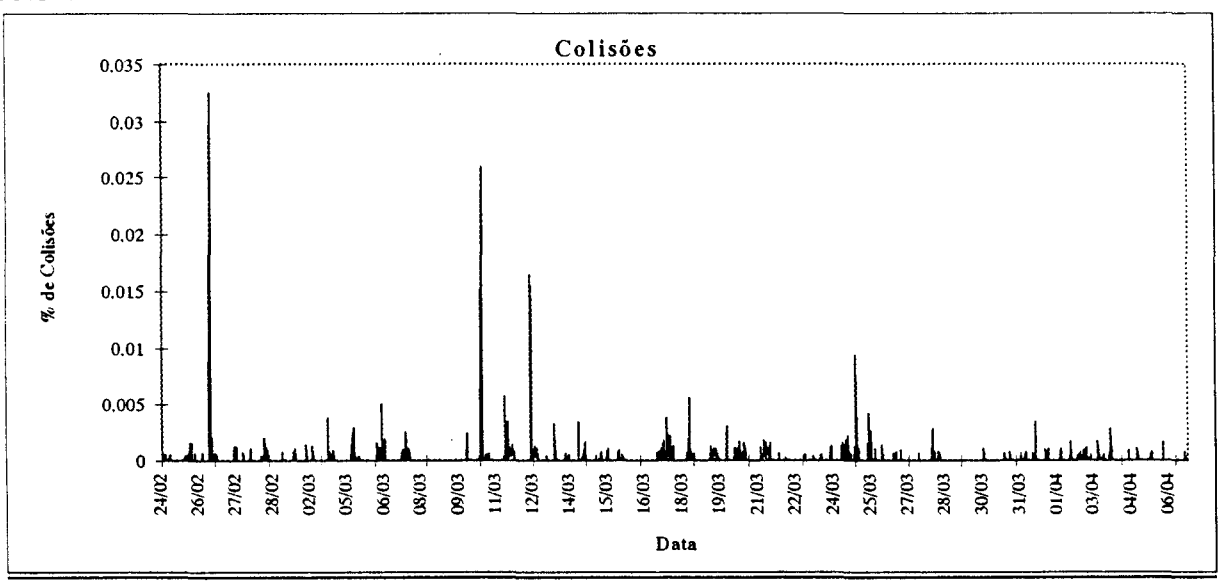


Figura 4.7 - Colisões Observadas

Pacotes Curtos

Neste trabalho entende-se pacotes curtos como sendo pacotes de tamanho menor ao limite inferior especificado em [ISO 8802.3] que é de 64 *bytes*. A média do nível de pacotes curtos durante o período observado foi de 0,0073% tendo como valor máximo 0,9% do número total de pacotes, valor este bastante baixo. Tal pico ocorreu em uma quinta-feira às 13 horas, neste momento, o nível de utilização era inferior a 2%, o nível de colisões era inferior a 0,005%, pacotes *broadcast* representavam menos de 0,6% do número de pacotes total e os de *multicast*, 0,06%, o número de octetos foi um dos mais baixos daquele dia, 16910898 octetos, e o número de pacotes era de $1{,}03*10^5$. A aplicação TELNET apresentou o maior nível de utilização neste momento (0,5%). No período observado, o horário das 6 às 18 horas foi o que apresentou o maior número de ocorrências deste tipo de pacote porém não fica caracterizado nenhum intervalo nele contido de maior ocorrência. Os finais de semana apresentaram valores abaixo dos observados durante os dias úteis. A Figura 4.8 apresenta a plotagem dos dados coletados.

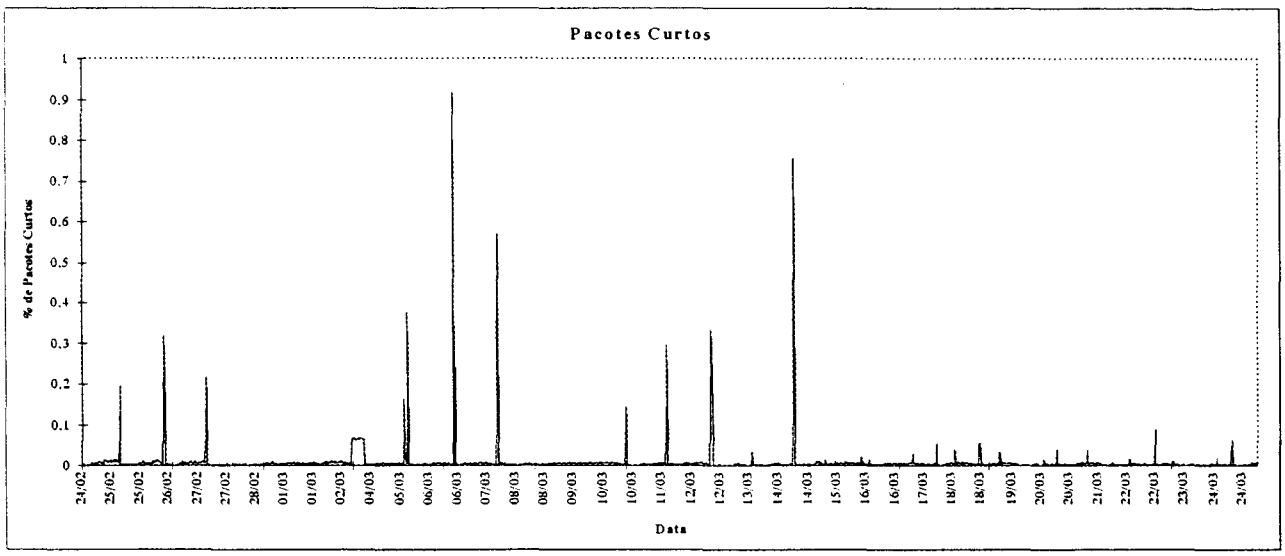


Figura 4.8 - Gráfico dos Pacotes Curtos Observados

Pacotes *Multicast*

O número mínimo de pacotes *multicast* observado foi de 0,006% e a média do período foi de 0,042% do total de pacotes. O valor máximo observado foi de 0,2% em uma quinta-feira às 14 horas e 30 minutos. Neste momento o nível de utilização era inferior a 6%, os pacotes *broadcast* correspondiam a menos de 0,4% e os pacotes curtos, menos de 01%, o número de pacotes curtos era desprezível e o número total de pacotes era de $2{,}53*10^5$, o número de octetos era inferior a $1{,}05*10^8$ e nenhuma das aplicações monitoradas apresentou um nível de utilização que se destacasse.

A distribuição deste tipo de pacote deu-se de forma bastante constante durante os diversos períodos do dia e os diferentes dias da semana. O gráfico dos pacotes *multicast* observados é apresentado na Figura 4.9.

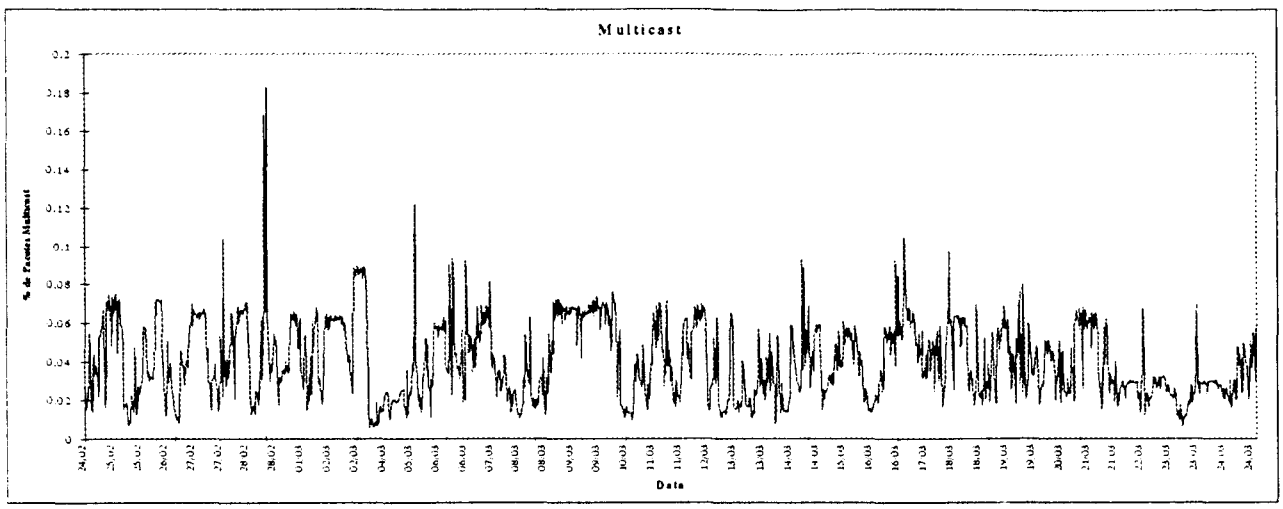


Figura 4.9 - Gráfico dos Pacotes *Multicast* Observados

Pacotes *Broadcast*

A média de pacotes *broadcast* durante a monitoração foi de 0,431% e o mínimo observado foi de 0,064%. O valor máximo observado foi de 1,2% do total de pacotes numa quinta-feira às 5 horas e 30 minutos, o nível de utilização era inferior a 1%, o nível de colisões e de pacotes curtos era de aproximadamente zero, o de pacotes *multicast* era inferior a 0,06%, o número de pacotes era de aproximadamente $1,03*10^5$ e o de octetos, $4,65*10^6$. Nenhuma das aplicações monitoradas apresentou um desempenho diferenciado.

Os maiores níveis deste tipo de pacote foram observados no período das 18 às 6 horas, fato este decorrente do baixo número de pacotes trafegando neste período. Durante o dia, a maior parte dos valores mais altos foi verificada no período entre 12 e 18 horas. Nos finais de semana, a ocorrência deste tipo de pacote é mais constante que durante o resto da semana, quando os níveis observados sofreram maiores variações. O gráfico dos pacotes broadcast é apresentado na Figura 4.10.

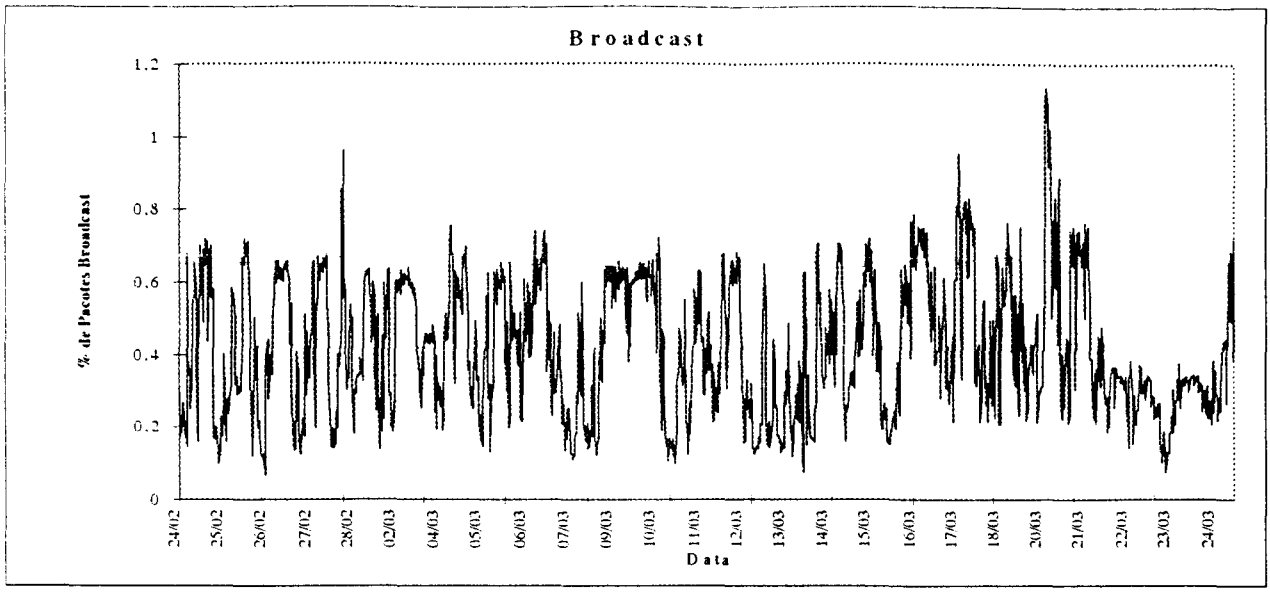


Figura 4.10 - Gráfico dos Pacotes *Broadcast* Observados

Utilização

Durante o período observado, a média de utilização do segmento foi de 1,99%, com o valor mínimo de 0,25% e o máximo de 11,73%, observado numa quinta-feira às 15 horas e 30 minutos. Neste momento o nível de pacotes broadcast era inferior a 0,2% e o de multicast menor que 0,02%, não havia colisões nem pacotes curtos, o número de pacotes era de $3,03*10^5$ e o de octetos foi o valor máximo observado para este parâmetro. A aplicação que mais se destacou neste momento foi a WWW, com aproximadamente 0,45% do nível de utilização.

Os níveis de utilização caracterizam-se por apresentar valores maiores no período das 12 às 18 horas, seguido pelo das 6 às 12 horas. Os níveis mais baixos foram constatados durante o final de semana. Dentre os dias úteis, a quarta-feira é o dia que apresenta um nível mais elevado de utilização e a sexta-feira, o nível mais baixo.

De uma forma geral o segmento monitorado apresenta um baixo nível de utilização. Não foram constatadas nenhuma situação anormal durante a coleta dos dados. Os níveis de pacotes broadcast, multicast, pacotes curtos e colisões são bastante baixos não inspirando maiores cuidados. Seu comportamento pode ser caracterizado por uma maior utilização durante os dias da semana e mais baixos nos finais de semana. O período das 12 às 18 horas apresentam maior atividade que os outros períodos do dia. A *baseline* desta variável pode ser observada no Anexo B.

4.4.2. *BASELINES* DO NÍVEL DE APLICAÇÃO

Semelhante às *baselines* coletadas até o nível de rede, foram coletadas também *baselines* em nível de protocolo de aplicação, buscando a caracterização da utilização das diversas aplicações, para que estas informações possam servir de subsídio ao gerenciamento pró-ativo em nível de aplicação.

Esta *baseline* é composta pelos seguintes parâmetros: número total de pacotes transmitidos, taxa de utilização do segmento e número total de octetos transmitidos. Destaca-se aqui, a dificultada encontrada em se obter informações sobre as aplicações que eram executadas, devido a natureza das aplicações, que foram desenvolvidas sem a preocupação com o desenvolvimento de interfaces apropriadas para o seu gerenciamento. Visando contornar este problema, foram monitorados parâmetros que pudessem ser obtidos a partir do tráfego gerado pelas aplicações.

A *baseline* de aplicação permite analisar o padrão de utilização das aplicações, verificando quais são as mais utilizadas, os horários de utilização e o impacto de cada aplicação no comportamento da rede. Um exemplo de análise de uma *baseline* de aplicação é apresentado a seguir e a plotagem da *baseline* coletada da taxa de utilização do segmento é apresentada no Anexo C.

Dentre as aplicações monitoradas, a que apresentou maiores níveis de utilização foi a WWW, seguida pelo FTP e TELNET e a que apresentou menores valores foi o DNS. Apesar de ter apresentado altos índices de utilização menos vezes que a aplicação WWW, o FTP foi responsável pelos maiores picos observados, chegando a 40% do nível de utilização. Constatou-se a existência de um processo periódico programado pelo usuário, a saber, uma aplicação FTP às 0 hora e 30 minutos durante vários dias. Não verificou-se nenhum outro padrão de comportamento das aplicações monitoradas além de uma maior utilização no período entre às 6 e 18 horas dos dias úteis, sendo que no período das 12 às 18 horas houve uma maior utilização das aplicações. Com relação aos finais de semana monitorados, merece destaque a maior utilização da aplicação TELNET, resultado do acesso remoto feito pelos usuários, porém a aplicação que apresentou taxas de utilização mais elevadas foi a WWW. A Tabela IV apresenta uma síntese porcentagem das aplicações monitoradas em relação ao nível de utilização do segmento.

	Mínimo	Média	Máximo
DNS	0	$9{,}625*10^{-5}$	0,0166
FTP	0	0,199	39,425
SMTP	0	0,025	2,279
TELNET	0	0,051	3,050
WWW	0	0,342	30,427

Tabela IV - Síntese das Aplicações Monitoradas

O perfil de funcionamento apresentado anteriormente é importante pois permite conhecer a operação normal da rede e poder identificar quando ocorrem variações que comprometam o seu desempenho.

4.5. MÉTRICAS DE DESEMPENHO

Neste trabalho, a identificação das variações que possam comprometer o desempenho do segmento dá-se da seguinte forma: utiliza-se os *thresholds* definidos por [NAS 94] para as variáveis apresentadas na Tabela V.

Variável	*Threshold*
Colisões	1%
Pacotes *Multicast*	8%
Pacotes *Broadcast*	8%
Pacotes com erro de CRC ou de alinhamento	2%
Pacotes Curtos	2%
Pacotes Fragmentados	2%
Pacotes *Jabbers*	2%

Tabela V - Parâmetros de Monitoração e *Thresholds* Associados

O nível de utilização da rede será analisado segundo os intervalos de confiança gerados utilizando-se a análise de série temporal apresentada no capítulo 6. Com esta técnica consegue-se estimar um intervalo para a média do nível de utilização do segmento para a próxima meia hora.

Por total falta de parâmetros para se estimar *thresholds* para as aplicações, foram utilizados os parâmetros apresentados na Tabela VI. A escolha destes valores deu-se através da observação do seu comportamento durante o período monitorado e pela prioridade que se deseja dar a determinadas aplicações.

Aplicação	*Threshold*
DNS	3%
SMTP	5%
TELNET	10%
FTP	20%
WWW	40%

Tabela VI - *Thresholds* para Aplicações

Com relação as variáveis até o nível de rede, a verificação da ocorrência de alguma situação que possa levar a degradação dá-se através da verificação de que os valores observados ultrapassaram os *thresholds* definidos pelo menos três vezes na mesma hora. Já os *thresholds* das variáveis em nível de aplicação são analisados de outra forma, pois a ocorrência de que algum valor observado ultrapassou um determinado valor pré-estabelecido não caracteriza necessariamente a ocorrência de uma situação que possa levar a degradação, tal conclusão só pode feita analisando-se também as variáveis até o nível de rede. Quando uma variável monitorada em nível de aplicação ultrapassa um *threshold* representa a alteração do comportamento daquela aplicação mas não necessariamente uma degradação do sistema.

4.6. Funções de Gerenciamento Pró-ativo

Um dos objetivos deste trabalho é a proposição de funções de gerenciamento pró-ativas baseadas nas informações colhidas sobre as aplicações que estão sendo executadas em um segmento.

Destaca-se apenas que estas funções tem como objetivo evitar que uma degradação no desempenho da rede venha a ocorrer. Elas não subtraem a necessidade de uma análise mais

detalhada das causas que levaram a ocorrência do erro, atividade esta que deve ser feita posteriormente.

4.6.1.NOTIFICAÇÃO AO GERENTE DA REDE

Uma das mais importantes funções é a notificação do gerente da rede sobre a ocorrência de situações que podem levar a degradação do sistema e as ações pró-ativas tomadas. Tais notificações funcionariam como um *log* do sistema, possibilitando ao gerente, uma análise das situações ocorridas visando identificar as causas.

Esta função pode ser implementada através do envio de mensagens do tipo *trap* à estação central de gerenciamento ou do envio de qualquer outro tipo de mensagem como, por exemplo, um correio eletrônico de notificação.

Esta função foi implementada no módulo Agente de aplicações remoto, vide seção 2.1.2.1.1, possibilitando que este possa informar ao gerente da ocorrência de uma situação que pode levar a degradação do sistema.

4.6.2.MUDANÇA DE ROTA EM UM ROTEADOR

Outra função pró-ativa proposta é a mudança de rota em um roteador quando um de seus canais de saída apresentar níveis de utilização que possam levar a degradação do sistema como um todo.

Baseando-se nas informações apresentadas na seção 2.1.2.1.1, tem-se que a **Função de Mudança de Rota de um Roteador** aplica-se em um roteamento estático onde as mudanças na tabela dependeriam de intervenção direta desta função sobre a tabela de roteamento. Esta intervenção poderia ser feita, por exemplo, através de um TELNET no roteador.

Esta tabela de roteamento utilizaria o método para a manutenção da sua estrutura de dados o encaminhamento adaptativo, onde informações utilizadas para a decisão de uma mudança de rota não seriam apenas as informações em nível de rede, mas também, sobre as aplicações que estão trafegando por aquela rota.

Tanto a atualização periódica destas informações quanto a forma de se efetuar o roteamento de pacotes poderia ser feita de forma distribuída, onde cada nó envia periodicamente aos outros nós, incluindo os *gateways*, informações locais sobre a carga da rede.

O algoritmo de roteamento a ser empregado, utilizaria-se de informações sobre as aplicações que estivessem sendo utilizadas e seus níveis de utilização, para determinar qual a melhor rota para o pacote de uma determinada aplicação.

4.6.3.CONTROLE DE FLUXO BASEADO EM APLICAÇÕES

Esta função tem como objetivo propor uma opção às formas de controle de fluxo atualmente existentes, baseando-se no tráfego de aplicações.

Na seção 4.8 é apresentada uma descrição detalhada desta função.

4.6.4.MONITORAÇÃO DO ESTADO DE UMA APLICAÇÃO

A função de **Monitoração do Estado de Aplicações**, apesar de não se basear no tráfego das mesmas, também é de grande valia.

Ela seria responsável pela monitoração do estado dos *daemons* que compõem a aplicação e, quando algum destes passasse para um estado inválido, ela encarregaria de matá-lo e iniciar um novo processo da aplicação.

4.6.5.CONTROLE DE CONGESTIONAMENTO BASEADO EM APLICAÇÕES

A função de Controle de Congestionamento em um segmento teria como objetivo promover uma política de descartes de pacotes baseada em aplicações.

A prioridade dada a cada aplicação seria previamente especificada. A técnica a ser aplicada privilegiaria as aplicações com mais alta prioridade, descartando mais pacotes das aplicações menos prioritárias.

4.6.6.LIMITAÇÃO DE ACESSO

Após a análise do comportamento do segmento, é possível especificar os horários de maior utilização. Dependendo de quais aplicações se deseja privilegiar, pode-se definir datas, horários e número de usuários que podem executar uma determinada aplicação através do comando **`limit`**, da seguinte forma: **`limit`** *class number times message* como, por exemplo: limit anonymous 100 any /etc/ftpmsgs/msg.toomany limita o número de usuários anônimos ao máximo de 100 durante todos os dias e períodos.

4.6.7.CONTROLE DE PRIORIDADE DE PROCESSO

Uma forma de controlar o tráfego gerado por uma aplicação é o controle da prioridade que este processo possui para executar. Processos com menos prioridade terão menos oportunidade de enviar pacotes para o segmento. Este tipo de solução se aplica a processos servidores, que tem como característica o recebimento de mensagens pequenas e gera como resposta uma certa quantidade de tráfego, como um servidor de ftp, por exemplo.

4.6.8.EXECUÇÃO CONDICIONADA DE PROCESSOS

O que se pretende com esta ação é limitar a geração de tráfego de aplicações que são executadas periodicamente como, por exemplo, o servidor de correio eletrônico, fazendo com que sua execução ocorra apenas nos horários de menor utilização do segmento ou que a sua execução em um determinado horário esteja condicionada ao nível de congestionamento do segmento naquele momento.

4.7.SISTEMA DE VERIFICAÇÃO

Nesta seção será descrita a implementação do protótipo do Sistema de Verificação, ilustrado na Figura . A comunicação entre estes módulos foi implementada utilizando-se as facilidades de *port* fornecidas pelo sistema operacional. Os programas foram desenvolvidos em linguagem C. Esta implementação foi feita em um computador Pentium de 133 Mhz de velocidade, com sistema operacional Linux Slackware 2.2.2.

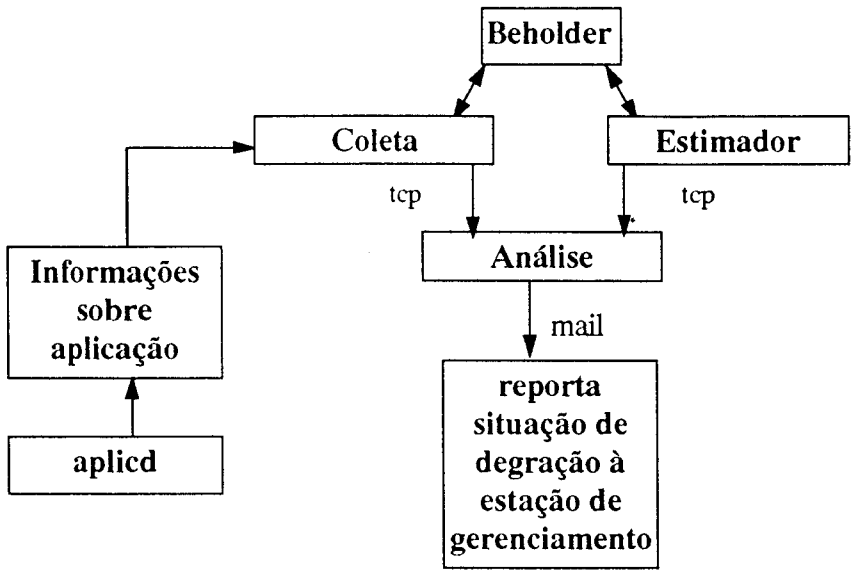


Figura 4.11 - Esquema do Sistema de Verificação

O sistema de verificação implementado é composto por três módulos: Coleta, Análise e Estimador. O módulo Coleta tem a tarefa de interagir com o programa *Beholder* e os programas de coleta de informações sobre aplicações, filtrar as informações coletadas e enviá-las ao módulo Análise. A Figura apresenta o esquema do seu funcionamento.

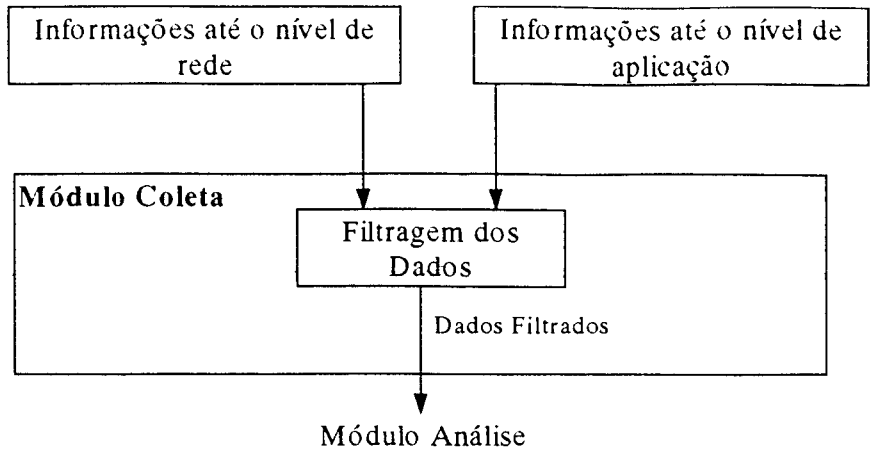


Figura 4.12 - Módulo Coleta

O módulo Estimador tem por função estimar de um limite superior para a média de utilização para a próxima meia hora de funcionamento do sistema. A estimativa calculada é transmitida ao módulo Análise. A Figura apresenta uma representação do seu funcionamento.

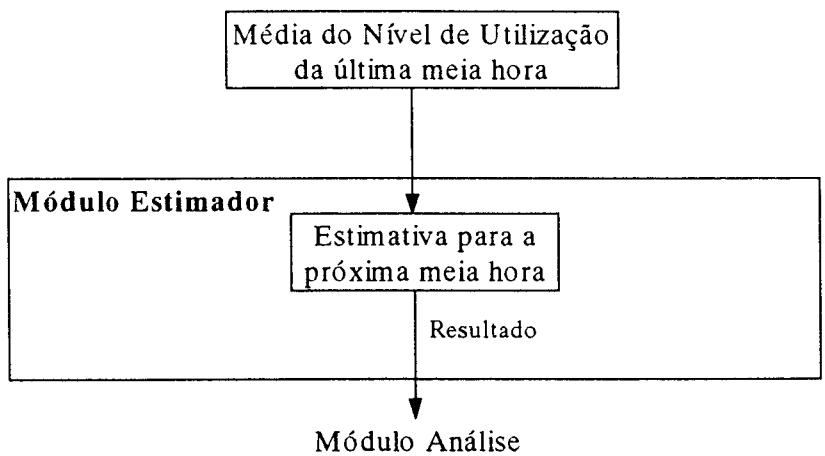


Figura 4.13 - Módulo Estimador

O módulo Análise é responsável pelo recebimento das informações do módulo Coleta e Estimador e, com base nestas informações, verificar a existência de tendência de degradação através da verificação de que o parâmetro monitorado ultrapassou o limite superior mais de três vezes na última meia hora. A Figura apresenta um esquema do seu funcionamento.

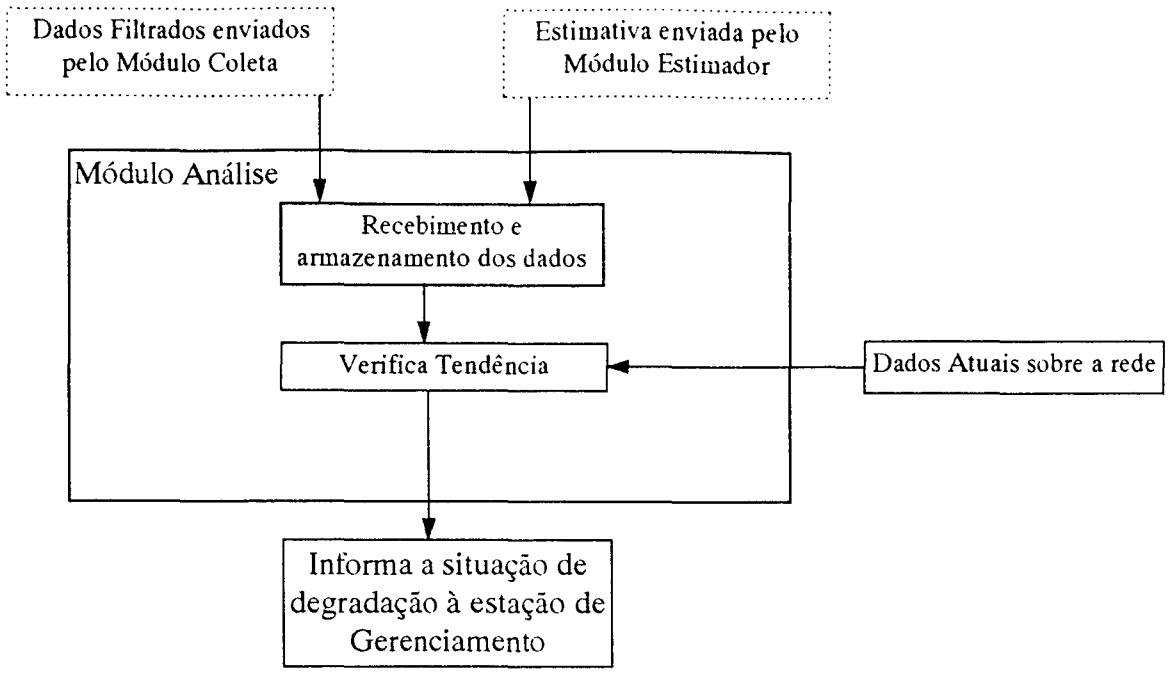


Figura 4.14 - Módulo Análise

4.8. FUNÇÃO DE CONTROLE DE FLUXO BASEADO EM APLICAÇÃO

Na seção 4.6 apresentou-se um conjunto de funções de gerenciamento pró-ativas baseadas nas informações sobre as aplicações que estão sendo executadas no segmento monitorado. Apresenta-se a seguir a especificação da função de controle de fluxo baseado em aplicação.

Este capítulo é composto por uma descrição do funcionamento da comunicação entre aplicação e a entidade de transporte, o controle de fluxo implementado no protocolo TCP, a situação existente com a inclusão de informações sobre as aplicações que estão trafegando no segmento e a forma de implementação do mecanismo proposto.

4.8.1. COMUNICAÇÃO ENTRE A CAMADA DE TRANSPORTE E A CAMADA DE APLICAÇÃO

A camada de transporte provê um serviço de comunicação confiável e eficiente aos seus usuários, utilizando-se das facilidades oferecidas pelas camadas inferiores. Os programas/equipamentos que implementam estas facilidades são chamados de ENTIDADES DE TRANSPORTE. Tal entidade pode estar localizada no sistema operacional, em um processo de usuário, em um pacote de bibliotecas ou na placa de rede. A Figura 4.15 apresenta um esquema de funcionamento entre a camada de aplicação e a de transporte.

Quando um processo servidor deseja receber informações ele executa uma primitiva LISTEN em um determinado Endereço de Transporte (também conhecido como *port*), normalmente chamando uma biblioteca de procedimentos que bloqueia o processo até que uma solicitação de estabelecimento de conexão chegue. Se um processo cliente deseja se comunicar com o servidor, ele executa uma primitiva CONNECT, que fará com que a entidade de transporte

bloqueie-o e envie a solicitação de conexão ao processo servidor. Ao receber uma mensagem de solicitação de conexão, a entidade de transporte verifica se o processo servidor está bloqueado aguardando mensagens. Em seguida, a entidade de transporte desbloqueia o processo servidor e envia uma mensagem de "conexão aceita" para o cliente, que é desbloqueado ao recebê-la e a conexão é estabelecida. A partir deste momento, os dados podem ser enviados ou recebidos através das primitivas SEND e RECEIVE. De forma transparente aos processos servidor e cliente, a entidade de transporte efetua a confirmação do recebimento das informações transmitidas, o controle das retransmissões e a verificação dos *timers*. Quando a conexão não é mais necessária, ela é desfeita através da utilização de uma primitiva DISCONNECT que encarrega-se de enviar uma mensagem a outra entidade solicitando o cancelamento da conexão. Ao receber tal mensagem, a entidade de transporte desfaz a conexão.

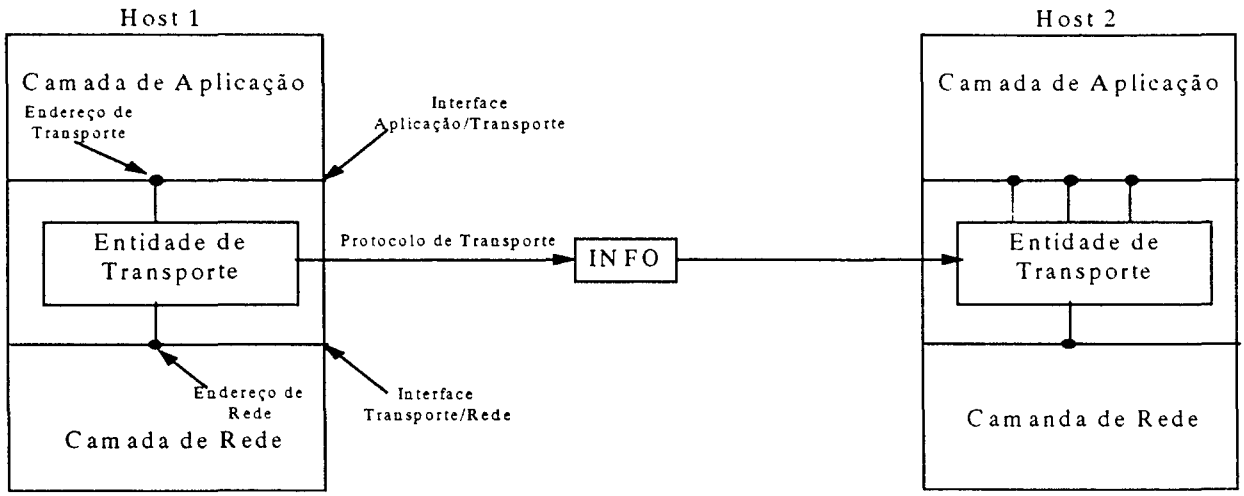


Figura 4.15 - Esquema de Funcionamento entre os Níveis de Aplicação e de Transporte

É de responsabilidade da entidade de transporte a multiplexação de várias conexões em nível de transporte em apenas uma conexão em nível de rede ou vice-versa. A Figura 4.16 apresenta um esquema desta atividade.

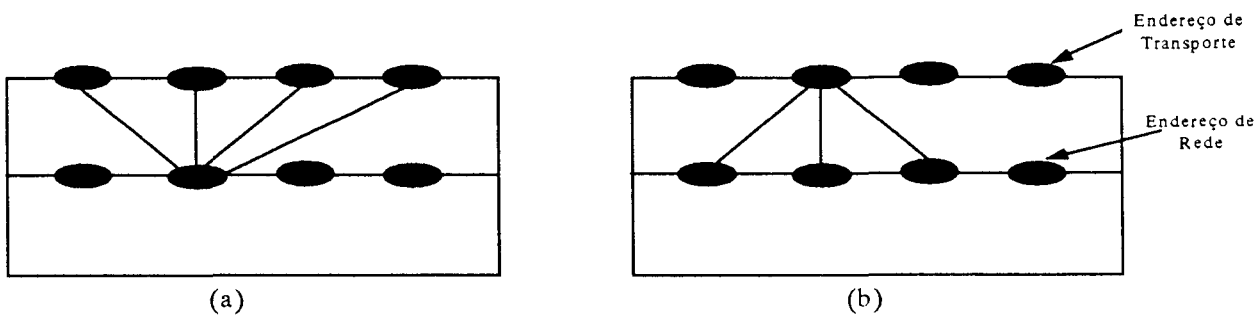


Figura 4.16 - Esquema de Multiplexação de Várias Conexões em Nível de Aplicação em Uma em Nível de Rede (a) e vice-versa (b)

4.8.2. POLÍTICA DE TRANSMISSÃO

Na implementação TCP, o gerenciamento do tamanho do *buffer* livre para receber dados, também conhecido como janela, não está diretamente associada à confirmação do recebimento de dados.

Quando uma entidade receptora recebe dados, ela envia à entidade emissora a confirmação do recebimento dos mesmos seguido da quantidade de espaço disponível no *buffer* para o recebimento de novos dados. Se o espaço disponível for zero, a entidade emissora deve esperar até que o processo de aplicação na máquina do receptor remova alguns dados armazenados e então a entidade receptora possa informar a existência de espaço disponível.

Se uma entidade emissora recebe a informação de que não existe espaço disponível para armazenar novos dados, ela só poderá enviar novas mensagens à entidade receptora em duas situações: dados urgentes, tais como a indicação para finalizar o processo, ou para que a entidade emissora informe novamente a última confirmação feita e o tamanho do *buffer* disponível.

A entidade emissora não está sujeita ao envio imediato dos dados provenientes do processo de aplicação e nem a entidade receptora é obrigada a enviar a confirmação do recebimento dos dados imediatamente.

4.8.3. ALGORITMO DE NAGLE

Um problema bastante comum em conexões de processos TELNET ou RLOGIN é o envio de apenas um *byte* de informação que resultará no tráfego de 41 *bytes* pelo segmento, resultante de 20 *bytes* do cabeçalho IP acrescido de 20 *bytes* do cabeçalho TCP e de 1 *byte* de dado.

Uma tentativa de se otimizar esta situação é atrasar a confirmação e a atualização da quantidade de espaço livre no *buffer* na esperança de que aumente a quantidade de informações a serem trafegadas ou de espaço livre no *buffer*. Infelizmente, esta tentativa reduz apenas a ineficiência da entidade receptora pois a emissora continuará enviando pacotes de 41 *bytes* de comprimento com apenas 1 *byte* de dado.

Uma forma de resolver este desperdício do lado da entidade emissora é conhecida como algoritmo de Nagle [NAG 84], que sugere a seguinte abordagem: uma conexão TCP pode possuir apenas um pequeno segmento ainda cujo recebimento ainda não foi confirmado. Nenhum segmento pequeno adicional poderá ser transmitido até que a confirmação seja recebida. Enquanto isto não ocorrer, a entidade TCP armazena as pequenas quantidades de dados e as transmite todas de uma só vez quando a confirmação for recebida. Uma característica interessante deste algoritmo é a sua autotemporização pois quanto mais rápido forem recebidas as confirmações, mais rápida será o envio dos dados [STE 94].

Este algoritmo é largamente utilizado nas implementações do protocolo TCP existentes mas em algumas situações sua utilização não é recomendável, por exemplo, quando uma aplicação transmite pela rede os movimentos do *mouse* para serem processados em outra máquina. Neste caso, a utilização de tal algoritmo fará com que o movimento do *mouse* ocorra de forma irregular.

4.8.4. ALGORITMO DE CLARCK

Outro problema que pode afetar o desempenho do protocolo TCP é a síndrome da *silly window* [CLA 82]. Este problema acontece quando a entidade emissora envia grandes quantidades de dados mas no lado da entidade receptora, a aplicação responsável pelo processamento de tais informações lê apenas 1 *byte* de cada vez. Na situação inicial, a entidade receptora está com o seu *buffer* vazio, que rapidamente é preenchido pela entidade emissora. Deste momento em diante, a aplicação no lado da entidade receptora retira apenas 1 *byte* por vez do *buffer* e a cada leitura, a entidade receptora encarrega-se de avisar a entidade emissora que existe 1 *byte* livre no seu *buffer*. A entidade emissora envia mais 1 *byte* e a entidade receptora envia a confirmação do recebimento informando que o seu *buffer* está cheio novamente. Este esquema é apresentado na Figura 4.17.

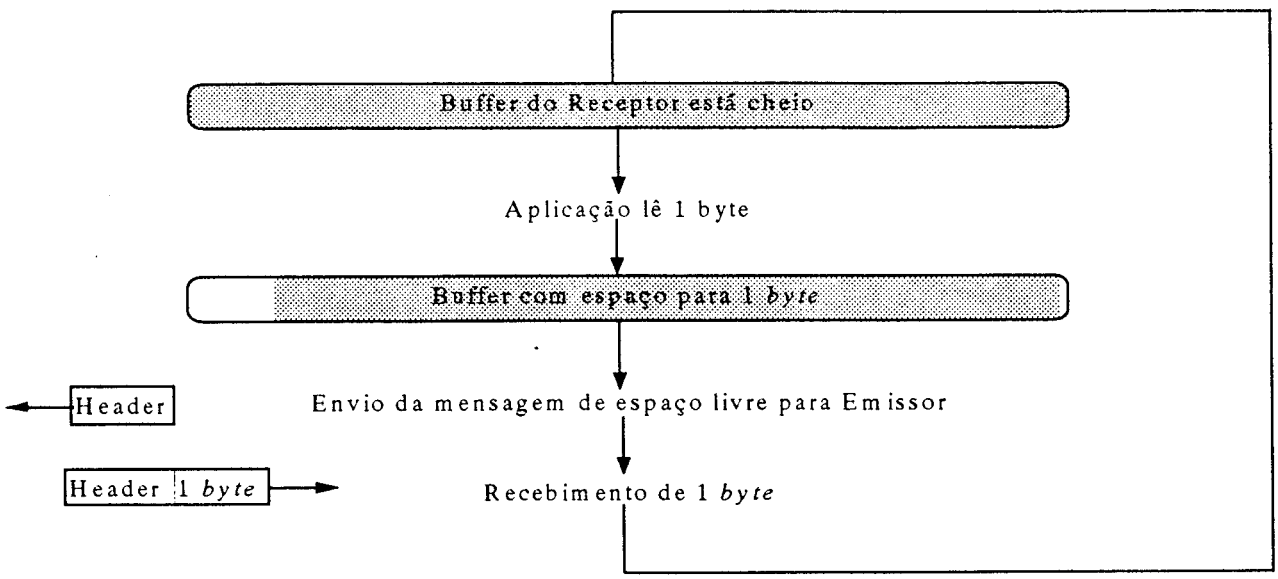


Figura 4.17 - Síndrome da *Silly Window*

A solução proposta por Clark sugere que a entidade receptora informe a existência de espaço livre em seu *buffer* à entidade emissora apenas quando este espaço for a metade do tamanho total do *buffer* ou quando for do tamanho máximo do segmento a ser transferido definido no momento do estabelecimento da conexão. Além disso, a entidade emissora deve aguardar até ser suficiente para se enviar um segmento cheio ou pelo menos, contendo a metade do tamanho do *buffer* da entidade receptora, a ser estimado baseando-se no padrão do tamanho do espaço livre informado anteriormente.

4.8.5. CONTROLE DE CONGESTIONAMENTO

Segundo [TAN 96], a verdadeira solução para o congestionamento é diminuir a taxa de transmissão de dados. Para se atingir este objetivo, emprega-se o princípio de não inserir um novo pacote na rede até que um pacote tenha sido retirado dela. No protocolo TCP, utiliza-se a manipulação dinâmica do tamanho da janela que informa o espaço livre no *buffer* do receptor.

Primeiro, precisa-se saber como detectá-la. Atualmente, a perda de pacotes durante a transmissão de dados é bastante rara devido a alta qualidade do meio físico. Conseqüentemente, a

UNICAMP
BIBLIOTECA CENTRAL

ocorrência de um *timeout* na transmissão de dados via Internet está relacionada com congestionamento. Todos os algoritmos TCP da Internet assumem que a ocorrência de tal evento está relacionada com congestionamentos.

Visando prevenir o congestionamento, no momento do estabelecimento de uma conexão é escolhido um tamanho razoável de janela a ser utilizado. A entidade receptora pode especificar um valor baseado no tamanho do seu *buffer*. Desta forma, os problemas referentes ao lado do receptor estarão resolvidos mas, mesmo assim, o congestionamento ainda pode ocorrer internamente na rede.

A solução encontrada parte do pressuposto que existem dois problemas em potencial que são tratados separadamente: a capacidade da rede e a do receptor. Cada emissor mantêm duas janelas: a janela que o receptor solicitou e uma segunda chamada de janela de congestionamento. O número de *bytes* que o emissor pode transmitir é o menor de ambos. O algoritmo utilizado chama-se de *slow start*.

Quando uma conexão é estabelecida, o emissor ajusta a janela de congestionamento com o tamanho máximo do segmento utilizado na conexão. Ele, então, envia um segmento com este tamanho, se este segmento é confirmado antes da ocorrência de um estouro no tempo de espera, o tamanho da janela de congestionamento é dobrado e dois segmentos deste novo tamanho são enviados. Este processo continua até que ocorra um *timeout* ou que chegue uma janela do receptor.

O algoritmo de controle de congestionamento Internet utiliza além dos parâmetros anteriores o de *threshold*. Quando um *timeout* ocorre, este parâmetro passa a valer a metade do tamanho atual da janela de congestionamento e esta passa a valer o tamanho máximo do segmento recebido. A seguir, é utilizado o algoritmo de *slow start* para se verificar qual o tamanho máximo que pode ser transmitido agora. O crescimento do tamanho da janela de congestionamento é exponencial até atingir o valor do *threshold* e, deste ponto em diante, passa a ser linear. A Figura 4.18 exemplifica este funcionamento.

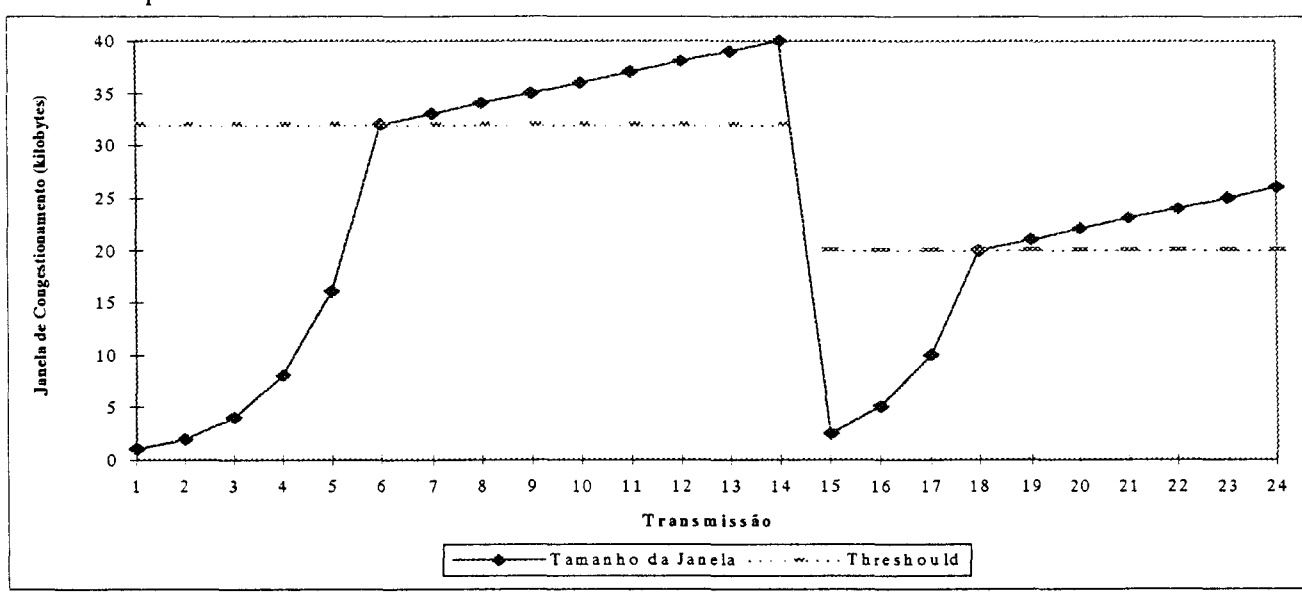


Figura 4.18 - Exemplo do Algoritmo de Congestionamento Internet

4.8.6. ESQUEMA PROPOSTO

Atualmente o controle de congestionamento implementado pelo protocolo TCP penaliza todas as conexões igualmente, independente da quantidade de tráfego gerada ou do tipo de aplicação que a esteja utilizando.

Utilizando-se o nível de informações disponibilizado pelo modelo proposto na seção 4.2, propõe-se que, ao ser detectada uma tendência de degradação do sistema ou a existência de congestionamento, o Sistema de Correções informe à entidade de transporte e esta ative a função de controle de congestionamento baseada em aplicações, penalizando as conexões baseando-se no tráfego gerado por cada aplicação e na prioridade atribuída a cada conexão ou aplicação sendo que esta ação independe da utilização do algoritmo *slow start*. Desta forma, a diminuição de geração do tráfego ocorreria de forma seletiva, favorecendo determinadas aplicações.

4.8.7. ASPECTOS DE IMPLEMENTAÇÃO

O primeiro ponto a ser tratado é como identificar quais são as conexões que serão penalizadas ou privilegiadas. Destacam-se aqui duas possibilidades:

- quando da solicitação de alocação de uma porta, o processo deverá especificar qual é a prioridade que deseja para a sua conexão, acarretando a inclusão de mais um parâmetro na atual primitiva de criação de conexão. Esta alternativa seria recomendável para as portas sem aplicações definidas para elas;
- a utilização de um arquivo de configuração no qual seriam especificados tanto números específicos quanto intervalos de portas e a prioridade associada a estas portas, o qual seria configurado pelo administrador da rede. Esta alternativa seria recomendável para as portas cuja utilização já está reservada a determinadas aplicações chamadas de *well-know ports*.

A utilização de uma destas alternativas não implica na exclusão da outra, porém, em se tratando de uma mesma porta definida em ambas, deveria prevalecer a prioridade definida no arquivo de configuração por se tratar de um esquema de prioridade de responsabilidade do administrador do sistema.

Duas seriam as formas de penalização a serem aplicadas às conexões:

- aumentar o intervalo entre leituras/escritas no *buffer* ou suspendê-las temporariamente o que levaria a uma diminuição do tráfego gerado pela aplicação. O atraso/suspensão poderia se dar até por um período do tempo pré-determinado, até que a situação de degradação estivesse resolvida, ou até que uma determinada quantidade do *buffer* estivesse livre para receber dados (no caso de processos clientes) ou para a escrita (no caso de processos servidores). Esta solução bastante simples seria implementada internamente no processo que controla o fluxo de informações entre a camada de transporte e aplicação.
- estabelecer prioridades diferentes para o processamento dos processos responsáveis pelas conexões. Esta solução poderia ser implementada utilizando-se os recursos de *threads*, disponibilizados em diversos sistemas operacionais, como apresentado na Figura 4.19. O controle da prioridade de cada processo poderia ser feito diretamente por cada um ou por um processo dedicado a esta tarefa.

A ativação de ambas estaria vinculada ao recebimento de um sinal que poderia ser enviado tanto pela entidade de transporte quanto pelo Sistema de Correções. A restrição de acesso ao *buffer* poderia ser aplicada tanto no processo cliente quanto no processo servidor, já o controle de prioridade surtiria um efeito maior quando aplicado no processo que gerar maior tráfego de informação seja ele servidor, no caso de uma aplicação de transferencia de arquivo, cliente ou quando ambos apresentarem taxas elevadas de tráfego.

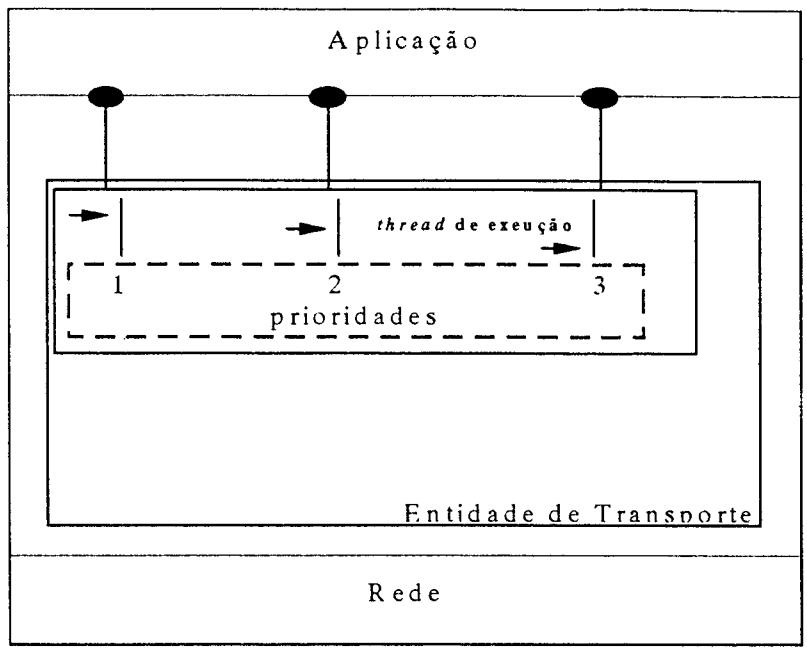


Figura 4.19 - Entidade de Transporte utilizando *Threads*

CAPÍTULO 5

UTILIZAÇÃO DA TÉCNICA DE SÉRIES TEMPORAIS NO GERENCIAMENTO PRÓ-ATIVO

As técnicas de simulação e inteligência artificial são complementares pois a técnica de simulação é utilizada na identificação de situações que levem a degradação e a técnica de inteligência artificial é empregada na identificação das causas do problema e possíveis formas de resolvê-los.

A técnica de inteligência artificial pode ser empregada na atividade de identificação das causas de uma situação que possa levar à degradação do sistema. Entre as pesquisas desenvolvidas nesta área destacam-se [ROC 96, SAE 96a, SAE 96b].

A aplicação da técnica de simulação no gerenciamento pró-ativo tem sido feita na geração da *baseline* da rede, como em [DEF 96a, DEF 96b].

Neste trabalho foi utilizada a técnica de modelagem matemática na atividade de verificação de situações que levem à degradação.

Atualmente, a tarefa de verificação de situações que podem levar a degradação do sistema dá-se através da monitoração de um ou vários parâmetros significativos e da verificação de que estes ultrapassem determinados valores significativos, também chamados de *thresholds*.

Uma nova forma de análise dos dados coletados e de previsão de situações que possam levar a degradação do sistema foi aplicada neste trabalho. Através de uma análise estatística destes dados pretende-se prever a ocorrência de valores que possam degradar o sistema antes que estes ocorram.

A técnica que se pretende utilizar é a de previsão aplicada a séries temporais[3].

Segundo [CHA 84], uma série temporal é uma coleção de observações feitas seqüencialmente no tempo. No contexto de gerência pró-ativa, duas seriam as principais séries temporais: a *baseline* da rede e as informações coletadas em tempo real.

Os valores coletados em tempo real compõem uma série temporal $x_1, x_2, ..., x_N$. O que se pretende é ser capaz de estimar um intervalo onde esteja inserido o valor de x_{N+Q} no tempo N do

[3] Tradução de *Time Series*

valor Q vezes à frente. Serão apresentados a seguir os estudos desenvolvidos sobre a teoria de séries temporais.

5.1. Fatores A Serem Considerados Na Escolha De Uma Técnica De Simulação

Segundo [BOW 87], a escolha de uma técnica de previsão deve levar em consideração os seguintes fatores:

1. **forma de previsão**. Basicamente esta pode ser feita de duas formas: previsão de um ponto ou de um intervalo;
2. **intervalo de tempo**. Identifica a granularidade de tempo que se deseja trabalhar já que as previsões são feitas para pontos no tempo;
3. **padrão dos dados**. Identificação de padrões, tendências e sazonalidades existentes nos dados;
4. **custo de previsão**. Dentre os fatores que devem ser levados em consideração, destacam-se: custo de desenvolvimento do modelo, o custo do armazenamento das informações necessárias à previsão e o custo do cálculo da previsão;
5. **precisão desejada**;
6. **disponibilidade de dados**;
7. **facilidade de operação e compreensão**.

Foram estabelecidas para este trabalho:

1. **forma de previsão**. A previsão de um intervalo de confiança onde o próximo valor a ser observado estará contido;
2. **intervalo de tempo**. Um intervalo de tempo de 30 minutos tanto para a coleta dos dados quanto para a previsão;
3. **padrão dos dados**. A análise dos dados e conhecimentos prévios levaram a identificação de uma sazonalidade na série temporal analisada entre as amostras;
4. **custo de previsão**. Por ser uma nova forma de aplicação desta técnica no campo do gerenciamento pró-ativo, o custo de desenvolvimento do modelo foi um pouco alto devido principalmente aos estudos realizados no campo estatístico. O custo de armazenamento das informações necessárias a previsão e o custo do cálculo da previsão podem ser desprezados;
5. **precisão desejada**. Trabalha-se com um intervalo de confiança de 95%, ou seja, 95% dos intervalos dos valores futuros previstos estarão corretos e que 5% destes intervalos previstos estarão errados;
6. **disponibilidade de dados**. A disponibilidade dos dados foi atendida com a utilização do agente Beholder e de programas desenvolvidos para a geração de informações sobre o nível de aplicação.

5.2. Estratégia de Modelagem

A metodologia de previsão, que está sendo aplicada foi desenvolvida por G. E. P. Box e G. M. Jenkins [BOX 70], consiste de quatro passos:

1. **Tentativa de Identificação**. Tentativa de identificação de um modelo através do emprego da função de auto-correlação da amostra (SACF - *Sample Autocorrelation Function)* e da

Função de Auto-correlação Parcial da Amostra (SPACF - *Sample Partial Autocorrelation Function*). Além disso, procura-se neste passo a compreensão do comportamento da série em estudo através de suas sazonalidades (horários onde os valores observados costumam elevar ou diminuir), tendências e situações anormais que possam ser identificadas tais como problemas nos equipamentos e na coleta dos dados.

2. **Estimativa**. Uma vez que o modelo tenha sido identificado, passa-se à estimativa dos parâmetros. Neste trabalho, esta tarefa foi feita com a utilização da linguagem de programação S+;
3. **Checagem de Diagnóstico**. É feita uma verificação das estimativas realizadas com o modelo especificado, através da análise dos resíduos associados ao modelo, ou seja, a diferença entre o comportamento da série observada e do modelo definido, além da utilização de testes de adequação do modelo especificado. Se o modelo se mostrar inadequado ele deve ser modificado e aprimorado.
4. **Previsão**. Quando o modelo já foi modificado e aprimorado, passa-se a aplicá-lo para a previsão de valores futuros. Esta etapa foi implementada também utilizando-se a linguagem de programação S+.

Esta metodologia foi aplicada neste trabalho por poder ser utilizada tanto em dados discretos quanto contínuos desde que as medidas dos dados contínuos sejam feitas em intervalos iguais e discretos. Pode ser aplicada também tanto sobre dados sazonais quanto não sazonais.

A linguagem S+ utilizada neste trabalho é voltada para o campo da estatística e possui uma enorme gama de funções embutidas. Sua utilização pode ser feita tanto de forma interativa quanto através de programas do tipo *batch* [EVE 94, SPE 94].

5.3. Conhecimentos Básicos para a Identificação de um Modelo

Durante o primeiro passo da estratégia de Box e Jenkins, procura-se identificar a forma de funcionamento da série temporal que se deseja modelar, identificando tendências, sazonalidades, fatos excepcionais ocorridos durante a coleta dos dados, etc. No campo do gerenciamento de redes, esta tarefa seria representada pela identificação de horários de pico, períodos onde a variável observada apresentou valores maiores/menores, etc. Estas informações auxiliam na escolha de um modelo que represente o comportamento da série temporal que está sendo estudada.

A identificação de um modelo para a representação do processo é feita através da comparação dos gráficos gerados pelas funções de auto-correlação (ACF) e auto-correlação parcial (PACF) (vide seções 5.3.3.1 e 5.3.3.2) da amostra com os das funções teóricas. Tenta-se identificar os gráficos das funções teóricas que mais se assemelham aos da amostra.

Os conceitos teóricos necessários para a modelagem de uma série temporal utilizando-se o modelo ARIMA (*Autoregressive Integrated Moving Average*) (vide seção 5.3.1) é apresentado a seguir. A modelagem ARIMA foi escolhida devido à simplicidade que a mesma apresenta em relação as outras modelagens existentes de séries temporais. Outra forma de modelagem de séries temporais é modelagem baseada em modelos de regressão dinâmica também conhecidos como modelos DR (*Dinamic Regression*).

5.3.1. MODELO ARIMA (AUTOREGRESSIVE INTEGRATED MOVING AVERAGE)

Uma equação deste modelo define um valor em uma série temporal como sendo linearmente relacionado ao seu próprio valor passado, dando-se maior importância aos valores mais recentes, ou seja, tenta-se verificar a influência dos valores anteriormente observados no valor atual de uma determinada variável monitorada. Qualquer previsão de um modelo deste tipo é uma média ponderada dos valores anteriormente observados.

Um processo ARIMA equivale ao mecanismo da população que gerou a série observada. Um modelo é baseado nos dados amostrados, no caso, as diversas variáveis que compõem a *baseline*. Espera-se que o modelo ARIMA construído seja uma aproximação útil dos processos verdadeiros mas não observáveis. Se o modelo é uma boa aproximação do processo, o modelo tende a imitar o comportamento do processo, desta forma as previsões deste fornecerão informações úteis sobre os valores da série, permitindo a identificação de situações que levem à degradação do sistema antes que estas ocorram.

No campo de monitoração de redes, este modelo assume que o valor de uma variável observado no momento t está relacionado aos valores desta variável observados anteriormente a este momento. Por exemplo, tomando-se a variável de utilização do segmento, assume-se que o valor de utilização do segmento em uma determinada hora h está relacionado com o nível de utilização à hora $h-1$ ou à mesma hora no dia anterior. A previsão de um intervalo onde o valor da variável de utilização do segmento estará localizado à hora $h+t$ é uma média ponderada dos valores observados nos horários anteriores.

5.3.2. MÉDIA DE VARIÂNCIA ESTACIONÁRIAS

A análise padrão ARIMA baseia-se na simplificação de que o processo que gerou a série a ser analisada é estacionário, ou seja, é um processo cuja média, variância e função de auto-correlação são constantes no tempo. Esta simplificação é de ordem prática pois permite o desenvolvimento de uma estrutura teórica relativamente simples e de estatísticas úteis [PAN 91]. Se a série observada não for estacionária, pode-se modificá-la para torná-la estacionária. Como as modificações são conhecidas, estas podem ser revertidas para que as previsões possam ser comparadas aos valores originais da série.

A verificação de cumprimento ou não deste requisito costuma ser feita através de observação tanto do gráfico da série propriamente dita quanto dos gráficos da função de auto-correlação (SACF) e de auto-correlação parcial (SPACF) da série em estudo.

Serão apresentadas a seguir as técnicas utilizadas para a estabilização da variância e da média de uma série.

5.3.2.1. ESTABILIZAÇÃO DA VARIÂNCIA

A indução de uma variância constante é feita através da transformação dos dados. Duas são as principais transformações a serem aplicadas:

1. Se o desvio padrão de uma série é proporcional ao seu nível, a utilização do logaritmo natural levará a uma série com a característica desejada;
2. Se a variância da série original é proporcional ao seu nível, a utilização da raiz quadrada levará a uma série com a característica desejada.

5.3.2.2. Estabilização da Média

Serão apresentadas as seguir, as técnicas empregadas na estabilização da média de uma série.

5.3.2.2.1. Diferenciação não Sazonal

Uma série z_t precisa estabilizar sua média quando ela não oscila em volta de uma média constante. Para que a sua média seja considerada estacionária, a série deve retornar muito rápida para uma média constante. Normalmente, quando isto não ocorre, cria-se uma nova série com média constante através do cálculo das mudanças sucessivas na série para todas as amostras, como apresentado na equação 5.1, chamada de primeira diferença não sazonal pois os valores utilizados são separados por apenas um período de tempo.

$$w_t = z_t - z_{t-1} \tag{5.1}$$

Se a série resultante w_t ainda não apresentar uma média geral constante, calcula-se a primeira diferença da primeira diferença para todo t, também chamada de segunda diferença não sazonal, (equação 5.2).

$$\begin{aligned} w_t &= w^*_t - w^*_{t-1} \\ &= (z_t - z_{t-1}) - (z_{t-1} - z_{t-2}) \end{aligned} \tag{5.2}$$

Seja d o grau da diferença não sazonal, então teremos $d = 1$ para a primeira diferença não sazonal, $d = 2$ para a segunda diferença não sazonal e assim, sucessivamente.

5.3.2.2.2. Diferenciação Sazonal

Através da observação do gráfico da série ou de conhecimentos prévios pode-se constatar a existência de variações periódicas ou sazonais: o nível de uma observação (w_t) é similar ou relacionado ao nível de outras observações que estão separadas dela por $s = 48$ períodos de tempo (w_{t-48},w_{t+48}), $2s$ períodos de tempo (w_{t-96},w_{t+96}), e assim por diante. Um exemplo de sazonalidade seriam os horários de expediente. Uma série pode ser diferenciada apenas não sazonalmente, apenas sazonalmente ou ambas. Seja D o grau de diferença sazonal, ou seja, o número de vezes que a diferença sazonal foi aplicada. Se d=0, uma série diferenciada sazonalmente (D=1) será da forma apresentada pela equação 5.3.

$$w_t = z_t - z_{t-s} \tag{5.3}$$

5.3.3. Auto-correlação

Na análise de séries temporais, procura-se identificar os padrões de auto-correlação dos dados: de que forma o valor corrente da série (z_t) está relacionado aos seus valores antecessores (z_{t-1}, z_{t-2}, z_{t-3},...) ou com seus valores futuros (z_{t+1}, z_{t+2}, z_{t+3},...). Esta auto-correlação mede a direção (positiva ou negativa) e a intensidade do relacionamento em uma única série temporal z_t

quando as observações são separadas por k períodos de tempo, para k=1,2,3,..,K. Neste caso, as séries z_t e z_{t+k} são tratadas como variáveis aleatórias.

5.3.3.1. FUNÇÃO DE AUTO-CORRELAÇAO DA AMOSTRA (SACF)

Dada a série temporal: z_1, z_2, ..., z_K. A função de auto-correlação da amostra no intervalo k (r_k) é:

$$r_k = \frac{\sum_{t=1}^{n-k}(z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=1}^{n}(z_t - \bar{z})^2} \tag{5.4}$$

$$\bar{z} = \frac{\sum_{t=1}^{n} z_t}{n} \tag{5.5}$$

Esta função mede o relacionamento linear entre as séries temporais observadas separadas por um intervalo de k unidades de tempo. Pode ser provado que r_k sempre terá um valor entre 1 e -1. Um valor de r_k próximo a 1 indica que as observações separadas por um intervalo de k unidades de tempo possuem uma forte tendência a se moverem juntas, ou seja, se a observação no tempo t está acima da média da série, a observação no tempo $t+k$ tende a estar também acima da média, enquanto um valor de r_k próximo a -1 indica que as observações possuem uma forte tendência a se moverem juntas de forma linear com uma inclinação negativa, ou se a observação no tempo t estiver acima da média, não existe a tendência da observação no tempo $t+k$ também estar acima da média. Em computação poderíamos interpretar este parâmetro como sendo a representação da tendência ou não de que se o nível de utilização do segmento monitorado no horário t está acima da média, o nível de utilização no horário $t+k$ também estará acima da média.

A SACF também é utilizada na verificação da estacionariedade de uma determinada série temporal, segundo [BOW 87]:

1. Se a SACF dos valores de uma série temporal desaparecem muito rapidamente ou decrescem muito rapidamente, então estes valores devem ser considerados estacionários.
2. Se a SACF dos valores de uma série temporal decrescem extremamente devagar, então estes valores devem ser considerados não estacionários.

5.3.3.2. FUNÇÃO DE AUTO-CORRELAÇÃO PARCIAL DA AMOSTRA (SPACF)

A função de auto-correlação parcial da amostra de intervalo k é:

$$r_{kk} = \begin{cases} r_1 & k = 1 \\ \dfrac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j} & \forall k \geq 2 \end{cases} \qquad r_{kj} = r_{k-1,j} - r_{kk} r_{k-1,k-j} \text{ para } j = 1, 2, ..., k-1 \tag{5.6}$$

No contexto de séries temporais, uma boa parte da correlação entre as observações z_t e z_{t-k} pode estar relacionada à correlação que estas variáveis possuem com as observações z_{t-1}, z_{t-2}, ...,

z_{t-k+1}. A auto-correlação parcial de intervalo k pode ser interpretada como sendo o coeficiente de regressão parcial ϕ_{kk} da seguinte representação: $z_t = \phi_{k1}z_{t-1} + \ldots + \phi_{kk}z_{t-k} + a_t$, que mede a correlação adicional entre z_t e z_{t-k} após terem sido feitos os ajustes para as variáveis intermediárias $z_{t-1} + \ldots + \phi_{kk}z_{t-k+1}$.

5.3.3.3. PROCESSOS ARIMA ESTACIONÁRIOS

Um modelo ARIMA é baseado nos dados coletados e sua representação teórica é um processo ARIMA que possui o seu próprio ACF e PACF associados.

A identificação de um modelo segue os seguintes passos:
1. Calcula-se o ACF e PACF da série temporal formada pelos dados disponíveis;
2. Compara-se o ACF e o PACF da série temporal da amostra com os ACFs e PACFs teóricos até que se encontre uma semelhança aceitável entre ambos;
3. Seleciona-se o processo que possui os ACFs e PACFs semelhantes como um modelo experimental ARIMA.

Serão apresentados a seguir, alguns processos estacionários ARIMA mais comuns e seus respectivos ACFs e PACFs.

5.3.3.3.1. PROCESSO AUTOREGRESSIVO

Dois processos ARIMA comuns e bastante simples são apresentados nas equações 5.7 e 5.8.

$$z_t = C + \phi_1 z_{t-1} + a_t \tag{5.7}$$

$$z_t = C + \phi_1 z_{t-1} + \phi_2 z_{t-2} + a_t \tag{5.8}$$

$$C = \mu_z \left(1 - \sum_{i=1}^{p} \phi_i\right) \tag{5.9}$$

Nestes processos, os valores de z passados com os seus coeficientes são chamados termos autoregressivos (AR). O termo C é uma constante que relaciona a média (μ_z) da série z_t e os coeficientes AR. A ordem (p) de um processo AR é o comprimento do maior intervalo entre os termos, por exemplo, a equação 5.7 é um processo AR de primeira ordem, designado por AR(1), enquanto a equação 5.8 representa um processo AR(2).

As ACFs e PACFs teóricas associadas aos processos AR(1) e AR(2) são apresentadas nas Figuras 5.1 e 5.2. Seu comportamento pode ser resumido da seguinte forma:
1. A ACF teórica decai exponencialmente, com o padrão de uma onda senoidal amortecida ou ambos;
2. A PACF teórica possui saltos até o intervalo p e em seguida torna-se zero.

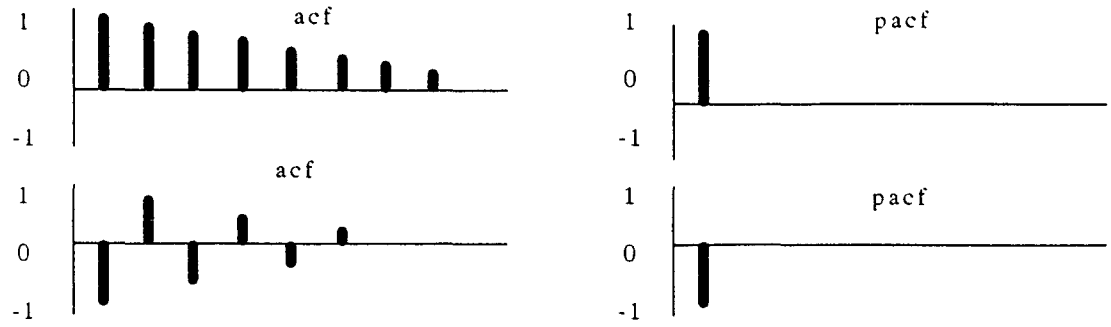


Figura 5.1 - ACFs e PACFs teóricos de AR(1)

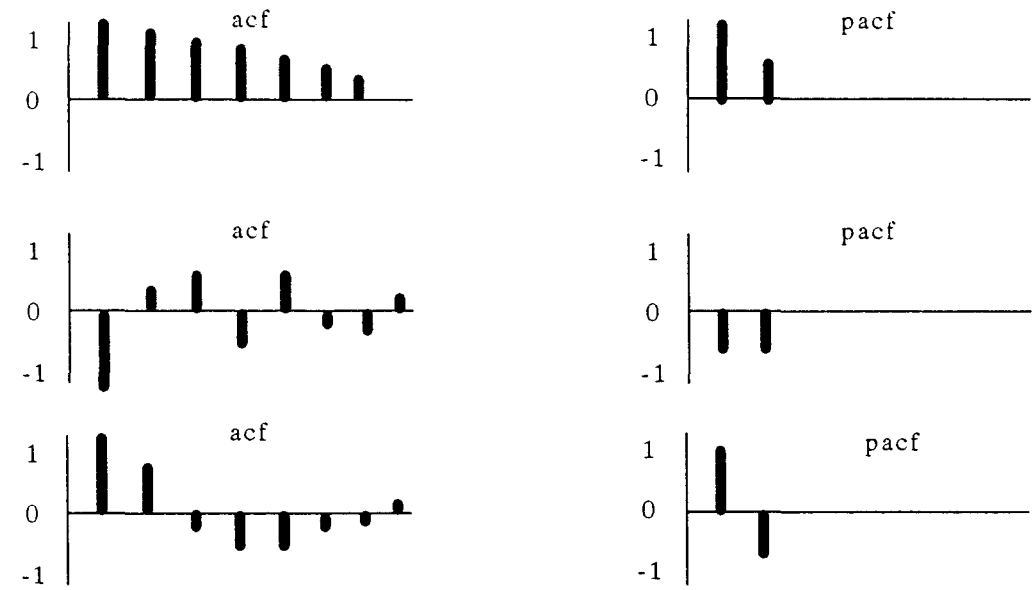


Figura 5.2 - ACFs e PACFs teóricos de AR(2)

No campo de gerenciamento de redes, estes processos poderiam ser interpretados como a representação da influência que o valor de uma variável coletado no tempo *t-p* influencia no valor observado no tempo *t*.

5.3.3.4. MÉDIAS MÓVEIS E PROCESSOS MISTOS

As equações 5.10, 5.11 e 5.12 apresentam os processos MA(1), MA(2) e ARMA(1,1). Nestes, o distúrbio aleatório passado (a_t) juntamente com os seus coeficientes são chamados termos de médias móveis (MA - *Moving Average*) que, por convenção, são escritos com sinais negativos. Merece destaque o fato de que um distúrbio aleatório passado a_{t-k} não é um valor passado de z_t, mas sim um distúrbio aleatório componente de z_{t-k}, logo um termo MA representa uma parte do valor passado de z_t. A ordem (q) deste processo é o tamanho do maior tamanho de intervalo nos seus termos. O processo exemplificado na equação 5.10 é um processo de primeira ordem, denotado por MA(1), já o apresentado na equação 5.11 é de segunda ordem, denotado por MA(2).

$$z_t = C - \theta_1 a_{t-1} + a_t \quad (5.10)$$

$$z_t = C - \theta_1 a_{t-1} - \theta_2 a_{t-2} + a_t \quad (5.11)$$

$$z_t = C - \phi_1 z_{t-1} - \theta_1 a_{t-1} + a_t \quad (5.12)$$

Um processo que combine os processos AR e MA é conhecido como um processo do tipo ARMA e chama-se média móvel autoregressiva. Quando as ordens deste processo são $p = 1$ e $q = 1$ têm-se um processo ARMA(1,1).

As ACFs e PACFs teóricas associados aos processos MA(1), MA(2) e ARMA(1,1) são apresentados nas Figuras 5.3, 5.4 e 5.5. Seu comportamento pode ser resumido da seguinte forma:

1. A ACF teórica de um processo MA possui saltos até o intervalo q e em seguida torna-se zero;
2. A PACF teórica de um processo MA decresce;
3. Ambos ACF e PACF teóricos de um processo misto decrescem.

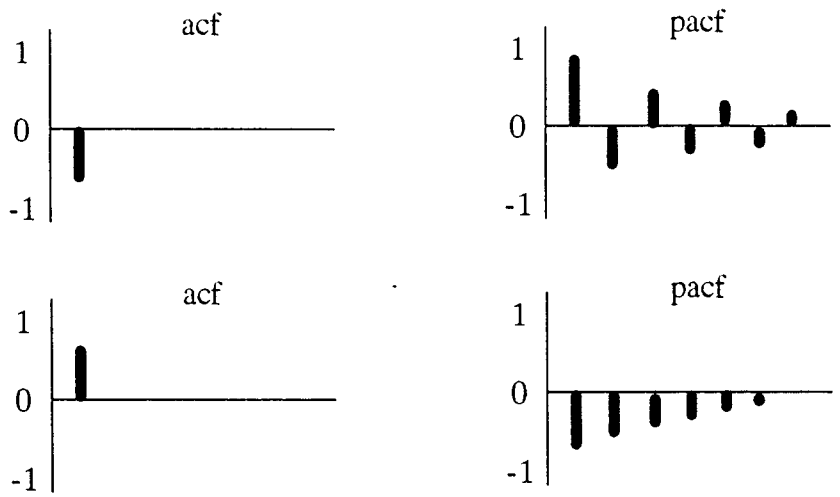


Figura 5.3 - ACFs e PACFs teóricos do Processo MA(1)

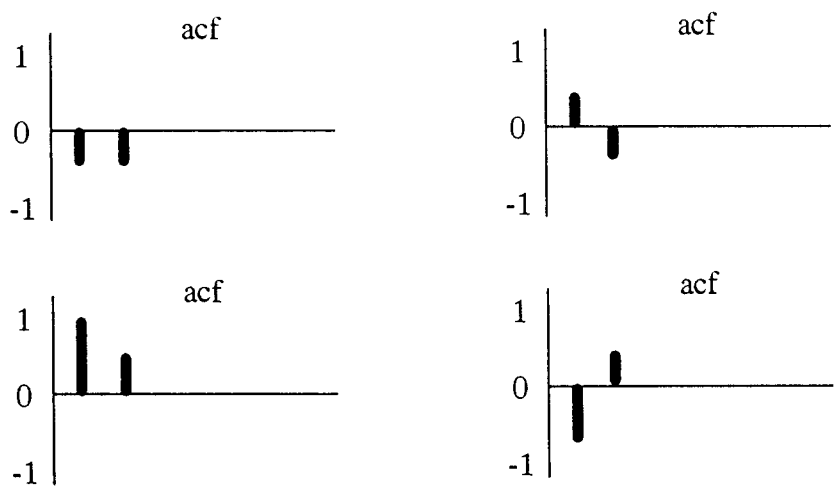


Figura 5.4 - ACFs teóricos do Processo MA(2)

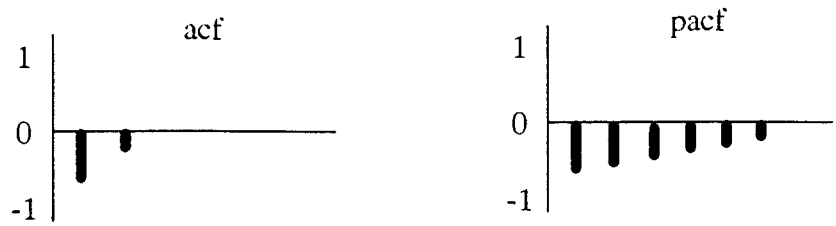


Figura 5.5 - ACF e PACF teóricos do Processo ARIMA(1,1,1)

No campo do gerenciamento de redes, os processos MA poderiam ser interpretados como uma representação do relacionamento existente entre o erro aleatório associado ao valor coletado no tempo t-q e ao associado ao valor coletado no tempo t.

5.3.3.5. PROCESSO INTEGRADOS

Todas as equações apresentadas até aqui são a representação de processos aplicáveis a séries temporais que possuem média estacionária. Para se aplicar tais conceitos a processos que não possuem tal característica, a única diferença é que o modelo será escrito para a série diferenciada gerada a partir da série original. Por exemplo, para um processo cujo $d = 1$ e $D = 0$ e cuja série diferenciada w_t seja descrita por um processo ARMA(1,1), isto gerará um processo ARIMA(p,d,q)=ARIMA(1,1,1) para a série original z_t .

5.3.4.MODELOS SAZONAIS

Os modelos ARIMA podem também ser aplicados na representação de padrões periódicos. As ordens de um processo sazonal ARIMA puro são expressas da forma ARIMA(P, D, Q)$_s$, onde Ps é o tamanho do maior intervalo dos termos AR sazonais, Qs é o tamanho do maior intervalo dos termos MA sazonais, D é o número de diferenciações sazonais realizadas na série original e s é a sazonalidade da série. Um dos modelos sazonais mais comuns existentes é o ARIMA(0,1,1)$_s$ que é apresentado na equação 5.13.

$$z_t = z_{t-s} + C - \theta_s a_{t-s} + a_t \tag{5.13}$$

Apesar dos padrões sazonais e não sazonais normalmente ocorrerem juntos na prática, é interessante procurar-se efetuar uma análise separada de ambos. As ACF e PACF de um processo teórico puramente sazonal de ordens P e Q são idênticas às funções ACF e PACF dos processo não sazonais de mesma ordem com uma exceção: para um processo puramente sazonal os padrões ocorrem nos intervalos s, 2s, 3s, ... Logo, as ACF e PACF de um processo puramente sazonais podem ser caracterizadas da seguinte forma:

1. A ACF decresce vagarosamente nos múltiplos do intervalo s, caso a diferença sazonal seja necessária;
2. Em um processo AR puramente sazonal estacionário de ordem P, a ACF decresce nos múltiplos do intervalo s enquanto a PACF possui saltos nos múltiplos do intervalo s e em seguida torna-se zero;
3. Em um processo MA puramente sazonal estacionário de ordem Q, a ACF apresenta saltos nos múltiplos do intervalo s e então torna-se zero, sendo que o último salto não nulo ocorre no intervalo Qs. A sua PACF decresce nos múltiplos do intervalo s;

4. Em um processo misto puramente sazonal de ordens P e Q, ambas as ACF e PACF decaem nos múltiplos do intervalo s.

Para o gerenciamento de redes, os modelos sazonais poderiam ser interpretados como sendo a representação de sazonalidades existentes nos dados coletados. Tais sazonalidades poderiam ser os diversos períodos de um dia de trabalho, os dias da semana ou os meses do ano.

5.3.5. NOTAÇÃO BACKSHIFT

Visando-se a simplificação na escrita de processos e modelos ARIMA, é utilizado o operador B^i da seguinte forma: quando este operador multiplica qualquer variável relacionada a tempo, este tempo é atrasado por i períodos, por exemplo:

- $B^i z_t = z_{t-i}$;
- $B^i a_t = a_{t-i}$;
- $B^i C = C$.

Esta notação também é empregada na representação de séries diferenciadas, por exemplo uma série diferenciada d vezes não sazonalmente e D vezes sazonalmente com intervalos de tamanho S é escrita: $w_t = (1-B^s)^D(1-B)^d z_t$.

Outros operadores também utilizados são:

1. $\nabla^d = (1-B)^d$ (operador de diferença de ordem d);
2. $\nabla^D{}_S = (1-B^S)^D$ (operador de diferença de ordem D e intervalo S);
3. $\phi(B) = (1 - \phi_1 B - \phi_2 B - ... - \phi_p B)$ (operador AR de ordem p);
4. $\theta(B) = (1 - \theta_1 B - \theta_2 B - ... - \theta_p B)$ (operador MA de ordem q).

5.3.6. PROCESSOS SAZONAIS E NÃO SAZONAIS COMBINADOS

Um processo ou modelo ARIMA pode ter elementos sazonais e não sazonais ao mesmo tempo, resultando em um processo ARIMA(p, d, q)(P, D, Q)s como mostra a equação 5.14.

$$\phi_{i,S}(B^S)\phi(B)\nabla_s^D\nabla^d z_t = C + \theta(B^S)\theta(B)a_t \tag{5.14}$$

$$C = \mu\left(1 - \sum_{i=1}^{p}\phi_i\right)\left(1 - \sum_{i=1}^{p}\phi_{i.S}\right) \tag{5.15}$$

Se d=D=0, então $\mu = \mu_z$; caso contrário $\mu = \mu_w$, ou seja, a média da série diferenciada $w_t = \nabla_s^D\nabla^d z_t$.

5.4. VERIFICAÇÃO DO MODELO

O passo seguinte à estimativa dos parâmetros de um modelo ARIMA é verificar se as suposições do modelo são satisfeitas e caso isto não ocorra, o modelo deve ser especificado novamente.

Uma boa forma de se verificar a adequação de um modelo é analisando-se os resíduos obtidos do modelo calculando a sua função de auto-correlação (RACF) e auto-correlação parcial

(RPACF). Sua análise é equivalente à feita para a função ACF e PACF da série temporal na modelagem, ou seja, comparam-se com os ACF e PACF dos modelos teóricos buscando alguma semelhança e, caso seja identificado algum modelo na série residual, refaz-se o modelo tentando representar mais este comportamento.

5.4.1. TESTE DE PORTMANTEAU

Uma forma de se verificar a adequação do modelo é a utilização da estatística de Ljung-Box, vista na equação 5.16. O processo de modelagem deve estimar o relacionamento entre os valores da série temporal. Quanto mais precisa for esta estimativa, menor será a auto-correlação dos resíduos e menor será o valor de Q^*, que mede esta auto-correlação. Se o valor Q^* calculado para o modelo for maior ao valor especificado na tabela desta variável para grau de liberdade e probabilidade α, o modelo pode ser considerado inadequado. A tabela desta estatística pode ser encontrada em [PAN 91].

$$Q^* = n'(n'+2)\sum_{l=1}^{K}(n'-l)^{-1} r_l^2(\hat{a}) \tag{5.16}$$

$$n' = n - (d+D) \tag{5.17}$$

5.4.2. IDENTIFICAÇÃO DE MODELOS APRIMORADOS

De uma forma geral, se a estatística Q^* indica que a adequação do modelo que descreve uma determinada série temporal deve ser rejeitada, deve-se utilizar o comportamento da RSAC e da RSPAC a nível sazonal e não sazonal para se tentar identificar um modelo que descreva este resíduos, utilizando-se os conceitos apresentados na seção 5.3. Em seguida são combinados o modelo que descreve o comportamento dos resíduos com o modelo que descreve a série original gerando um novo modelo que descreve o comportamento da série original. Efetua-se o teste de adequação a este novo modelo.

Exemplificando o que foi descrito acima, suponha que o modelo $z_t = C + (1 - \theta_{1,12}B_{12})a_t$ (a) foi rejeitado pela estatística Q^*. Isto implica que os componentes aleatórios não são estatisticamente independentes. Substitui-se em (a) a_t por η_t . Passa-se então a identificação de η_t resultando na série $z_t = C + (1 - \theta_{1,12}B_{12})\eta_t$ (b). Após a análise, constata-se que a série (b) pode ser representada por $(1 - \phi B - \phi_2 B^2 - \phi_3 B^3)$ (c). Chega-se então a um modelo de teste para o resíduo: $(1 - \phi B - \phi_2 B^2 - \phi_3 B^3)\eta_t = a_t$ (d). Combinando-se (b) e (d), obtém-se um novo modelo $(1 - \phi B - \phi_2 B^2 - \phi_3 B^3)z_t = C + (1 - \theta_{1,12}B_{12})a_t$ (e). Em seguida, testa-se a adequação de (e) e constata-se que ele não foi rejeitado pela estatística Q^*.

No campo do gerenciamento de redes, tal procedimento seria responsável pela caracterização de sazonalidades faltantes no modelo inicial. Seria também a tentativa de se modelar a relação entre os valores observados em diferentes períodos.

5.5. CONSIDERAÇÕES SOBRE A APLICAÇÃO DA TÉCNICA DE SÉRIES TEMPORAIS NA ANÁLISE DO COMPORTAMENTO DE UM SEGMENTO

Neste trabalho foram aplicados conceitos de séries temporais, mais especificamente a modelagem ARIMA, na interpretação do comportamento de um segmento monitorado.

Apesar do modelo ARIMA gerar como resultado a média ponderada da série temporal de entrada, em comparação com uma forma de previsão mais simples, por exemplo a média simples da série temporal, ele apresenta a vantagem de permitir que leve-se em consideração várias periodicidades, tais como horários e dias da semana, levando a resultados mais próximos da realidade.

A médio prazo, com a disponibilização de uma maior quantidade de dados e a aplicação de outros conceitos existentes no campo de séries temporais, é possível fazer vários estudos mais aprofundados sobre o comportamento de um segmento como por exemplo a verificação de alteração do padrão de funcionamento do mesmo quando da troca de equipamentos, a verificação de tendências de crescimento.

5.6.Como Aplicar a Técnica de Séries Temporais no Gerenciamento de Redes

Este trabalho propõe que os seguintes passos sejam seguidos para a aplicação de séries temporais na monitoração de redes:

1. Geração da *baseline*;
   Neste passo é feita a coleta de dados, devem ser definidos com que intervalo de tempo serão feitas as coletas de dados e o programa necessário para esta atividade.
2. Análise da *baseline*,
   Após a *baseline* ter sido coletada, deve-se escolher os parâmetros que pretende-se analisar segundo a técnica de séries temporais. Para esta escolha, deve ser levada em consideração a relevância da mesma para o gerenciamento de redes e a aleatoriedade da variável pois quanto maior for a aleatoriedade, menor será a precisão do modelo, podendo chegar ao ponto de ser impossível modelar a variável utilizando-se séries temporais. A geração de gráficos dos dados coletados é peça fundamental nesta fase. Os gráficos são importantes tanto para se verificar a aleatoriedade de uma variável (série temporal) quanto para a verificação de sazonalidades, tendências e períodos. A apresentação de todos os dados coletados em apenas um gráfico pode ajudar na identificação de tendências de crescimento ou de queda da variável ao longo do tempo. A geração de um gráfico para cada dia da semana ou para cada mês ajuda na identificação de sazonalidades e padrões de comportamento da variável como, por exemplo, os dias da semana ou semanas do mês que registram os valores mais altos. Já o gráfico que apresenta vários dias da semana juntos serve para a identificação de periodicidades com uma granularidade menor, como os horários onde aparecem os maiores valores ou o padrão de comportamento da variável durante o dia. O Anexo B apresenta um exemplo destes tipos de gráficos. Estes gráficos também são úteis na identificação da relação entre os valores observados, por exemplo, para verificar se a existência de um valor elevado em um tempo t influencia ou não nos valores futuros.

3. Efetua-se as diferenciações sazonal e não sazonal da série, quando necessário, e a geração dos gráficos da função ACF e PACF da série;
   A geração dos gráficos da função ACF e PACF normalmente é feita com a utilização de um programa de estatística que possui também as facilidades para se fazer a estabilização da média e da variância da série.

4. Identificação de modelo-teste;

   Neste passo é utilizado o conhecimento sobre o comportamento da série adquirido no passo 2. Os processos AR representam a relação entre o valor da variável em um determinado tempo *t* com os seu valores anteriores, ou seja, ele procura representar a tendência de um determinado valor permanecer acima ou abaixo da média em relação ao seus valores anteriores. Já os processo MA representam a correlação existente entre o distúrbio aleatório componente do valor observado no tempo *t* e seus antecessores. Por exemplo, os processos de primeira ordem (AR(1), MA(1), ARMA(1,1)) indicam que o valor da variável no tempo *t* sofre influência apenas do seu valor anterior, no gerenciamento de redes eqüivaleria a dizer que o nível de utilização em uma determinada hora está relacionado apenas com o nível de utilização da hora anterior, ou ainda que o nível de utilização observado agora não influenciará em nada o valor observado daqui a 2 observações. As sazonalidades do processo são representadas também através dos processo AR, MA e da integração de ambos. Neste caso, eles estarão representando a correlação existente entre o valor da variável observado agora e o observado a *s* períodos de tempo. Neste caso, *s* pode representar a correlação entre os horários de um dia ou entre os dias de uma semana. Para o gerenciamento de redes, seria a representação no modelo da relação existente entre os horários de pico ou entre os dias de maior utilização do segmento, por exemplo. Um processo de ordem 1 representa a correlação existente entre um determinado horário de um dia e o respectivo horário no dia anterior mas caracterizando que o mesmo horário a dois dias atrás não teria influencia sobre o atual. As funções ACF e PACF irão auxiliar na identificação do relacionamento entre os valores da série temporal que está sendo analisada. Procura-se comparar os gráficos de ACF e PACF da série observada com os gráficos dos modelos teóricos (AR, MA, ARIMA), buscando identificar os que mais se assemelham. Esta comparação é bastante empírica e muitas vezes pode levar a vários modelos que devem ser testados posteriormente;

5. Estimativa dos valores da função que representa o modelo-teste e a verificação da adequação do modelo;

   Este passo normalmente é feito com o auxílio de um programa estatístico. Com este programa são calculados os valores da equação que representa o modelo especificado e são realizados os testes de adequação do modelo ao processo. Caso o modelo seja rejeitado, indica que existe alguma correlação entre os valores que ainda não foi representada, ou seja, a correlação entre os horários de um dia ou entre os dias da semana não está corretamente adequada. Neste caso o modelo dever ser melhorado através da tentativa de representação da correlação existente entre os diversos valores que compõem o resíduo do modelo de teste.

6. Estimativa de valores futuros.

   Neste passo é feita a estimativa dos valores futuros da série. Sua precisão está relacionada com o nível de adequação do modelo ao processo, ou seja, com o nível de detalhamento que se procurou representar no modelo e com o nível de aleatoriedade da variável.

CAPÍTULO 6

ANÁLISE DA UTILIZAÇÃO DE UM SEGMENTO MONITORADO ATRAVÉS DA TÉCNICA DE SÉRIES TEMPORAIS

Levando em consideração os passos apresentados no Capítulo 5, é apresentada a seguir a aplicação dos estudos teóricos através da análise do nível de utilização de um segmento monitorado utilizando-se a técnica de séries temporais.

A série temporal que será utilizada como base de dados para este estudo terá como dados o nível de utilização de um segmento da rede do Instituto de Computação. As coletas foram feitas em intervalos de 30 minutos durante 39 dias. Os gráficos diários dos dados encontram-se no Anexo B.

6.1. IDENTIFICAÇÃO

A Figura 6.1 apresenta a plotagem da série lida, o ACF e o PACF desta amostra com $d=D=0$. Observa-se que o seu ACF decai vagarosamente para zero, sugerindo que a média geral da série não é estacionária. Isto parece consistente com o comportamento do gráfico da plotagem da série. Faz-se então a primeira diferença não sazonal $w^*_t = \nabla z_t$ ($d=1$) desta série, apresentado na Figura 6.2 juntamente com seu ACF e PACF. Analisando-se o gráfico de plotagem, observa-se que este apresenta uma média global constante, fato este confirmado pelo gráfico ACF que decai para zero bastante acentuadamente, além de uma acentuada variação sazonal.

Com o auxílio dos gráficos diários apresentados no Anexo B, pode-se verificar as seguintes características:

1. Os dias com maior utilização são as segundas, terças, quartas e quintas;
2. As sextas, sábados e domingos a utilização é mais baixa;
3. O horário compreendido entre as 8 horas e as 18 é o que apresenta o comportamento mais estável entre a segunda e quinta, caracterizando-se por uma tendência de crescimento das 8 às 13 horas seguindo de uma tendência de queda até às 23:30;
4. O intervalo de tempo compreendido entre a 0 hora e as 8 horas apresenta um nível de aleatoriedade bem maior que os outros intervalos.

Baseando-se nestes pontos, pode-se constatar que a série que está sendo analisada apresenta alguma sazonalidade levando-se em consideração os diferentes horários do dia e uma sazonalidade um pouco mais fraca em relação aos dias da semana. Estas informações nos fornecem idéia que o modelo poderá apresentar uma parte sazonal e uma não sazonal.

O gráfico da série diferenciada (d=1 e D=0) nos indica a existência de sazonalidades, vindo a confirmar a análise dos dados feita anteriormente. Aplica-se então a diferença sazonal (D = 1). Esta nova série, $w^*_t = \nabla_{48} w^*_t = \nabla_{48} \nabla z_t$ é apresentada na Figura 6.3 juntamente com a função ACF e PACF calculadas para a nova série. Esta apresenta uma sazonalidade bem menor que a série anterior. Conclui-se que não é mais necessário efetuar qualquer outra diferenciação sazonal. Comparando-se os gráficos da função ACF e PACF da série em análise com os apresentados na seção 5, pode-se estimar que a parte não sazonal se assemelha aos de um modelo ARMA(1,1), se tomarmos como interpretação dos gráficos que ambos decaem.

Procura-se então, analisar os gráficos tentando dar mais atenção aos valores sazonais. Mais uma vez, pode-se estimar que eles podem ser representados por um processo AR(2), caracterizado pelo salto no intervalo 48 e 96 do gráfico da função PACF e pela forma de decrescer do gráfico ACF, que sazonalmente não aparenta um comportamento muito forte pois a observação no 48 de -0,03 pode ser considerada irrelevante se comparada com o valor no intervalo 0 que é igual a 1. A sazonalidade com valor 48 é explicada pela periodicidade com que é coletada a amostra: de 30 em 30 minutos ou 48 amostras por dia.

O modelo que representa o comportamento da série observada é ARIMA(1,1,1)$(2,1,0)_{48}$ que é representado na equação 6.1.

$$\phi(B^{48})\phi(B)\nabla_{48}\nabla z_t = C + \theta(B)a_t \qquad (6.1)$$

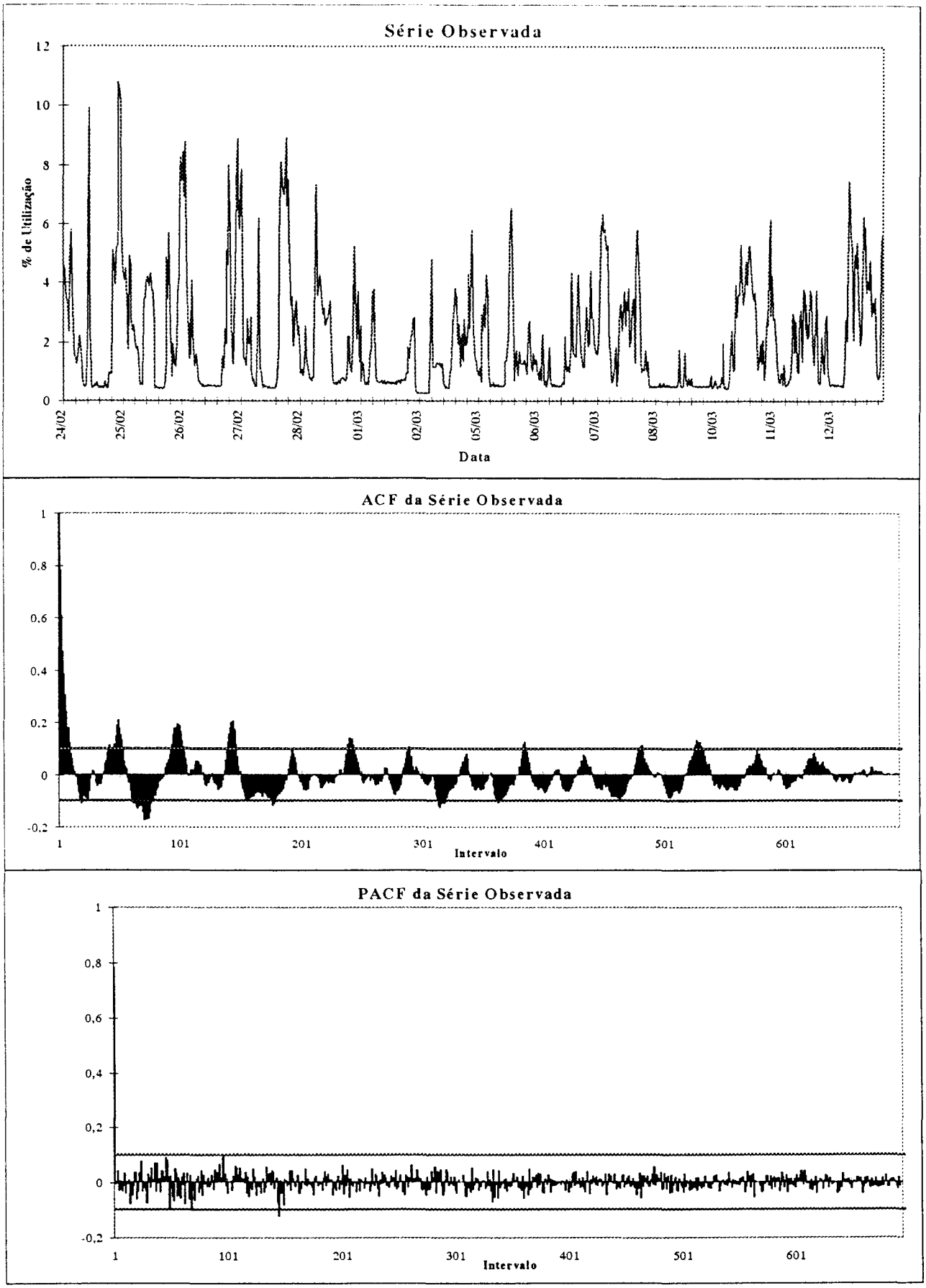


Figura 6.1 - Série Lida, seu ACF e PACF

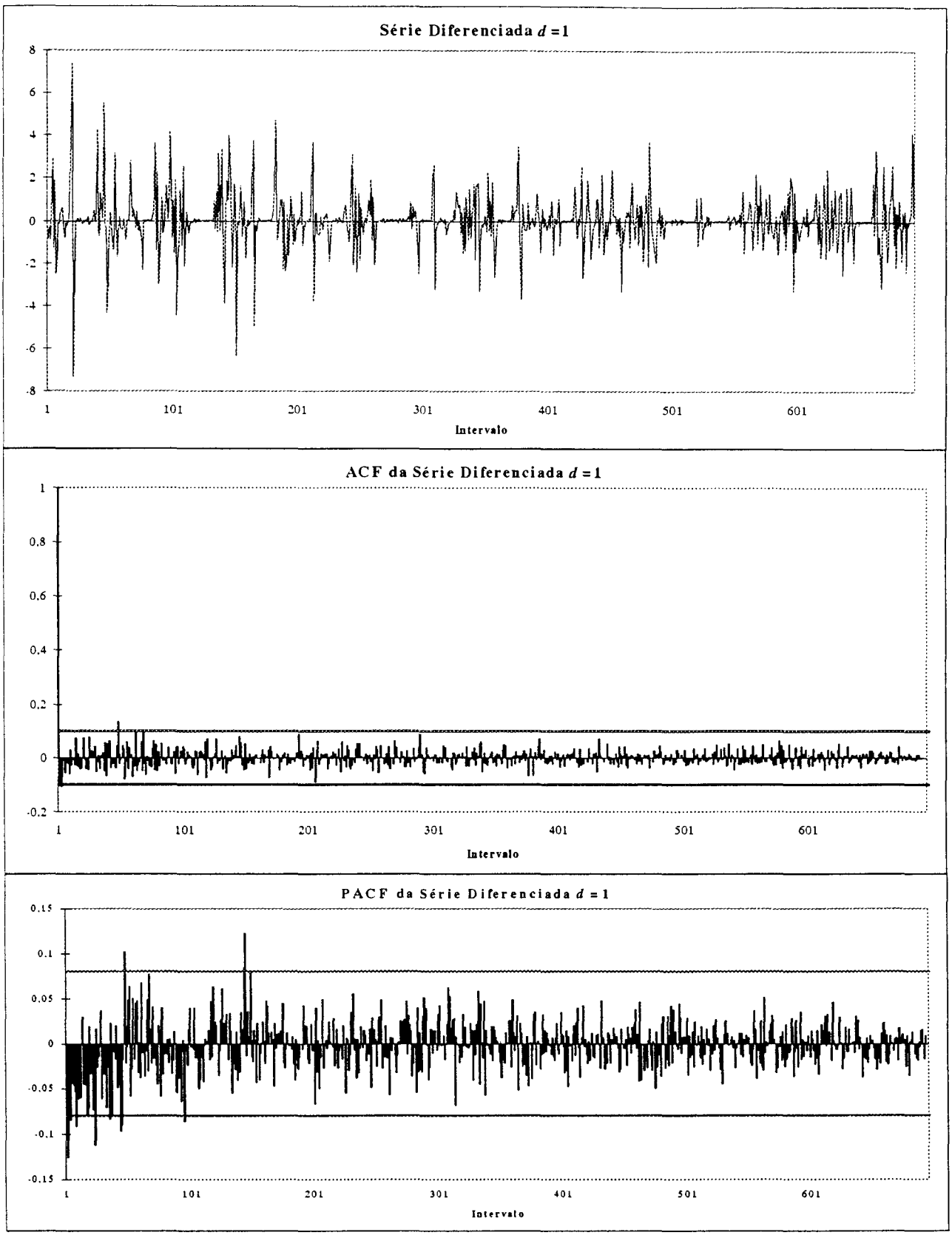


Figura 6.2 - Plotagem da Série com *d*=1, seu ACF e PACF

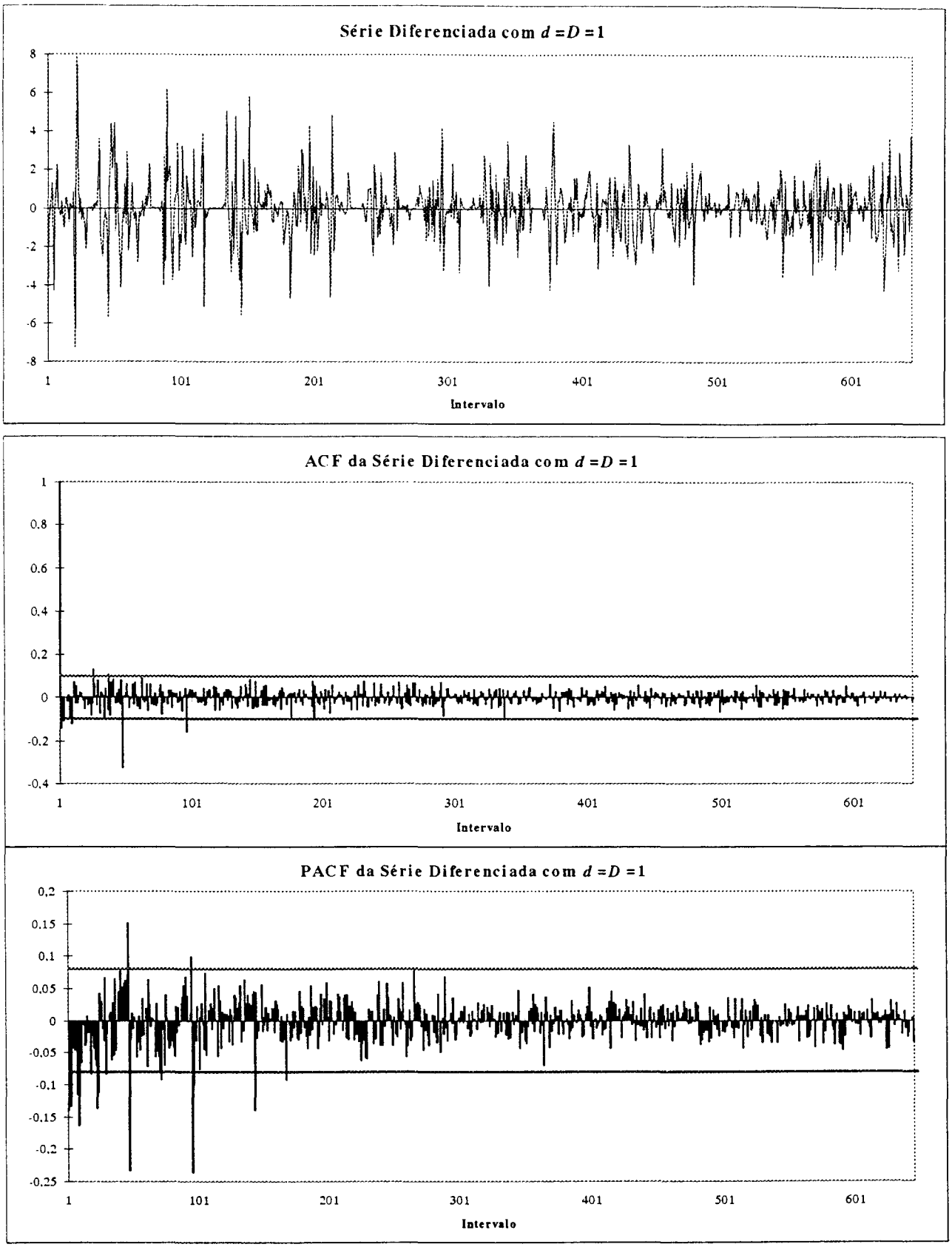


Figura 6.3 - Plotagem da Série com d=D=1, seu ACF e PACF

6.2. Estimativa dos parâmetros e Checagem do Modelo

A estimativa dos parâmetros do processo e a verificação da adequação do modelo foram feitas com a utilização da linguagem de programação para estatística, chamada de S+.

O modelo apresentou estatística $Q^* = 6,9$ para $\alpha=0,5$ (95% de confiabilidade no modelo) e um grau de liberdade de 10, além disso, a plotagem do resíduo do modelo apresentou-se bastante aleatória e a sua função ACF também não apresenta nenhum padrão aparente. Logo, pode-se dizer que o modelo é adequado ao processo.

6.3. Previsão

Esta seção apresenta os valores futuros da série analisada, obtidos através da utilização da linguagem S+. O intervalo de confiança escolhido foi de 95%, ou seja, em média 95% dos intervalos dos valores futuros previstos estarão corretos e que 5% destes intervalos previstos estarão errados. A Tabela VII apresenta o resultado da estimativa dos próximos 20 pontos, ilustrada na Figura 6.4. Analisando-se a plotagem do resíduo contata-se que este apresenta valores bastante baixos e que podem ser desprezados dado que o modelo mostrou-se adequado segundo a estatística Q^*, além disso, a função ACF do resíduo apresenta valores baixos o suficiente param serem desprezados. A plotagem dos valores lidos juntamente com os valores previstos e o intervalo de confiança mostra que o modelo consegue uma representação bastante próxima do padrão de funcionamento da variável em análise: nível crescente no período matutino e decrescente no período vespertino, além dos baixos valores encontrados no período da 0 às 8 horas. As estimativas feitas para o horário das 21 horas à 0 hora apresentaram um maior nível de erro. Merece destaque o fato de que todos os valores observados estavam contidos no intervalo de confiança calculado.

6.4. Considerações sobre os Resultados Obtidos

O modelo apresenta uma boa estimativa de intervalos onde os valores futuros da série temporal poderão estar localizados. Devido ao considerável grau de aleatoriedade apresentado pela variável não apenas com relação aos horários mas também nos diversos dias da semana, a estimativa dos valores futuros não pode ser feita, porém é possível estimar um intervalo onde este valor poderá estar incluído, atendendo à forma de previsão estabelecida neste trabalho (vide seção 5.1). O período das 8 horas às 18 horas foi o que apresentou menor diferença entre os valores estimados e os observados, já o horário das 20 horas até as 6 horas possui um maior erro na estimativa devido a um maior aleatoriedade da variável neste período.

Valor Observado	Valor Calculado	Intervalo de Confiança - 95%
2.88	4.79725118	7.15397991
0.93	3.89187912	6.83033668
0.59	3.55292952	6.78168098
0.48	3.16735202	6.56220226
0.51	2.9783915	6,47766043
0.53	2.61841845	6.18889173
0.57	3.0361933	6.65893646
0.48	2.45709438	6.12081162
0.52	2.41498976	6.11268295
0.49	2.30650491	6.03369923
0.53	2.4241686	6.17792552
0.48	2.37325046	6.15159569
0.51	2.72605109	6.52763058
0.47	2.95862863	6.7824951
0.51	2.70424675	6.54972372
0.47	3.60944631	7.47603964
0.94	3.63949843	7.52683788
2.75	6.22632736	10.1341275
2.03	6.78053154	10.7085653
5.37	6.14308623	10.0911671
7.47	5.97283345	9.94080303
5.92	4,82403849	8.81175829
5.19	4.47697052	8.48431611
2.06	5,75203858	9,77889561
2.38	5,74555869	9.7918202
4,97	6.23668068	10,3022452
4,52	5,66763587	9,75240598
5,36	4.68729203	8,79117366
3,44	5,11770659	9,24060823
1.88	5.94385208	10.0856844
2,27	6,70983023	10.8705057
3,64	5,90107125	10.0805039
6,24	6,39792431	10,5960297
5,12	5,42362384	9,64031884
5,86	5,00848265	9,24368535
3,66	4.13632909	8,38995874
4,04	4,78127898	9,05325595
3,76	4,22280778	8,5130535
4,75	4,22211303	8,53054998
2,86	3,768874	8.09542563
3,31	4.26160375	8.6061945
3,02	2.65066247	7.01321773
3,48	3.08538413	7.46583021
1.08	3,30866095	7,70692506
0,95	3,22541937	7,64142961
0,71	4.43858252	8.87226786
0,8	7.0202393	11,4715295
1,22	5,61673472	10.0855605

Tabela VII - Previsão

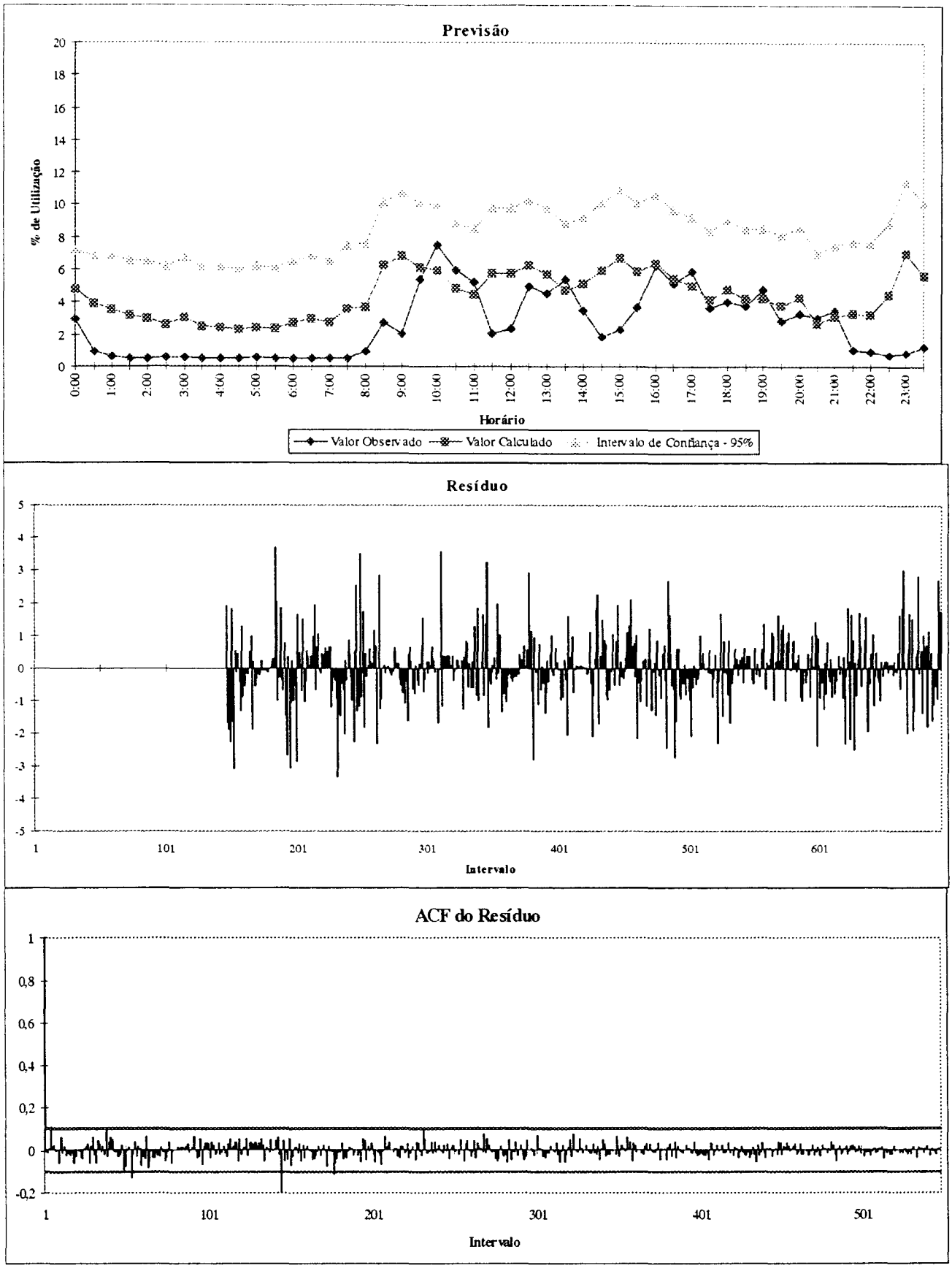


Figura 6.4 - Plotagem dos Valores Lidos e Previstos, Resíduos e ACF do Resíduo

CAPÍTULO 7

CONCLUSÃO

Em um mundo onde a globalização é uma realidade que vem modificando toda a sociedade, o perfeito funcionamento dos sistemas de computação, um dos principais alicerces deste fenômeno, é cada vez mais imprescindível. Neste contexto, o gerenciamento pró-ativo, ou seja, o paradigma de gerenciamento que procura estimar a degradação do sistema e evitar que ela aconteça, é imprescindível.

Como os sistemas de computação são utilizados através das aplicações, o seu gerenciamento vem sendo cada vez mais necessário, levando a uma elevação no grau de detalhamento sobre a forma de se compreender os sistemas de computação.

Este trabalho contribuiu para o gerenciamento pró-ativo de três formas: a utilização da teoria de séries temporais na estimativa de possíveis degradações nos sistemas, a proposta de um modelo de gerenciamento pró-ativo, que trata de informações em nível de aplicação e novas funções de gerenciamento pró-ativas para o módulo Sistema de Correção Automática.

A aplicação da teoria de séries temporais a curto prazo permite uma compreensão bastante apurada do padrão de funcionamento do sistema possibilitando a identificação de mudanças no seu comportamento. A aplicação desta teoria a médio prazo permite a análise do impacto de mudanças no sistema, como por exemplo, a inclusão de novos equipamentos. Além disso, com a disponibilização de um volume maior de informações, a modelagem do sistema pode conter um elevado nível de detalhamento.

A disponibilização de informações sobre as aplicações que estão sendo utilizadas no segmento na forma de *baseline* possibilitou um maior nível de detalhamento sobre o sistema monitorado, permitindo uma maior compreensão do seu comportamento.

A especificação de um conjunto de funções de gerenciamento pró-ativas que se utilizam das informações disponíveis sobre as aplicações contribui para o alargamento dos horizontes do gerenciamento pró-ativo, abrindo novas áreas de pesquisas. Além disso, esta atividade demonstrou quanto são fechadas as aplicações utilizadas no tocante a sua administração.

Quando da implementação do sistema de coleta de informações em nível de aplicação, constatou-se uma grande inadequação das aplicações à atividade de monitoração. Foi principalmente neste fase que verificou-se quão grande era a sobrecarga causada pelo programa *Beholder*, que implementa a RMON MIB, no equipamento em que ele estava sendo executado. A resolução de tal problema resultou na utilização de um equipamento dedicado principalmente a tarefa de coleta de dados. Uma alternativa para a coleta de dados em nível de aplicação bastante eficiente é a utilização do programa desenvolvido em [SIL 95], solução esta não adotada neste trabalho porque foram utilizadas diversas facilidades disponibilizadas pelo programa *Beholder*. Com relação às amostras coletadas sobre as aplicações, constata-se a grande aleatoriedade das mesmas, dificultando sua análise e não permitindo sua modelagem através de séries temporais.

A utilização do programa *Beholder* levou a uma série de pesquisas sobre a sua configuração, funcionamento e utilização. Além disso o emprego das facilidades de filtragem e armazenamento de pacotes, normalmente pouco utilizadas devido a complexidade existente na sua configuração e principalmente na compreensão do seu funcionamento, mereceu uma considerável carga de esforço. Como resultado destas pesquisas, criou-se de um conjunto de programas do tipo *batch* capazes de configurar os grupos da RMON 1 MIB, implementados no programa *Beholder*, utilizados neste trabalho.

Através da aplicação do modelo de gerenciamento pró-ativo proposto e dos estudos sobre séries temporais foi implementado um protótipo do sistema de verificação. Com a utilização da RMON MIB e de programas complementares, foi monitorado um segmento do Instituto de Computação com o objetivo de compor um perfil de funcionamento da rede, tanto até o nível de rede quanto a nível de aplicação. Visando aplicar os conceitos de séries temporais foi implementado um módulo estimador que busca estimar um intervalo para o parâmetro nível de utilização do segmento. Este intervalo mostrou-se bastante apropriado para o intervalo de tempo que foi modelado (30 minutos). Em intervalos menores (30 segundos), as estimativas de limite superior demonstraram ser muito baixas dada a variância que apresenta a variável modelada, duas seriam as soluções para este fato: a utilização de um intervalo de confiança menor, que elevaria o limite superior estimado ou o aumento da tolerância aos avisos de que o nível de utilização do segmento ultrapassou o limite superior.

Como continuidade das pesquisas, sugere-se o aprimoramento do sistema de verificação com o melhora da estimativa de situações que possam degradar a rede. O avanço nos estudos sobre a aplicação de séries temporais no campo do gerenciamento de redes poderia ser feito através de modelagens baseadas em um maior volume de informações e da caracterização de um número maior de sazonalidades tais como semanas de um mês e meses do ano.

Ainda no tocante a trabalhos futuros, propõe-se continuar os estudos sobre a especificação do sistema de correções automáticas e das funções de gerenciamento pró-ativas, tratando de problemas como a definição de um algoritmo de roteamento baseado em aplicações e a implementação da função de controle de congestionamento baseado em aplicações.

REFERÊNCIAS BIBLIOGRÁFICAS

[ALB 92] ALBITZ, Paul, LIU, Cricket. "DNS and Bind". O'reilly & Associates, Inc. 1992.

[BIE 97] BIERMAN, A., Iddon, R. "Remote Network Monitoring MIB Protocol Indentifiers", RFC 2074, janeiro, 1997.

[BOR 93] BORENSTEIN, N.; FREED, N. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies". RFC 1521, setembro, 1993.

[BOW 87] BOWERMAN, Bruce L., OÇONNEL, Richard T. "Time Series Forecasting - Unified Concepts and Computer Implementation". PWS Publishers, 1987.

[BOX 70] BOX, GEORGE P., JENKINS, Gwilym M. "Time Series Analysis - forecasting and control". Holden-Day, 1970.

[BRISA 93] BRISA - SOCIEDADE BRASILEIRA PARA A INTERCONEXÃO DE SISTEMAS ABERTOS, "Gerenciamento de Redes - uma abordagem de sistemas abertos". Makron Books do Brasil Editora Ltda., 1993.

[BUR 93] BURDICK, Matthew J. . "Continuous Monitoring of Remote Networks: The RMON MIB". Hewllet-Packard Journal, abril, 1993.

[CAR 94] CARRILHO, José Aparecido. "Um Modelo para o Gerenciamento do Protocolo FTP baseado em Domínios". Tese de Mestrado, Unicamp, 1994.

[CAS 93] CASE, J., McCLOGHRRIE; K., ROME, M.; WALDBUSSER, S. "Protocol Operations of version 2 of the Simple Netwok Management Protocol (SNMPv2)". RFC 1448, abril, 1993.

[CHA 84] CHATFIELD, Christopher. "The Analysis of Time Series ". Chapman and Hall, 1984.

[CLA 82] CLARCK, D. D. "Window and Acknowledgement Strategy in TCP" RFC 813, julho, 1982.

[DAU 91] DAUBER, Steven. "Finding Fault". *Byte*, março, 207-214, 1991.

[DEF 96a] DE FRANCESCHI, Analúcia Schiaffino Morales. "Aplicação de Desempenho para Validar a Gerência Pró-ativa de Redes", Tese de Mestrado, UFSC, 1996.

[DEF 96b] DE FRANCESCHI, Analúcia Schiaffino Morales, KORMANN, Luiz Fernando, WESTPHALL, Carlos Becker. " A Performance Application for Proactive Network Management". Second IEEE Workshop on Management Systems, junho,19-21, 1996.

[DNP 94] DNPAP Group. "Remote Network Monitoring, MIB, RFC 1271". ftp dnpap.et.tudelft.nl /pub/Whietepaper.ps . Delft University of Technology, The NetherLands, 1994.

[EVE 94] EVERITT, Brian S. "A Handbook of Statistical using S-plus". Chapman & Hall Publisher, 1994.

[FER 83] FERRARINI, Domenico; SERAZZI, Giuseppe; ZEIGNER, Alessandro. "Measurement and Tunning of Computer Systems". Prentice-Hall, USA, 1983.

[FRONTIER] FRONTIER *Software* Development, Inc. "The Integration of RMON and SNMP Technologies for Managing Enterprise Network". http://www.frontier.com/whitepap/index.html

[GRI 93] GRILLO, P.; WALDBUSSER, S. "Host Resources MIB". RFC 1514, setembro, 1993.

[HAN 95] HANDLEY, Mark; CROWEROFT, Jon. "The World-Wide Web: How Servers Work", ConneXions, Vol. 9, n° 8, agosto, 12-24, 1995.

[HUG 94] HUGHES, Jeff. "The RMON Standards for Network Monitoring", ConneXions, Vol. 8, n° 1, janeiro, 1994.

[HUN 92] HUNT, Craig. "TCP/IP Network Administration". Sebastopol, California, USA, O'Reilly & Associates, Inc., 1992.

[ISO 8802.3] ISO/IEC 8802.3 - Information Processing Systems - Local Area Networks - Part 3: Carrier Sense Multiple Access Method and Physical Layer Specification, International Organization for Stantardization /International Electrotechnical Commission, 1992.

[ISO 10040] ISO/IEC DIS 10040 - Information Technology - Open Systems Interconnection - System Management Overview, International Organization for Stantardization /International Electrotechnical Commission, maio, 1994.

[JAN 93] JANDER, Mary. "Proactive LAN Management". Data Communications, março, 1993.

[KIL 94a] KILLE, S. "Network Services Monitoring MIB". RFC 1565, janeiro, 1994.

[KIL 94b] KILLE, S. "Mail Monitoring MIB". RFC 1566, janeiro, 1994.

[KOO 94] KOOIJMAN, R., et all. "RMON Implementation Issues for Managers and Agents". http://dnpap.et.tudelft.nl/~richard/research/research.html, setembro, 1994.

[KRU 95a] KRUPCZAK, Bobby. "Systems Management and the Internet Management Framework". ConneXions, n° 8, 1995.

[KRU 95b] KRUPCZAK, Cheryl. Applicability of *Host* Resources MIB to Application Management. NetWorld+Interop BOF - Report to group, outubro, 1995.

[KRU 96] KRUPCZAK, Bobby, HULTQUIST, Steve. "Managing the InteropNet™". ConneXions, vol. 10, maio, 1996.

[LIU 94] LIU, Cricket, PEEK Jerry, JONES, Russ, BUUS, Bryan, NYE, Adrian. "Managing Internet Information Services". O'reilly & Associates, Inc. 1994.

[MAN 94] MANSFIELD, G.; KILLE, S. "X.500 Directory Monitoring MIB". RFC 1567, janeiro, 1994.

[MUL 95] MULLER, Nathan J., "SNMP's Remote Monitoring MIB -- RMON". http://iquest.com/nmuller/rmon2.html.

[NAG 84] NAGLE, J. "Congestion Control in TCP/IP Internetworks", Computer Communications Review, vol. 14, págs. 11-17, outubro, 1984.

[NAS 94] NASSER, Dan. "Network Optimization and Troubleshooting" Achieve Maxium Network Performance. NRP, Indianapolis, 1994.

[NEM 92] NEMZOV, Martin A. W. "The Internet Management Guide: keeping the link", 2nd. Edition, McGraw Hill, 1992.

[PAN 91] PANKRATZ, Alan. "Forecasting with Dinamic Regression Models". Wiley-Interscience Publication, 1991.

[PHA 94] PHAAL, Peter. "LAN Traffic Management". Prentice-Hall International (UK) Ltd. 1994.

[PIC 95] PICOLO, Carlos; VEIGA, Pedro. "Management of a WWW Server using SNMP". http://smurfland.cit.buffalo.edu/NetMan/Papers.html, maio, 1995.

[POS 83] POSTEL, J. B., Reynolds, J. K. . "Telnet Protocol Specification". RFC 854, maio, 1983. Obsoletes RFC 765.

[POS 85] POSTEL, J. B. . "File Transfer Protocol". RFC 959, outubro, 1985.

[POS 94] POSTEL, J. B. . "Assigned Numbers". RFC 1700, outubro, 1994.

[ROC 96] ROCHA, Marco Antonio da., FERNANDEZ, Luiz Fernando Nunes, WESTPHALL, Carlos Becker. "Gerência Pró-ativa de Redes de Computadores usando Agentes e Técnicas de Inteligência Artificial". Anais do 14° Simpósio Brasileiro de Redes de Computadores, Fortaleza, 97-117, 1996.

[ROS 91] ROSE, Marshall T.. "The simple book - an introduction to management of TCP-IP internets". Prentice-Hall, 1991.

[ROS 93] ROSE, Marshall T., Lynch, Daniel C.. "Internet system handbook", USA, Addison-Wesley Publishing Company, Inc., 1993.

[SAE 96a] SÁENS Artola, Esmilda. "Sistema Especialista para a Gerência Pró-ativa de Redes". Tese de Mestrado, UFRGS, 1996.

[SAE 96b] SÁENS Artola, Esmilda. " Um Sistema Especialista para a Gerência Pró-ativa Remota". Anais do 14° Simpósio Brasileiro de Redes de Computadores, Fortaleza, 118-139, 1996.

[SAL 95] SALAMONE, Salvatore. "Standards-based remote monitoring takes hold". Byte Magazine, junho, 1995.

[SAP 96] SAPERIA, Jonathan; et all. "System Application MIB". Internet Draft (work in progress), abril, 1996.

[SIL 95] SILVEIRA, Carlos Kelner. "Um esquema para o gerenciamento do tráfego de aplicações em redes TCP/IP". Tese de Mestrado, DCC-Unicamp, abril, 1995.

[SOA 95] SOARES, Luiz Fernando G., et all. "Redes de Computadores: Das LANs, MANs e WANs às Redes ATM". Rio de Janeiro, Campus, 1995.

[SOL 81] SOLLINS, K. R. "TFTP Protocol (Revision 2)". Request for Comments 783, junho, 1981.

[SPE 94] SPECTOR, Phill. "An Introduction to s and S-plus". Duxburg Publisher, 1994.

[STA 93] STALLINGS, William. "SNMP, SNMPv2, and CMIP: the practical guide to network management standards", USA, Addison-Wesley Publishing Company, Inc., 1993.

[STA 94] STALLINGS, W. "Packet Filtering in the SNMP Remote Monitor". Dr. Dobb's Journal , novembro, 1994.

[STA 96] STALLINGS, William. "RMON 2 - The Next Generation of Remote Network Monitoring". ConneXions, maio, 1996.

[STE 91] STERN, Hal. "Managing NFS and NIS". O'reilly & Associates, Inc. 1991.

[STE 94] STEVENS, W. Richard. "TCP/IP Illustrated: the protocols", Addison-Wesley Company, Inc., 1994.

[STE 95a] STEVENSON, Douglas W. "Network Management - What is and What isn't". Whitepaper. http://smurfland.cit.buffalo.edu/NetMan/Papers.html, abril, 1995.

[STE 95b] STEINKE, Steve. "A Diamont in the Rough". LAN Magazine, outubro, 1995.

[TAN 96] TANENBAUM, Andrew S. "Computer Networks". Prentice-Hall, 1996.

[WAL 91] WALDBUSSER, S. "Remote Network Monitoring Management Information Base". RFC 1271, novembro, 1991.

[WAL 95a] WALDBUSSER, S. . "Remote Network Monitoring Management Information Base". RFC 1757, fevereiro, 1995.

[WAL 95b] WALDBUSSER, S. "*Token Ring* Extensions to the Remote Network Monitoring Management Information Base". RFC 1513, fevereiro, 1995.

[WAL 97] WALDBUSSER, S. "Remote Network Monitoring Management Information Base Version 2 usin SMIv2". RFC 2021, janeiro, 1997

[WAT 96] WATZKO-NETO, Francisco; MARTINS, Marcelo M.; MADEIRA, Edmundo R. M. "Gerenciamento de Aplicações". *Position Paper* publicado no 1° Workshop de Gerenciamento de Redes de Computadores, Fortaleza, 1996.

ANEXOS

ANEXO A - PROGRAMA DE CONFIGURAÇÃO DOS FILTROS DO BEHOLDER

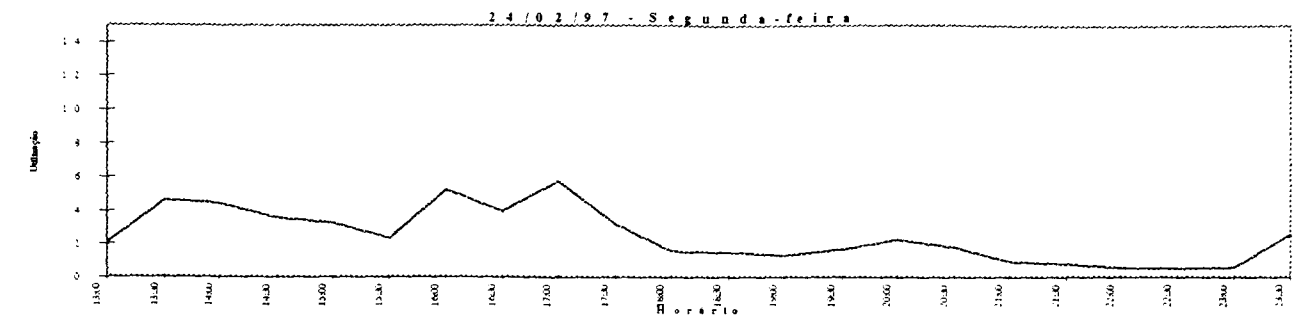
```
# setagem:
# Offset: 14,
# Numero de campos do header: 5,
# Protocolo: tcp (06),
# Source Port: www 80/tcp  (00:50),

echo 'filterStatus[1]=2'|snmp-set itajai secret     # solicita a criacao do filtro
echo 'filterStatus[2]=2'|snmp-set itajai secret     # solicita a criacao do filtro
echo 'filterPktDataOffset[1]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[2]=14'|snmp-set itajai secret
echo'filterPktData[1]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:50'|snmp-set itajai
        secret
echo 'filterPktData[2]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:50'|snmp-set
        itajai secret
echo 'filterPktDataMask[1]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
        itajai secret
echo 'filterPktDataMask[2]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
        set itajai secret
echo 'filterChannelIndex[1]=1'|snmp-set itajai secret # associa o canal n. 1 ao filtro n. 1
echo 'filterChannelIndex[2]=1'|snmp-set itajai secret # associa o canal n. 1 ao filtro n. 1
echo 'channelStatus[1]=2'|snmp-set itajai secret  # solicita a criacao do canal
echo 'channelDataControl[1]=1'|snmp-set itajai secret # permite que os pacotes fluam pelo canal
echo 'bufferControlStatus[1]=2'|snmp-set itajai secret # solicita a criacao do buffer
echo 'bufferControlDownloadSliceSize[1]=45'|snmp-set itajai secret      # n. de bytes trazidos por snmp-get/nxt
echo 'bufferControlFullAction[1]=1'|snmp-set itajai secret # faz com que o buffer pare qdo. cheio
echo 'bufferControlChannelIndex[1]=1'|snmp-set itajai secret # associa o canal n. 1 ao buffer
echo 'bufferControlDownloadOffset[1]=14'|snmp-set itajai secret
echo 'filterStatus[1]=1'|snmp-set itajai secret # ativa o filtro
echo 'filterStatus[2]=1'|snmp-set itajai secret # ativa o filtro
echo 'channelStatus[1]=1'|snmp-set itajai secret # ativa o canal
echo 'bufferControlStatus[1]=1'|snmp-set itajai secret # ativa o buffer de captura de pacotes

# setagem:
# Offset: 14,
# Numero de campos do header: 5,
# Protocolo: tcp (06),
# Source Port: smtp 25/tcp  (00:19),

echo 'filterStatus[3]=2'|snmp-set itajai secret     # solicita a criacao do filtro
echo 'filterStatus[4]=2'|snmp-set itajai secret     # solicita a criacao do filtro
echo 'filterPktDataOffset[3]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[4]=14'|snmp-set itajai secret
echo 'filterPktData[3]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:19'|snmp-set itajai
        secret
echo 'filterPktData[4]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:19'|snmp-set
        itajai secret
```

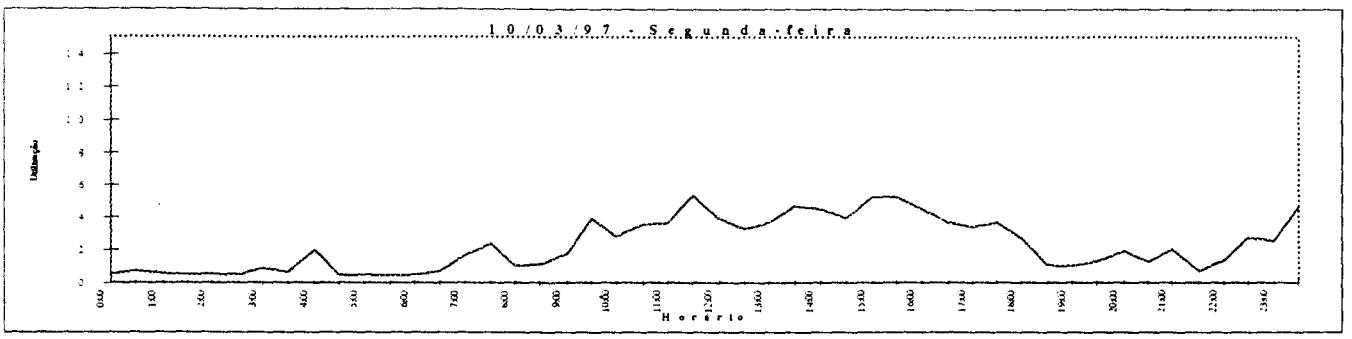
```
echo 'filterPktDataMask[3]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
        itajai secret
echo 'filterPktDataMask[4]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
        set itajai secret
echo 'filterChannelIndex[3]=2'|snmp-set itajai secret # associa o canal n. 2 ao filtro n. 2
echo 'filterChannelIndex[4]=2'|snmp-set itajai secret # associa o canal n. 2 ao filtro n. 2
echo 'channelStatus[2]=2'|snmp-set itajai secret  # solicita a criacao do canal
echo 'channelDataControl[2]=1'|snmp-set itajai secret # permite que os pacotes fluam pelo canal
echo 'bufferControlStatus[2]=2'|snmp-set itajai secret # solicita a criacao do buffer
echo 'bufferControlFullAction[2]=1'|snmp-set itajai secret # faz com que o buffer funcione como um buffer
        circular
echo 'bufferControlChannelIndex[2]=2'|snmp-set itajai secret # associa o canal n. 2 ao buffer
echo 'bufferControlDownloadSliceSize[2]=45'|snmp-set itajai secret      # n. de bytes trazidos por snmp-get/nxt
echo 'bufferControlDownloadOffset[2]=14'|snmp-set itajai secret
echo 'filterStatus[3]=1'|snmp-set itajai secret # ativa o filtro
echo 'filterStatus[4]=1'|snmp-set itajai secret # ativa o filtro
echo 'channelStatus[2]=1'|snmp-set itajai secret # ativa o canal
echo 'bufferControlStatus[2]=1'|snmp-set itajai secret # ativa o buffer de captura de pacotes

# setagem:
# Offset: 14,
# Numero de campos do header: 5,
# Protocolo: tcp (06), udp(17)
# Source Port: DNS 53/tcp/udp  (00:35),

echo 'filterStatus[5]=2'|snmp-set itajai secret     #solicita a criacao do filtro n. 3
echo 'filterStatus[6]=2'|snmp-set itajai secret     #solicita a criacao do filtro n. 3
echo 'filterStatus[13]=2'|snmp-set itajai secret    #solicita a criacao do filtro n. 3
echo 'filterStatus[14]=2'|snmp-set itajai secret    #solicita a criacao do filtro n. 3
echo 'filterPktDataOffset[5]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[6]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[13]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[14]=14'|snmp-set itajai secret
echo 'filterPktData[5]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:35'|snmp-set itajai
        secret
echo 'filterPktData[6]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:35'|snmp-set
        itajai secret
echo 'filterPktData[13]=05:00:00:00:00:00:00:00:00:17:00:00:00:00:00:00:00:00:00:00:00:35'|snmp-set itajai
        secret
echo 'filterPktData[14]=05:00:00:00:00:00:00:00:00:17:00:00:00:00:00:00:00:00:00:00:00:35'|snmp-set
        itajai secret
echo 'filterPktDataMask[5]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
        itajai secret
echo 'filterPktDataMask[6]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
        set itajai secret
echo 'filterPktDataMask[13]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
        itajai secret
echo filterPktDataMask[14]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
        set itajai secret
echo 'filterChannelIndex[5]=3'|snmp-set itajai secret # associa o canal n. 3 ao filtro n. 5
echo 'filterChannelIndex[6]=3'|snmp-set itajai secret # associa o canal n. 3 ao filtro n. 6
echo 'filterChannelIndex[13]=3'|snmp-set itajai secret # associa o canal n. 3 ao filtro n. 5
echo 'filterChannelIndex[14]=3'|snmp-set itajai secret # associa o canal n. 3 ao filtro n. 6
echo 'channelStatus[3]=2'|snmp-set itajai secret# solicita a criacao do canal n. 3
echo 'channelDataControl[3]=1'|snmp-set itajai secret   # permite que os pacotes fluam pelo canal
echo 'bufferControlStatus[3]=2'|snmp-set itajai secret   # solicita a criacao do buffer n. 3
echo 'bufferControlFullAction[3]=1'|snmp-set itajai secret          # faz com que o buffer funcione como um
        buffer circular
echo 'bufferControlChannelIndex[3]=3'|snmp-set itajai secret    # associa o canal n. 3 ao buffer
```

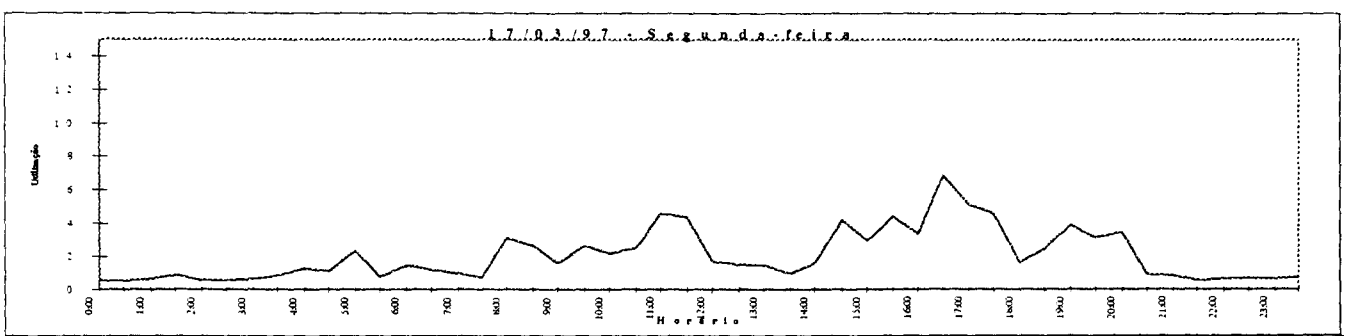
```
echo 'bufferControlDownloadSliceSize[3]=45'|snmp-set itajai secret      # n. de bytes trazidos por snmp-get/nxt
echo 'bufferControlDownloadOffset[3]=14'|snmp-set itajai secret
echo 'filterStatus[5]=1'|snmp-set itajai secret     # ativa o filtro n. 3
echo 'filterStatus[6]=1'|snmp-set itajai secret     # ativa o filtro n. 3
echo 'filterStatus[13]=1'|snmp-set itajai secret    # ativa o filtro n. 3
echo 'filterStatus[14]=1'|snmp-set itajai secret    # ativa o filtro n. 3
echo 'channelStatus[3]=1'|snmp-set itajai secret# ativa o canal n. 3
echo 'bufferControlStatus[3]=1'|snmp-set itajai secret    # ativa o buffer de captura de pacote

# setagem:
# Offset: 14,
# Numero de campos do header: 5,
# Protocolo: tcp (06),
# Source Port: TELNET 23/tcp  (00:17),

echo 'filterStatus[7]=2'|snmp-set itajai secret     #solicita a criacao do filtro n. 4
echo 'filterStatus[8]=2'|snmp-set itajai secret     #solicita a criacao do filtro n. 4
echo 'filterPktDataOffset[7]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[8]=14'|snmp-set itajai secret
echo 'filterPktData[7]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:17'|snmp-set itajai
        secret
echo 'filterPktData[8]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:17'|snmp-set
        itajai secret
echo 'filterPktDataMask[7]=0F:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
        itajai secret
echo 'filterPktDataMask[8]=0F:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
        set itajai secret
echo 'filterChannelIndex[7]=4'|snmp-set itajai secret # associa o canal n. 4 ao filtro n. 4
echo 'filterChannelIndex[8]=4'|snmp-set itajai secret # associa o canal n. 4 ao filtro n. 4
echo 'channelStatus[4]=2'|snmp-set itajai secret# solicita a criacao do canal n. 4
echo 'channelDataControl[4]=1'|snmp-set itajai secret    # permite que os pacotes fluam pelo canal
cho 'bufferControlStatus[4]=2'|snmp-set itajai secret     # solicita a criacao do buffer n. 4
echo 'bufferControlFullAction[4]=1'|snmp-set itajai secret          # faz com que o buffer funcione como um
        buffer circular
echo 'bufferControlChannelIndex[4]=4'|snmp-set itajai secret     # associa o canal n. 4 ao buffer
echo 'bufferControlDownloadSliceSize[4]=45'|snmp-set itajai secret      # n. de bytes trazidos por snmp-get/nxt
echo 'bufferControlDownloadOffset[4]=14'|snmp-set itajai secret
echo 'filterStatus[7]=1'|snmp-set itajai secret     # ativa o filtro n. 4
echo 'filterStatus[8]=1'|snmp-set itajai secret     # ativa o filtro n. 4
echo 'channelStatus[4]=1'|snmp-set itajai secret# ativa o canal n. 4
echo 'bufferControlStatus[4]=1'|snmp-set itajai secret    # ativa o buffer de captura de pacotes

# setagem:
# Offset: 14,
# Numero de campos do header: 5,
# Protocolo: tcp (06),
# Source Port: FTP Data 20/tcp  (00:14),
# Source Port: FTP Control 21/tcp  (00:15),

echo 'filterStatus[9]=2'|snmp-set itajai secret
echo 'filterStatus[10]=2'|snmp-set itajai secret
echo 'filterStatus[11]=2'|snmp-set itajai secret
echo 'filterStatus[12]=2'|snmp-set itajai secret
echo 'filterPktDataOffset[9]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[10]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[11]=14'|snmp-set itajai secret
echo 'filterPktDataOffset[12]=14'|snmp-set itajai secret
echo 'filterPktData[9]=05:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:14'|snmp-set itajai
        secret
```

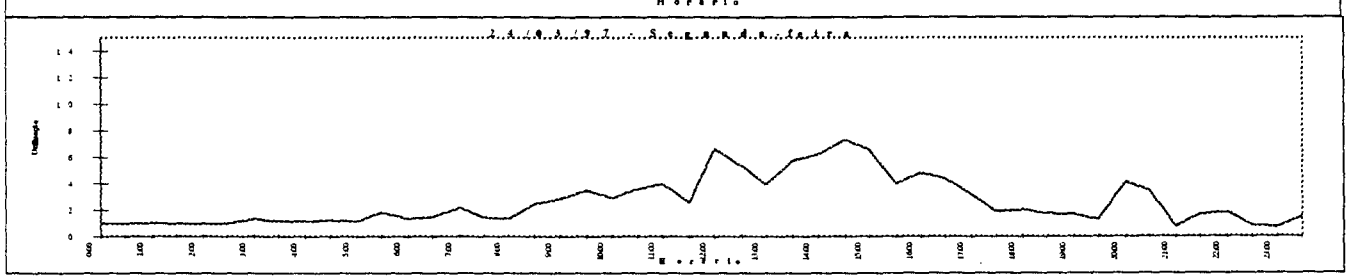
```
echo 'filterPktData[10]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:00:00:14'|snmp-set
    itajai secret
echo 'filterPktData[11]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:00:00:15'|snmp-set itajai
    secret
echo 'filterPktData[12]=05:00:00:00:00:00:00:00:00:00:06:00:00:00:00:00:00:00:00:00:00:00:00:00:00:15'|snmp-set
    itajai secret
echo 'filterPktDataMask[9]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
    itajai secret
echo'filterPktDataMask[10]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-
    set itajai secret
echo 'filterPktDataMask[11]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
    itajai secret
echo 'filterPktDataMask[12]=0F:00:00:00:00:00:00:00:00:00:FF:00:00:00:00:00:00:00:00:00:00:00:00:FF:FF'|snmp-set
    itajai secret
echo 'filterChannelIndex[9]=5'|snmp-set itajai secret
echo 'filterChannelIndex[10]=5'|snmp-set itajai secret
echo 'filterChannelIndex[11]=5'|snmp-set itajai secret
echo 'filterChannelIndex[12]=5'|snmp-set itajai secret
echo 'channelStatus[5]=2'|snmp-set itajai secret
echo 'channelDataControl[5]=1'|snmp-set itajai secret
echo 'bufferControlStatus[5]=2'|snmp-set itajai secret
echo 'bufferControlFullAction[5]=1'|snmp-set itajai secret
echo 'bufferControlChannelIndex[5]=5'|snmp-set itajai secret
echo 'bufferControlDownloadSliceSize[5]=45'|snmp-set itajai secret
echo 'bufferControlDownloadOffset[5]=14'|snmp-set itajai secret
echo 'filterStatus[9]=1'|snmp-set itajai secret
echo 'filterStatus[10]=1'|snmp-set itajai secret
echo 'filterStatus[11]=1'|snmp-set itajai secret
echo 'filterStatus[12]=1'|snmp-set itajai secret
echo 'channelStatus[5]=1'|snmp-set itajai secret
echo 'bufferControlStatus[5]=1'|snmp-set itajai secret
```

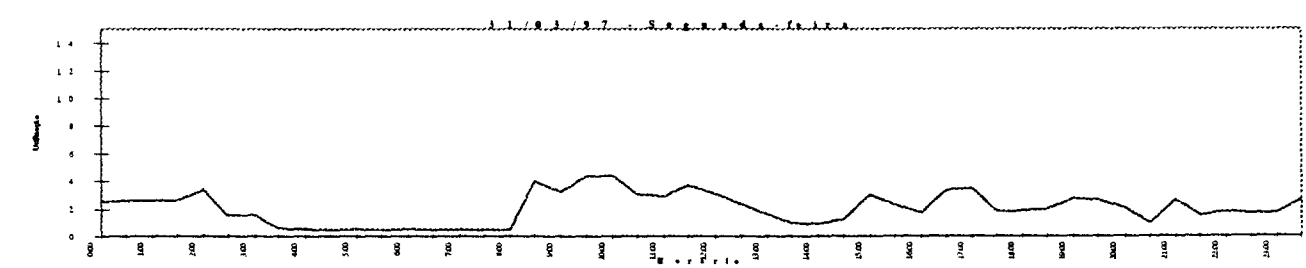
ANEXO B - *BASELINE* DO NÍVEL DE UTILIZAÇÃO DO SEGMENTO MONITORADO

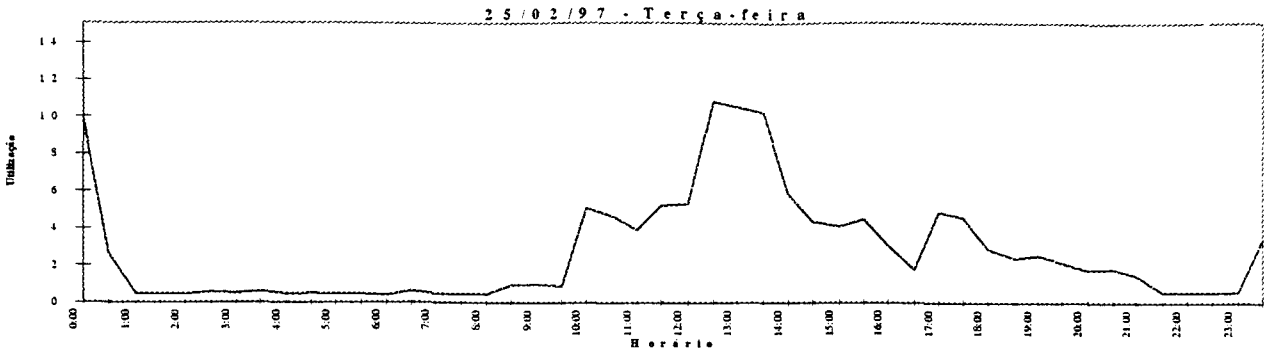


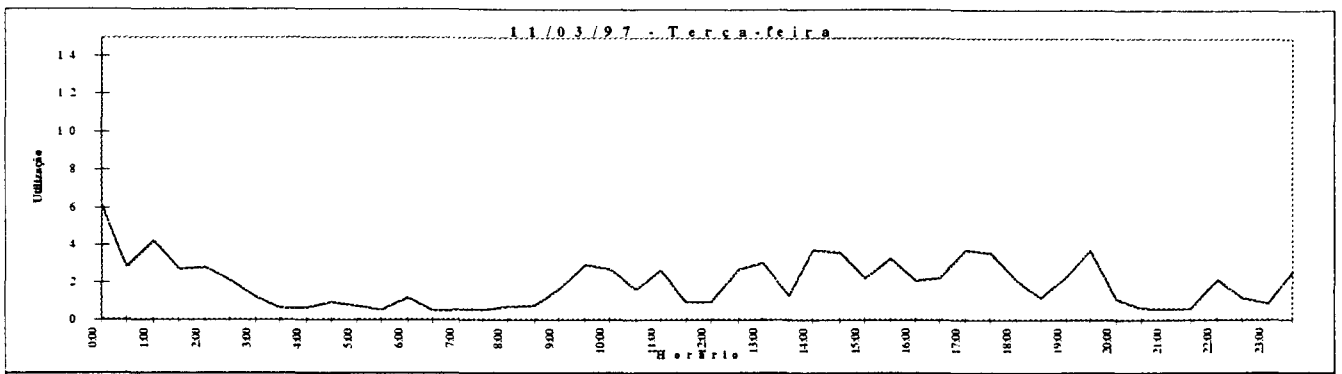


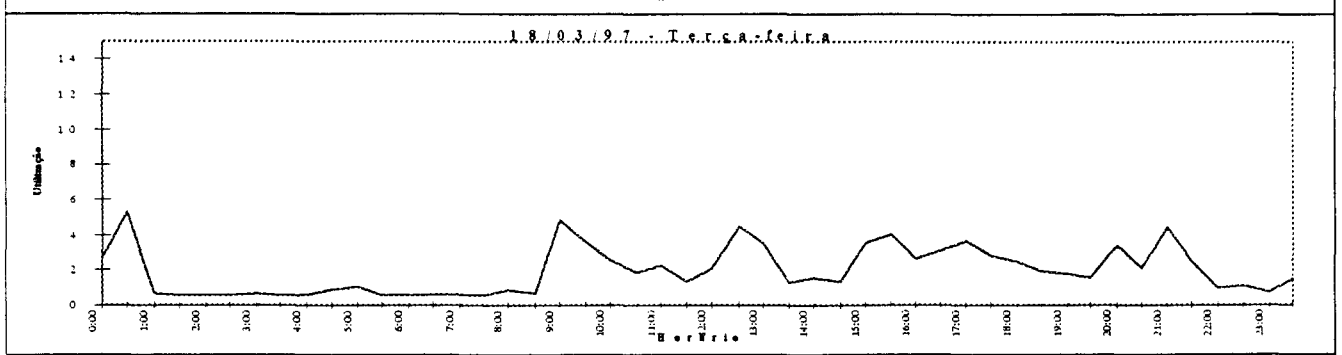


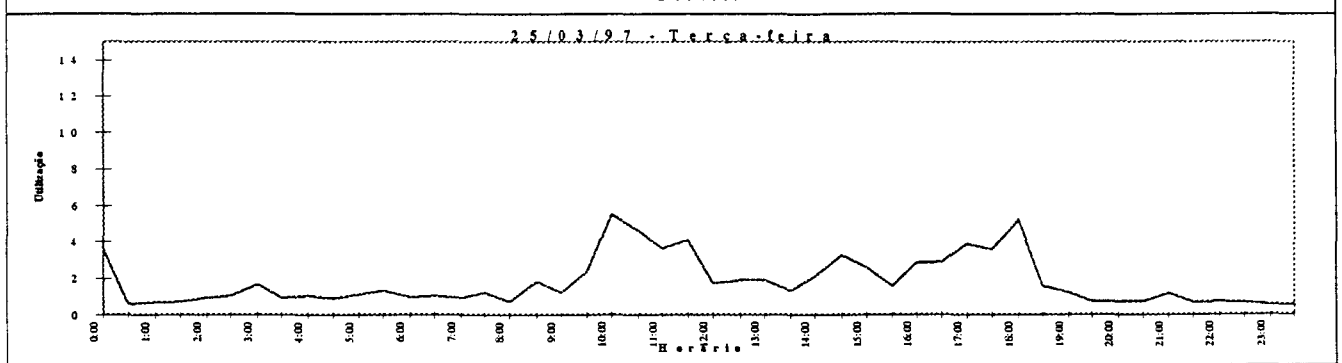


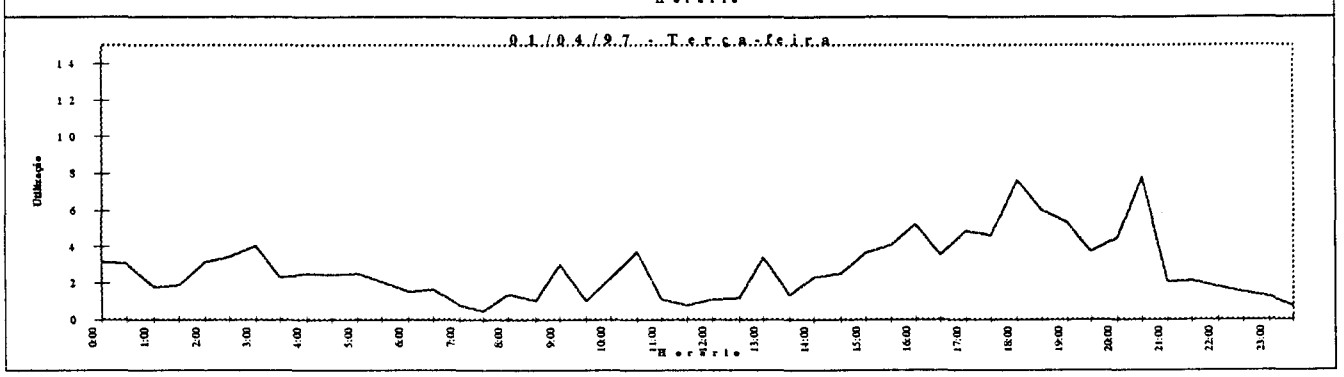


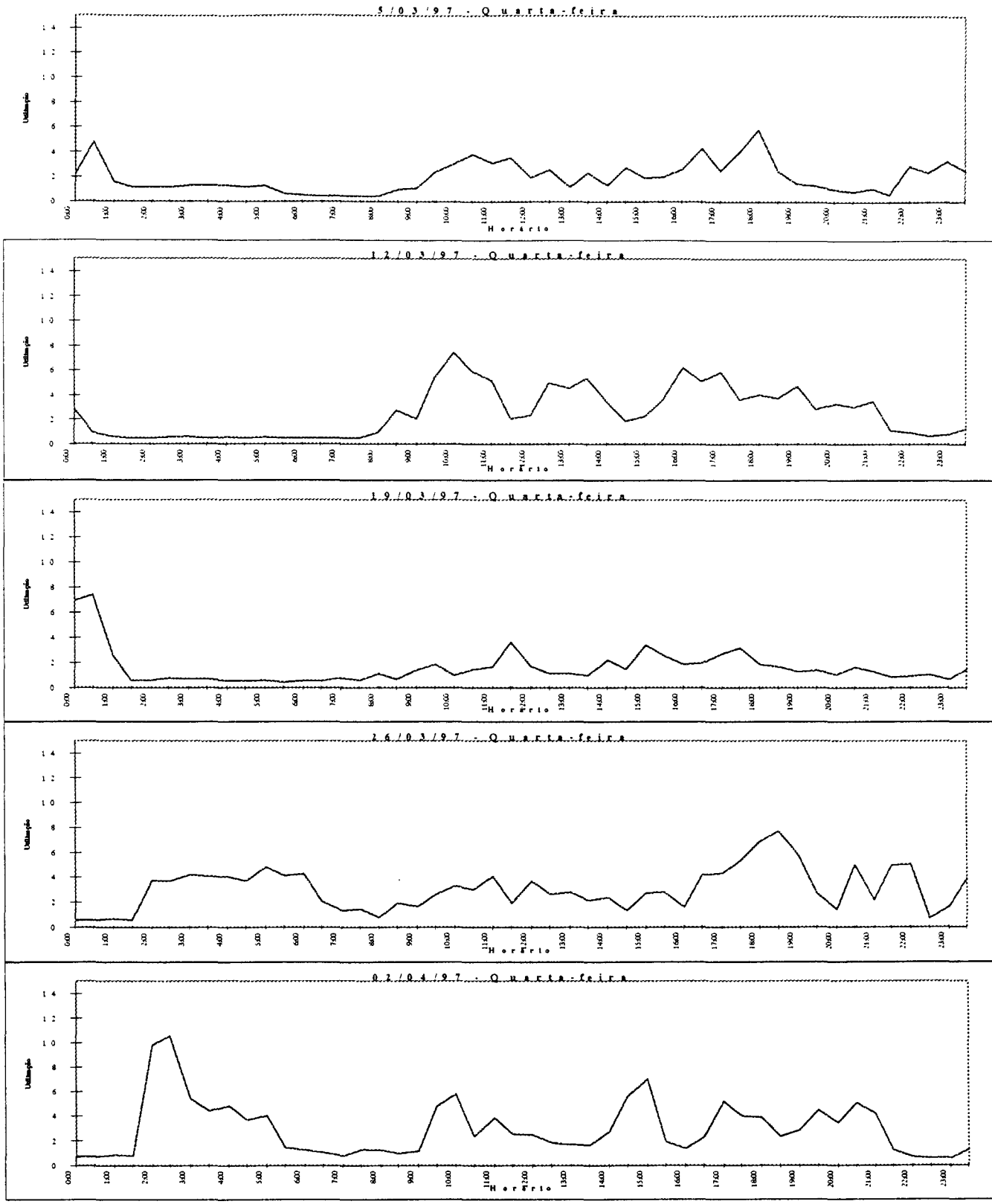

25/02/97 - Terça-feira
Utilização
Horário

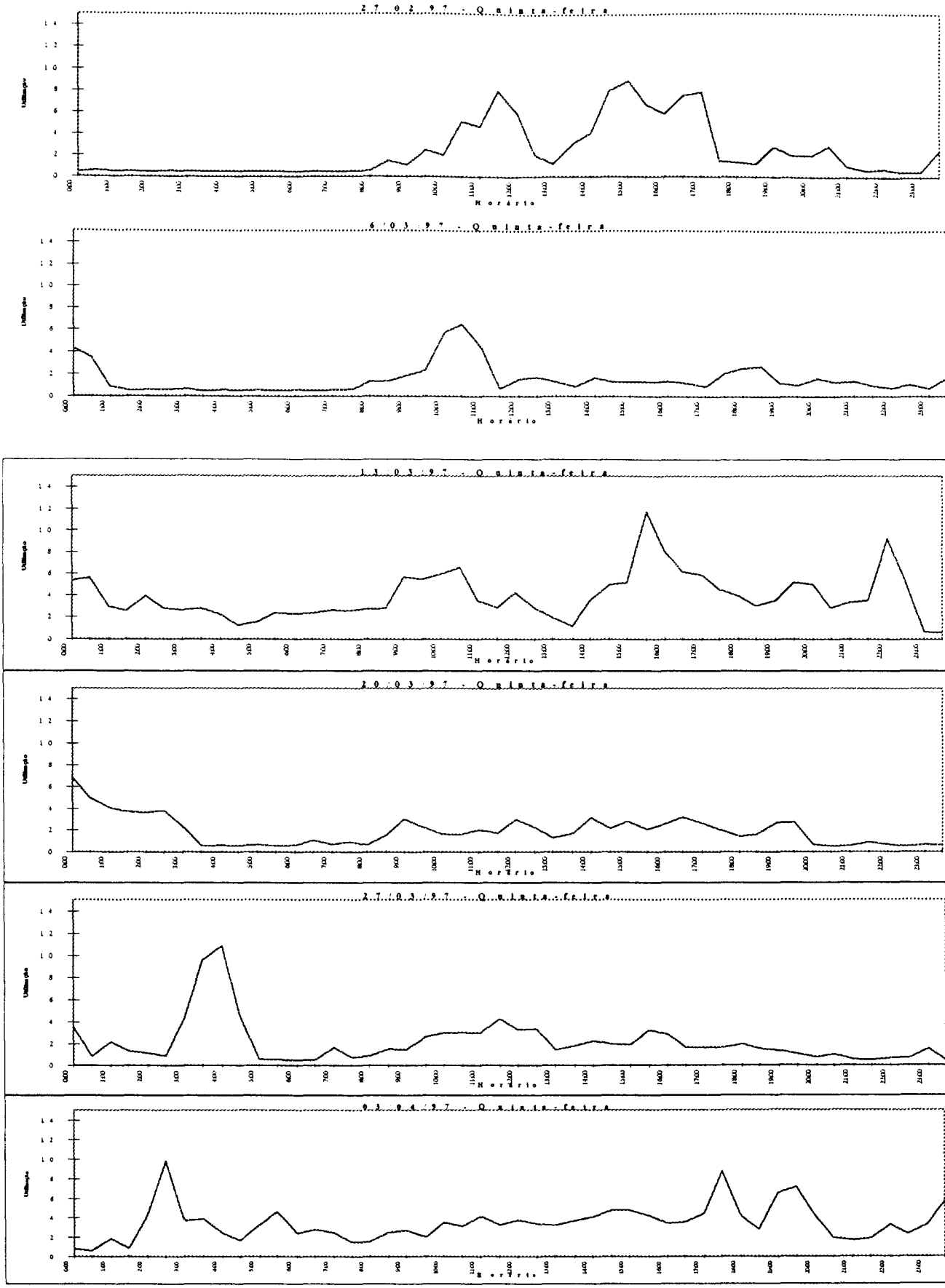

11/03/97 - Terça-feira
Utilização
Horário

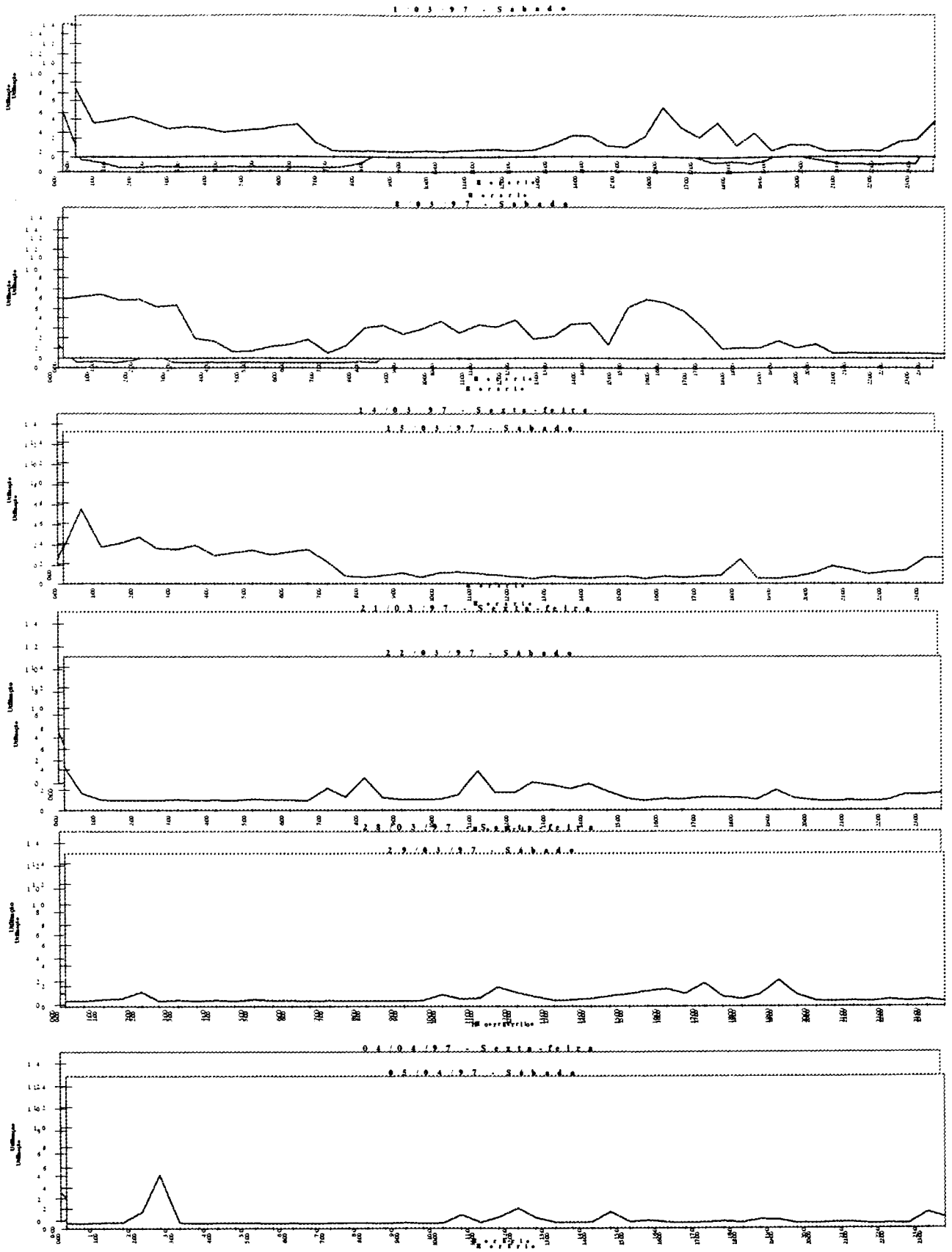

18/03/97 - Terça-feira
Utilização
Horário

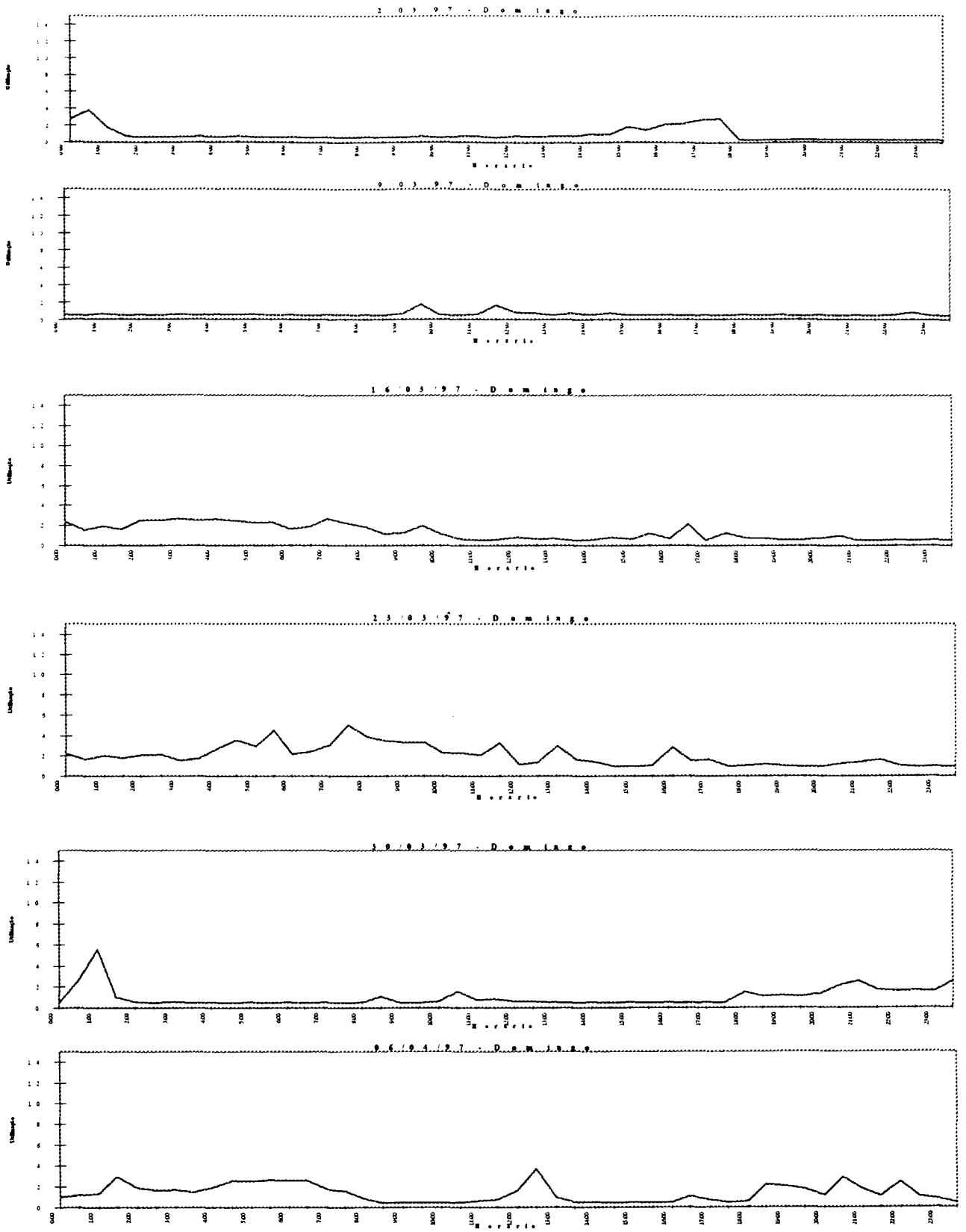

25/03/97 - Terça-feira
Utilização
Horário


01/04/97 - Terça-feira
Utilização
Horário

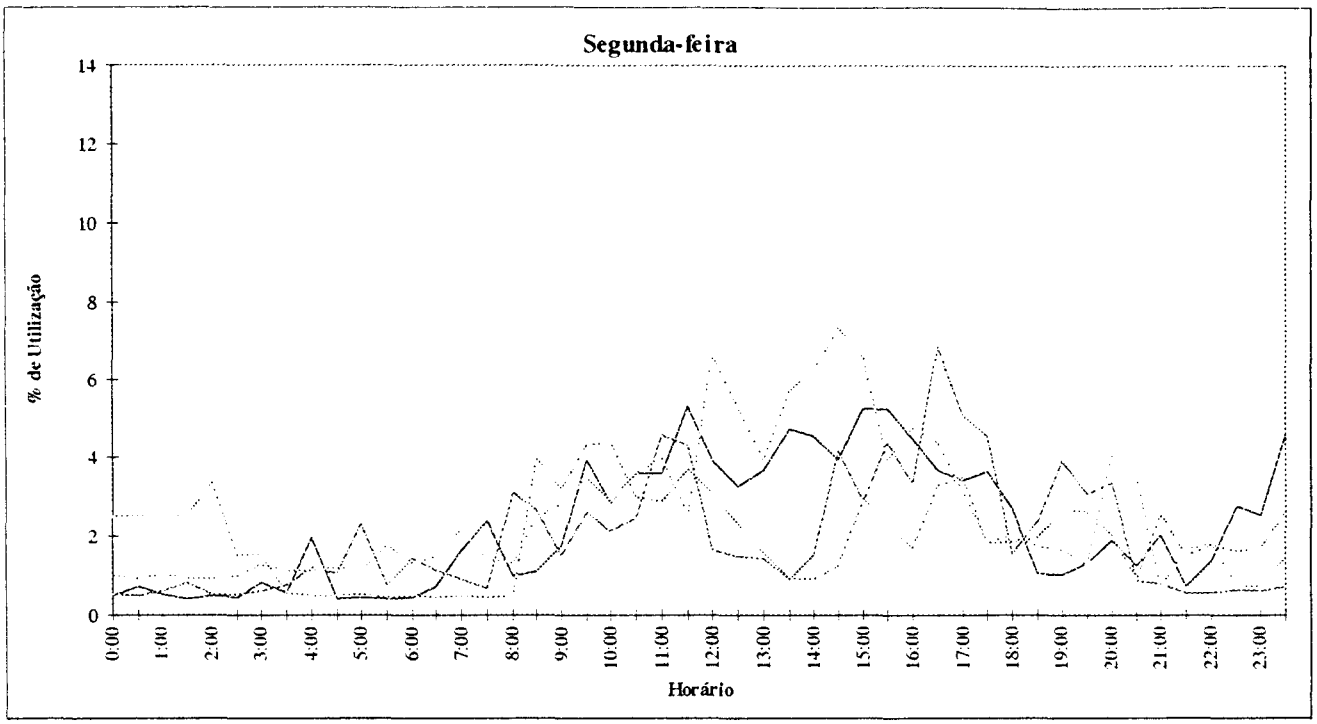

5/03/97 - Quarta-feira
12/03/97 - Quarta-feira
19/03/97 - Quarta-feira
26/03/97 - Quarta-feira
02/04/97 - Quarta-feira
Utilização
Horário

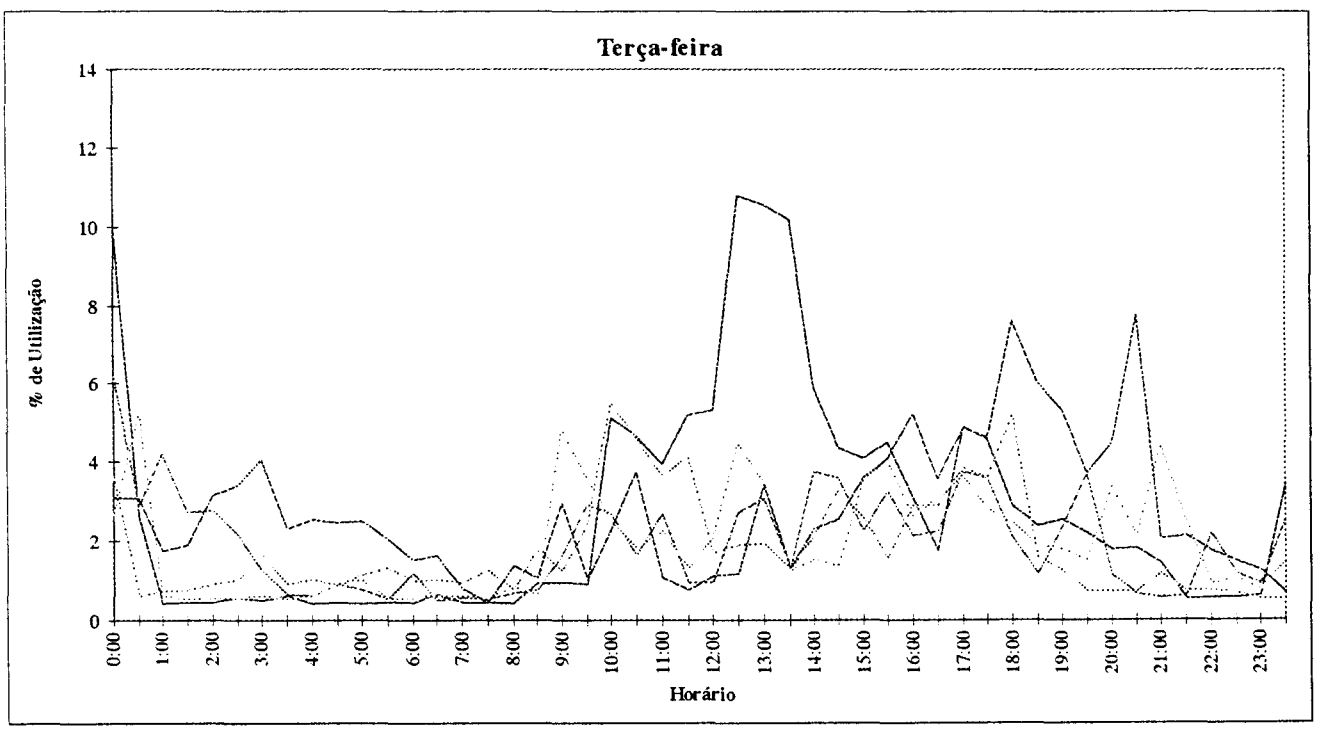

27/02/97 - Quinta-feira
6/03/97 - Quinta-feira
13/03/97 - Quinta-feira
20/03/97 - Quinta-feira
27/03/97 - Quinta-feira
03/04/97 - Quinta-feira
Horário

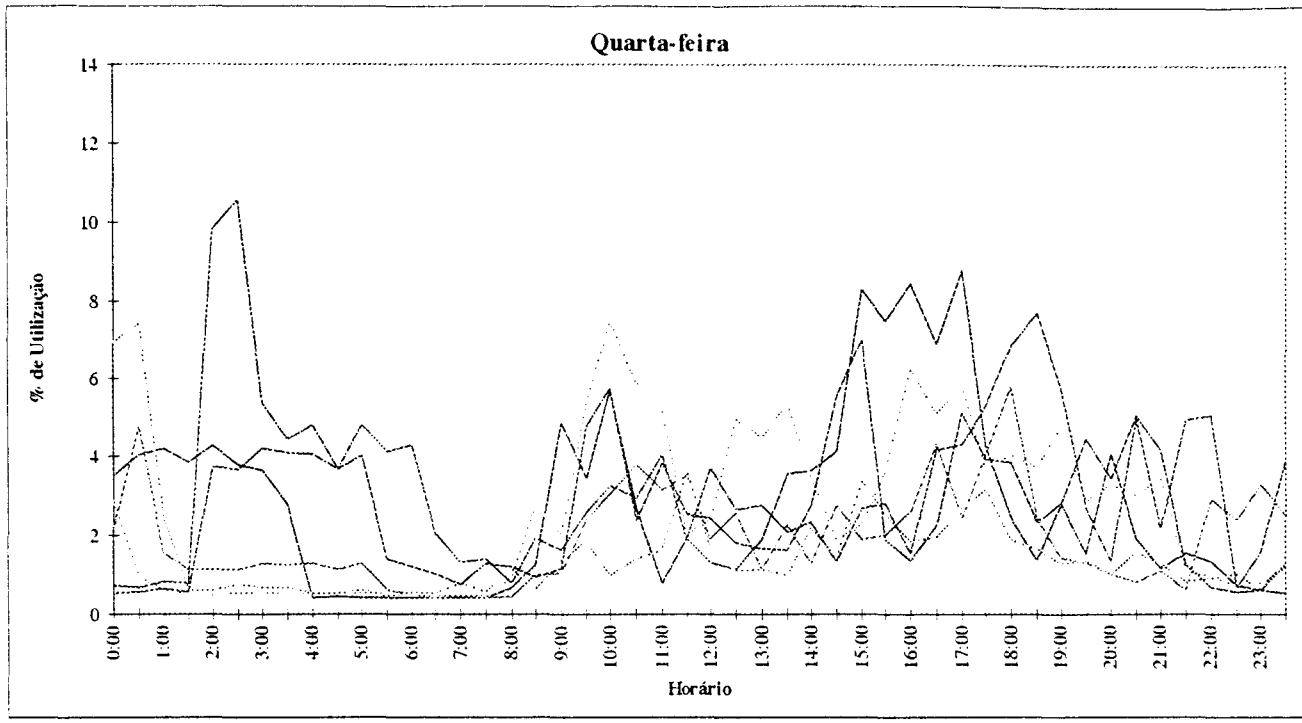


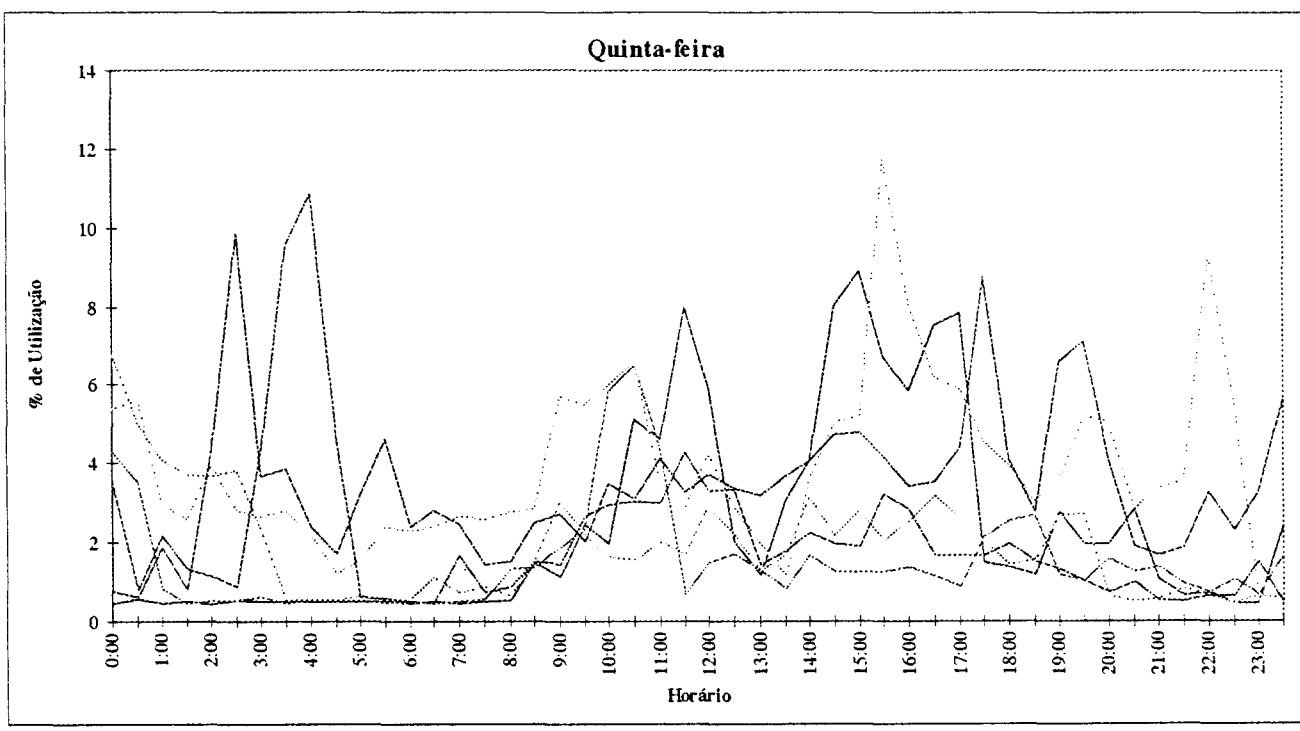


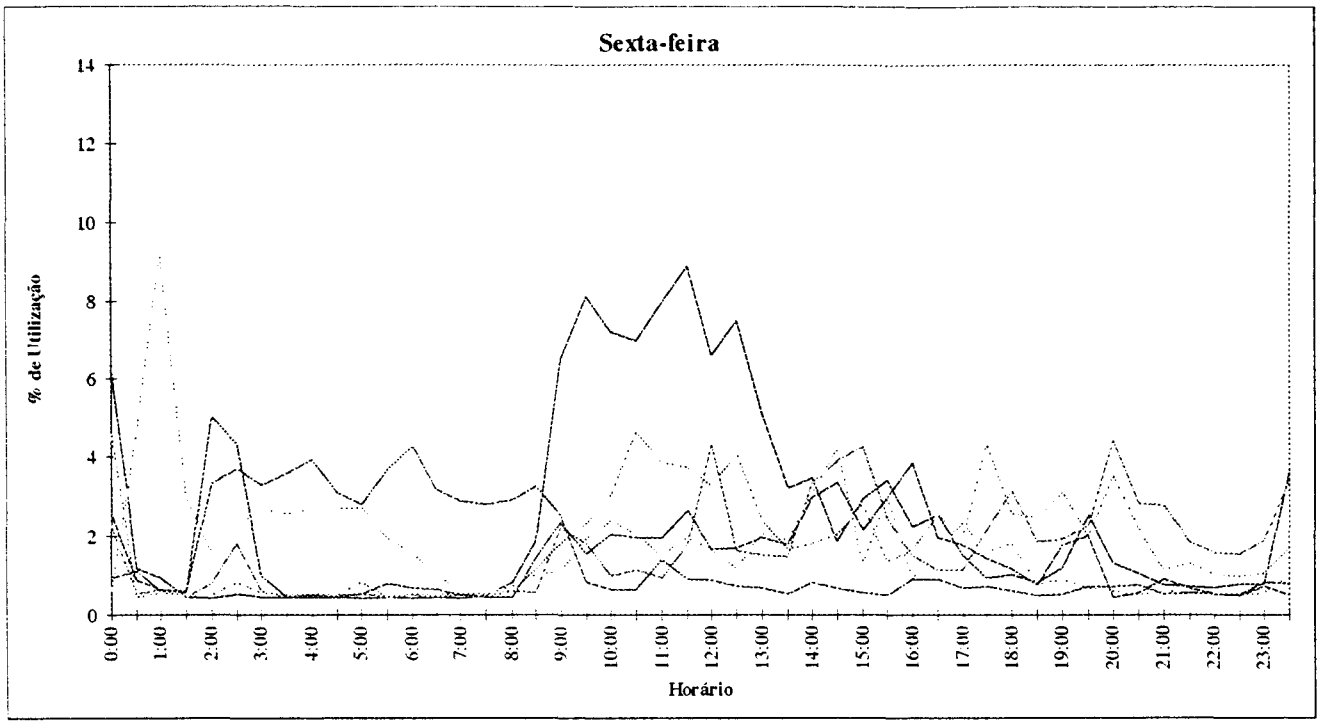
Serão apresentados a seguir os gráficos do nível de utilização do segmento monitorado separado por dia da semana.

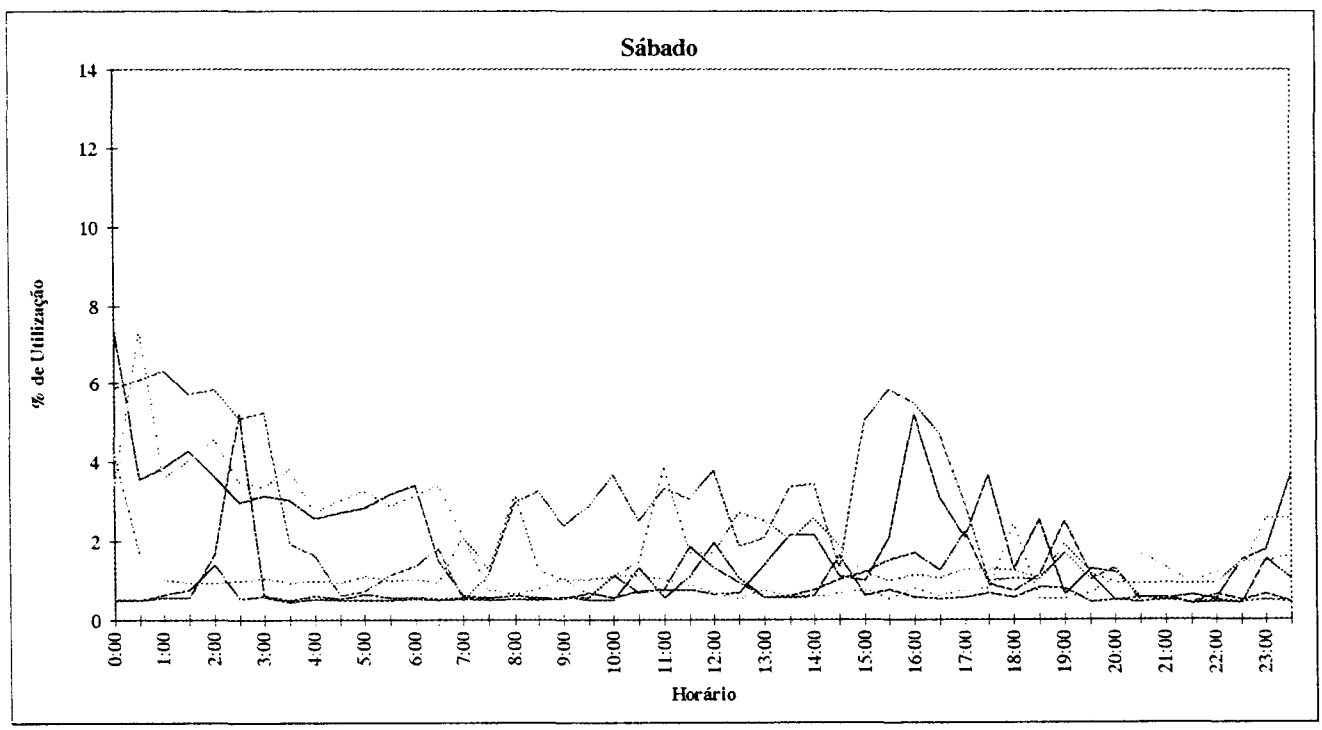

Segunda-feira
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário

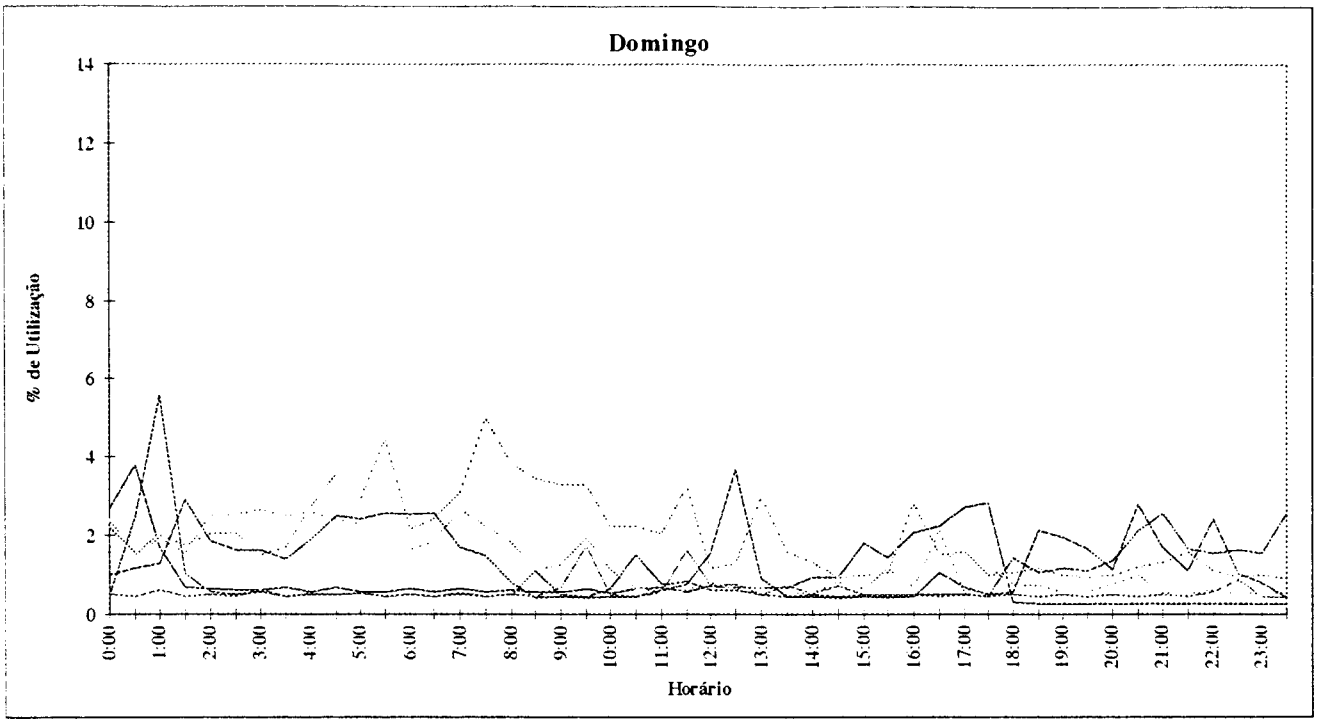

Terça-feira
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário


Quarta-feira
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário


Quinta-feira
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário

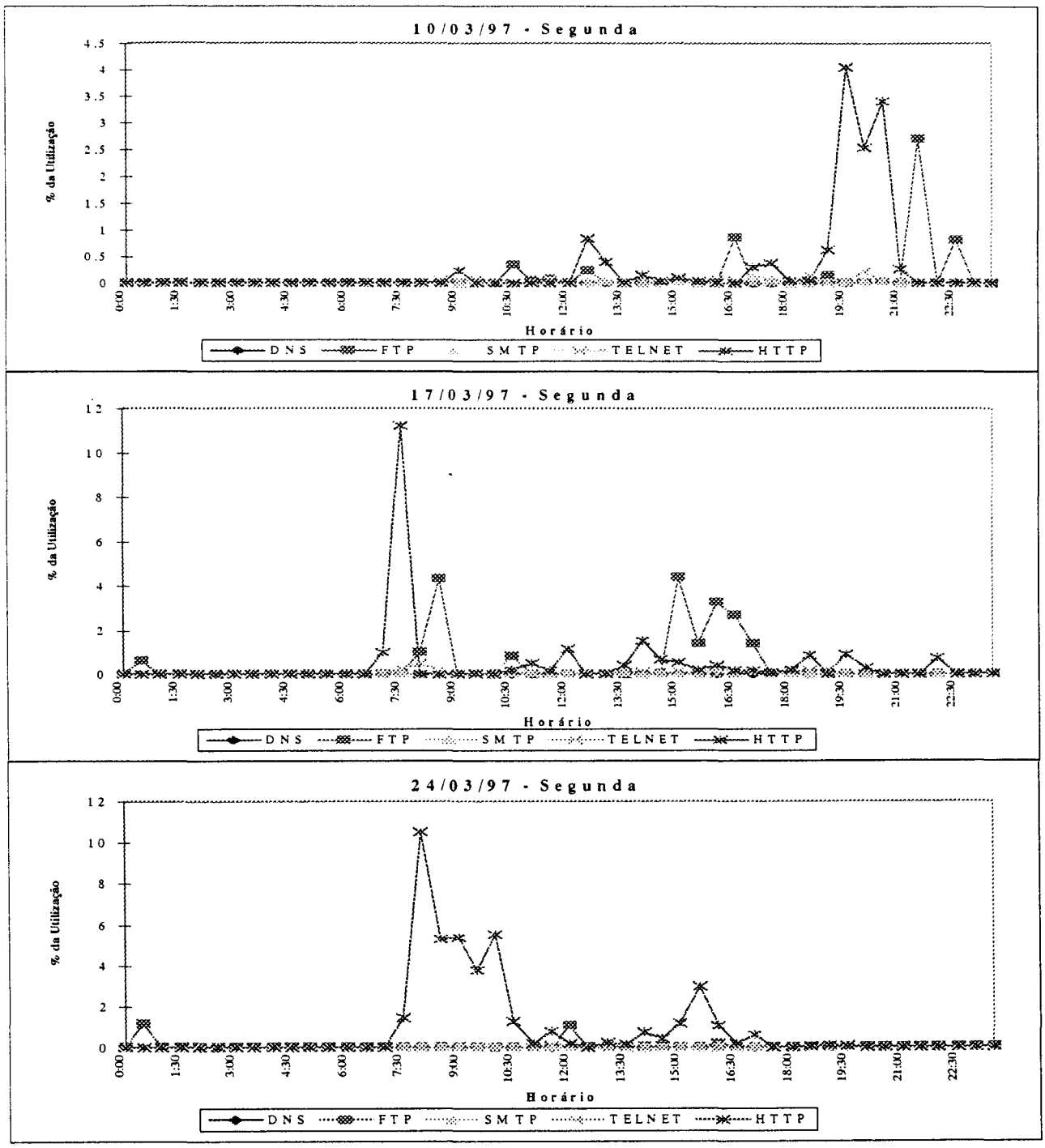

Sexta-feira
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário

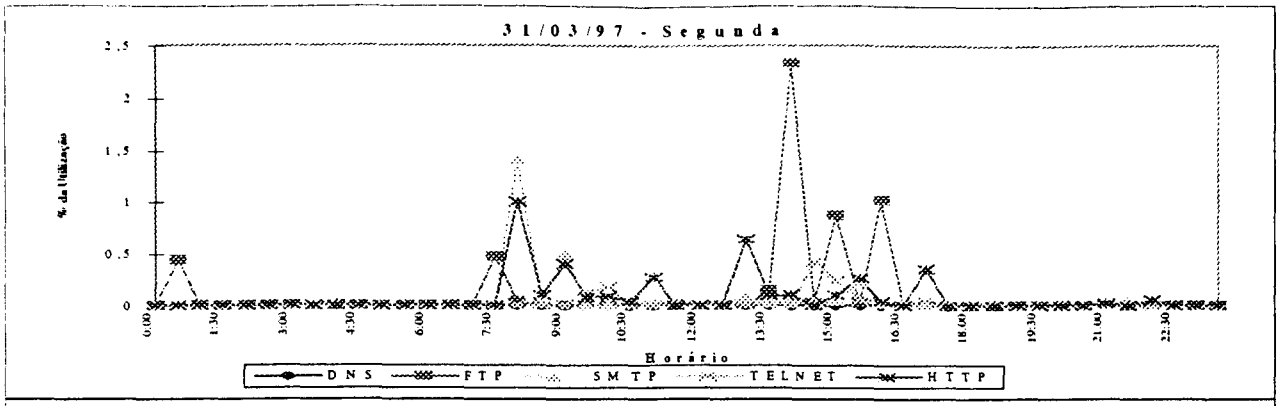

Sábado
% de Utilização
0
2
4
6
8
10
12
14
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário

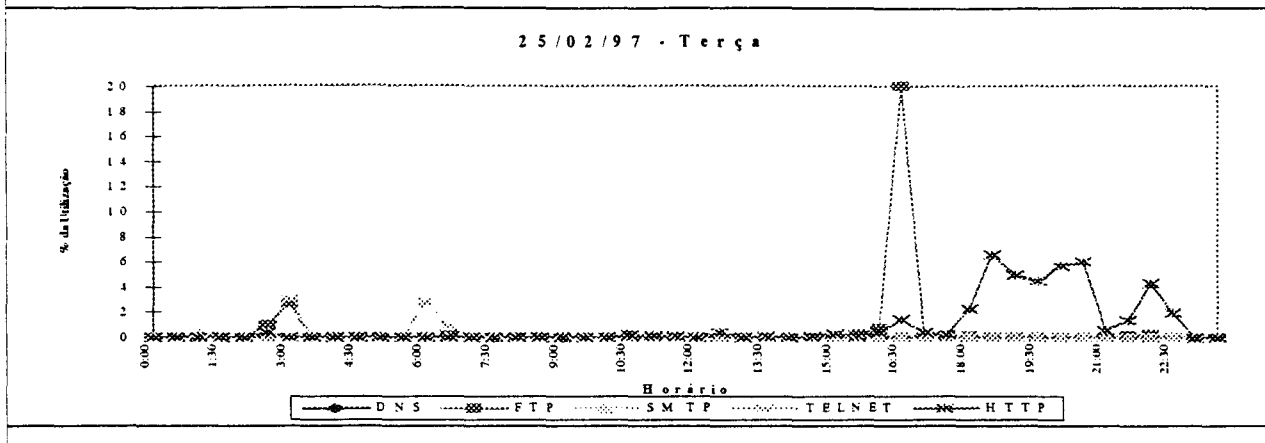

Domingo
% de Utilização
14
12
10
8
6
4
2
0
0:00
1:00
2:00
3:00
4:00
5:00
6:00
7:00
8:00
9:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Horário

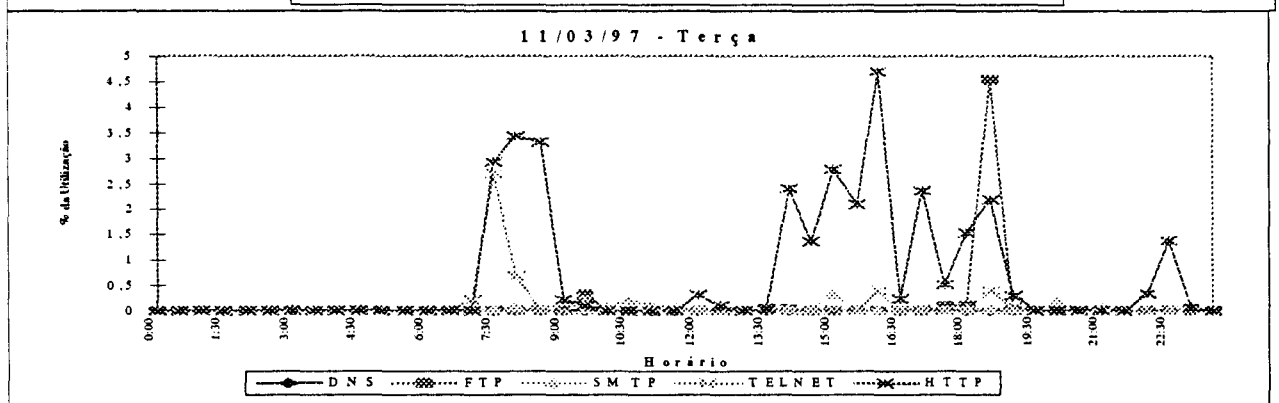
ANEXO C - *BASELINES* COLETADAS EM NÍVEL DE APLICAÇÃO

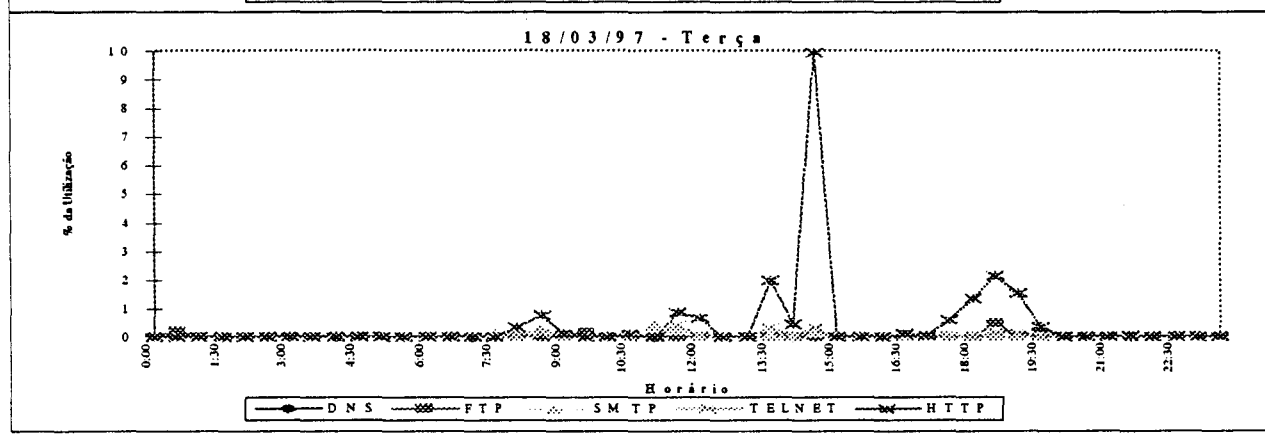
Serão apresentadas a seguir as *baselines* em nível de aplicação coletadas diariamente.

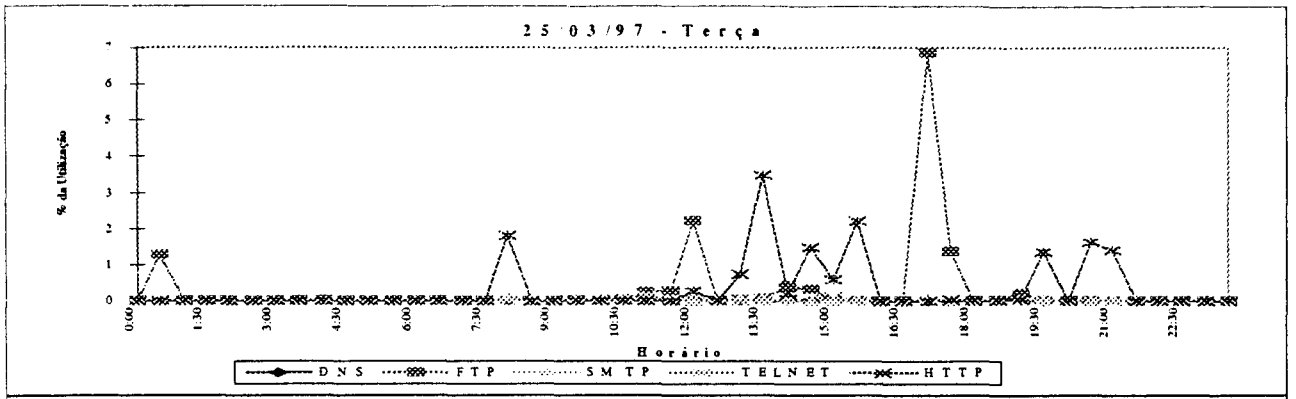


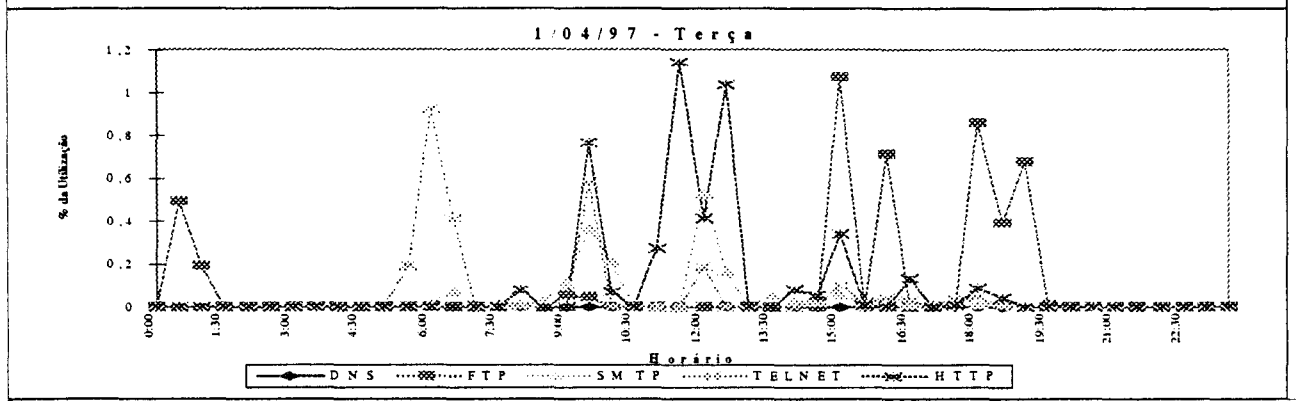

31/03/97 - Segunda
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

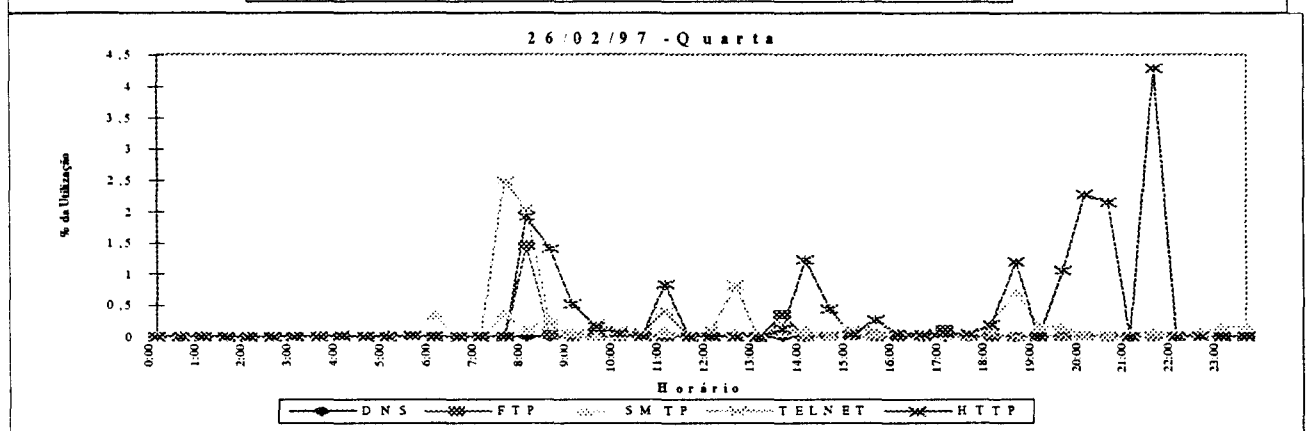

25/02/97 - Terça
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

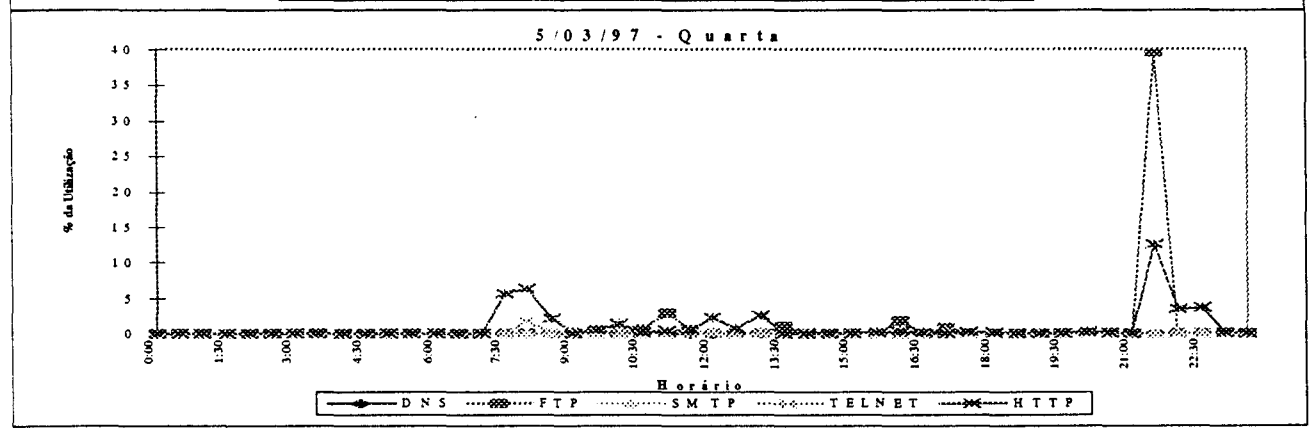

11/03/97 - Terça
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

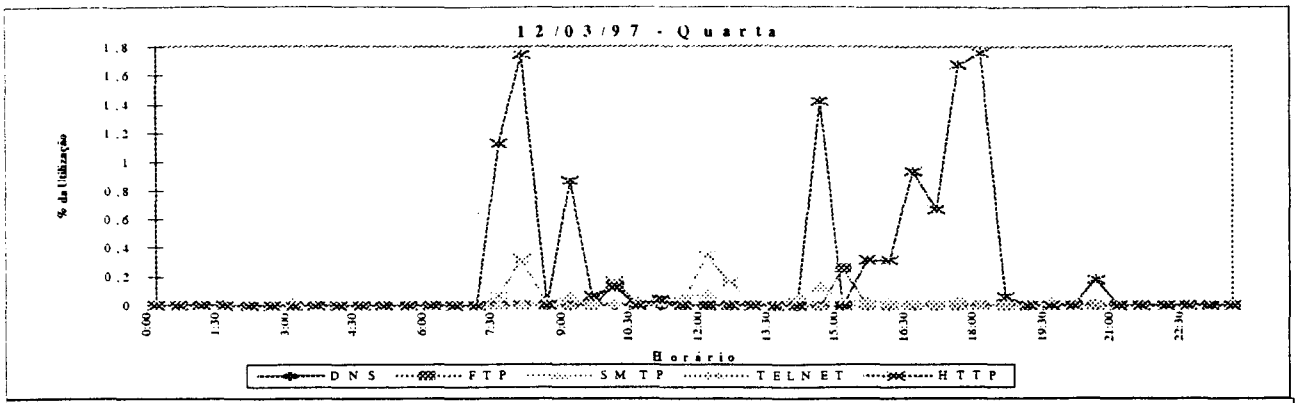

18/03/97 - Terça
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

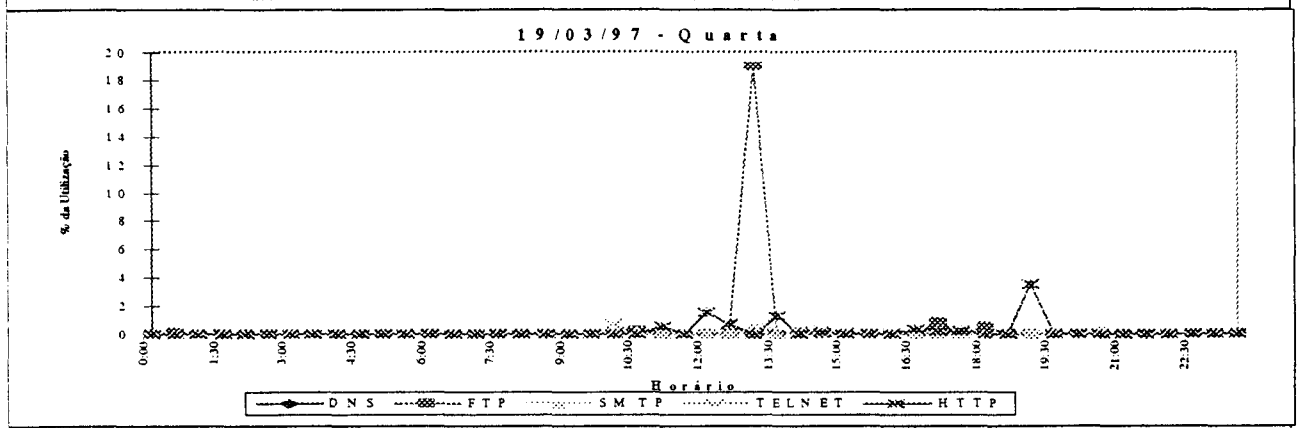

25/03/97 - Terça
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

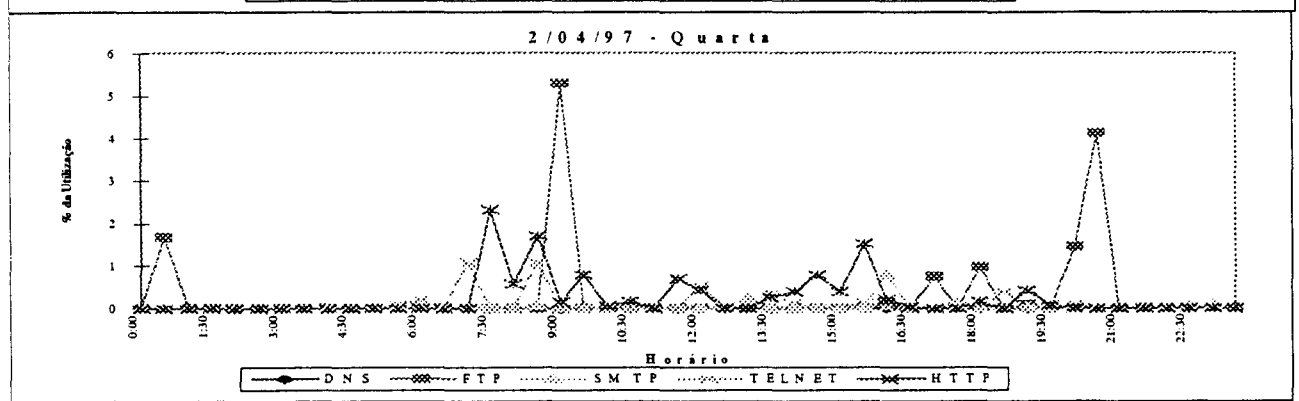

1/04/97 - Terça
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

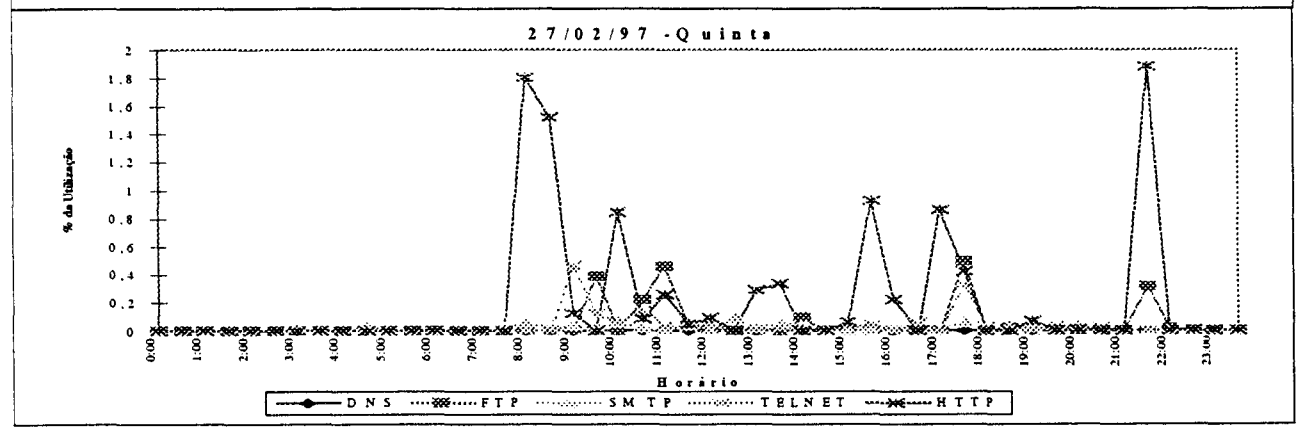

26/02/97 - Quarta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

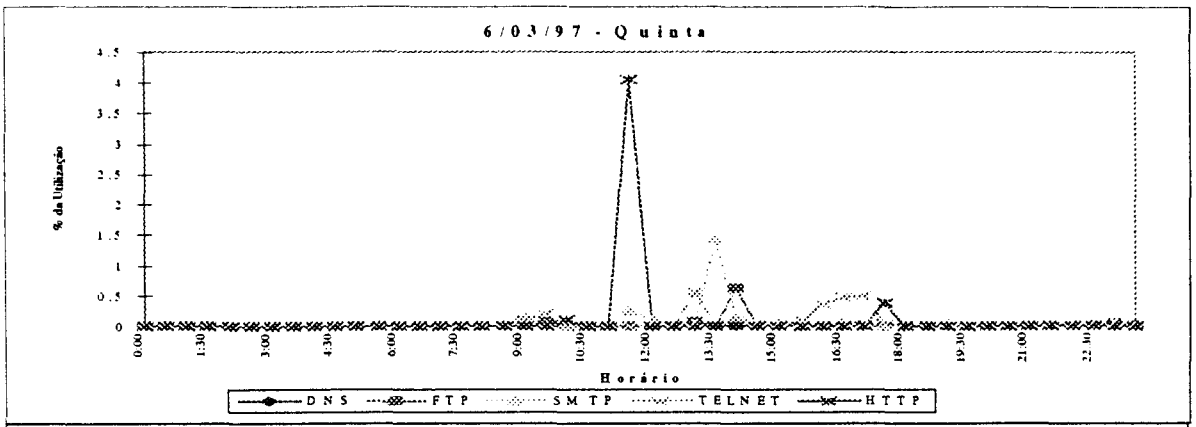

5/03/97 - Quarta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

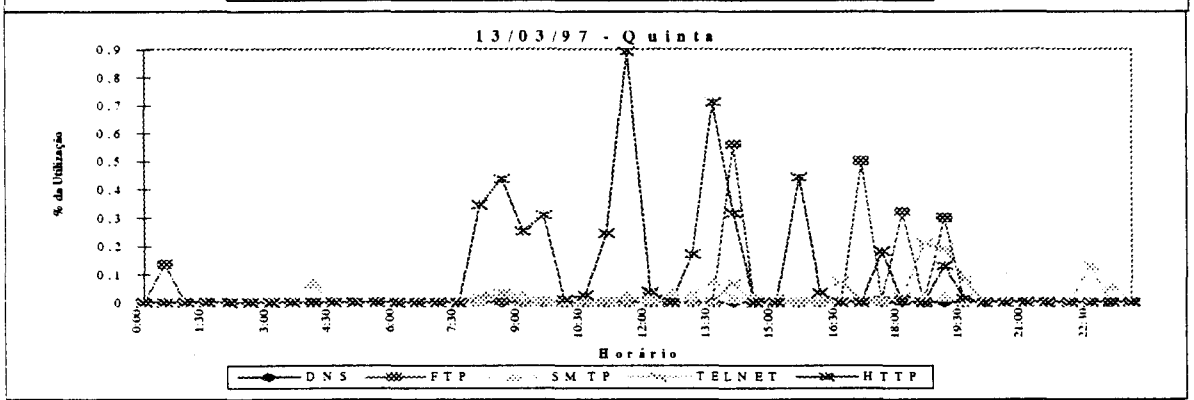

12/03/97 - Quarta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

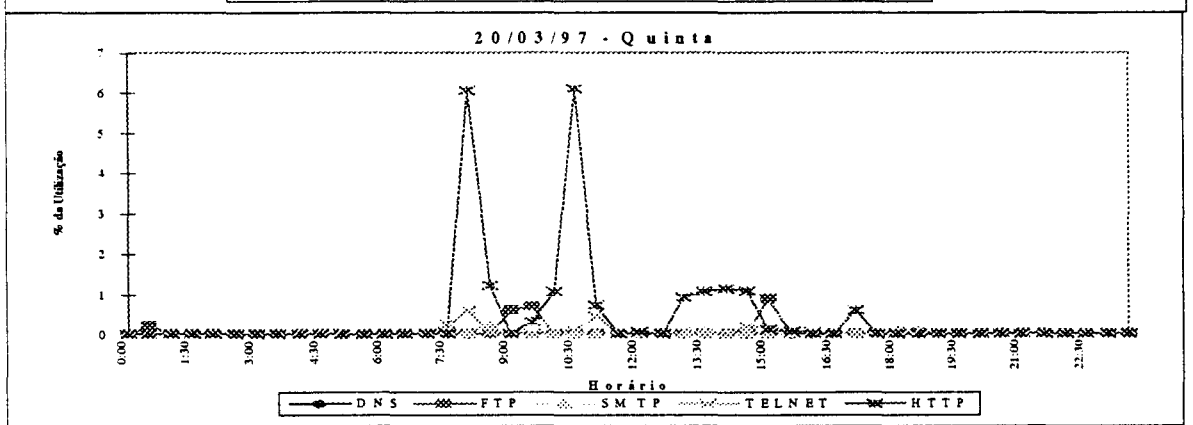

19/03/97 - Quarta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

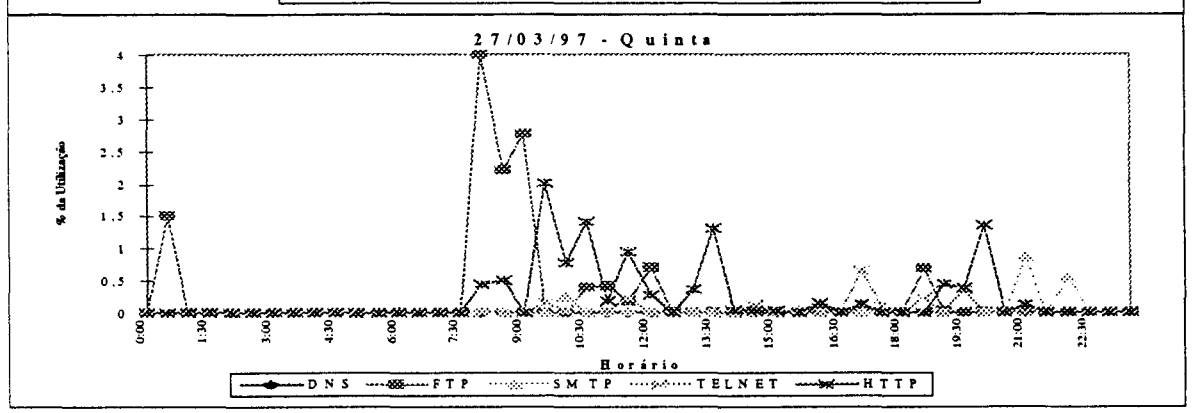

2/04/97 - Quarta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

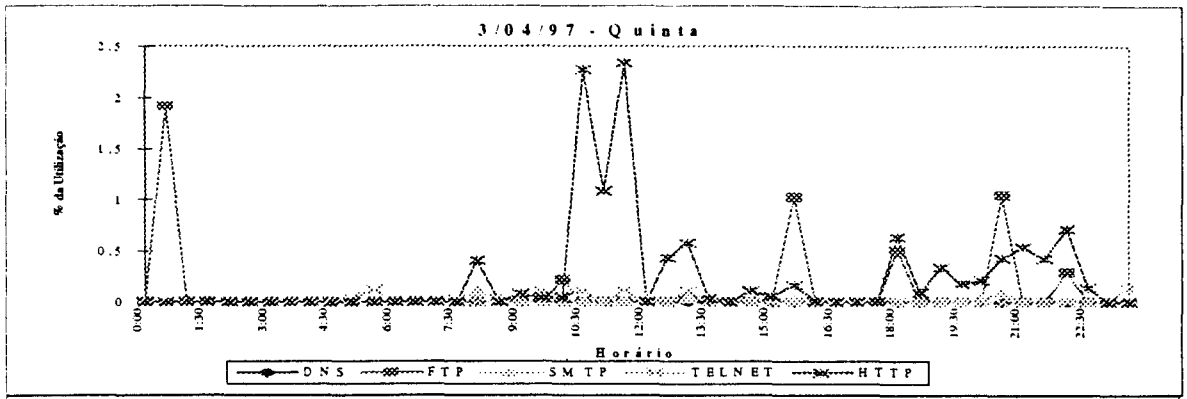

27/02/97 -Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

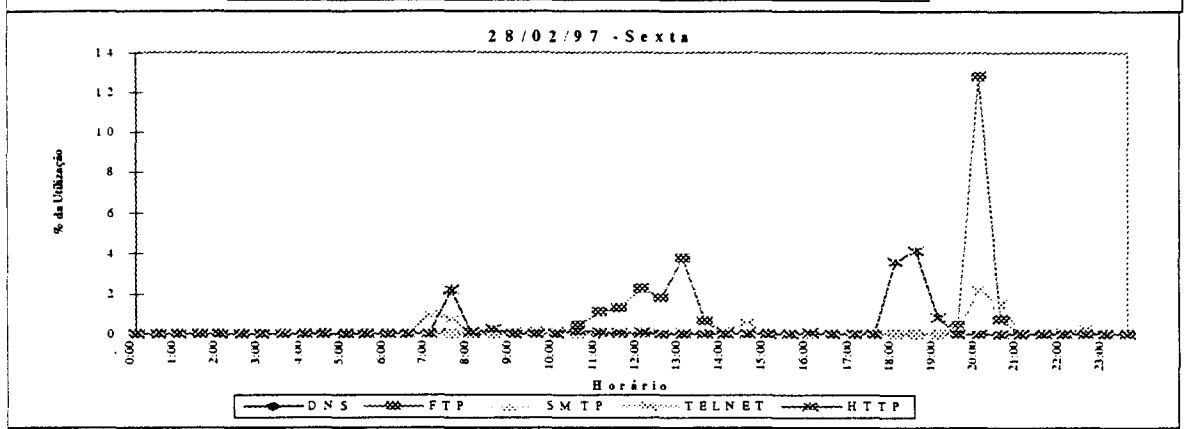

6/03/97 - Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

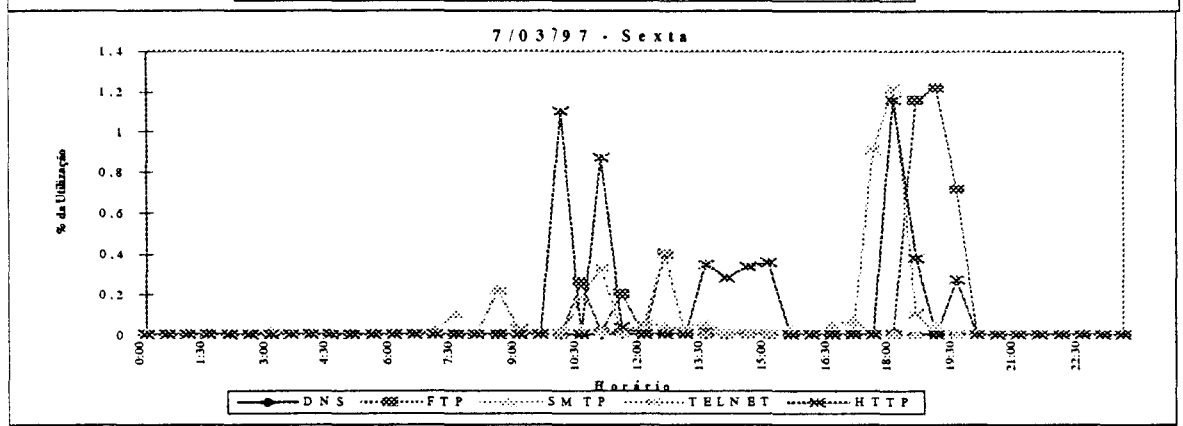

13/03/97 - Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

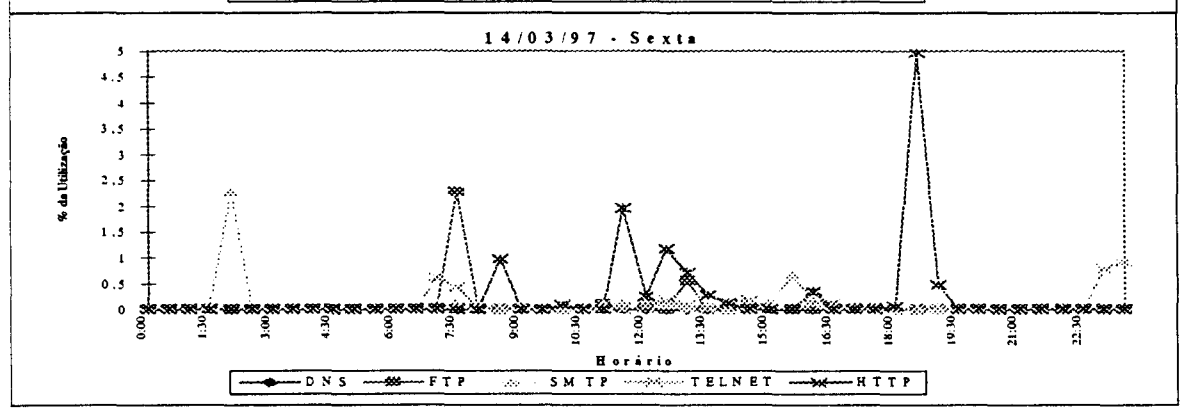

20/03/97 - Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

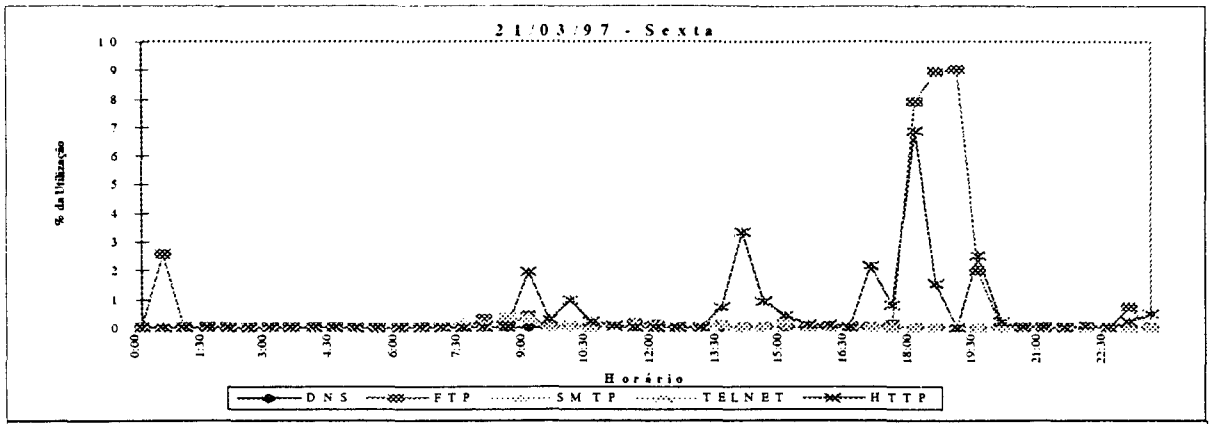

27/03/97 - Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

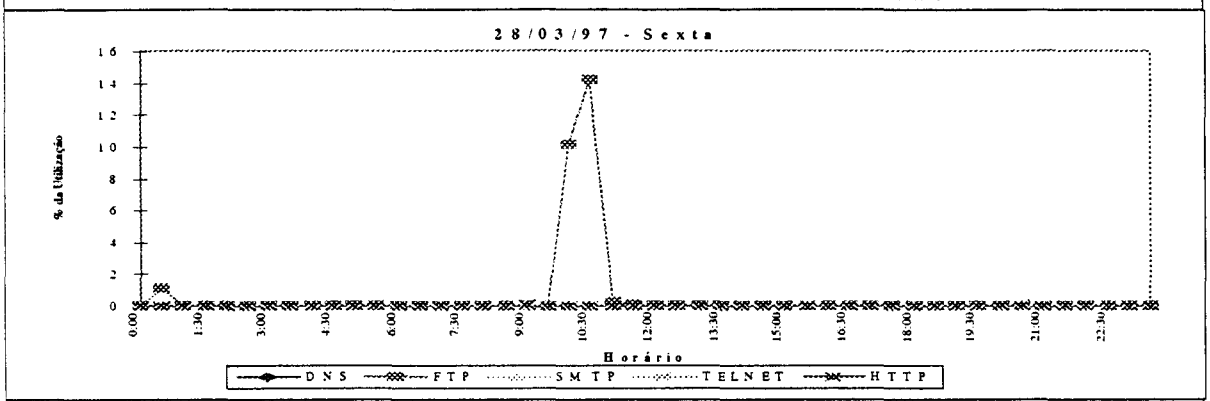

3/04/97 - Quinta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

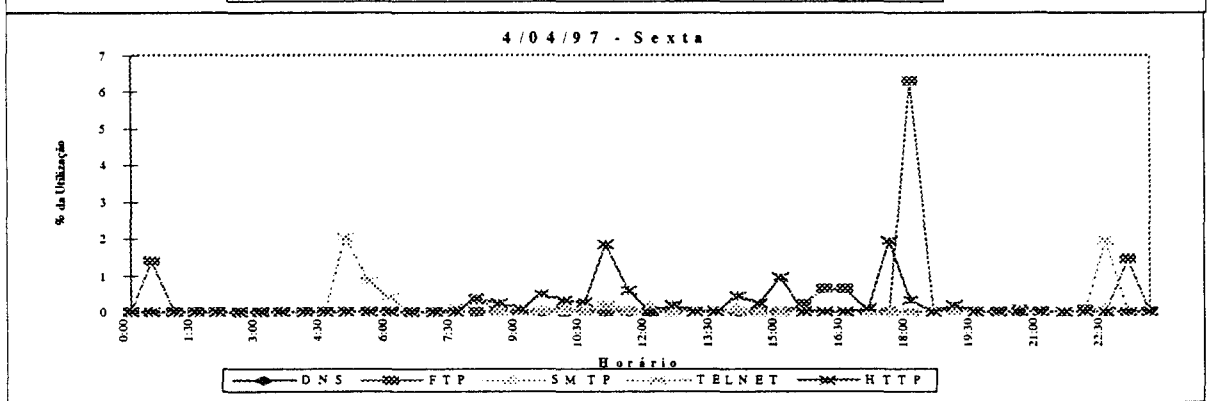

28/02/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

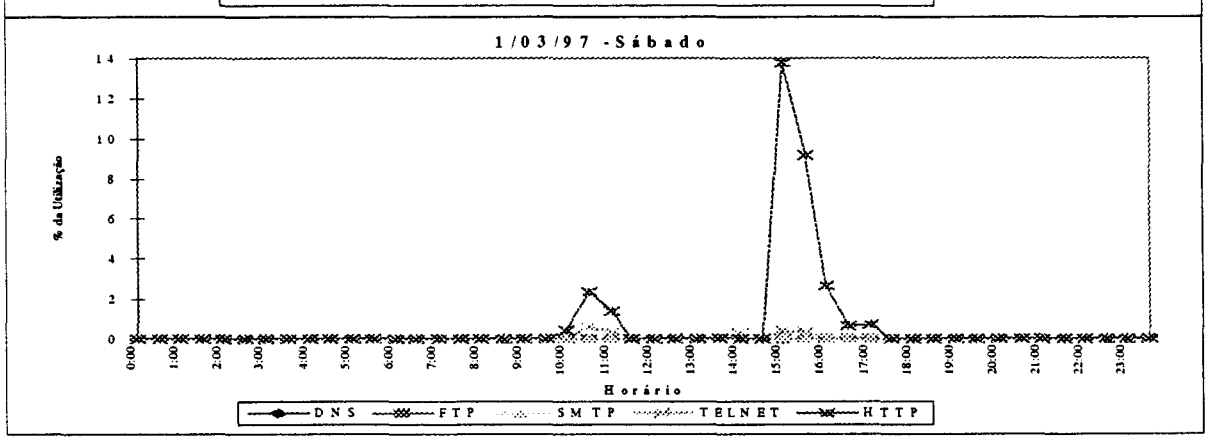

7/03/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

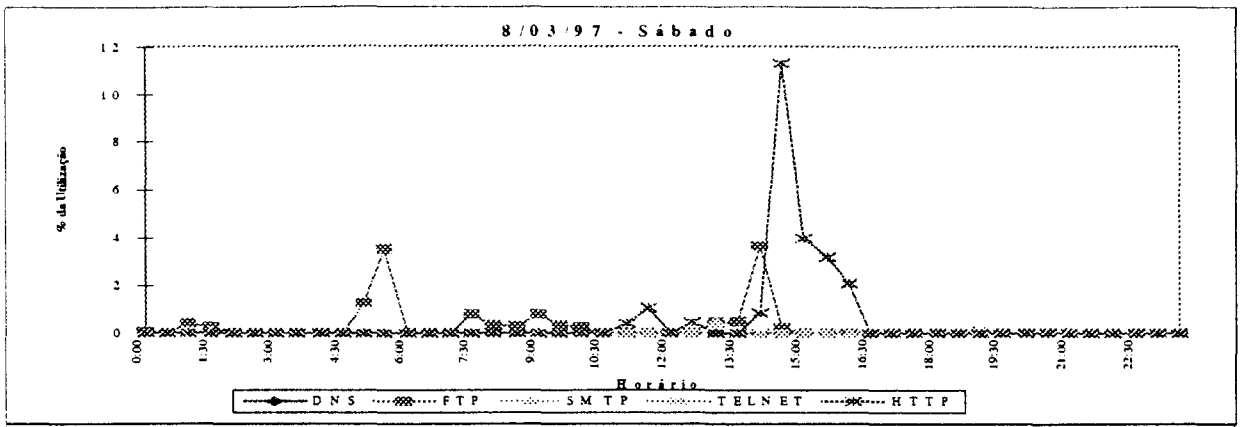

14/03/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

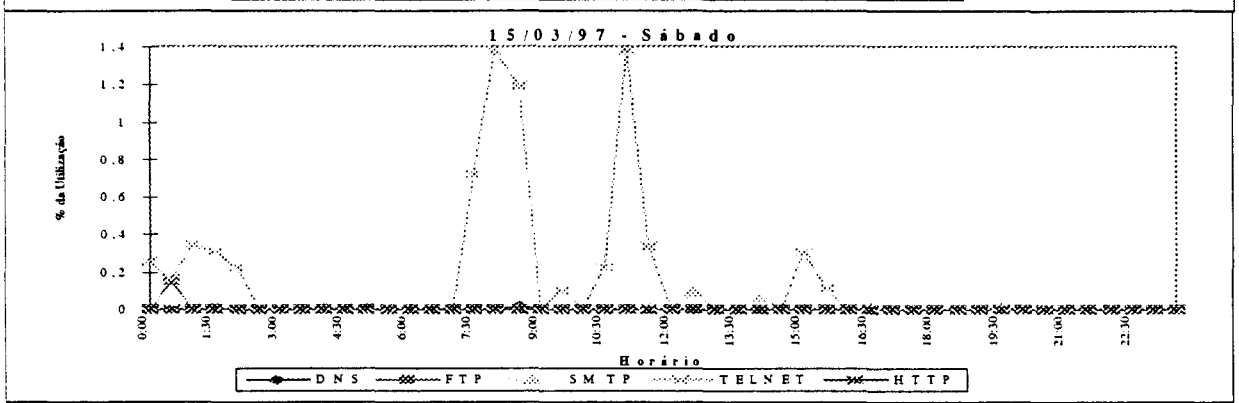

21/03/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

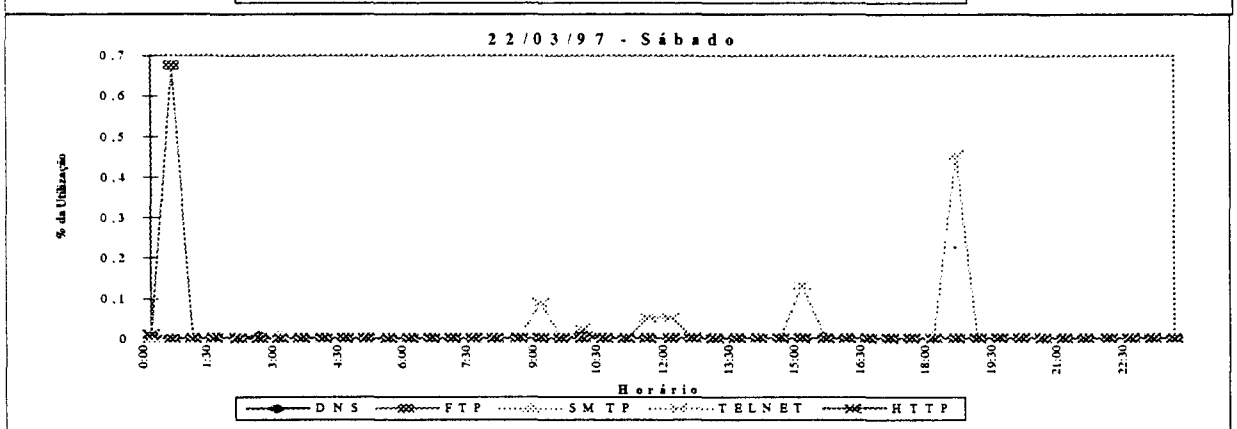

28/03/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

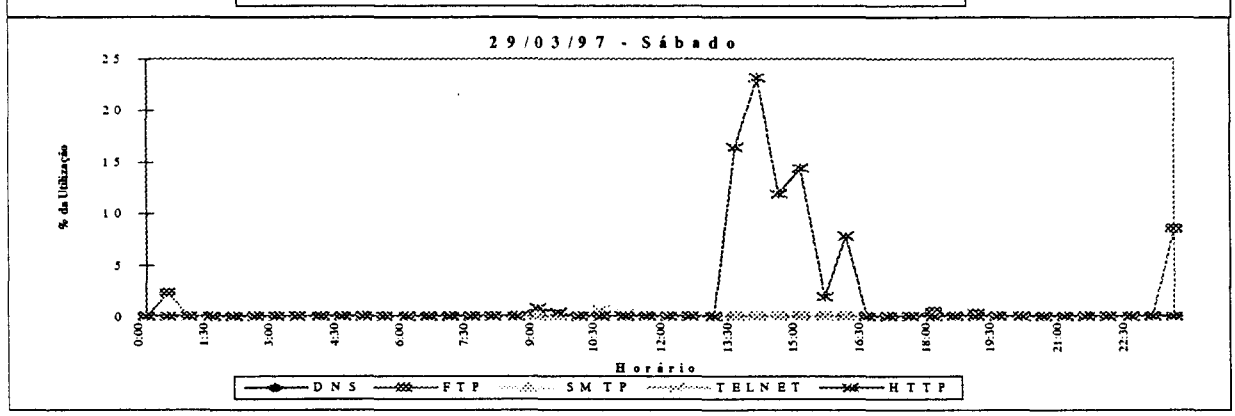

4/04/97 - Sexta
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

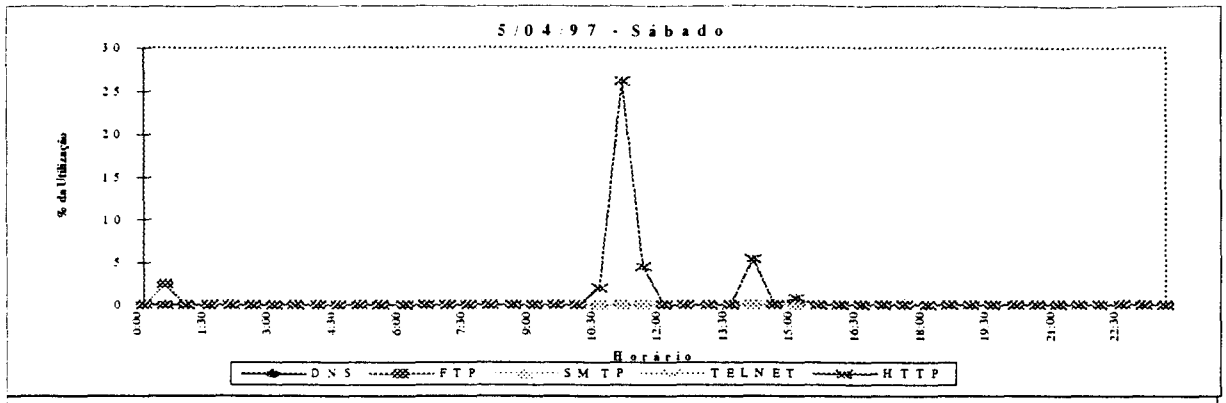

1/03/97 -Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

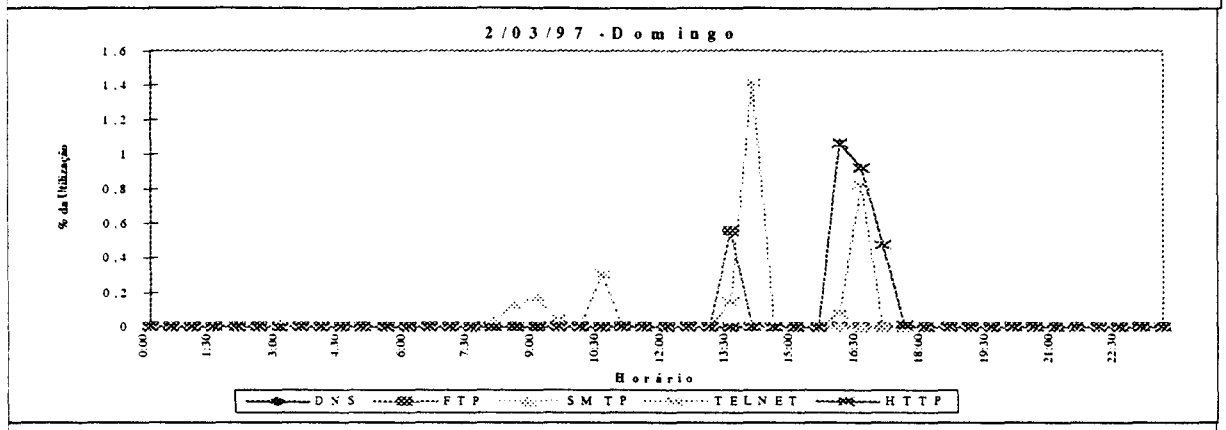

8/03/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

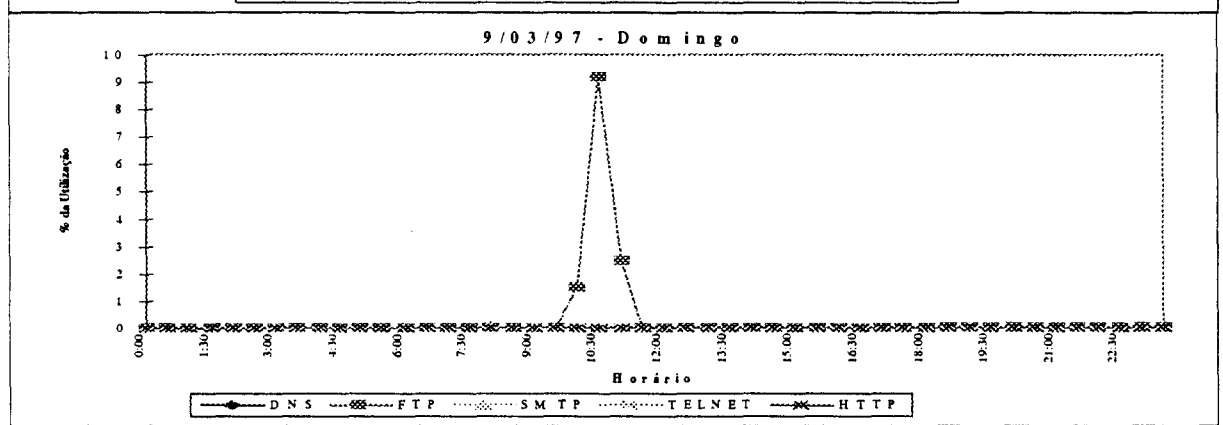

15/03/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

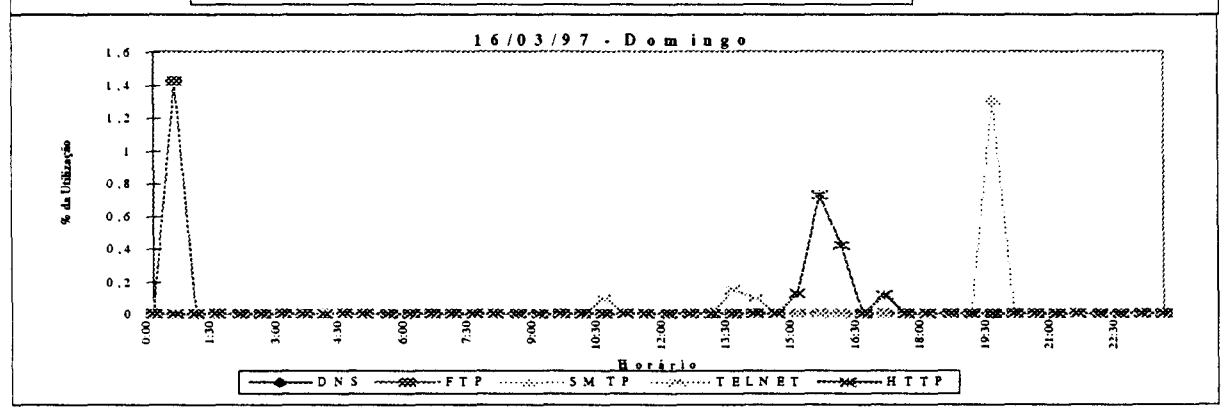

22/03/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

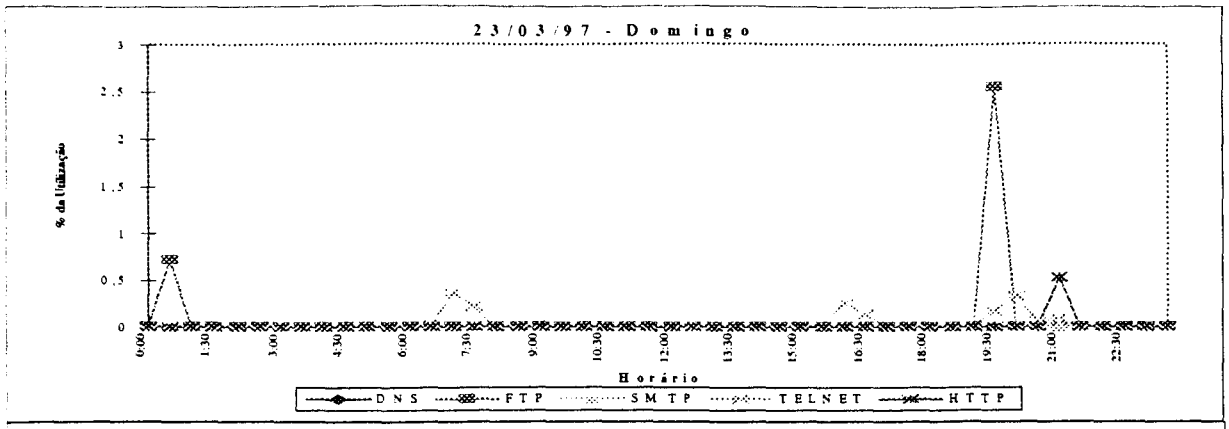

29/03/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

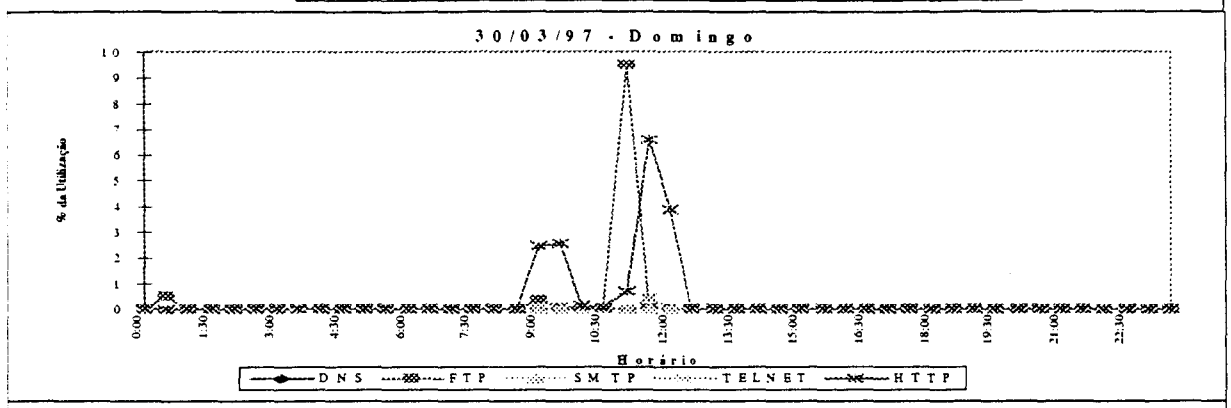

5/04/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP

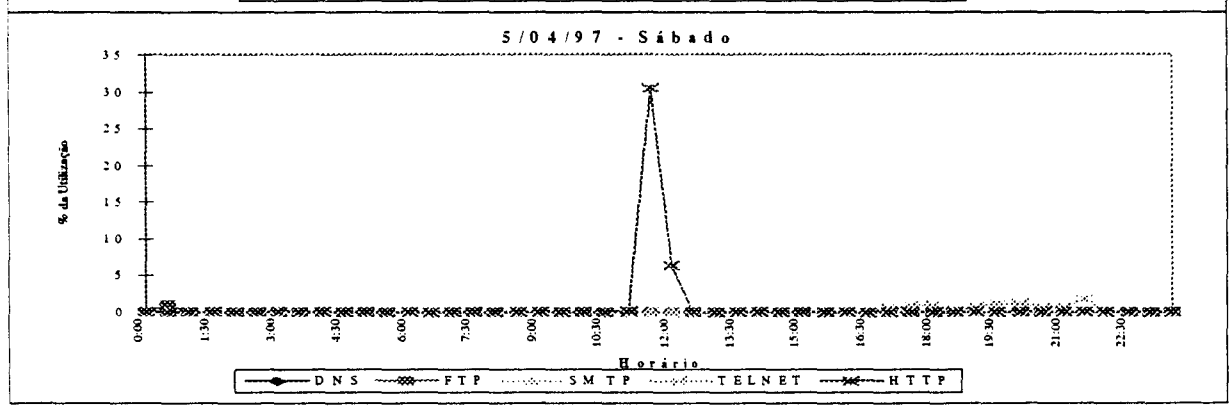

2/03/97 - Domingo
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP


9/03/97 - Domingo
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP


16/03/97 - Domingo
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP


23/03/97 - Domingo
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP


30/03/97 - Domingo
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP


5/04/97 - Sábado
% da Utilização
Horário
DNS
FTP
SMTP
TELNET
HTTP